VOL. 14, NO. 15, AUGUST 2019 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ROBUST IMPULSIVE NOISE SUPPRESSION TECHNIQUE FOR ITERATIVE TIMING RECOVERY

Chuan Hsian Pu, Ezra Morris Abraham Gnanamuthu, Fook Loong Lo and Mau-Luen Tham
Lee Kong Chian Faculty of Engineering and Science,Universiti Tunku Abdul Rahman
Jalan Sungai Long, Bandar Sungai Long Cheras, Kajang, Selangor, Malaysia
E-Mail: chpu2001@hotmail.com

ABSTRACT

Channels typically introduce amplitude and phase distortions as well as intersymbol interference (ISI) to the received signals. Therefore, timing recovery is crucial when sampling the received waveforms at the correct instants. Specifically, we consider an iterative timing recovery technique, which incorporates the error correction codes (ECC) owing to its robustness against poor signal-to-noise ratios (SNRs). These gains, however, are purely derived under the assumption of the additive white Gaussian noise (AWGN). The bit error rate (BER) performance of iterative timing recovery methods may suffer in non-Gaussian channels where impulsive noise exists. In view of this, we propose a novel suppression method to remove impulsive noise from the non-Gaussian channel before feeding the received signal to the iterative timing recovery unit. Results demonstrate that the proposed scheme can perform well, even in non-Gaussian channels.

Keywords: error correction codes, robustness, iterative timing recovery, impulsive noise, suppression, turbo equalization.

1. INTRODUCTION

SYNCHRONIZATION is an essential mechanism for any communication systems. In digital communication systems, various channel impairments such as mismatch of clock frequencies, multipath propagations and inherent error-prone nature of channel environments render the optimal detection a challenging task. Typically, matched filter (MF) is adopted for coherent detection to maximize the signal-to-noise ratio (SNR) in the presence of Gaussian noise. The MF works in the optimal condition when the receiver is synchronized to the transmitter. The optimality of the MF filtering deteriorates when its output is not sampled at the precise timing instants at the receiver.

Timing recovery is a standard method to minimize the timing offset between the transmitter's and receiver's clock. It consists of two phases: timing error detection (TED) and timing acquisition (TA). The former is to compute the timing error that occurs between transmitter and receiver. The latter is to perform tracking on timing instants when the transmitter and receiver are phase-locked with each other. Conventionally, timing recovery works such as [1] does not take into consideration the presence of error correction codes (ECCs). Applying it with prevalent ECCs such as turbo code in a sequential pattern will lead to performance degradation [2].

An iterative timing recovery method, which jointly solves the timing recovery and error-control decoding was proposed in [3]. Specifically, the proposed solution combines Mueller & Müller (M&M) TED [4], TA and turbo equalizer. Different from the original mechanism in [4], which requires training sequence, the work in [3] exploit the tentative sequence extracted from the feedback of turbo decoder. Thanks to the fact that the corrected tentative decision from the turbo equalizer's output can be used as feedback to guide the M&M TED to produce more reliable estimates of the timing error in multiple iterations, these timing error estimates are fed to the TA to obtain digital samples from received waveforms. In this way, their performance improves significantly in the low SNR region. However, their work is based on additive white Gaussian noise (AWGN) channels.

In real-world, many channels are non-Gaussian such as power-line channels (PLC), digital subscriber lines (DSL), indoor wireless links, underwater acoustic channels, ultra-wideband radio channels, etc [2]. Impulsive noise is one of the common types of non-Gaussian channels; it possesses impulses in the form of outliers in channel modellings. Works that take into account the presence of impulsive noise for timing recovery in digital subscriber loop (DSL) can be found in [3]. Blind sampling clock adjustment technique was proposed in [4] for orthogonal frequency division multiplexing (OFDM) powerline channels which exhibit the nature of the impulsive channels.

In [8], the authors proposed a framework called "matched myriad filtering" (MMyF) framework, in order to suppress the presence of outliers at the receiver. It consists of TED, TA and a turbo equalizer where each module operates orderly and independently. Their results demonstrated bit error rate (BER) performance gains over impulsive noise channels for various impulsive indices and convergence performances of timing recovery. However, as mentioned earlier, the separation in timing recovery unit from ECCs results in the timing recovery does not benefit from the soft-information generated by the iterative decoder. Hence, the separated design does not benefit the timing recovery at the low SNR region.

Our earlier work in [9] presented some preliminary ideas of jointly performing timing recovery and error-correction decoding for non-Gaussian channels. The decoded bits from the turbo equalizer is used to aid the timing recovery to acquire more precise timing instants for sampling the received signals. The timing recovery and error-correction decoding processes are iterated a number

of times in order to minimize the timing errors and to lower the BER at the receiver. In this paper, we provide an in-depth analysis of a novel suppression technique for the timing recovery method, particularly on its BER performance. Results demonstrate that the proposed scheme can perform well in low SNR region for non-Gaussian channels.

The rest of the paper is organized as follows. Section II describes the system overview and problem formulation. In Section III, we present the channel model. Section IV details the Gaussian-based iterative timing recovery with MF. Section V presents the proposed impulsive noise suppression technique for iterative timing recovery with MMyF. Section VI provides simulation results. The conclusion is drawn in Section VII.

2. SYSTEM OVERVIEW AND PROBLEM FORMULATION

Figure-1 shows a baseband iterative timing recovery model. The model is divided into three parts, namely, transmitter, channel and receiver. The transmitter is formed by the Recursive Systematic Convolutional (RSC) encoder (k, n, K) with k input bits, n output bits and K constraint length, interleaver and transmitting filter which is an MF with pulse shape of square root raised cosine. The RSC encoder encodes the message bits $u_k \in \{0,1\}$ into coded bits $c_m = c_{(k,l)} \in \{-1,1\}$ by using convolutional encoding with code rate k/n, where k is the time index, $l = 1, \dots, n$, and each of the message bit u_k generates $\boldsymbol{c_m} = c_{(k,l)}$ coded sequence and $m = k + l$. The interleaver maps c_m to d_m.

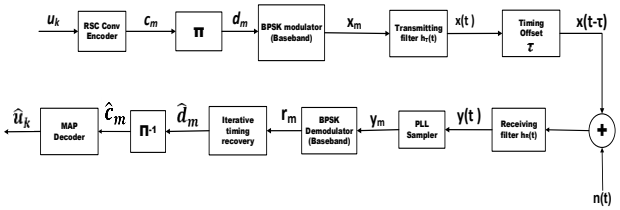

Figure-1. The communication system model of iterative timing recovery.

The output d_m of the interleaver π is modulated with binary phase-shift keying (BPSK) as in (1), where interleaved coded bits d_m are modulated with either $+1$ and -1, where $k = 1,2,3, \dots \dots N$

$$x_m = (-1)^{d_m} e^{j2\pi f_c t} \quad (1)$$

where the interleaved coded bit $d_m \in \{0, 1\}$ is mapped to the BPSK symbol $x_k = \{1, -1\}$, f_c is the carrier frequency. Since the $e^{j2\pi f_c t}$ is a fixed term for the BPSK modulation, hence, we can omit it for simplicity $x_k = (-1)^{d_m}$, where the interleaved coded bit $d_m \in \{0, 1\}$ is mapped to the BPSK symbol $x_m = \{1, -1\}$ for BPSK baseband modulation.

The role of the transmitting filter is to serve as MF to shape the modulated symbol x_m into square-root-raised-cosine waveforms for bandwidth limited channel.

The channel output $y(t)$ waveform can be given as in (2).

$$y(t) = \sum_{i=-\infty}^{\infty} x_m h(t - mT - \tau_m) + n(t) \quad (2)$$

where τ_m is the unknown timing offset. τ_m is the delay caused by the channel at time instant $m\,T$. Noise $n(t)$ can be Gaussian or non-Gaussian that the channel introduced to the transmitted signal $x(t)$. Non-Gaussian noise can be α-stable noise, which is one type of non-Gaussian random process, and which we will elaborate in more details in section (III). $h(t)$ is the cascade of the impulse response for the overall end-end system given in (3)

$$h(t) = h_T(t) * h_c(t) * h_R(t) \quad (3)$$

where $h_T(t)$, $h_c(t)$, and $h_R(t)$ denote the impulse responses of the transmitting filter, channel, and the receiving filter, respectively.

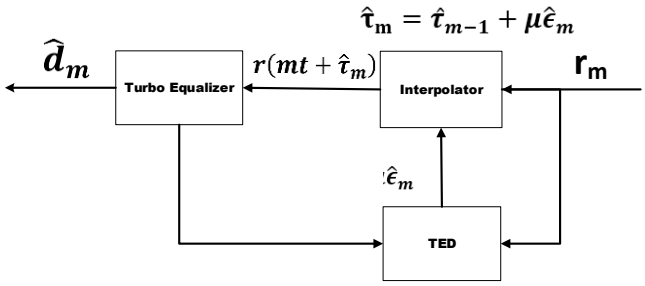

Figure-2. Iterative timing recovery block.

The uncertain timing offset τ_m is defined as the random walk model [1] with the zero mean normal distribution as in (4),

$$\tau_m = \tau_{m-1} + N(0, \sigma_T^2) \ . \quad (4)$$

where $\tau_m \in (-\infty, \infty)$. The random walk model uses σ_T to indicate the severity of the timing offset.

Iterative timing recovery block is shown in Figure-2; it consists of turbo equalizer, interpolator and TED. The interpolator uses (5) to for TA. The interpolator is controlled by the timing error estimate $\hat{\epsilon}_m$ from the M&M TED. The M&M TED generates the required timing error estimate $\hat{\epsilon}_m$ for the interpolator to produce the new timing estimates $\hat{\tau}_{m+1}$ to sample the received signal $y(t)$ at time instant $m + 1$ to obtain the sample r_{m+1} during the tracking phase via interpolator.

$$\hat{\tau}_{m+1} = \hat{\tau}_m + \mu\,\hat{\epsilon}_m \quad (5)$$

where μ is the step-size which controls the agility of the interpolator.

3. CHANNEL MODELS

Gaussian distribution has been widely used in modelling communication channels, and it is a well-

known modelling method for a noisy channel. However, in many practical applications, Gaussian modelling for the noisy channel may not describe the underlying noise phenomena adequately. The practical channels may experience non-Gaussian interference and should be modelled by non-Gaussian distributions. The popular non-Gaussian distributions are Gaussian mixture or contaminated Gaussian model [5], Middleton's Class A, B, C models [6], and Alpha-Stable model [7]. These noise models were developed to describe the required statistical characteristics for the non-Gaussian interference in the channels. We use symmetric α-stable (SαS) model to describe the non-Gaussian distribution for the noise source $n(t)$ shown in Figure-1. There are several advantages for using the α-stable process; first ,i t provides good empirical fits to an impulsive random process. Second, it has been shown by researchers that it satisfies strong theoretical justification, and satisfies the generalized center limit theorem [5].

In general, the α-stable random process can be described by its characteristic function, as shown in (6).

$$E(exp(itX)) = \begin{cases} exp\left(-\gamma^{\alpha}|t|^{\alpha}\left[1 + i\delta sign(t)tan\left(\frac{\pi\alpha}{2}\right)((\gamma|t|)^{1-\alpha} - 1) + i\beta_0 t\right]\right) \; for\; all\; \alpha \neq 1 \\ exp\left(-\gamma|t|\left[1 + i\delta sign(t)\frac{2}{pi} ln(\gamma|t|) + i\beta_0 t\right]\right) for\; \alpha = 1 \end{cases} \quad (6)$$

The δ controls the skewness of the α-stable process. If $\delta = 0$, the α-stable process is called the symmetric α-stable (SαS) process, then the distribution is symmetric about the location parameter β. Its characteristic function can be described as (7).

$$\phi(x) = exp(-\gamma^{\alpha}|x|^{\alpha}) \quad (7)$$

where $-\infty < x < \infty$. The γ is the dispersion parameter, which is a positive constant related to the scale of the distribution. The variable $\alpha \in [0,2]$ is the index of stability. The α represents the degree of impulsiveness for the α-stable distribution. It controls the thickness of the tails of the pdf (probability density function). If $\alpha = 1$, the α-stable process is equivalent to Cauchy distribution. If $\alpha = 2$, the α-stable process is equivalent to zero-mean Gaussian distribution with variance $2\,\gamma^{2/3}$. For all other α values, they correspond to heavy-tailed distributions. The dispersion parameter γ is related to the scale of the density distribution, and it is a positive constant value. The α-stable random process is generally lacked closed form expressions [5] except for limited cases, such as for the Gaussian distribution ($\alpha = 2$), the Cauchy distribution ($\alpha = 1$), and the Lévy ($\alpha = 1/2$) distribution. If $\alpha = 1$, the density function of the follows Cauchy distribution and can be given in (8).

$$f(x) = \frac{1}{\pi}\frac{\gamma}{\gamma^2 + (x-\beta)^2} \quad (8)$$

where $-\infty < x < \infty$.

Given its lacking of the closed-form expressions for the density functions, it is problematic to evaluate the variance and mean for the α-stable processes. Without the knowledge of the noise variance, it is difficult to measure the performance of the communication system over the α-stable channel. In such a situation, the Geometric signal-to-noise ratio (GSNR) can be used as a parameter of measurement instead of the conventional E_b/N_0. When $\alpha < 2$, the density function in (7) will have infinite variance and undefined mean. As a result, GSNR is the best tool to deal with the case where $\alpha < 2$, where the second-order moment does not exist. GSNR can be used to estimate the noise power of α-stable noise as given in (8):

$$GSNR = \frac{1}{2C_g}\left(\frac{E_b T}{S_0}\right) \quad (8)$$

where E_b is bit energy, T is the symbol period, $C_g \approx 1.78$ is the exponential of the Euler constant. S_0 is the geometric noise power.

4. GAUSSIAN-BASED ITERATIVE TIMING RECOVERY WITH MF

A. Iterative timing recovery structure

Figure-3 shows the structure of the conventional iterative timing recovery block of Figure-1, used in [8] for systems with Gaussian channels. For its structure, iterative timing recovery is made up of an M & M TED, an interpolator and a turbo equalizer and a decoder. For the turbo equalizer, it consists of a SISO (Single Input Single Output) component equalizer, a SISO component decoder, an interleaver π and a de-interleaver π^{-1}. First of all, the analogue received signals are sampled by using a free-running PLL (Phase-Lock-Loop). Subsequently, the digital samples from the output of the PLL are interpolated by evaluating the timing estimate $\hat{\tau}_m$ at m^{th} time instant to obtain the received sequence $y_m = y(m\,T + \hat{\tau}_m)$. Initially, assume that the timing estimate $\hat{\tau}_m$ is zero during startup. After the interpolator, the interpolated received sequence is fed to the turbo equalizer for ISI equalization and decoding; the process is iterated several times. Typically, eight times of iterations are sufficient for optimal performance.

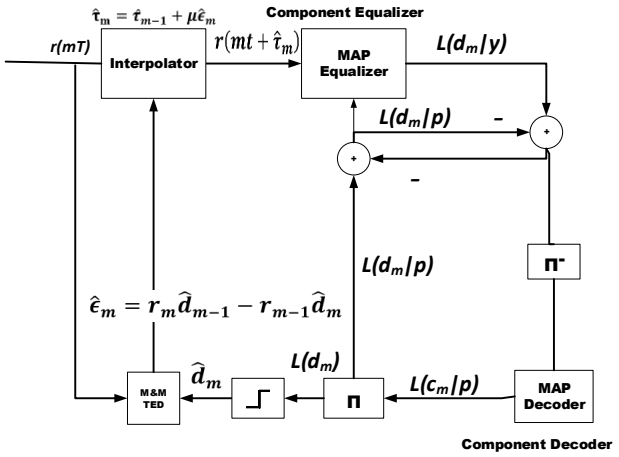

Figure-3. Turbo Equalizer block diagram.

The component equalizer and component decoder perform equalization and decoding based on message passing algorithm with Maximum-a-posteriori (MAP) computations to generate probabilistic APP soft-information as its output. The soft-information is known as log-likelihood ratios (LLRs) as it is a convenient method for the decoder to produce the hard-decision output by determining the sign of the LLR's value. The soft-LLR information is exchanged internally between the component equalizer and component decoder of the turbo equalizer to achieve performance gain for the conventional iterative timing recovery.

At the beginning of the operation, the interpolated received samples $r_m = r(m\,T)$ with $\hat{\tau}_m = 0$ from the interpolator is fed to the component equalizer which equalizes the ISI from the dispersive AWGN channel. The equalizer computes the a-posteriori LLRs of $L(d_m|\boldsymbol{r})$ for interleaved bit d_m using MAP. MAP algorithm calculates the a-posteriori probability (APP) for each state of the trellis to generate the required LLR values.

A-priori information of a bit is the information known before the decoding or equalization commences, and this information is associated with the source of the information bits. Initially, the a-priori information is set to zero in the logarithmic domain as we assume the source generates the information bits as I.I.D (independent and identically distributed).

When performing soft-information passing by using soft LLR values, it is crucial to avoid creating a direct positive correlation between the component equalizer and decoder. Therefore the role of the interleaver π and the de-interleaver π^{-1} is to scramble the channel LLRs to avoid error bursts. For iterative decoding, it is imperative that the soft LLR values sent back to the equalizer or decoder do not contain its own decisions or high positive correlation for better performance. The direct positive correlation of the LLR values would lead to the fast saturation and unstable operation of the turbo equalizer. Hence, this brings forth the notion of intrinsic and extrinsic information. Extrinsic information is generated to be used as the priori input for the SISO components in the turbo equalizer. Therefore, the output LLR $L(d_m|\boldsymbol{r})$ of the equalizer is subtracted by the a-priori values $L(d_m)$ to generate extrinsic LLRs to be passed on to the decoder. The extrinsic LLRs of the equalizer is given as in (9).

$$L_{ext}(d_m|\boldsymbol{r}) = L(d_m|\boldsymbol{r}) - L(d_m) \tag{9}$$

Subsequently, the extrinsic values $L_{ext}(d_m \mid \boldsymbol{r})$ from the component equalizer is fed to the de-interleaver π^{-1} to be de-interleaved before being sent to the component decoder as local evidence LLRs $L_{ext}(c_m|\boldsymbol{p})$ about the coded bit c_m. The $L_{ext}(c_m|\boldsymbol{p})$ [9] is used by the decoder to compute $L(c_m|\boldsymbol{p})$ for the coded bit c_m by setting the channel observation input to zero $L(c_m|\boldsymbol{r}) = 0$.The extrinsic information of the decoder could be computed as in (10).

$$L_{ext}(c_m|\boldsymbol{p}) = L(c_m|\boldsymbol{p}) - L_a(c_m) \tag{10}$$

where $L_a(c_m)$ is the a-priori information to the component decoder, **p** is the dummy vector for the completeness of the APP.

The $L_{ext}(c_m|\boldsymbol{p})$ is interleaved by the interleaver π to produce $L_{ext}(d_m|\boldsymbol{p})$ and it is used as a-priori information for the equalizer. Hence, in this way, the turbo equalizer completes an iterative cycle of equalization and decoding for the input sample u_k.

The equalizer and decoder exchanged their extrinsic LLR values to improve the performance gain in several iterations. Turbo equalizer applies MAP algorithm to calculate the APP values $L(d_m|\boldsymbol{r})$ for the component equalizer and APP $L(c_m|p)$ for the component. Hence, the soft LLR information is exchanged between the equalizer and decoder for each iterative cycle. By using the APP values of $L(d_m|\boldsymbol{r})$ and $L(c_m|\boldsymbol{p})$, the turbo equalizer can determine the most likely bits that have been transmitted from the noisy channel. The hard-decision information $\hat{c}_m$ for coded bits could be obtained from the decoder's output with the following decision rule.

$$\hat{c}_m = \begin{cases} 1, & L(c_m|\boldsymbol{p}) \geq 0 \\ -1, & L(c_m|\boldsymbol{p}) < 0 \end{cases} \tag{11}$$

After completing a cycle, the hard-decision estimates of the coded bits $\hat{c}_m$ from the decoder are interleaved again as $\hat{\mathrm{d}}_m = \Pi(\hat{c}_m)$ and is used by M&M TED to compute the timing error estimate $\hat{\epsilon}_m$as in (12), where $\Pi(\boldsymbol{c})$ is the interleaving operator.

$$\hat{\epsilon}_m = r_m\hat{\mathrm{d}}_{\mathrm{m-1}} - r_{m-1}\hat{\mathrm{d}}_m \tag{12}$$

The next timing estimate $\hat{\tau}_{m+1}$ can be obtained with (13).

$$\hat{\tau}_{\mathrm{m+1}} = \hat{\tau}_m + \mu\hat{\epsilon}_m \tag{13}$$

where μ is the step size which controls the agility of the interpolator. The new timing error estimate $\epsilon_{\mathrm{m+1}}$ from the output of the M&M TED is fed to the interpolator to

provide more accurate information to interpolate the received vector with (13).

As the number of iterations increases with more reliable LLR values $L(c_m|\boldsymbol{p})$ available from the decoder's output, the improved hard-decision information of $\hat{\mathrm{d}}_m$ from LLRs can provide a more accurate estimation on the timing error estimate $\hat{\epsilon}_m$which adjusts the next timing offset estimate $\hat{\tau}_{m+1}$of the interpolator to sample the received signal r_m more accurately. In short, the iterative process that gleans the extrinsic information is used to refine the equalization and decoding processes to produce a reliable measure of the input channel information. Then, the improved channel estimates from the component equalizer are subsequently used by M&M TED to minimize the timing errors and to sample the received signal more accurately. For better performance, a reasonable trade-off between latency and response time, the sampling/interpolating, equalization, decoding processes are repeated eight times for excellent convergence.

B. Maximum-a-posterior algorithm (MAP) for component equalizer

As mentioned earlier, (MAP) [10] is applied to the SISO units to compute the LLR values for the turbo equalizer. In our context, the channel is presumably memoryless and based on hidden Markov chain transition. For memoryless Markov chain, the future state is only dependent on the current state but not the past states and the inputs. For hidden Markov chain, typically the sequence of states are unobservable, but the outputs coded bits of the states are observable to the receiver.

The outputs of the coded bits represent the signals emitted from the RSC encoder when the state on the trellis of the encoder transits from the previous state $S_{k-1} = s`$ to the current state $S_k = s$ by giving data bitu_k, where $k = 1,2, \ldots \ldots K$ is the time index for data bit. The output of the RSC encoder is the coded sequence $\boldsymbol{c_m}$, where $m = 1,2 \ldots \ldots \ldots, N$. m is the time index of the coded bit. The code rate is $R = \frac{k}{N}$.

For the component equalizer, the a-priori LLRs input d_m can be defined as merely the logarithmic ratio of the probability for interleaved coded bit $\mathrm{d_m} = \{\pm 1\}$ taking on its two possible values, The LLR for the interleaved coded bit d_m can be expressed as in (14).

$$L(d_m) = ln\left(\frac{P(\mathrm{d_m}=+1)}{P(\mathrm{d_m}=-1)}\right) \tag{14}$$

where d_m is the interleaved coded bit, and $m = 1,2,3, \ldots \ldots, N$ is the time index.

The soft information from the output of the component equalizer is represented in the form of a posteriori log-likelihood ratios (APP-LLRs). It is defined as in (15).

$$L(\mathrm{d_m}\,|\boldsymbol{r}) = ln\left(\frac{P(d_m=+1|\boldsymbol{r})}{P(d_m=-1|\boldsymbol{r})}\right) \tag{15}$$

where $\boldsymbol{r}$ is the noisy received vector from the output of the interpolator.

Component equalizer takes samples of the received signal from the interpolator, then calculates the maximum likelihood of the interleaved coded bit from the trellis. For each state, only two possible transitions can take place, which is interleaved coded bit $d_m = +1$ and $d_m = -1$. Hence, the probability for $d_m = +1$ is the probability for the transition from the previous state $S_{m-1} = \grave{s}$ to the present state $S_m = s$, which is one set of the four possible transitions that can take place when the input coded bit $d_m = +1$ and it is vice-versa for $d_m = -1$. Due to the set of transitions being mutually exclusive as only one of them could have taken place at each stage in the trellis, the probability that any one of them occurring is equivalent to the sum of the probabilities causing the transition of the $d_m = +1$ for the numerator, and of the $d_m = -1$ for the denominator. Hence, the output of the equalizer in APP form can be expressed as (16).

$$L(d_m|\boldsymbol{r}) = ln\left[\frac{\sum_{(\grave{s},s)=>d_m=+1} P(\grave{s}\cap s\cap \boldsymbol{r})}{\sum_{(\grave{s},s)=>d_m=-1} P(\grave{s}\cap s\cap \boldsymbol{r})}\right] \tag{16}$$

$(\grave{s}, s) => d = +1$ is the set of transitions from the previous state $S_{m-1} = \grave{s}$ to the present state $S_m = s$ where the interleaved coded bit is $d_m = +1$ and vice-versa for $(\grave{s}, s) => d_m = -1$.

The probability $P(\grave{s} \cap s \cap \boldsymbol{r})$ can be split into three terms; namely, forward recursion $\alpha_k(s`)$, transition probability $\gamma_k(\grave{s}, s)$ and backward recursion $\beta_k(s)$. With the use of Bayes' rule, the three probability terms can be used to obtain the APP-LLR output $L(d_m|\boldsymbol{r})$ for the equalizer as in (17),

$$L(d_m|\boldsymbol{r}) = ln\left(\frac{\sum_{d_m=+1} \alpha_{m-1}(s`)\gamma_m(s`,s)\beta_m(s)}{\sum_{d_m=-1} \alpha_{k-1}(s`)\gamma_m(s`,s)\beta_m(s)}\right) \tag{17}$$

Likewise, for forward recursion $\alpha_k(s)$ probability term can be computed recursively as (18).

$$\alpha_m(s) = \sum_{all\ s`} \alpha_{m-1}(s`)\gamma_m(s`, s) \tag{18}$$

where $\alpha_{m-1}(s`)$ is the forward probability term that the trellis is in the previous state $S_{k-1} = \grave{s}$ and the received channel sequence up to this time instant is $r_{j<m}$. As the trellis starts from the initial state S_0, the initial conditions of the forward recursion can be given as in (19).

$$\alpha_0 = \begin{cases} 1, & if\ S_0 = 0 \\ 0, & if\ S_0 \neq 0 \end{cases} \tag{19}$$

Note that $\alpha_0(s` = 0) = 1$ and $\alpha_0(s` \neq 0) = 0$.

The $\beta_m(s)$ is the backward recursive probability that the trellis is in state s at time index m for the interleaved coded bits. The future received channel

sequence will be $r_{j>m}$. The backward recursion $\beta_m(s)$ can be split into the joint APP probabilities of the current received channel sequence r_m and future received channel sequence $r_{j>m}$ conditioned on the previous state $\dot{s}$. The backward recursion of the component equalizer can be formulated with Bayes' rule in (20).

$$\beta_{m-1}(\dot{s}) = \sum_{\text{all s}} \beta_m(s)\gamma_m(\dot{s}, s) \quad (20)$$

To calculate the backward probability$\beta_{m-1}(\dot{s})$ recursively, again the branch transition probability $\gamma_m(\dot{s}, s)$ is needed as part of the recursive terms for the backward recursion $\beta_{m-1}(\dot{s})$ at time index $m-1$. The branch transition $\gamma_m(\dot{s}, s)$ metrics can be computed from the received channel sequence r_m conditioned on the noiseless channel sequencex_m. For memoryless Markov chain, the branch transition metrics $\gamma_m(\dot{s}, s)$ can be given in (21)

$$\gamma_m(\dot{s}, s) = P(r_m|\{\dot{s} \cap s\}) \cdot P(s|\dot{s})$$
$$\gamma_m(\dot{s}, s) = P(r_m|d_m) \cdot P(d_m) \quad (21)$$

where $(\dot{s}, s) \Rightarrow d_m = +1$ is the set of transitions from the previous state $S_{k-1} = \dot{s}$ to the current state $S_k = s$ that is caused by the positive interleaved coded bit $d_m = d_{k,l} = +1$, where $m = k \times l$. Similarly, this can be applied to the transition from$(\dot{s}, s) \Rightarrow d_{k,l} = -1$. $P(d_m)$ is the a-priori probability of the interleaved coded bit.

For a discrete system, the received signal $y(t)$ is sampled by a free-running PLL. Hence, for a discrete channel model with impulse response length of $L = 3$, the discrete channel's output after PLL can be shown in (22):

$$r_m = \sum_{l=0}^{L} x_{m-l}\, h_l + n_m \quad (22)$$

where $x_m = x(m\,T + \tau_m)$ is the BPSK modulated signal, the real-valued coefficients $h_l = h(l\,t + \tau_l)$ are the sampled impulse response of the cascade of the filters $h = h_T(t) * h_c(t) * h_R(t)$, and n_m is the sampled noise at time instant mT.

For an AWGN channel, the transition probability $\gamma_m(\dot{s}, s)$ can be calculated as (23)

$$\gamma_m(\dot{s}, s) = C \exp\left(\frac{d_m L(d_m)}{2}\right) \exp\left(\frac{L_c}{2} \sum_m^n (r_m - v_m)^2\right) \quad (23)$$

where v_m is the discrete channel noiseless output. $L_c = 4a\frac{E_b}{2\sigma^2}$ is defined as channel reliability, where a is the fading amplitude (for a non-fading channel, $a = 1$). Assuming that the sampled impulse response $L = 3$, then the noiseless output can be expressed as in (24),

$$v_m = \sum_{l=0}^{L} d_{m-l} h_l \quad (24)$$

From (23), we can see that the transition probability $\gamma_m(\dot{s}, s)$ is associated with the quadratic square of the difference of received sample y_m and noiseless sample v_m. The transition probability $\gamma_m(\dot{s}, s)$ is over-emphasised in the situation with the outlier presence, which has a characteristic function given in (6). Hence, the Gaussian-based iterative timing recovery fails to recover the signal's exact sampling instants in the non-Gaussian channels.

5. PROPOSED IMPULSIVE NOISE SUPPRESSION TECHNIQUE FOR ITERATIVE TIMING RECOVERY WITH MMYF

We propose a robust technique to suppress the presence of outliers to the conventional timing recovery. MMyF framework is introduced to replace the MF residing on the front-end of the receiver. MMyF filtering framework has shown its capability in dealing with the outliers with its tunable parameter K. Hence, the introduction of the MMyF filtering can significantly improve robustness and reliability for timing estimates to the conventional iterative timing recovery.

We propose the algorithmic MMyF framework the timing recovery as follows:

Step 1. Filter the received signals with MMyF

Consider the received signal *y(t)* in Figure-1, which is sampled by a free-running PLL at a constant rate. A window-size of N samples are captured as the vector of the received signal $\boldsymbol{y} = \{y_m\}_{m=1}^{N}$. The discrete signal samples $\boldsymbol{y}$ from the PLL is demodulated to produce$r_m = r(mT)$, which is connected between the PLL and the turbo equalizer. The interpolator starts with $\hat{\tau}_m = 0$ in the first iteration. Consider the received m^{th} sample corresponding to m^{th} the coded bit, which can be represented in (25),

$$r_m = y(mT + \hat{\tau}_m),\ m = 1,2,\dots,N \quad (25)$$

MMyF filtering framework uses the objective function in (26) to perform estimation to find out the correct $\hat{\beta}_m^N$ of the received samples which minimize the sum of the cost function ρ(x).

$$\hat{\beta}_m^N = arg\ min_\beta \sum_{k=0} log[\rho(r_m^N - \beta)], k = 1,2,\dots,N \quad (26)$$

where N is the length of the MMyF filter.

The cost function can be defined as (27)

$$\rho(x) = \sum_{k=0} -log[f(y_m^N - \beta)] \quad (27)$$

Substituting (27) into (26), we can get (28) as one form of the M-estimator, which is the objective function of the MMyF filtering framework. Its objective is to minimize the distance of the received samples $\{r_m\}_{m=1}^{N}$ to the location parameter. It is called the location parameter as it determines the location of the α-stable distribution.

$$\hat{\beta} = arg\ min_\beta \sum_{m=1}^{N} -log[f(y_m - \beta)] \quad (28)$$

Real-valued weights are introduced to the filtering process, where the weights are identical and matched to the impulse equivalent response used in (2). With the above M-estimator cost function and the given

set of N real-valued weights $\boldsymbol{w} = \{w_1, w_2, w_3 \ldots, w_N\}$, we can express the output of the MMyF filtering framework as in (29),

$$\beta_K(W,r) = myriad\{|w_k| \circ (sgn(w_m)r_m - \beta)\}|_{k=1}^{N} = \arg\min_{\beta}(\sum_{m=1} \log[K^2 + |w_m|^2(sgn(w_m)r_m{}^N - \beta)^2] \quad (29)$$

where $Q(\beta)$ is the objective function $Q(\beta)$ of the MMyF and can be expressed in (30).

$$Q(\beta) = \sum_{m=1} log[K^2 + |w_m|^2(sgn(w_m)r_m - \beta)^2 \quad (30)$$

where the weight $w(t)$ is matched to the impulse response of the equivalent channel response $h(t)$ in (3), hence, $w_m = h(mT - t)$ and $w \circ r$ represents the weighting operation, and K is the linearity parameter of the Myriad operator, which can be used to adjust the robustness of the estimator. Consider constant real-valued weights $\{w_m\}_{m=1}^{N}$. As $K \to 0$, MMyF filtering has mode-like behaviour, where it is sensitive to the local extrema. However, as the $K \to \infty$, the MMyF filtering behaves as a constrained linear FIR filter. Tunable linearity parameter K can provide appropriate resistance against the outliers for MMyF filtering by merely decreasing the linearity parameter K to a finite value.

Despite the excellent characteristics against outliers, (30) poses a challenge to solve the objective function $Q(\beta)$. The fixed-point search algorithm [18] has been employed to perform estimation continuously by using a sliding-widow approach to solve the above objective function. The output of the objective function $Q(\beta)$ represents the output of the MMyF filter that minimizes the distance of the received channel sequence $\boldsymbol{r}$ to the β at m^{th} instant with (31),

$$T(\beta) \triangleq \frac{\sum_{m=1}^{N} h(\beta)\, sgn(w_m)x_m}{\sum_{m=1}^{N} h_m(\beta)} \quad (31)$$

where sgn is the signum function to extract sign bit from weight at time index m , and $h_m(\beta)$ is defined as follow:

$$h_m(\beta) \triangleq \frac{2|w_m|}{K^2 + |w_m|\left(sgn(w_m)\cdot x_m - \beta\right)^2} \quad (32)$$

For practical application in the non-Gaussian channel, the linearity parameter K [7] can be empirically set to (33),

$$\mathrm{K} = \sqrt{\frac{\alpha}{2-\alpha}}\gamma^{1/\alpha} \quad (33)$$

Step 2 Sample the output of the MMyF $\beta_K(W,r)|_{k=1}^{N}$ vector with a free-running PLL to get received samples $\boldsymbol{y}$.

Step 3 Equalize the received vector $\boldsymbol{y_k}$ of dimension N, which is the size of sliding window N using MAP component equalizer with the following transition probability $\gamma_m(\dot{s}, s)$. Assume that the channel is memoryless, then the transition probability $\gamma_m(\dot{s}, s)$can be shown as in (34),

$$\gamma_m(\dot{s}, s) = C \cdot exp\left(-\frac{E_b}{2\sigma^2}\ (r_m - \sum_{l=1}^{N} \mathrm{d_l} \cdot h_{m-l})^2\right) \quad (34)$$

Where C is a constant, E_b is the bit energy, σ^2 is the variance or noise power, h is the sampled channel impulse response. Subsequently, this transition metric is used by MAP to calculate APP-LLRs $L(d_m|\boldsymbol{r})$. The APP-LLRs $L(d_m|\boldsymbol{r})$ that are produced by the component equalizers for estimation of the likelihood of the interleaved coded sequence d_m can be shown in (35).

$$L(\mathrm{d_m}|\mathbf{r}) = \ln\left(\frac{\sum_{(\dot{s},s)=>\ \mathrm{d_m}=+1} \alpha_{\mathrm{k}-1}(\dot{s})\,\gamma(\dot{s},s)\,\beta_{\mathrm{k}}(s)}{\sum_{(\dot{s},s)=>\ \mathrm{d_m}=-1} \alpha_{\mathrm{k}-1}(\dot{s})\,\gamma(\dot{s},s)\,\beta_{\mathrm{k}}(s)}\right) \quad (35)$$

The output of the component equalizer as given in (35) is the conditional LLR on the received sequence $\boldsymbol{r}$ in recursive form. $L(d_m|\boldsymbol{r})$ is subtracted from its a-priori LLRs $L(d_m)$ to generate the extrinsic LLRs $L_{ext}(d_m|\boldsymbol{y})$ as given in

$$L_{ext}(d_m|\boldsymbol{r}) = L(d_m|\boldsymbol{r}) - L(d_m)$$

The $L_{ext}(d_m|\boldsymbol{r})$ is de-interleaved to become the a-priori input $L(c_m|\boldsymbol{p})$for the component decoder.

Step 4 Decode the a-priori input $L(d_m|\boldsymbol{p})$ with the MAP component decoder with the use of the MAP algorithm from (15) – (23) to get the soft estimate of the coded bit $\lambda_m = L(d_m|\boldsymbol{p})$. For this case, the SISO observation input for the component decoder is grounded with LLRs $L(d_m|\boldsymbol{r}) = 0$ for all iterations.

$$\lambda_m = L(d_m|\boldsymbol{p}) = ln\left(\frac{\sum_{(\dot{s},s)=>\ d_m=+1} \alpha_{m-1}(\dot{s})\,\gamma(\dot{s},\ s)\,\beta_m(s)}{\sum_{(\dot{s},s)=>\ d_m=-1} \alpha_{m-1}(\dot{s})\,\gamma(\dot{s},\ s)\,\beta_m(s)}\right) \quad (36)$$

Step 5 Estimate the timing error estimate $\hat{\boldsymbol{\varepsilon}}_{\boldsymbol{m}}$. Before the finding of timing error estimate $\hat{\varepsilon}_m$, one needs to compute the soft estimate $\tilde{d}_m$ for the interleaved coded symbols.

$$\tilde{d}_m = \Pi(E[d_m|y_m]) = \Pi\left(tanh(\frac{\pi(\lambda_m)}{2\gamma})\right) \quad (37)$$

where Π is the interleaving operator. The timing error estimate $\hat{\varepsilon}_m$ can then be calculated with the substitution of the soft-estimate $\tilde{d}_m$ into M&M TED as in (38),

$$\hat{\epsilon}_m = \left(y_m\tilde{d}_{m-1} - y_{m-1}\tilde{d}_m\right) \quad (38)$$

where $y_m$and $y_{m-1}$are the received channel samples from the interpolator for the m^{th} and $(m-1)^{th}$ received symbols, whereas $\tilde{d}_{m-1}$ and $\tilde{d}_m$ are the estimates of the interleaved coded bits for time index $m-1$ and m respectively. The next timing estimate can be calculated from the time error estimate $\hat{\varepsilon}_m$ with the gradient descent algorithm as in (39),

$$\tau_{m+1} = \tau_m + \mu\hat{\varepsilon}_m \quad (39)$$

where m is the timing index, and μ is the step size.

Step 6 Interpolate the received samples with τ_m to get the new received sample r_{m+1} as shown in (40),

$$r_{m+1} = r((m+1)\,T + \tau_{m+1}) \quad (40)$$

Step 7 Repeat the steps from step 2 to step 6 for the next iteration.

6. RESULTS AND ANALYSIS

The following results were simulated with MATLAB simulation package. The square-root-raised-cosine pulse shape $h(t)$ is the equivalent channel impulse response. $h(t)$ is assumed to have roll-off factor α = 0.5. The bit energy to noise spectral density is $\frac{E_b}{N_0} = GSNR/2$.

The interleaved coded bits d_m were converted to polar form $\{\pm 1\}$. Consider the case where the interleaved coded bits d_m were corrupted by *SαS* noise with α = *2*, which is equivalent to AWGN noise. The first subplot in Figure-4 shows the signal waveforms from the output of the AWGN channel. The signal waveform is fed to the MF and MMyF filters at the receiver for filtering. The second subplot shows the output of the MF filtering, and the third subplot shows the output of MMyF filtering. We can observe that both MF and MMyF filters have comparable performance under the AWGN noise. The sampling strobes that were generated by the iterative timing recovery with M&M TED were overlaid with the waveforms in the subplots. It shows that in both cases under the influence of the AWGN noise, the iterative timing recovery can correctly track the received samples accurately regardless of the filters used.

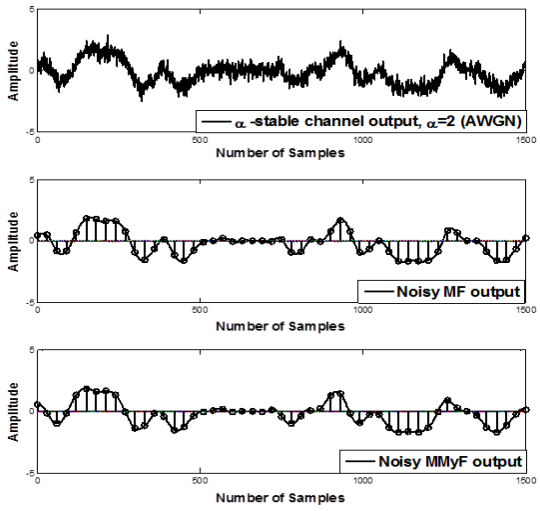

Figure-4. Input and output coded waveforms of the MF and MMyF with α = *2* (Gaussian distribution), E_b/N_0 = 7.

Figure-5 shows the case where α = *1* and the *SαS* random process is no longer Gaussian. Voltage spikes or outliers are present at the output of the *SαS* non-Gaussian channel. Assume that the outliers follow the Poisson-distribution with arrival rate λ. The second subplot of Figure-5 shows the output from the MF filtering, where occasionally, outliers can be observed from the channel outputs with varying amplitudes. The third-subplot shows that the outliers are effectively removed via MMyF filtering. The output waveform from the MMyF filter is smoothened as compared to the second subplot. As a result, the smoothened output waveform can provide more reliable inputs to the iterative timing recovery circuit to produce more reliable timing offset estimates.

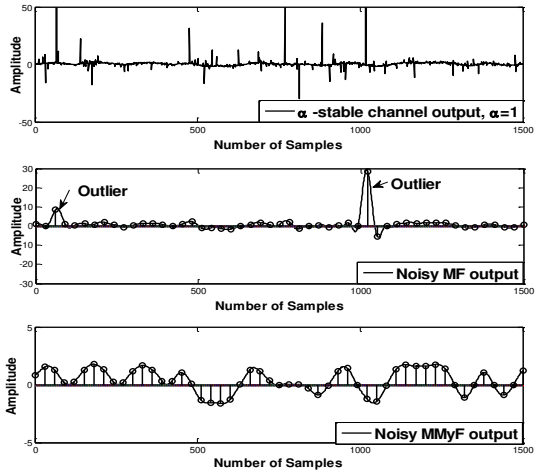

Figure-5. Input and output coded waveforms of the MF and MMyF in α = *1* (Cauchy distribution), E_b/N_0 = 7.

Figure-6 and Figure-7 show the normalised reliability plots of the output turbo equalizer. The reliability plots are used to indicate the convergence of the APP-LLRs from the output of the turbo equalizer through different numbers of iterations. In our case to ease of analysis, we use the normalised reliability value instead of directly using the APP-LLRs, as after several iterations, the output soft-output APP-LLRs of the turbo equalizer increase monolithically with more iterations performed. The reliability values in each bit position are formed by the 60 overlapping APP-LLRs $L(c_m|p)$ from the output of the de-interleaver from the turbo equalizer, which multiplied to its corresponding coded bits $c_m \in \{\pm 1\}$. The effect of reliability values can show the convergence of the turbo equalizer after a number of iterations with increasing reliability value for the correct decision. Hence, with the correct decoding soft-decisions $L(u_k|p)$, the reliability values formed as the vector product to its corresponding information bits can result in positive reliabilities whereas the negative reliabilities indicate the erroneous decoding soft-decisions $L(u_k|p)$. The reliability plots obtained in *SαS* impulsive noise ($\alpha = 1$) at $\frac{E_b}{N_0} = 5\,dB$ with MF filtering with 1.2% of timing error is given in Figure-5. We can observe that iterative timing recovery failed to converge after the 10th iteration due to the high degree of impulsiveness introduced to the transmitted coded bits.

The coded bits which are corrupted severely could move into the regions of other valid codewords to make the coded bits undecodable. The impulses from the hostile channel could not be suppressed by MF alone because its quadratic metric tends to place more emphasis on large amplitudes of errors in comparison to small amplitudes errors.

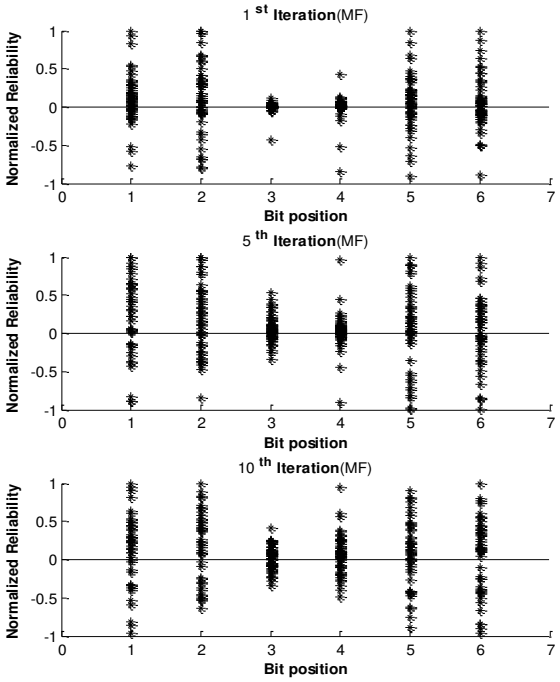

Figure-6. Reliability plots in *SαS* impulsive noise ($\alpha = 1$) at $\frac{E_b}{N_0} = 5\ dB$ with MF filtering.

As a result, the biased quadratic metrics that place a high degree of emphasis on the large amplitudes of the errors from the impulses could be detrimental to the iterative timing recovery. Hence, the quadratic metrics of the turbo equalizer could be problematic in the non-Gaussian channel without using MMyF.

Figure-7 shows the reliability plots of the turbo equalizer's output with the use of the effective impulse suppression MMyF framework. Initially, after the 1st iteration, turbo equalizer does not provide a noticeable reduction of the low confidence negative-valued reliabilities. As the number of iterations increases, the low confidence negative-valued reliabilities were improved with the increasing correctness from the output of the turbo equalizer as more errors were corrected. The performance improvement can be attributed to the capability of MMyF that is placed before the receiving end that effectively suppresses the outliers that were superimposed on the transmitted waveforms. Hence, the MMyF can effectively purge the outliers and safeguard the receiver and its timing recovery unit against the impulses from the *SαS* impulsive channel.

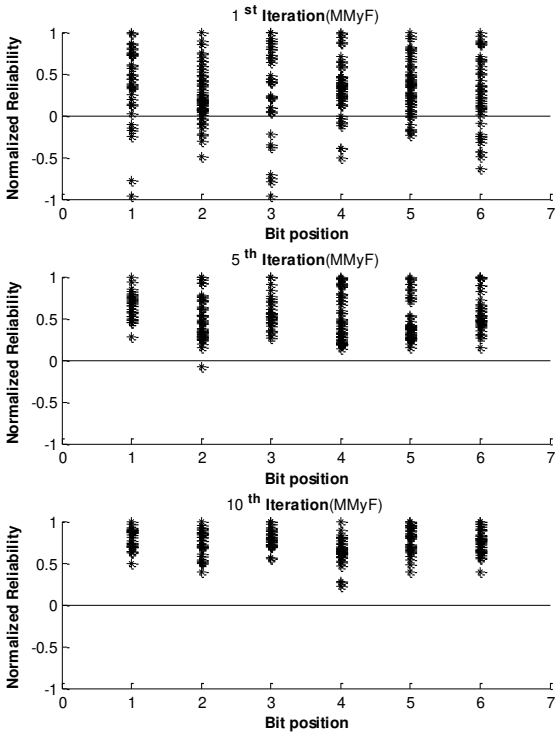

Figure-7. Reliability plots in *SαS* impulsive noise ($\alpha = 1$) at $\frac{E_b}{N_0} = 5\ dB$ with MMyF filtering.

The following simulations and its results are obtained by simulating the iterative timing recovery module with the parameters, as shown in Table-1:

Table-1. The parameters used in simulations.

No	Parameters	Values
1	PLL Gain *g*	0.05
2	Eb/N0 (dB) range	0 to 10 dB
3	Number of symbols used	1000 symbols
4	Bits per symbol	1000 bits
5	Transmitted waveform	Square root raised cosine pulse shape p(t)
6	Roll-off factor	0.5
7	MF or MMyF filtering span in the symbol duration	6
8	Timing error percentage	1.2%

Timing error percentage is defined as $\frac{\tau}{T_b}\% = 1.2\%$ where τ is the arbitrary timing delay introduced to the transmitted waveforms from the noisy channel, and T_b is the bit period.

Bit error rate (BER) allows us to measure the performance of the digital communication in terms of bit

error ratio (on vertical axe) against the independent variable bit energy to noise spectral density ratio $\frac{E_b}{N_0}(dB)$ (on horizontal axe). The BER is the measurement to determine the effectiveness of the particular ECC scheme in combating and correcting the errors of the transmitted signal over a noisy channel. Figure-8 shows that the turbo equalizer equipped with MF and MMyF filtering demonstrates identical BER performance over a noisy *SαS* impulsive noise ($\alpha = 2$) channel, which is equivalent to a Gaussian channel with no impulses introduced.

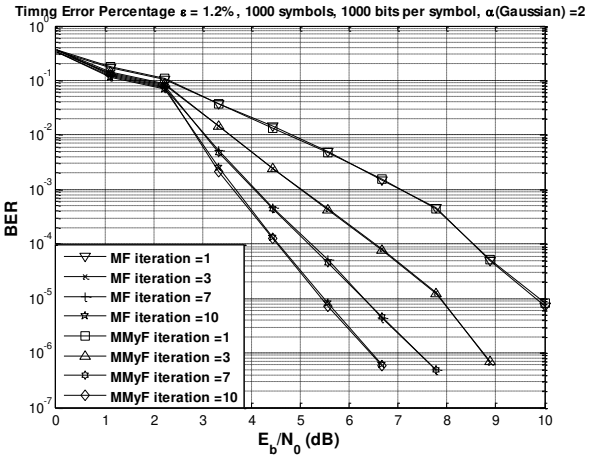

Figure-8. Bit error rate (BER) performance of iterative timing recovery in impulsive noise (α = 2) with timing error percentage = 1.2%.

Figure-9. shows the BER performance of turbo equalizer in *SαS* noise with $\alpha = 1.3$ which introduces a mild degree of impulsiveness to the *SαS* channel. Hence, the channel is becoming more hostile with the presence of outliers occasionally shown on the transmitted waveforms from the output of the channel.

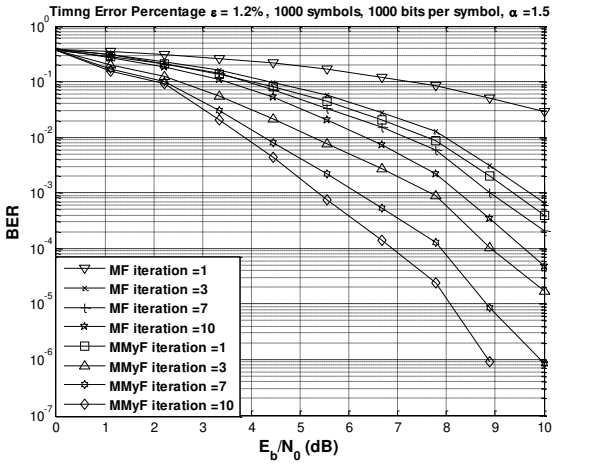

Figure-9. Bit error rate (BER) performance of iterative timing recovery in impulsive noise (α = 1.3) with timing error percentage = 1.2%.

The BER performance of both MF and MMyF are affected and deteriorates as more bits received were in burst errors. We can see that turbo equalizer with MF's BER performance is worse than that of turbo equalizer with MMyF by observing the BER. The deterioration in performance for MF is due to the presence of the outliers, which caused more erroneous decisions (APP-LLRs) are propagated between the component equalizer and component decoder. Increasing the number of iterations contributes little in performance gain for timing recovery with MF as front-end receiving filter. However, timing recovery with the MMyF at the receiving end can perform reasonably well and able to achieve $BER = 10^{-6}$ at $\frac{E_b}{N_0} = 9\ dB$ after the 10^{th} iteration.

Figure-10 shows the BER performance of both turbo equalizer with MF and MMyF filtering in *SαS* channel ($\alpha = 1$), which is equivalent to the Cauchy distribution channel. Significant amplitude-valued outliers with Poisson arrival are introduced to the transmitted waveforms. Because of the worsened channel environment with the frequent arrival of impulses with massive amplitudes, the turbo-equalizer failed to correct the error bursts as a result of the increasing hostility of the *SαS* channel. On the other hand, turbo equalizer with MMyF can still function and perform error corrections with increasing BER performance as the iterations increase. The robustness of using MMyF filtering framework is because the impulsive outliers are kept outside the receiver with the use of the MMyF filtering framework. Thus the received signals are in a more decodable form. It can be handled by the conventional iterative timing recovery scheme.

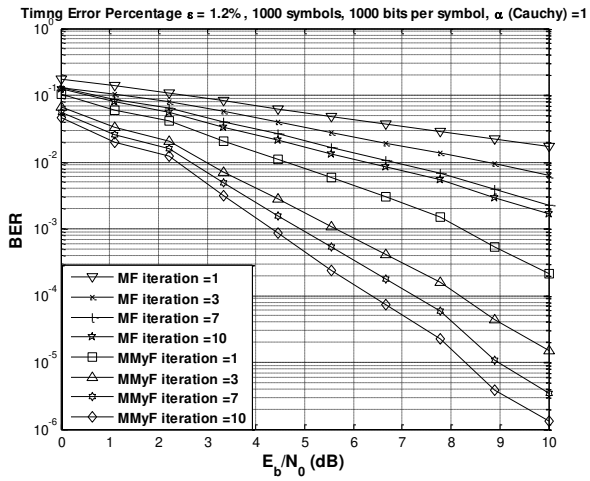

Figure-10. Bit error rate (BER) performance of iterative timing recovery in impulsive noise (α = 1) with timing error percentage = 1.2%.

7. CONCLUSIONS

As a conclusion, MMyF filtering framework can improve BER performance when used as a suppression technique for α–stable impulsive non-Gaussian channel. The performance of the conventional iterative timing recovery to operate over the non-Gaussian channel was studied. We proposed the MMyF framework to suppress the outliers from the non-Gaussian noise channel for conventional iterative timing recovery. The MMyF has been shown to significantly suppress the outliers from the *α*–stable channel at the receiver. Hence, the output from the MMyF filter can be sent to for iterative timing recovery. The received signals after removing outliers enable the timing recovery to better estimate the timing delay and provide accurate sampling strobes to sample the received signals via the interpolator. Therefore the proposed MMyF filtering framework that was incorporated into iterative timing recovery has been shown to demonstrate better performance over the impulsive *α*–stable channel compared to just the iterative timing recovery circuit alone. The suggested scheme is shown to retain the optimality and excellent performance as if it is operating over a Gaussian channel.

REFERENCES

[1] F. M. Gardner. 1986. A BPSK/QPSK Timing-Error Detector for Sampled Receivers, Commun. IEEE Trans. 34(5): 423-429.

[2] M. Zimmermann and K. Dostert. 2002. Analysis and modeling of impulsive noise in broad-band powerline communications, IEEE Trans. Electromagn. Compat. 44(1): 249-258.

[3] E. Panayirci. 1994. Symbol timing recovery in digital subscriber loops in the presence of residual echo and impulsive noise, in Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS ’94, 3: 45-48.

[4] S. Aghajeri, M. R. Gholami and H. Nikopoor. 2007. Blind sampling clock adjustment in OFDM power line communication, 9th International Symposium on Signal Processing and Its Applications. 1-4.

[5] G. R. Arce. 2005. Nonlinear Signal Processing: A Statistical Approach. Wiley.

[6] D. Middleton. 1999. Non-Gaussian noise models in signal processing for telecommunications: New methods and results for class A and class B noise models, IEEE Trans. Inf. Theory. 45(4): 1129-1149.

[7] C. L. Nikias and M. Shao. 1995. Signal processing with alpha-stable distributions and applications. Wiley-Interscience.

[8] J. R. Barry, A. Kavcic, S. W. McLaughlin, A. Nayak and W. Zeng.2004. Iterative timing recovery, IEEE Signal Process. Mag. 21(1): 89-102.

[9] R. Koetter, A. C. Singer and M. Tuchler.2004. Turbo equalization. IEEE Signal Process. Mag. 21(1): 67-80.

[10] T. C. T. H. L. Lajos Hanzo, B. L. Yeap, R. Y. S. Tee, Soon Xin Ng. 2001. Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart-Aided Near-Capacity Designs for Wireless Channels. Wiley-IEEE Press. 676.