A Frequency Offset Estimation Scheme for a T-DMB Software Baseband Receiver

Byungjik Keum, Moohong Lee, Jeong Han Jeong, Kyungsu Ko, Suhyun Cha, Young Serk Shim, and Hwang Soo Lee, Member, IEEE

MMPC and the School of EECS, KAIST, Daejeon 305-701, Korea
coolcoolbj@mmpc.kaist.ac.kr

Abstract — To avoid inter-carrier interference caused by frequency offsets (FOSs) in a T-DMB receiver based on OFDM, accurate FOS estimation is essential. Since the T-DMB system transmits only a phase reference symbol (PRS) in each frame, accurate FOS estimation of data symbols in each frame is difficult when a T-DMB baseband receiver is implemented in software. This paper presents a FOS estimation scheme for a T-DMB software baseband receiver. The FOS can be estimated by monitoring an abrupt change in fractional FOSs for adjacent symbols. Results demonstrating the validity of the proposed FOS estimation scheme are given.

Keywords — Frequency offset, Software Receiver, Terrestrial Digital Multimedia Broadcasting.

1. Introduction

Commercial service of Terrestrial Digital Multimedia Broadcasting (T-DMB) [1], which is based on the Eureka-147 Digital Audio Broadcasting (DAB) standards [2], was started in 2005 in Korea. Since then, diverse multimedia services such as audio, video, and data services have been available to users through T-DMB. In particular, mobile TV service on handheld devices and car navigation systems has become a major application of T-DMB. The T-DMB system uses orthogonal frequency division multiplexing (OFDM) as a transmission technique, because OFDM is robust in multipath fading channels and is bandwidth-efficient [3]. However, OFDM-based receivers are sensitive to frequency synchronization, because the frequency offset (FO) of each OFDM symbol in the input frames causes inter-carrier interference (ICI) [3].

Many types of DAB and T-DMB receivers have been implemented in hardware such as chipsets [4]-[7] and their FOS estimation and compensation are based on a closed-loop scheme [4]-[5]. In the closed-loop scheme, FO is first estimated and the estimated FO value is used to compensate the FO by shifting the frequency of the local oscillator in the front-end tuner to compensate it by shifting the frequency of the local oscillator. Frequency offset tracking algorithms have been studied in [8]-[9]. Several types of DAB and T-DMB receivers have been implemented in software running on a personal computer (PC) or a digital signal processor (DSP) [10]-[11]. In such software-based receivers, a closed-loop scheme to estimate and compensate the FO of each OFDM symbol is implemented in software. As a result, the estimated FO does not need to be sent to the local oscillator for FO compensation. However, FOS estimation and compensation for each OFDM symbol, whose data are stored in the input buffer of the software-based receiver, should be performed independently. Hence, accurate FOS estimation for every OFDM symbol in the input frames is necessary. On the other hand, as the T-DMB system transmits only a phase reference symbol (PRS) in each frame, it is difficult to accurately estimate the FOs of OFDM data symbols in each frame. This leads to inaccurate FO compensation in the T-DMB software baseband receiver.

In this work, a FOS estimation scheme for a T-DMB software baseband receiver is presented. The fractional FO (FFO) is first estimated and the estimated FFO is then added to the integral FO (IFO) of the PRS. The decision to change the IFO of the PRS in the summation process is determined by monitoring an abrupt change in FFOs for adjacent OFDM symbols.

An overview of a T-DMB software baseband receiver and a FOS estimation problem occurring in the T-DMB software baseband receiver are presented in Section 2. Section 3 describes the proposed FOS estimation scheme in detail. Results showing the validity of the proposed FOS estimation scheme are given in Section 4. Finally, we present conclusions in Section 5.

2. Overview of a T-DMB Software Baseband Receiver

A. T-DMB Transmission Frame

The T-DMB system [1], which is based on the Eureka-147 standards [2], transmits audio, video, and data signals using the transmission format shown in Fig. 1 over a multiplex channel with 2 MHz bandwidth. A T-DMB transmission frame is made up of three different physical channels. The first channel, called the synchronization channel, carries a Null symbol with no energy and a phase reference symbol (PRS) that is used as a phase reference for DQPSK demodulation used in the T-DMB system. These first two symbols are used for time and frequency synchronization at the receiver. The second channel is the fast information channel (FIC) composed of three OFDM symbols. This channel is used to carry control and system information. The last channel, called the main service channel (MSC), has four common interleaved frames (CIFs). One CIF is composed of 18 OFDM symbols and contains audio, video, and data service channels. Each of 76 OFDM symbols including the PRS in a transmission frame consists of a cyclic prefix and a useful OFDM symbol. The cyclic prefix (CP), which is a copy of the last part of the useful OFDM symbol, may be used for synchronization at the
receiver. The useful OFDM symbol carries data traffic for audio, video, and data channels. In this work, we call each OFDM symbol in a frame a symbol, a data symbol, or the PRS, depending on the situation.

Figure 1. Transmission frame structure for the T-DMB system.

B. Conventional FOS estimation and compensation scheme

The conventional T-DMB software baseband receiver [8], whose structure is shown in Fig. 2 (a), is made up of a radio frequency (RF) tuner, an analog-to-digital converter (ADC), a baseband processing unit, an audio and video (A/V) codec, and a data decoder.

Figure 2. The structure of a T-DMB software baseband receiver using a conventional FOS estimation and compensation scheme.

The low noise amplifier (LNA) in the RF tuner amplifies the small input signal and the mixer (MXR) then converts the RF input frequency to an intermediate or baseband frequency. The desired channel is selected by the bandpass filter (BPF) with the aid of the local oscillator, which generates a local frequency. The analog input signal of the selected channel is transformed into a digital signal by the ADC and stored in the input buffer. Subsequently, a time offset (TOS) \( \Delta t \), a FFO \( \Delta f \), and an IFO \( \Delta f \) for the input data are estimated by a TOS estimator, a FFO estimator, and an IFO estimator, respectively. The input data is then compensated by a time and frequency compensator using these estimated TOS and FOS (FOS compensation in the OFDM-based system is normally divided into IFO estimation and FOS estimation). The IFO in the OFDM-based system is defined as

\[
IFO = \arg \max_d \left\{ \mathcal{L}_p(k) \right\}
\]

In (1), \( Z_p(k) \) represents the \( k \)th subcarrier of the received PRS and \( Z_p(k) \) represents the \( k \)th subcarrier of the reference PRS. One of the conventional integral frequency synchronization algorithms such as that delineated in (1) [12]. This algorithm is based on partial correlation of the reference PRS and received PRS, which finds an IFO \( \Delta F \) within \( \pm 0.5 \) times of subcarrier spacing [12].

FOS estimation in the OFDM-based system is normally divided into IFO estimation and FFO estimation. The IFO is defined as a multiple of subcarrier spacing \( L_s \) in the OFDM-based system (\( L_s = 1 \) kHz in the T-DMB system). The IFO in the T-DMB software baseband receiver can be estimated using one of the conventional integral frequency synchronization algorithms based on the CP of a symbol such as that defined in (1) [12].

\[
IFO = \arg \max_d \left\{ \sum_{k=0}^{K-1} \sum_{d=0}^{L_s-1} \left[ Z_p(k + mB_s)Z_p \left( k + mB_s + d \right) \right] \right\}
\]

In (1), \( Z_p(k) \) represents the \( k \)th subcarrier of the reference PRS and \( Z_p(k) \) represents the \( k \)th subcarrier of the received PRS, \( K = B/B_s \) is the number of divided blocks, \( B_s \) is a partial bandwidth, and \( B \) is the multiplex channel bandwidth.

The FFO in the OFDM-based system is defined as

\[
-0.5L_s < FFO < +0.5L_s.
\]

One of the conventional fractional frequency synchronization algorithms based on the CP of a symbol such as that defined in (3) can be used to find a FFO within \( \pm 1 \% \) of subcarrier spacing in the T-DMB receiver [13].

\[
FFO = \frac{1}{2\pi} \tan^{-1} \left\{ \sum_{n=0}^{N_e-1} z_r(-N_e + n) \cdot z^*_r(N_e - N_e + n) \right\}
\]

In (3), \( z_r(n) \) represents the \( n \)th sample of the received PRS or the received data symbol. \( N_e \) is an effective OFDM symbol length, \( N_e \) is a cyclic prefix length, and the symbol * denotes a complex conjugate operation.

As the transmission frame of the T-DMB system has a PRS in each frame, as shown in Fig.1, the IFO for each frame can only be estimated once using the PRS. If the T-DMB baseband
receiver is implemented by hardware using chipsets [6] and FOS estimation and compensation based on the closed-loop scheme [4]-[5] are employed, the IFO of the PRS in a frame is sufficient to compensate FOSs for every data symbol. The reason for this is that once the IFO of the PRS is estimated, the estimated IFO along with the estimated FFO is sent to the local oscillator to shift the frequency of the local oscillator for FOS compensation of succeeding data symbols. As a result, the succeeding data symbols do not have IFOs but only FFOS. However, if the T-DMB baseband receiver is implemented in software running on a PC or a DSP, FOS estimation and compensation based on a closed-loop composed of the RF tuner and the DSP does not guarantee reliable FOS compensation performance due to slow response time.

When this closed-loop scheme based on FOS estimation and compensation is applied to a software baseband receiver [8], the estimated FOS is not sent to the local oscillator to shift its frequency, as shown in Fig. 2. Instead, an internal closed-loop, which is composed of a frequency compensator, OFDM demodulator, a FFO estimator, and an IFO estimator, is used under the assumption that the IFO of the PRS does not change at least over two adjacent frames. Thus, the IFO of the PRS belonging to the previous frame can be applied to the FOS compensation of the current frame. However, the FOS estimation and compensation for every symbol in each frame should be performed independently in the software-based receiver. Hence, the IFO of the PRS in each frame is not sufficient to compensate FOSs for every data symbol in the frame. There should be some way to estimate the IFO for each data symbol in every frame. A simple method to obtain the necessary FOS for each symbol is to add the FFO of each symbol, which is estimated using (3), to the IFO of the PRS for the frame. In this work, the FOS, which is obtained by simply adding a FFO to the IFO of the PRS, is referred to as a carrier frequency offset (CFO).

However, this simple method to obtain a CFO of each data symbol for FOS compensation cannot lead to error in the FOS estimation. If the actual FFO for a data symbol is outside the range in (2), the estimated CFO, which is obtained by adding the FFO estimated by (3) to the IFO of the PRS, will not be correct. This phenomenon is well explained using several FFO values shown in Fig. 3. The algorithm in (3) always produces a FFO for a symbol within the range in (2) regardless of the IFO value owing to the property of the function \(\tan^{-1}(\cdot)\). Therefore, if the actual FFO is either \(\text{FFO}_1\) or \(\text{FFO}_2\), as shown in Fig. 3, the estimated FFO will also be \(\text{FFO}_1\) or \(\text{FFO}_2\), respectively. However, if the actual FFO is either \(\text{FFO}_1\) or \(\text{FFO}_2\), the estimated FFO will be \(\text{FFO}_2\) or \(\text{FFO}_1\) instead of \(\text{FFO}_2\) or \(\text{FFO}_1\), respectively. As a result, adding \(\text{FFO}_1\) or \(\text{FFO}_2\) to the IFO of the PRS will result in an erroneous CFO. However, this error does not occur for the CFO of the PRS, because the IFO of the PRS is directly estimated using the PRS. For example, if the actual FOS of a data symbol is a value at the point \(a_1\), the CFO for the data symbol will be \(f_m + \text{FFO}_2\), where \(f_m\) is assumed to be the IFO of the PRS. However, if the actual FOS of the PRS is a value at the point \(a_1\), then the algorithm in (3) will estimate \(\text{FFO}_1\), which is assumed to be the IFO of the PRS. As a result, the CFO of the PRS will be \(f_m + \text{FFO}_1\) instead of \(f_m + \text{FFO}_2\).

Figure 3. Various FFO values around the subcarrier carrier \(f_m\); some of them can lead to error in estimating the CFO of each data symbol.

3. Proposed FOS Estimation Scheme

To correct the aforementioned FOS estimation errors occurring in the T-DMB software baseband receiver, we propose a new FOS estimation scheme, as illustrated in Fig. 4. The TOS estimator estimates the TOS \(\Delta T\) to find the start time of data decoding using [4]. TOS compensation is then performed by shifting data sample points using the estimated TOS. To estimate the FOS for each symbol, the T-DMB receiver runs in one of two different modes, a periodic mode or a tracking mode, depending on whether the symbol to be processed is the PRS or data symbols. The symbols in each frame are compensated using the estimated FOS \((\Delta F + \Delta T + \Delta f)\) depending on whether they are the PRS or data symbols. \(\Delta F\) denotes the IFO of the PRS and \(\Delta F\) denotes the corrected IFO for each data symbol. OFDM demodulation and channel decoding are then performed using the FOS compensated data.

Figure 4. Proposed FOS estimation scheme for a T-DMB software baseband receiver.

In the synchronization block of the T-DMB receiver shown in Fig. 3, FFO estimation using (3) is first performed by the FFO estimator for each symbol including the PRS in a frame. In the periodic mode, which works only on the PRS, the switch (SW) is connected to port 1 (P1) and the IFO of the PRS in each frame is estimated using (1) by the IFO estimator. In the tracking mode, which works on data symbols, the switch (SW) is connected to port 2 (P2). Even though the IFO of a data symbol cannot be estimated directly because there is no known information available to the receiver on each data symbol, the IFO can be indirectly estimated by monitoring an abrupt change in FFO values for adjacent two symbols. The CFO estimator can use FFOSs, which are estimated by the FFO estimator for symbols in each frame, to determine whether the
IFO of each symbol should be changed from that of the PRS due to frequency drift or incorrect FFO estimation for the symbol. For this determination regarding the IFO of each data symbol, the CFO controller uses the algorithm described in Fig. 5 under the assumption that there is no abrupt change in actual CFOs for two adjacent symbols.

The operation of the proposed FOS estimation algorithm shown in Fig. 5 is as follows: First, the CFO of the mth symbol, \(\text{CFO}_m\), obtained by adding the estimated FFO (\(\text{FFO}_m\)) to the IFO of the PRS. The first condition, i.e., that \(\text{CFO}_{m+1} - \text{CFO}_m > \text{threshold 1 (Th1)}\), is then checked to assess if it is true. If the first condition is not met, we assume that there is no error in the estimated CFO and use the IFO of the PRS as that of the \(m\)th symbol. If the first condition is met, we assume that there is an error in the estimated CFO. We then check whether the second condition, i.e., that \(\text{CFO}_{m+1} - \text{CFO}_m > \text{threshold 2 (Th2)}\), is true. If the second condition is not met, we assume that there was some error in estimating the CFO. We next check the third condition, i.e., that \(\text{CFO}_{m+1} > 0\) and \(\text{FFO}_m < 0\). If the third condition is met, it is assumed that a positive drift from the IFO of the PRS is required to obtain the IFO for the \(m\)th symbol and therefore the IFO of the PRS is increased by one. If the third condition, i.e., that \(\text{CFO}_{m+1} > 0\) and \(\text{FFO}_m < 0\), is not met, we assume that a negative drift in the IFO of the PRS is needed and decrease the IFO of the PRS by one.

The threshold Th1 and Th2 are critical for the reliability of the FOS estimation algorithm. If the Th1 is large, the CFO drift, which is required if the FFO value estimated by (3) exceeds the range in (2), cannot be performed. If the Th2 is small, the wrong CFO estimation by (3) can be addressed by a positive or negative IFO drift. Both cases cause an error in the FOS estimation for the \(m\)th symbol, which leads to performance degradation of the receiver. In this work, thresholds 1 and 2 are set to be 0.3 and 0.8, respectively. The validity of this algorithm can be easily checked using the FFO values shown in Fig. 3.

Some numerical examples illustrating how FOS estimation errors in a T-DMB input frame can be corrected by the proposed FOS estimation scheme are presented in Table 1. Table 1 shows actual CFOs, actual IFOs, actual FFOs, estimated FFOs (e-FFOs), and estimated CFOs (e-CFOs) for symbols in a frame. The second row indicates that the CFO, IFO, and FFO of the PRS are 3.3, 3, and 0.3, respectively.

In the case of the 2nd symbol, since the actual FFO is 0.4, which is within the range of (1), the e-FFO is 0.4 and the e-CFO is correct, i.e. 3.4. However, in the case of the 4th symbol, since the actual FFO is 0.6, which is outside the range of (1), the e-FFO is -0.4. Therefore, the e-CFO becomes 2.6, i.e. \(3 - 0.4 = 2.6\), which is seen to be incorrect when compared with the actual CFO of 3.6. Similar errors in the estimated CFOs will occur if the actual FFOs are beyond the range in (1), as shown in Table 1 for the 5th, 73rd, and 74th symbols.

The last two column in Table 1 show the corrected CFOs (c-IFOs) and CFOs (c-CFOs), which can be obtained by the proposed FFO estimation algorithm. The erroneous CFOs of the 4th, 5th, 73th, and 74th symbols were all corrected and their IFOs are increased by one. As a result, their IFOs become 3 instead of 3, which is the IFO of the PRS. Subsequently, they are used for FOS compensation of each data symbol, as shown in Fig. 4, where \(\Delta F\) denotes the IFO of the PRS and \(\Delta F\) denotes a corrected IFO for each data symbol. From the results listed in Table 1, we can conclude that accurate FOS estimation in the T-DMB software baseband receiver can be obtained with the proposed FOS estimation scheme.

### Table 1. Numerical Examples showing erroneous CFOs and corrected CFOs for various OFDM data symbols

<table>
<thead>
<tr>
<th>OFDM symbol</th>
<th>CFO</th>
<th>IFO</th>
<th>FFO</th>
<th>PRS CFO</th>
<th>e-FFO</th>
<th>e-CFO</th>
<th>c-IFO</th>
<th>c-CFO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st (PRS)</td>
<td>3.3</td>
<td>3</td>
<td>0.3</td>
<td>3</td>
<td>0.3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2nd</td>
<td>3.4</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>3rd</td>
<td>3.4</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4th</td>
<td>3.6</td>
<td>3</td>
<td>0.6</td>
<td>3</td>
<td>-0.4</td>
<td>2.6</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>5th</td>
<td>3.7</td>
<td>3</td>
<td>0.7</td>
<td>3</td>
<td>-0.3</td>
<td>2.7</td>
<td>4</td>
<td>3.7</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73rd</td>
<td>3.6</td>
<td>3</td>
<td>0.6</td>
<td>3</td>
<td>-0.4</td>
<td>2.6</td>
<td>4</td>
<td>3.6</td>
</tr>
<tr>
<td>74th</td>
<td>3.6</td>
<td>3</td>
<td>0.6</td>
<td>3</td>
<td>-0.4</td>
<td>2.6</td>
<td>4</td>
<td>3.6</td>
</tr>
<tr>
<td>75th</td>
<td>3.4</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
<td>0.4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>76th</td>
<td>3.3</td>
<td>3</td>
<td>0.3</td>
<td>3</td>
<td>0.3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

### 4. Performance Verification

To confirm the validity of the proposed FOS estimation scheme, we have implemented a T-DMB software baseband receiver running on a DaVinci DSP processor [14] mounted on a digital board. The implemented software baseband receiver is composed of an input buffer, TOS and FOS compensators, an OFDM demodulator, a channel decoder, and a synchronization block including a CFO estimator, as shown in Fig. 4. A commercial RF tuner was used to receive a T-DMB video signal with a rate of 512 kbps in transmission mode I, which is generated by a commercial T-DMB generator. We also built a
To monitor how accurately the proposed FOS estimation scheme estimates FOSs of symbols in the T-DMB input signal, we configured the T-DMB signal generator to sweep its output frequency so that a T-DMB output signal with a frequency offset varying between 0 kHz and 6 kHz is transmitted, as shown in Fig. 6. The trace showing changing FOSs of the T-DMB input signal is marked as Generated FOS.

Even though the estimated FOSs roughly follow the Generated FOS trace, as shown in Fig. 6 (b), some FOS estimation errors occur at a somewhat regular interval due to the property of the function \( \tan^{-1}(\cdot) \) in (3), as expected. The third algorithm used in the proposed FOS estimation scheme accurately estimates the FFOs of data symbols in each frame and thus the trace of the estimated FOSs coincides precisely with the Generated FOS trace, as shown in Fig. 6 (c).

A computer simulation was also performed to evaluate the effect of FOS estimation accuracy, which is provided by the three FOS estimation algorithms, on the bit error rate (BER) performance of the T-DMB software baseband receiver. The following conditions were used in the simulation. The number of test bits transmitted to obtain a reliable BER at each carrier to noise ratio (CNR) was \( 5 \times 10^6 \) and a FFT size of 2048 was used. The BER was measured after the channel decoder under an AWGN channel while T-DMB frames with slowly changing FOSs between 0 kHz and 6 kHz were transmitted.

The BERs of the T-DMB software baseband receiver, which was implemented on a PC, using the three FFO estimation algorithms noted above are plotted in Fig. 7. As expected, the T-DMB receiver using the first algorithm shows the worst performance, because of severe ICI among the subcarriers. The BER performance of the T-DMB receiver using the second algorithm is also considerably degraded due to ICI, compared to that of the T-DMB receiver that decodes input frames without FOS. However, when the proposed FOS estimation scheme was used, the BER performance was not deteriorated, as shown in Fig. 7, because the FOS estimation for every symbol in each frame is accurate. The simulation result shows that if accurate FOS estimation for symbols in each input frame is not properly performed, the BER performance of the T-DMB software baseband receiver will considerably deteriorate due to the ICI problem, which is caused by inaccurate FOS compensation using incorrect FOSs.
5. Conclusion

We presented a FOS estimation scheme for a T-DMB software baseband receiver. To accurately estimate FFOs for symbols in each T-DMB input frame, the FFO of each symbol is first estimated by the conventional algorithm and the estimated FFO is then added to the IFO of the PRS to obtain an FOS for each data symbol. The decision to change the IFO of the PRS, which is used as the IFO of each data symbol, is determined by monitoring an abrupt change in FFOs for adjacent symbols. We presented FOS estimation results measured on a digital board and simulation results for BER performance to demonstrate the validity of the proposed FOS estimation scheme.

ACKNOWLEDGEMENT

This work was supported by the IT R&D program of MKE/IITA under Contract No. 2005-S-609-04, an International Joint Project between Texas Instruments and KAIST, Korea.

REFERENCES