EXPERIMENTAL VALIDATION OF AN ITERATIVE RECEIVER FOR ENERGY-EFFICIENT COMMUNICATIONS

Qasim Chaudhari (qchaudhari@unimelb.edu.au)[1] and Brian Krongold (bsk@unimelb.edu.au)[1]

[1]affiliation: Centre for Energy Efficient Telecommunications (CEET), Melbourne, VIC, Australia

ABSTRACT

Rising demand for wireless connectivity anywhere anytime has led the scientific community to focus on the energy efficiency of wireless networks. It has been realized that powerful error correcting codes like turbo and LDPC are making it possible to establish wireless communications at low SNRs. However, synchronization and channel estimation errors, which are inevitable at such SNRs, erode much of the achieved gains. Further expanding the turbo concept through an iterative receiver – which brings synchronization and equalization modules inside the loop – can help, but this solution is prohibitively complex and it is not clear what can and what cannot be a part of the iterative structure. This paper fills two important gaps in this field: (1) as compared to previous research which either focuses on a subset of the problem assuming perfect remaining parameters or is computationally too complex, we propose a proper partitioning of algorithm blocks in the iterative receiver for manageable delay and complexity, and (2) to the best of our knowledge, this is the first physical demonstration of an iterative receiver based on experimental radio hardware. We have found that for such a receiver to work, (1) iterative timing synchronization is impractical, iterative carrier synchronization can be avoided by using our proposed approach, while iterative channel estimation is essential, and (2) the SNR gains claimed in previous publications are validated in indoor channels.

1. INTRODUCTION

Wireless is the least energy efficient medium due to its unguided nature. Increasing customer demand and the ensuing carbon footprint put conflicting demands on the design of modern wireless communication systems. As these systems achieve higher rates, the need to optimize their energy efficiency makes better economical sense and therefore, research is currently being driven towards green communications in wireless perspective without compromising the data rates.

To improve energy efficiency, we want to reduce the transmit power as the power amplifier is very inefficient and a key contributor to the wasted energy. The negative result is then a relatively low SNR at the receiver.

The physical layer of a receiver system consists of three major parts, namely the frontend, inner receiver, and outer receiver. The job of the inner receiver is to estimate and compensate for gain control, synchronization (timing, carrier frequency and phase), channel estimation and equalization through application of DSP and estimation theory, and present almost perfect symbols perturbed by additive white Gaussian noise only to the decoder. Conventional methods to acquire these signal parameters are

Data-aided that exploit a training sequence and hence are more accurate but less spectrally efficient

Decision-directed that utilize previous decisions and hence suffer from error propagation

Non-data-aided that are spectrally efficient but suffer from slow convergence.

At low SNRs, acquisition is difficult as the signal is not clearly differentiated from the noise. In this regime, data-aided algorithms prove costly for bandwidth, decision-directed approach is unreliable and non-data-aided techniques simply fail to work. It seems that the only way to reduce the noise is significant averaging, which leads to the idea that soft decisions from decoders can be utilized to refine the unknown nuisance parameters. The tradeoffs to manage in this case are computational complexity and feedback delays.

The basic principle of iterative receiver – whether in the form of iterative synchronization or iterative channel estimation – lies in squeezing the maximum amount of information from the received samples before their quantization in any form. Viterbi, in his famous 1991 paper [26], summarized the essence of Shannon theory for digital communications in the form of three lessons. The first lesson was "Never discard information prematurely that may be useful in making a decision until after all decisions related to that information have been completed". The applications of this lesson lead to soft-decision decoding in error control codes, maximum-likelihood sequence estimation in high-quality wireline modems and partial-response maximum-likelihood detection in magnetic recording. The concept of utilizing the error correction capability of channel coding to refine unknown parameters is just another manifestation of the same principle.

It is well known that at high data-rates, reduced symbol duration results in an increased number of equalizer taps, which proves to be the bottleneck for communication at that rate. Therefore, most modern wireless communication systems have

Table 1: A summary of iterative receiver contributions

Iterative Processing Scope	References
Channel estimation	[1], [2], [3], [4], [5], [6], [7]
Carrier phase synchronization	[8], [9], [10], [11]
Timing synchronization	[12], [13], [14], [15]
Joint carrier phase and frequency synchronization	[16], [17]
Joint carrier phase and timing synchronization	[18], [19]
Joint carrier frequency and channel estimation	[20], [21], [22], [23], [24]
Joint frame detection, carrier frequency and channel estimation	[25]

adopted multi-carrier techniques, particulary orthogonal frequency division multiplexing (OFDM), for their physical layer operation. In time domain, multi-carrier systems utilize symbol parallelism to increase the symbol duration and reduce the effective channel length, thus greatly simplifying the equalizer design. In frequency domain, this amounts to dividing a highly frequency-selective channel into a fixed number of frequency flat subchannels, each requiring only a single-tap equalizer.

Code-aided synchronization particularly suits OFDM systems as Intersymbol interference (ISI) and inter-carrier interference (ICI) arise due to inaccurate timing and frequency estimates. The knowledge of channel estimates is also essential for coherent detection. Sensitivity of OFDM receivers to synchronization errors is a thoroughly researched topic [27].

2. RELATED WORK AND CONTRIBUTIONS

Past work in this field focuses on one of the subsets shown in Table 1 where the scope describes the receiver components included in iterative processing. Note that only a few important references for each category are included in the above table due to bibliography constraints. As is evident, some publications deal with iterative channel estimation while others target iterative timing, phase or frequency synchronization. Joint techniques of iterative synchronization (mainly carrier frequency) and channel estimation have also been investigated.

The major focus is on constructing soft symbols from the log-likelihood ratios provided by the decoder while introducing some minor variations. For example, [6], [8] and [9] utilize the soft estimates of symbols without any quantization while [2] also considers the hard mapping for the next iteration. In most of the publications (e.g., [2], [5]), the channel estimate is found through the least squares technique. However, [6] and [7] consider the mean of the channel frequency response instead of forming the mean of the data symbol. There are other techniques based on a threshold test and weighted sum of new and old channel estimates.

In summary, almost all of the research is based on individually useful schemes but not on how they fit together within a larger framework. The only comprehensive solution taking into account frame detection, carrier frequency synchronization and channel estimation has been proposed in [25] which is prohibitively complex for a practical implementation. In addition, all the publications on this topic are based solely on simulation results and an experimental demonstration is necessary to validate them, as quoted by Dr. Klein Gilhousen, co-Founder of Qualcomm Inc., "If you haven't tested it, it doesn't work".

The main contributions of this paper are as follows.

- To the best of our knowledge, this is the first published physical demonstration of an iterative receiver structure using experimental hardware. We use Universal Software Radio Peripheral (USRP) B210 boards by Ettus Research as our hardware platform.
- We take into account the simultaneous effects of all impairments including timing synchronization, carrier frequency synchronization, channel estimation and equalization, and propose a practical algorithm partition in an iterative setup.
- We propose an alternative approach for carrier frequency synchronization that does not require embedded pilots and eliminates the need of bringing the frequency correction within the iterative loop. In addition, we propose a change in iterative processing update structure that results in slightly improved performance for PSK modulations.

Remember that our use of the term *iterative receiver* involves the utilization of log-likelihood ratios (LLRs) to form soft estimates of the transmitted symbols. This is different than a *turbo receiver*, in which equalization and decoding tasks are repeated on the received data in such a way that the feedback information from the decoder is incorporated into the equalization process and vice versa.

3. SYSTEM MODEL

The transmitter block diagram is shown in Fig. 1 and works as follows:

An information segment of length B bits is input to a turbo-encoder of rate r with two recursive systematic convolutional (RSC) encoders. The encoded bit stream is subsequently bit interleaved by a pseudo-random interleaver to make the coded bits approximately independent from each other, as needed in the decoding stages at the receiver. This bit stream is QPSK modulated with a Gray-coded symbol mapping. This symbol stream

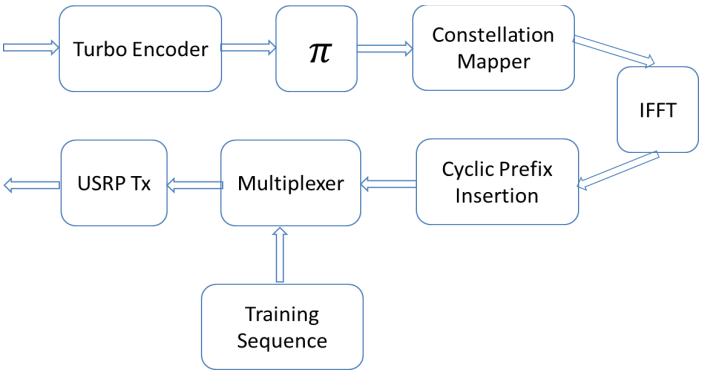

Figure 1: Block diagram of the transmitter

is grouped into segments of N_c subcarriers and an N_c-point inverse Discrete Fourier Transform (IDFT) is performed to form the time domain signal. A few subcarriers, N_g, are left as guards on both edges of the band to ease the filtering requirements. Afterwards, the last N_{cp} samples of an OFDM segment are inserted as a cyclic prefix at the start to avoid intersymbol interference (ISI) and to convert the linear convolution between the transmitted signal and the channel to circular convolution. The length of the cyclic prefix, N_{cp}, is chosen to be larger than the delay spread of the channel. Next, the resulting symbols are multiplexed with a training sequence T_k of length N_p required for the acquisition stage of the receiver. The training sequence is similar to IEEE 802.11a systems with two identical halves. Finally, N such segments are assembled to form a complete OFDM frame and the samples are sent to the USRP B210 hardware through Universal Hardware Driver (UHD), which provides a range of Application Programming Interfaces (APIs) for interfacing the host computer with USRP boards.

Thus, the relationship between the signal to noise ratio and E_b/N_o is given by $E_s/N_o = 2rE_b/N_o$. The signal is transmitted over a frequency selective channel given by

$$h(t) = \sum_l \alpha_l \delta(t - \tau_l)$$

where α_l is the complex amplitude distributed as $\mathcal{CN}(0, \sigma_l^2)$ and τ_l is the delay of l^{th} tap with the maximum delay less than N_{cp} samples. We assume a quasi-static block-fading channel that remains constant during each frame, but varies independently for each subsequent frame.

Considering the above system model, the sampled received signal collected from the USRP hardware through UHD is given as

$$y(n) = e^{j2\pi\epsilon n/N_c} \sum_{l=0}^{L-1} h(l)x(n-l) + w(n) \qquad (1)$$

where ϵ is the carrier frequency offset (CFO) between the transmitter and receiver (normalized by the symbol rate), $h(n)$ is the sampled channel response with L number of taps and $x(n)$ is the transmitted signal.

For each OFDM symbol, the received signal Y_k after taking the DFT at the receiver can be written as

$$Y_k = \frac{sin(\pi\epsilon)}{N_c\, sin(\pi\epsilon/N_c)} e^{j\pi\epsilon(N_c-1)/N_c} H_k X_k + \text{ICI}(k) + W_k \qquad (2)$$

where $k = 0, 1, \cdots, N_c - 1$ is the subcarrier index, X_k is the transmitted symbol, $H_k = \sum_{l=0}^{L-1} h(l) e^{-j2\pi lk/N_c}$ is the channel transfer function at subcarrier k, and W_k is the additive white Gaussian noise, respectively, all in frequency domain. The inter-carrier interference term, ICI(k), term is given by

$$\text{ICI}(k) = \frac{1}{N_c} \sum_{i=0}^{N_c-1} \sum_{m \neq k} H_m X_m e^{j2\pi \frac{i(m-k+\epsilon)}{N_c}} \qquad (3)$$

The above equations assume a normalized CFO less than 0.5, which is achieved through coarse frequency acquisition.

4. PROPOSED RECEIVER STRUCTURE

There are significant challenges to be overcome for the practical implementation of an iterative receiver. We define three such criteria here:

1. The computational complexity of the overall structure should not be prohibitively complex.
2. Any operation inside the iterative loop has to be simple. The decoder, for example, cannot be made a part of any phase-locked loop (PLL), whether time or phase, due to the effect of its transport delay. This is because longer feedback loops may allow the phase to change by a significant amount before the correction is applied, and hence require wider loop bandwidth which in turn results in more noise entering the system to corrupt the estimates.
3. The convergence to global optima is not guaranteed in iterative algorithms, which makes them very sensitive to their initialization point.

Based on these criteria, we partition the algorithm blocks for manageable delay and complexity in the following way. We can

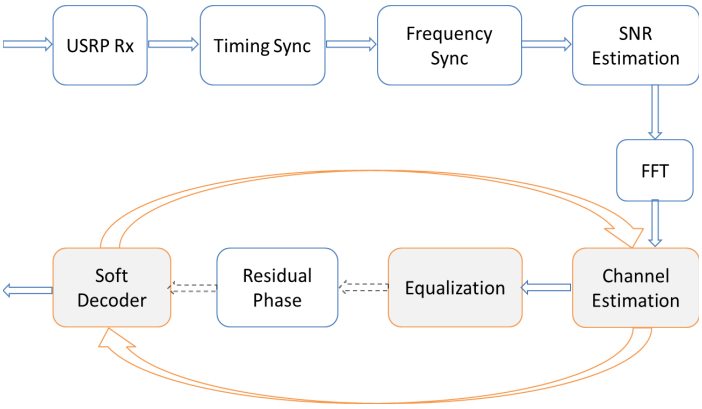

Figure 2: Block diagram of the receiver.

conclude that timing synchronization cannot be done iteratively because repeated filtering and interpolation in single-carrier systems and repeated FFT operations in multi-carrier systems violate both criteria 1 and 2 above (though it does make sense in systems requiring offline processing).

On the other hand, frequency synchronization can be included within the iterative framework. However, requiring the soft symbols to help in estimating the frequency offset constrain the capture range within a very narrow margin. This can be understood from the fact that the frequency offset must be small enough to change the phase within acceptable limits, say $\pi/4$, at the far end of the frame. Later, we propose a two-step solution to correct this CFO in a manner that can be kept outside the iterative section of the system. Finally, iterative phase synchronization is automatically incorporated as phase is just a part of the channel.

This leaves the channel estimator and equalizer embedded within the turbo decoder. Due to their simplicity, frequency-domain equalizers are better suited to this kind of framework. As channel estimates become more reliable with each iteration, updating them can be stopped after the first few iterations while the decoder continues. In order to satisfy criterion 3 above, conventional data-aided techniques can be used at the start with the help of a training sequence.

The overall receiver operation is shown in Fig. 2 and is described as follows. First, the starting sample of the receive sequence is identified through a frame synchronization algorithm (or coarse timing in OFDM), which also helps determine the coarse frequency offset. Next, fine timing and fine frequency are estimated followed by SNR estimation. After stripping the cyclic prefix and taking the DFT, channel estimation is performed in the frequency domain, through which the received signal is equalized. Before the first iteration, any residual phase offset arising from remaining frequency offset is removed, which is necessary for relatively long block lengths of iterative decoders. Finally, the iterations between the soft decoder, channel estimation and equalizer are executed as shown in Fig. 2, where the dotted line represents the residual CFO correction being performed only once before the first iteration and the shaded boxes represent the modules participating in the iterative framework.

5. ITERATIVE RECEIVER ALGORITHMS

Below, we give a short description of the receiver algorithms starting with the synchronizer for signal acquisition. Synchronization includes frame detection (equivalent to coarse timing acquisition in OFDM systems), coarse frequency synchronization, fine timing recovery and fine frequency adjustment.

5.1. Coarse Synchronization

Our frame detection mechanism is based on [28], which utilizes two autocorrelators to exploit the periodic structure of the training sequence. Based on eq. (1), an autocorrelator metric $J(m)$ can be constructed as

$$\begin{aligned} J(m) &= \sum_{n=0}^{N_c-1} y(n-m)\, y^*(n-m-N_w) \\ &= e^{j2\pi\epsilon N_w/N_c} \times (\texttt{Magnitude Term}) \end{aligned} \quad (4)$$

where N_w is the window length of operation chosen equal to at least the minimum repetition length of the training sequence [1], the phase term arises due to similar complex samples in training sequence but with the normalized CFO adding a phase difference, and the magnitude term is a function of sequence alignment, ICI and their cross-terms. Now, the particular shape of the plateau can be used to identify the starting point of a frame in terms of a differentiator defined by

$$J'(m) = |J(m)|^2 - |J(m - N_c/2)|^2 \quad (5)$$

The differentiator results in a metric that is slowly rising until the start of the frame and then falling with the same slope. As described in [28] and verified in our experiments, the Automatic Gain Control (AGC) severely breaks the peak detection algorithms. Hence, this algorithm uses an instantaneous peak

[1] Most wireless OFDM systems have a preamble with several repetitions

detector and a group peak detector in addition to the differentiator. The instantaneous peak detector is basically a combination of a comparator and a counter operating sample-wise, while the group peak detector is also a comparator used to detect the falling edges in the differentiator. However, as the name implies, the input signal is accumulated in groups of six samples and the present group is compared with the previous one effectively filtering the differentiator output with a moving average filter.

We found that the above algorithm alone was not sufficient to ensure proper frame detection because the receiver – which is just collecting noise samples – on rare occasions shows unipolar spikes within the bipolar noise. This might be due to rebalancing or overall change in gain along the RF and baseband path, causing a change in DC offset that is then zeroed out by DC offset compensation or highpass response. These one-sided samples have extremely high correlation and hence destroy all correlation-based frame detection mechanisms. This problem can be resolved through discarding the false flag raised by a very high correlation. We avoided this problem by not utilizing a data set within which such an anomaly occurs.

In general, an initial coarse frequency estimate can be formed at this point by observing the phase of the coarse timing reference. However, as we operated at relatively lower SNR, the coarse frequency estimate is unreliable to the extent that it need not be estimated at all, and instead rely on the cross-correlation in the next section for a finer timing estimate, and corresponding finer CFO estimate.

5.2. Fine Synchronization

For fine timing synchronization, a cross-correlation is computed between the stored noiseless training sequence and the received sequence adjusted by the coarse frequency estimate, starting a few samples before the frame boundary estimate and ending a few samples after it. There are two reasons for not using such a cross-correlation in the first stage of the receiver. First, the received sequence is distorted by the CFO which is absent in the locally stored copy of the training sequence, and hence cross-correlation does not give actual results. Second, auto-correlation can be implemented much faster compared to cross-correlation as it can be computed iteratively by discarding one sample of the first cross-product and adding one sample of the last cross-product.

The fine timing estimate can be offset by a few samples due to the channel convolution. Therefore, actual frame start is taken some samples before the estimated marker as there is a safety margin to the left of the OFDM symbol boundary due to the cyclic prefix. This offset later appears as a phase shift after the DFT and becomes part of the unknown channel. A fine frequency offset is also calculated based on this absolute timing reference.

Having known the boundaries, the received training sequence is separated from the data part. A CFO offset estimate can now be generated by using the two repetitive portions of this sequence. From eq. (1), the CFO can be straightforwardly estimated as

$$\hat{\epsilon} = \frac{N_c}{2\pi N_w} \arctan\{J(\hat{m})\} \tag{6}$$

In the above equation, $\hat{m}$ is the starting sample of the sequence. For our purpose, we keep N_w equal to N_c as our training contains two repetitions of the same sequence.

5.3. SNR Estimation

Next, the SNR is estimated based on [29], where the signal power is calculated from I and Q samples of the received training as

$$\hat{P}_S = \frac{1}{N_c/2} \sum_{n=0}^{N_c/2-1} \Big[y_I(n) y_I(n+N_c/2) + y_Q(n) y_Q(n+N_c/2) \Big]$$

Similarly, the noise power can be written as

$$\hat{P}_W = \frac{1}{2N_c/2} \sum_{n=0}^{N_c/2-1} \Big[\{y_I(n) - y_I(n+N_c/2)\}^2 + \{y_Q(n) - y_Q(n+N_c/2)\}^2 \Big]$$

The SNR estimate is then given by $\hat{P}_S/\hat{P}_W$. Separate estimates of signal and noise power are also required as an input to the turbo decoder.

5.4. Channel Estimation

At the first pass, the standard channel estimates are given using the two repetitive portions of the received training as

$$\hat{H}_k = \frac{0.5(Y_k + Y_{k+N_c/2})}{T_k} \tag{7}$$

The subsequent data symbols in frequency domain are equalized by dividing them with their respective channel estimates and the resulting signal is input to the turbo decoder.

5.5. Residual CFO Tracking

At this stage, there is still some frequency offset left due to slight mismatch between the actual and estimated fine frequency offset. There are two reasons why this residual CFO is crucial in our scenario. First, the signal at low SNR does not produce better frequency estimates on average as it suffers from the well-known threshold effects. Second, even if the estimated CFO is close to the actual value, long block lengths – as in the case of iterative decoders – introduce enough phase shift across the frame to render further turbo processing meaningless.

Reference [30] details the importance of an accurate initial estimate for such a purpose. It was shown that the data-aided

synchronizer followed by an iterative soft decision directed estimator performs very closely to the ideal behaviour but at an unacceptable expense of 34% overhead training length. Reducing the length of the training sequence any further causes a serious degradation for such an estimator. As an alternative, the authors incorporated a chain of data-aided, non-data-aided and then soft decision-directed carrier frequency acquisition in an iterative loop which improves the performance if initialized through a training sequence equal to 20% of the frame length.

To save power and bandwidth, we implemented a different solution that exploits the rotation of constellation symbols. In multi-carrier systems or single-carrier systems with frequency-domain equalization, the actual signal is already divided into multiple segments. For conventional single-carrier systems, the received signal can be partitioned into several segments for such a purpose. As explained in [31], a frequency offset not only adds an ICI term to the received symbols due to sampling in frequency at incorrect instants, but also rotates the frequency domain symbols by a time-variant phasor $e^{j2\pi\epsilon[n(N_c+N_{cp})+N_{cp}]/N_c}$, where n is the OFDM symbol index. Therefore, the phase increment from one OFDM symbol to the next is given by the angle

$$\theta_F = 2\pi\epsilon\frac{(N_c+N_{cp})+N_{cp}}{N_c} \tag{8}$$

At high SNR, these subcarrier symbol rotations caused by residual local offsets are usually small enough to be tracked by the channel estimator. However, this is not the case at low SNR. Instead of recomputing the frequency offset that has already been refined, we exploit this property by utilizing a non-data-aided Viterbi and Viterbi phase estimator on segment-by-segment basis.

$$\begin{aligned}\gamma &= |Y_k|^2 e^{j4\angle Y_k} \\ \hat{\theta}_F &= -\frac{1}{4}\angle\sum\{\Re(\gamma)+j\Im(\gamma)\}\end{aligned} \tag{9}$$

This feedforward estimation implies that the observation intervals should be short enough to incur not a large phase shift in frequency domain symbols but long enough to keep the computational complexity under control. We compared our results for each segment length equal to (a) a single OFDM symbol and (b) two OFDM symbols, and found little difference between their performance.

Since the estimated phase offset is non-data-aided for QPSK modulation, it is restricted to the interval $\pi/4 \le \hat{\theta}_F \le \pi/4$. To remove this modulo $\pi/2$ operation to track the phase shift, it is necessary to unwrap the estimates by

$$\tilde{\theta}(n+1) = \tilde{\theta}(n) + \text{mod}\left(\hat{\theta}_F(n) - \tilde{\theta}(n) - \pi/4, \pi/2\right) - \pi/4$$

This method of removing the effect of residual CFO not only eliminates the need of bringing the frequency correction within the iterative loop, but also shields against a large residual CFO. The reason is that for the residual CFO corection to be inside the iterative loop, it has to be small enough such that the decoder still converges to approximately true soft symbols, which is not required in our approach. Finally, this solution requires no extra training or pilot overhead and can be implemented as a non-data-aided approach even at low SNRs.

5.6. Iterative Processing

In keeping the receiver structure simple, we have left only channel estimation and equalization inside the iterative loop as shown in Fig. 2. The soft decoder computes the LLR of both the information and the parity bits $L(a_i)^{(q)}$ and $L(b_i)^{(q)}$, where the subscript q denotes the iteration number, and a_i and b_i represent the two bits Gray-mapped onto the i^{th} QPSK symbol. The LLR of each individual bit is obtained by

$$\text{LLR}(a_i) = \log\frac{\sum_{S_{a_i}\in S_0} e^{-|Y_k - H_k S_{a_i}|^2/\hat{P}_W}}{\sum_{S_{a_i}\in S_1} e^{-|Y_k - H_k S_{a_i}|^2/\hat{P}_W}} \tag{10}$$

where $\hat{P}_W$ is the estimated noise power, and S_0 and S_1 are the set of symbols in the original QPSK constellation corresponding to the bit a_i being 0 and 1, respectively. Then, the soft estimates of received symbols are computed through

$$\begin{aligned}\hat{X}_k^{(q)} &= \sum_{a_i,b_i} P_{a_i}^{(q)} P_{b_i}^{(q)} Z_{a_i,b_i} \\ &= \tanh\left(\text{LLR}(a_i^{(q)}/2)\right) + j\tanh\left(\text{LLR}(b_i^{(q)}/2)\right)\end{aligned} \tag{11}$$

where P_{a_i} and P_{b_i} are the a posteriori probabilities of bits a_i and b_i obtained from the decoder for the i^{th} symbol and Z_{a_i,b_i} is the corresponding constellation symbol. Note that the presence of the interleaver has allowed us to write this symbol probability as a product of individual bit probabilities.

Using the updated soft symbol, the channel estimate at the k^{th} subcarrier can be updated as

$$\hat{H}_k^{(q)} = \frac{Y_k}{\hat{X}_k^{(q)}}$$

We propose an improvement in the iterative update structure in the following manner: as zero-forcing and MMSE equalization enhance the noise excessively (former more than the latter), better performance can be gained for a constant modulus constellation like QPSK by keeping the amplitudes untouched while correcting for the phase offset in every iteration:

$$\hat{Y}_k^{(q+1)} = Y_k \cdot e^{-j\angle\hat{H}_k^{(q)}}.$$

With this kind of update, we found improvement in block error rates during our simulations at an order of 0.5 dB with the exact figure depending on the channel.

As our experiments were performed indoors in a static environment, the normalized fade rate of the channel is slow. Following the arguments in [2], the complexity of Wiener filtering can

be avoided by using a moving average filter with equal weights. More importantly, this does not require the knowledge of fade or autocorrelation of the channel.

For reducing computational complexity, the channel updates and subsequent equalization can be stopped either after a fixed number of iterations, or after reaching some predetermined criterion for convergence. In our experiments, we adopted the former approach by stopping the updates after a fixed number of iterations.

6. PERFORMANCE EVALUATION

A widely adopted hardware platform for Software Defined Radio (SDR) experimentation in the academic community is the Universal Software Radio Peripheral (USRP) by Ettus Research. In a typical application, baseband signal processing algorithms run in software on a host computer, while the analog and digital frontends are implemented on USRP and connect to the host computer through a high-speed link. The open source UHD driver is used to interface with the USRP making it almost a plug-and-play solution, while the GNU Radio software suite is employed as the processing engine to create software-defined radio systems.

We have used the USRP B210 boards for our experiments. USRP B210 is a single-board platform with frequency coverage from 70 MHz to 6 GHz. It combines a single chip direct-conversion transceiver using Analog Devices AD9361 RFIC, MIMO (2 Tx and 2 Rx) operation with up to 56 MHz of real-time bandwidth (61.44 MS/s quadrature), a reprogrammable Spartan6 XC6SLX150 FPGA, and fast USB 3.0 connectivity.

As shown in the experimental setup of Fig. 3, one desktop and one laptop were used to connect to their respective USRP B210s and run the experiments. The USRP with the desktop was configured as the transmitter while the USRP with the laptop ran the receiver. The ISM band at 2.4 GHz was used for the transmission with a bandwidth of 8 MHz. The transmitter and receiver block diagrams are shown in Fig. 1 and Fig. 2 respectively, where the main parameters are set according to Table 2. Since the turbo decoding algorithms are computationally complex and cannot be implemented in realtime on a general purpose processor, we have implemented the baseband signal processing in Matlab through offline processing instead of using GNU Radio libraries.

The complex baseband time samples are generated in Matlab on the transmitter side and stored in a file. A GNU Radio interface is run to send these samples to the USRP hardware through UHD. The hardware performs Digital Up-Conversion (DUC) to the specified sample rate, while the amplification and conversion to the carrier frequency are performed by the analog frontend. On the receiver side, the above process is reversed where the RF frontend amplifies and downconverts the received signal which is sampled and decimated by the Digital Down-Conversion (DDC). The UHD is again responsible for feeding the received samples from the hardware to the PC through GNU Radio and stored in a file that is then read in Matlab for offline baseband processing .

There are two types of receiver implementations in our system:

1. Iterative receiver that implements the iterative processing framework discussed above, and
2. Conventional receiver that uses the same acquisition algorithms followed by turbo decoding without any feedback to the channel estimation and equalization units.

All the other parameters are kept the same for both receivers including the number of turbo decoder iterations which is set to 8 in each case (although the parameter update in iterative receiver is for the first 5 iterations).

To validate the theory of iterative receiver design, one target of this research was to match the gains in BLock Error Rate (BLER) curves against the simulation results generated by us and other publications. The simulation as well as experimental parameters are mentioned in Table 2. As mentioned above, the channel is re-estimated at the start of each OFDM frame and the SNR is computed using the technique in Section 5.3.

Table 2: System parameters used in the experiments

Parameter	Value
Carrier frequency	2.4 GHz
Bandwidth	8 MHz
Information segment size, B	1502 bits
Code rate, r	1/2, 1/3
Modulation, M	QPSK
RSC generator	$(13)_8, (15)_8$
DFT size, N_c	64
Cyclic prefix N_{cp}	16
No. of iterations	8
Training sequence length, N_p	128
Pilots/symbol	0
Decoding algorithm	MAP

The problem of residual CFO basically divides this issue into two sections:

1. As explained in Section 5, the estimated frequency offset at low SNRs is highly unreliable and rotates the received frequency-domain symbols by effectively random offsets throughout the frame. No amount of turbo decoding or iterative channel estimation then is enough to decode the frame correctly resulting in complete loss of a block.

   Since the purpose of this first experiment was to compare the effect of residual CFO only, we placed the transmitter and the receiver in relatively close proximity with each other in order to reduce the effect of the channel. Multipath propagation was still there due to the indoor office environment, but it was a Rician channel instead of a Rayleigh one.

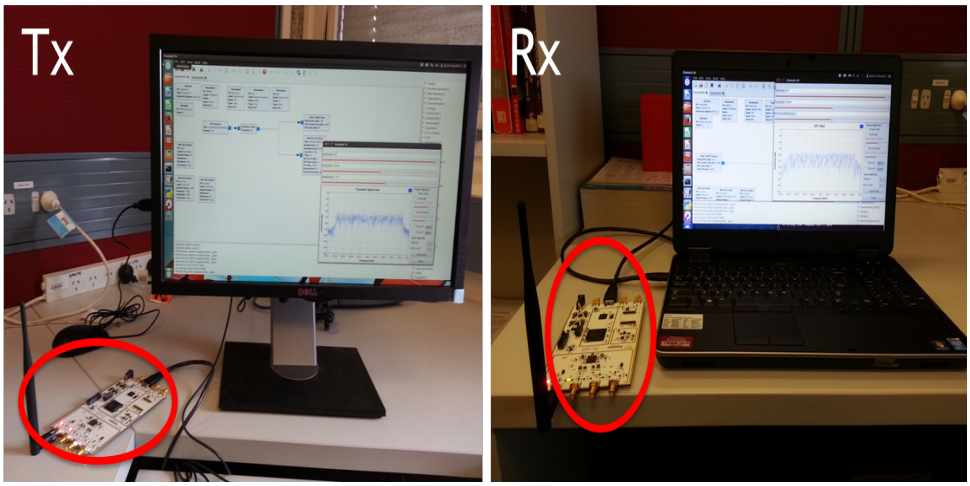

Figure 3: Experimental setup with USRP B210s

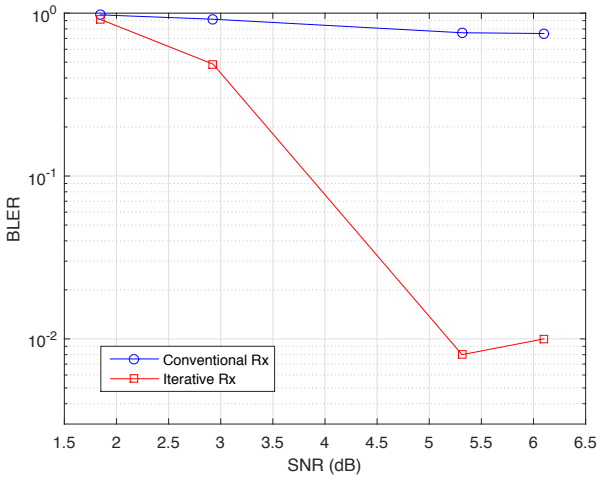

Figure 4: Block error rate with residual CFO tracking in iterative Rx

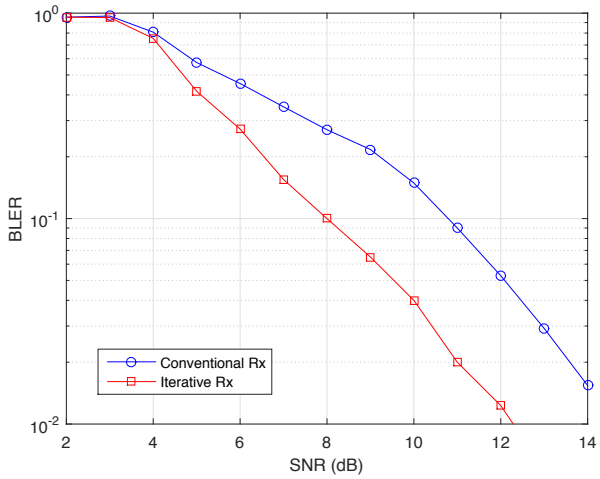

Figure 5: Simulation BLER with residual CFO tracking in both Rx

The BLER for code rate 1/2 is shown in Fig. 4 where the iterative receiver performs significantly better in the low-SNR regime. The curves for code rate 1/3 follow a very similar pattern (not shown here). Observe that the gap is more than 2 dB at operating SNRs of 3-5 dB. After a particular SNR, the CFO estimate should get better and the blue curve should fall with a similar slope as in red curve. The reason why we did not project the conventional receiver BLER for further SNRs is that CFO estimate is a random variable and for this lower SNR range, even a few poor estimates destroy the overall accuracy, particularly more so for receivers with iterative decoders.

2. Next, we plot simulation results as shown in Fig. 5 while catering for the residual CFO in both receivers. To get a rough idea of a bad scenario, the channel was set as $h = [0.407 \quad 0.815 \quad 0.407]$ which is the widely known Proakis channel B. It has been extensively used in communications and iterative receiver literature due to its high frequency selectivity. It is evident that the error rate curves exhibit no gap at the start for very low SNR, then gradually diverge from each other and converge at approximately 2 dB apart for medium to high SNR.

Subsequently, we set the receiver further apart by placing it in an open-door extension of the hall. By taking care of the residual CFO in both receivers through the technique in Section 5.5., we get the BLER shown in Fig. 6. The gap of 1.5-2 dB between the conventional and iterative receiver is in line with the published research and confirms the simulation results from many papers in Table 1. Minor differences arise due to a number of factors such as the simulated channel profile, constraint length at the encoder,

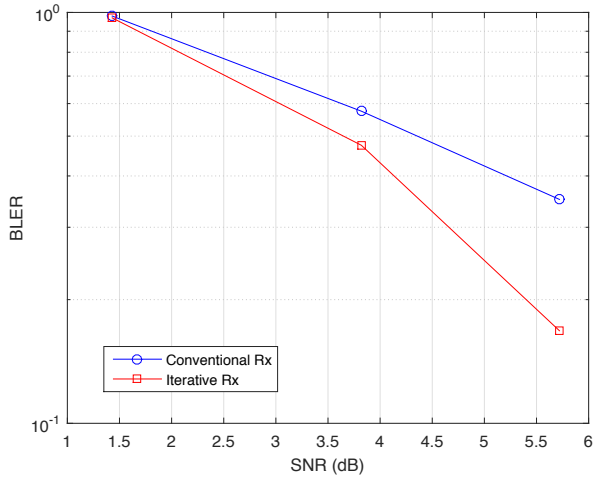

Figure 6: Experimental BLER with residual CFO tracking in both Rx

the generator polynomials and number of iterations at the decoder to name a few.

7. CONCLUSIONS

In this paper, we proposed a practical framework for the implementation of iterative receivers in wireless communication systems through proper partitioning of algorithm blocks for manageable delay and complexity. We also proposed a technique to compensate for the residual frequency offset as well as an improved iterative processing update structure. Finally, we demonstrated experimental results for our implementation on USRP B210 software defined radios which, according to our knowledge, is the first experimental verification of an iterative receiver. It was found that the SNR gains achieved in indoor channels matched closely with the previous simulation-based research.

REFERENCES

[1] Y. Liu, Z. Tan, H. Hu, L. Cimini, and G. Ye, "Channel estimation for ofdm," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, 2014.

[2] M. Valenti and B. Woerner, "Iterative channel estimation and decoding of pilot symbol assisted turbo codes over flat-fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 9, 2001.

[3] H. Wymeersch, F. Simoens, H. Steendam, and M. Moeneclaey, "Code-aided channel tracking for ofdm," in *4th International Symposium on Turbo Codes Related Topics*, 2006.

[4] H. Hafez, Y. Fahmy, and M. Khairy, "Iterative channel estimation and turbo decoding for ofdm systems," in *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2011.

[5] J. Bonnet and G. Auer, "Optimized iterative channel estimation for ofdm," in *IEEE Vehicular Technology Conference*, 2006.

[6] Y. Lee, A. Ashikhmin, and J. Chen, "Impact of soft channel construction on iterative channel estimation and data decoding for multicarrier systems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, 2008.

[7] F. Sanzi, S. Jelting, and J. Speidel, "A comparative study of iterative channel estimators for mobile ofdm systems," *IEEE Transactions on Wireless Communications*, vol. 2, no. 5, 2003.

[8] M. Dervin, M. Boucheret, G. Mesnager, and A. Ducasse, "A soft decision directed phase detector suited to satellite communications at very low signal to noise ratio," in *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, 2005.

[9] V. Lottici and M. Luise, "Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations," *IEEE Transactions on Communications*, vol. 52, no. 4, 2004.

[10] L. Zhang and A. Burr, "Iterative carrier phase recovery suited to turbo-coded systems," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, 2004.

[11] N. Wu, H. Wang, Z. Li, and J. Kuang, "Performance analysis of code-aided iterative carrier phase recovery in turbo receivers," *IET Communications*, vol. 17, no. 6, 2012.

[12] J. Bao, Y. Zhan, and J. Lu, "Iterative timing recovery via soft decision metrics of low-density parity-check decoding," *IET communications*, vol. 14, no. 4, 2010.

[13] R. Barry, A. Kavcic, W. LcLaughlin, A. Nayak, and W. Zeng, "Iterative timing recovery," *IEEE Signal Processing Magazine*, vol. 21, no. 1, 2004.

[14] C. Herzet, V. Ramon, and L. Vandendorpe, "Iterative soft-decision directed timing estimation for turbo receivers," in *IEEE 9th Symposium on Communication and Vehicular Technology*, 2003.

[15] C. Herzet, H. Wymeersch, M. Moeneclaey, and L. Vandendorpe, "On maximum-likelihood timing synchronization," *IEEE Transactions on Communications*, vol. 55, no. 6, 2007.

[16] N. Noels, H. Steendam, and M. Moeneclaey, "Pilot-symbol assisted iterative carrier synchronization for burst transmission," in *IEEE International Conference on Communications*, 2004.

[17] N. Noels, H. Steendam, M. Moeneclaey, and H. Bruneel, "A maximum-likelihood based feedback carrier synchronizer for turbo-coded systems," in *IEEE 61st Vehicular Technology Conference*, 2005.

[18] E. Vallés, R. Wesel, J. Villasenor, and C. Jones, "Carrier and timing synchronization of bpsk via ldpc code feedback," in *IEEE Asilomar Conference on Signals, Systems and Computers*, 2006.

[19] L. An and L. Wu, "Simplified iterative symbol timing and carrier phase recovery scheme for ldpc-coded systems," in *IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, 2006.

[20] P. Kim, K. Choi, Y. Song, B. Kim, D. Oh, and H. Lee, "Joint carrier recovery and turbo decoding method for tdma burst modem under very low snrs," in *IEEE Vehicular Technology Conference*, 2006.

[21] Y. Rahamim, F. Avraham, and R. Arie, "Ml iterative soft-decision-directed (ml-isdd): a carrier synchronization system for short packet turbo coded communication," *IEEE Transactions on Communications*, vol. 56, no. 7, 2008.

[22] J. Lee, H. Jae, and K. Seong-Cheol, "Joint carrier frequency synchronization and channel estimation for ofdm systems via the em algorithm," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, 2006.

[23] F. Merli and G. Vitetta, "Iterative ml-based estimation of carrier frequency offset, channel impulse response and data in ofdm transmissions," *IEEE Transactions on Communications*, vol. 56, no. 3, 2008.

[24] R. Yen, H. Liu, and C. Tsai, "Iterative joint frequency offset and channel estimation for ofdm systems using first and second order approximation algorithms," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012:341, 2012.

[25] M. Guenach, F. Simoens, H. Wymeersch, H. Steendam, and M. Moeneclaey, "Code-aided bayesian parameter estimation for multi-carrier systems," *European Transactions on Telecommunications*, vol. 17, no. 6, 2006.

[26] A. Viterbi, "Wireless digital communication: a view based on three lessons learned," *IEEE Communications Magazine*, vol. 29, no. 9, 1991.

[27] X. Wang, T. Tjhung, Y. Wu, and B. Caron, "Ser performance evaluation and optimization of ofdm system with residual frequency and timing offsets from imperfect synchronization," *IEEE Transactions on Broadcasting*, vol. 49, no. 2, 2003.

[28] T. A., M. K., K. M, G. E., J. U., and K. R., "Efficient inner receiver design for ofdm-based wlan systems: Algorithm and architecture," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, 2007.

[29] A. Ijaz, A. Awoseyila, and B. Evans, "Low-complexity time-domain snr estimation for ofdm systems," *Electronics Letters*, vol. 47, no. 20, 2011.

[30] N. Noels, H. Steendam, and M. Moeneclaey, "Carrier and clock recovery in turbo- coded systems: Cramer-rao bound and synchronizer performance," *EURASIP Journal on Applied Signal Processing*, 2005.

[31] M. Speth, A. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using ofdm. i," *IEEE Transactions on Communications*, vol. 47, no. 11, 1999.