Synchronization Considerations in Meteor-Burst Communications

Scott L. Miller
Dept. of EE
University of Florida
Gainesville, FL 32611

Laurence B. Milstein
Dept. of ECE
University of California, San Diego
La Jolla, CA 92039

Abstract

In this paper, we consider the effect of frame synchronization on the performance of a meteor-burst communication system. It is shown that for a system which can tolerate a reasonable size frame sync pattern on each packet, synchronization losses should not contribute much to overall performance degradation. That is, it is shown that, at random noise, synchronization losses should not be too large for satisfactory operation even in the presence of perfect synchronization.

1. Introduction

For typical MB systems, three levels of synchronization are required, those of carrier, bit and frame synchronization. However, because the modulation format that we consider is binary frequency shift keying with non-coherent demodulation, we do not need a coherent phase reference. Also, it is assumed that each packet contains a certain amount of overhead at the beginning of the packet for bit synchronization purposes and that this is adequate for all SNR's encountered. Hence in this paper, we concentrate on frame synchronization.

One simple way to provide frame synchronization is to insert a marker sequence at the beginning of each packet. When the receiver detects this sequence (often called a unique word (UW)), it assumes that what is to follow is the information portion of the packet. For systems with short packets (such as the COMET system [1]), the amount of overhead used for synchronization may become significant and thus a more efficient scheme must be used. However, for the protocols considered in [2], and upon which this work is based, we assume packets that are long enough so that inserting even a fairly lengthy marker at the beginning of each packet only adds a minimal amount of overhead. Thus, for the purposes of this paper, a unique word is used to provide packet synchronization.

In Section 2, an approximate analysis is presented to determine the probability of correctly receiving the packet. Numerical results are presented in Section 3, along with final conclusions.

2. Analysis

The structure of the packet is shown in Figure 1. The packet starts with \( h \) bits to provide bit synchronization, and this is followed by an \( m \)-bit \( UW \). The last \( n \) bits then constitute the information portion of the packet. Note that this last segment may contain, in addition to data, items such as packet number, error detection bits, and any other overhead that may be needed. The information is assumed to be coded with an \( (n,k) \) linear block code, where \( k \) is equal to \( n \) if no code is used. Upon reception of the packet, the receiver correlates the input with the \( UW \), and if the difference is less than some threshold \( h \), which is known as the error tolerance, it declares acquisition and treats the next \( n \) bits as the information portion of the packet.

<table>
<thead>
<tr>
<th>( h ) bits</th>
<th>( m ) bits</th>
<th>( n ) bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>bit sync</td>
<td>UW</td>
<td>information (possibly coded)</td>
</tr>
</tbody>
</table>

Figure 1 Packet structure

With this packet structure, it is seen that a packet error can occur in one of two ways. Either the information segment is not decodable because of too many random errors, or incorrect synchronization causes the information segment to be totally unintelligible. Thus, the probability of correctly receiving a packet, can be written as

\[
P_r = Pr(\text{correctly receiving packet} | \text{correct sync}) Pr(\text{correct sync}).
\]

(1)

The first quantity was calculated in [2], and the second term can be approximated as follows. First, for simplicity, assume that the SNR is constant for the duration of a \( UW \). This is certainly reasonable since at the rates being considered here, a typical \( UW \) lasts only a few milliseconds and thus the SNR varies by at most a few tenths of a dB. Using the approach of [3], the receiver achieves correct synchronization if it detects the \( UW \) correctly and if there are no false alarms (false acquisitions) immediately preceding the \( UW \). Thus,

\[
Pr(\text{correct sync}) = Pr(\text{correct detection of UW} | A) Pr(A),
\]

(2)

where \( A \) is the event that there are no false alarms that cause the correct sync pattern to be missed. Recall that once acquisition is declared, the receiver treats the next \( n \) bits as information and does not look for the \( UW \) at those \( n \) positions. Thus, a false alarm causes incorrect synchronization (or none at all) if it occurs in any of the \( n \)
positions preceding the correct position of the UW. The event \(A\) is then equivalent to having no false alarms in the \(n\) bits previous to the correct position of the UW, so that (2) can be bounded by

\[
Pr(\text{correct sync}) \geq Pr(\text{correct detection} | \text{no false alarms in } n \text{ bits})
\]

\[
\cdot Pr(\text{no false alarms in } n \text{ bits})
\]  

The inequality is due to the fact that the receiver may not always check for the UW in all the previous \(n\) positions, and thus there may be no chance for a false alarm in many of the \(n\) positions preceding the correct position of the UW.

Let \(P_{fa}\) be the probability of declaring acquisition on an incorrect position (of random data) and \(P_{fa}\) be the probability of declaring acquisition on the correct position. If it is assumed that the probability of declaring acquisition on either an out-of-phase position of the UW or a portion of the bit sync sequence is less than that of declaring acquisition on random data, then the last term in (3) can be bounded by

\[
Pr(\text{no false alarms in } n \text{ bits}) \geq 1 - nP_{fa}
\]  

In addition, it is assumed that, for a well-designed UW, the first term in (3) can be (approximately) lower bounded by \(P_{fa}\), since the fact that there are no false alarms would tend to indicate fewer errors. For such a case, Equation (3) becomes

\[
Pr(\text{correct sync}) \geq P_{fa}(1 - nP_{fa})
\]  

where \(P_{fa}\) and \(P_{fa}\) are given by

\[
P_{fa} = \sum_{k=0}^{m} \binom{m}{k} \left( \frac{1}{2} \right)^m
\]  

and

\[
P_{fa} = \sum_{k=0}^{m} \binom{m}{k} pFA^{m-k}
\]

respectively. In Equations (6) and (7), \(h\) is the error tolerance of the frame synchronization algorithm, and \(pFA\) is the bit error rate present during the time that the UW is transmitted. Equations (1) and (7) can now be used in conjunction with the results of [2] and [4] to study the performance of a system using the previously described synchronization scheme.

3. Numerical Results and Conclusions

Before looking into the performance of a particular protocol, the length of the marker and the error tolerance must be chosen. Towards this end, consider Figure 2, which shows the probability of achieving correct synchronization on a packet as a function of the marker length for several SNR’s. It should be noted that the error tolerance, \(h\), used to plot Figure 2, was the optimum value of \(h\) (found numerically) for each particular SNR and marker length. Thus, for the 8 dB curve, for example, the optimum \(h\) for \(m=20\) is \(h=1\), while the optimum \(h\) for \(m=40\) is \(h=4\). Even for a fixed \(m\), the optimum \(h\) is still a function of SNR. This is of particular importance in a MB system where it is required to achieve synchronization over a variety of different SNR’s with a fixed marker and error tolerance.

Since, for a particular marker, there is not necessarily a value of \(h\) that will maximize the probability of correctly achieving sync for all SNR’s, it appears to be most beneficial to choose a value of \(h\) that is optimum over, in some sense, a range of "most commonly encountered SNR’s". Consider, for example, Figure 3, which shows the probability of correct sync as a function of SNR for a marker of length 20 and three different values of \(h\). In [2], it is shown that for uncoded systems, most of the throughput is due to packets that have SNRs greater than about 10 dB. For this case then, the best choice is \(h=0\), since it is optimum for all SNR’s greater than about 10.5 dB.

On the other hand, for some coded systems analyzed in [4], a significant amount of the throughput is due to packets that have SNRs as low as 7-8 dB. Thus, it is unclear whether \(h=0\) or \(h=1\) is the best choice. For such cases, it may be that the only way to determine the best choice is to evaluate the performance of the system for several choices of \(h\). Table 1 shows the throughput for a rate 0.9 (2047, 1383, 19) BCH code for the 3 different values of \(h\) plotted in Figure 3. It is seen from this table, that, even for the coded system, \(h=0\) is still the best choice. This may be explained by noting that, for the packet lengths being used here, even those packets with SNRs as low as 7-8 dB at the beginning of the packet where the UW occurs.

<table>
<thead>
<tr>
<th>(h)</th>
<th>(m)</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>20</td>
<td>94.3</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>90.6</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>54.8</td>
</tr>
<tr>
<td>0</td>
<td>40</td>
<td>93.5</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>93.7</td>
</tr>
<tr>
<td>4</td>
<td>40</td>
<td>93.6</td>
</tr>
<tr>
<td>6</td>
<td>40</td>
<td>92.8</td>
</tr>
</tbody>
</table>

Table 1. Performance of synchronization schemes for (2047,1383,19) BCH code.

Also shown in Table 1 is the throughput for the same system with a UW of length 40 for several different values of \(h\). The probability of correct sync for this marker is shown in Figure 4. Note that the performance of this system (as given in Table 1) is about the same for any \(h\) less than about 6, and that the performance is slightly worse than that for the shorter, \(m=20\), marker using an \(h=0\). Thus, it is seen that for the MB systems considered here, it is most beneficial to design the frame sync marker so that it provides a high probability of correct sync for high SNR’s.

Another important conclusion that can be drawn from Figure 4 is that the ability to synchronize is not a limiting factor on the performance of the system. This can be seen from the \(h=6\) curve in Figure 4, in which the probability of correct sync is essentially constant for all SNR’s greater than about 6 dB. Thus, this sync scheme will work well for SNR thresholds as low as 4 dB, since the UW is at the beginning of the packet and thus will have a SNR about a few dB above the threshold. But, from Figure 6 of [4], it is seen that the performance of the MB system levels off at a threshold of about 7 dB. On the other hand, it should be pointed out that for a protocol which uses much shorter packets (such as a selective repeat ARQ scheme), the 40 bits used for a UW may be too excessive and thus a shorter marker must be used. In such a system, synchronization may indeed turn out to be a limiting factor.

Lastly, consider Figure 5, which shows the performance of some of the codes used in [4] with the \(m=20\) marker. Comparison of Figure 6 of [4] and Figure 3 of this paper shows that the results including synchronization affects are almost identical to the results under the assumption that the receiver cannot synchronize correctly. This is due to the fact that when the SNR is so low that the receiver cannot synchronize, the packet will, with high
probability, be filled with too many errors for the code to handle and thus the packet will be lost anyway.

4. References


Figure 2. Probability of achieving correct sync vs. marker length.
Figure 3. Probability of achieving correct sync vs. SNR; marker length = 20.

Figure 4. Probability of achieving correct sync vs. SNR; marker length = 40.
Figure 5. Performance of various BCH codes including synchronization effects; $m = 20, h = 0$. 

<table>
<thead>
<tr>
<th>SNR Threshold (dB)</th>
<th>Throughput (bits/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>100</td>
</tr>
<tr>
<td>6</td>
<td>90</td>
</tr>
<tr>
<td>8</td>
<td>80</td>
</tr>
<tr>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>12</td>
<td>60</td>
</tr>
</tbody>
</table>

- uncoded
- (2047,1948,9)
- (2047,1838,19)
- (2047,1717,30)