FRAME SYNCHRONIZATION IN PCM TELEMETRY

BERNARD PEAVEY
NASA-Goddard Space Flight Center
Greenbelt, Maryland

Summary Frame synchronization of PCM telemetry data can be accomplished very effectively by employing the fixed threshold method with a unique strategy. Probability of false sync acquisition and maintenance can be made negligibly small for data having bit error rates less than 10%, at a small sacrifice in acquisition time. Furthermore, experimental results indicate that frame synchronization is significantly affected by the frame sync code length rather than by the code pattern itself, i.e., any pseudo-random code is just as good. The difference in performance between “optimum” codes and any pseudo-random code is negligible.

Introduction In PCM telemetry, the serial stream of binary data (bits) is composed of words (groups of bits) which originate from one source, or are time multiplexed from several sources. In order to identify these words and sources at the receiving end it is necessary to append to each word a unique identification code, or group a number of these words into blocks (frames or groups) and append only one identification code to the entire block. The process of word or frame identification is called word or frame synchronization, respectively. Reliable word synchronization by means of word sync codes is extremely inefficient in terms of data bandwidth. Therefore, word synchronization is used very seldom in general, and is not used at all by Goddard Space Flight Center.

By inserting the same frame sync code in each frame of data, and establishing a fixed frame length in terms of an integral number of words, it is possible to precisely identify each word and its source with very small overhead, usually less than 5%. Thus, this paper deals with the process of frame synchronization based on fixed frame length and random data occurring in each frame. Since it is possible to achieve frame synchronization (sync) and, thereby precisely identify each data word, the central question is: under what conditions and what percentage of the time can this be done? To answer this question properly, one needs to define and characterize the performance of a given frame synchronization method (frame synchronizer) in useful and measurable terms. Although the literature contains descriptions of specific frame synchronizers, and methods for determining the probability of false sync acquisition (also referred to as mean time to sync), the results have not been verified experimentally, and the conditions for which these results have been obtained are not all inclusive from a practical point of view.

In addition to frame sync performance characteristics, one ought to consider the implementation aspects also. For example, a full fledged maximum likelihood frame synchronizer is considerably more complex to implement (and almost impossible for high data rates) than a fixed threshold type, which is very simple, while the performance of the fixed threshold is only slightly inferior to that of the maximum likelihood, which is theoretically optimum.

In this summary paper, an attempt is made to define frame sync performance characteristics which can be measured under realistic operating conditions, explain the frame sync process, and present test data on the performance of frame synchronizers and frame sync codes. Application of frame sync methods in specific cases is also considered in light of frame sync code selection, frame length, and data recovery.

Frame Sync Performance Characteristics In order to correctly recover telemetry data it must be bit and frame synchronized. It is only under these conditions that each frame of data can be identified and, therein, each data word can be properly recognized. Thus, one obvious characteristic of a frame synchronizer should be the average time required to correctly establish frame sync. This is called frame sync acquisition (FSA), and is measured in terms of frames. Once frame sync has been acquired correctly, it must be maintained for the duration of data. This is called frame sync maintenance (FSM), and is measured in terms of frames maintained, or in terms of frames lost (FSL), i. e. , the average number of frames retained falsely.

To measure these performance characteristics the frame synchronizer under test must always receive bit synchronized data. This data must be random, and contain bit errors in the amount of the expected bit error rate of actual telemetry data. The test instrument must be capable of controlling the input to the frame synchronizer, count the number of frames required to achieve frame sync, and verify the validity of the acquired frame sync prior to registering the frame count. Following acquisition, the test instrument causes the frame synchronizer to revert to the search mode and repeat the same process a sufficient number of times in order to accumulate adequate statistical data. For FSL measurements, the test instrument simply counts the number of frame sync losses versus the total number of frames correctly maintained.

Since the main purpose of frame synchronization is to enable the data to be decommutated and, thus recovered, it is very important to know the FSA and FSM of the frame synchronizer used in the data handling system. In fact, FSA and FSM are the only two characteristics which directly affect data recovery.

Frame Synchronization For a fixed frame length data format, frame synchronization consists of detecting the frame sync code (FSC) and observing it in the same location of each frame for the entire duration of the telemetry data. In an ideal case, one would wish to obtain frame sync immediately upon having received the first frame of data, and maintain it thereafter until the end of data. Indeed, it is always desirable to minimize FSA and maximize FSM. However, since random data may contain groups of bits which are replicas of the FSC, false detection of the proper FSC is certainly possible. This fact suggests that there is a tradeoff between FSA and the probability of false frame sync detection, PFS , depending on the frame length and the FSC length (the FSC itself is found to have little or no adverse effect). Clearly, longer frames of random data contain more data words resembling short FSC than shorter frames do - a fact of life which must be accepted.

Frame synchronization, therefore, encompasses the selection of an appropriate FSC, frame length (the number of data bits between which the FSC is inserted and is expected to occur at the receiving end), and the procedure for correctly detecting this FSC not only in error free data but also in data containing bit errors.

Frame Sync Code One of the parameters in the frame synchronization process is the FSC. FSC is characterized by length (K bits), and a particular pattern which, presumably, exhibits a correlation property such that PFS is minimized when this code is searched on in a random stream of data having a given bit error rate (BER), usually less than 10%. Thus, attempts have been made to find "optimum" FSC on the basis of a specific frame sync procedure (strategy) such as described in Ref. 1. As shown in Ref. 1, these codes were developed under the conditions that (1) the allowed number of bits in error is always two (for all FSC lengths), (2) the BER is 10%, and (3) the frame sync is established on the first "hit", i.e., without verification. When these codes were subjected to exhaustive tests, which simulated the expected data and the assumed BER, it was determined that (1) these were not unique codes, and (2) the P_{FS} given in Ref. 1 was lower by 2-4 orders of magnitude. In fact, further experimental investigation indicated that any pseudo-random FSC of 16 bits or longer was equally reliable and efficient. Furthermore, the FSC length, rather than the pattern, is directly affecting the P_{FS} , which is considerably decreased with longer FSC.

Using P_{FS} as a measure of "goodness" of frame sync codes (FSC), it is seen from experimental investigations, carried out under conditions representing a wide range of data formats and BER, that almost any pseudo-random code of reasonable length, say longer than 16 bits, will perform equally well. Table 2 shows the P_{FS} of several different 24 bit FSC. Although the 5th FSC produced a higher P_{FS} than the other four codes, when used with a frame sync strategy requiring two consecutive hits, even this code will produce the same negligible P_{FS} as the other codes. It should be noted that the P_{FS} in Table 2 were obtained on the basis that frame sync was detected on the first hit, and the number of allowed bit errors, E_s, in the FSC was “2”. However, the particular strategy used is not important as long as it is the same for all codes.

Frame Sync Code Length As mentioned previously, P_{FS} is very much dependent on the FSC length, K. Table 3 shows, for example, that a 10:1 improvement (decrease) in P_{FS} can be achieved by increasing K from 18 to 30 bits. This is a significant improvement in situations where FSA must be very small, say one frame. Indeed, regardless of the synchronization method, the longer the FSC the smaller PFS and FkA will be. This is due to the fact that in random data longer words resembling FSC occur less frequently than shorter words do. In most cases, if not all, it is seldom necessary to use more than 30 bits. This applies to formats having 4000 bits per frame or less. Experimental data indicates that with K = 30, BER < 10%, and frame length, $L \leq 4000$ bits ($L > 4000$ was not tested, however, there is no reason to suspect significant degradation for $L < 10000$), $P_{FS} < 2\%$ when frame sync is acquired on the first hit, and if $E_s = 3$, the attendant FSA = 1. A fixed threshold frame synchronizer requiring only one verification (2 consecutive hits) has been demonstrated to acquire frame sync with P_{FS} less than 10^{-5} and FSA <3 for BER <10% using 18< K <31.

Frame Length In general, P_{FS} has been observed to increase approximately linearly with L. However, the significance of L should be viewed not in terms of P_{FS}, rather in terms of data recovery. Since FSA is given in frames, it is clear that when L is large a substantial amount of data will be lost during the frame sync acquisition period. This is particularly true when frame sync cannot be maintained due to loss of bit sync, and reacquisition becomes frequent. This consideration certainly outweighs that of P_{FS} which may increase from 1% to 2% over a span of 3000 bits, as shown in Table 4.

Frame Synchronization Method Frame synchronization can be accomplished in several ways with rather similar performance characteristics and different degrees of sophistication. Some of the techniques are: (1) maximum likelihood, (2) maximum correlation, (3) variable threshold, and (4) fixed threshold. All of these techniques are designed to accomplish the same function, viz. , detection of the a priori known FSC in the telemetry data. Each of these methods performs this function with somewhat different P_{FS} and, therefore, varying FSA. Since the optimum method can achieve FSA ≤ 1, the other methods are not expected to be superior. Indeed, their performance falls into the range of FSA < 3, for BER < 10%. Although these methods exhibit some differences in FSA performance, there is virtually no difference observed in the performance of FSM. This is true because each method lends itself to the utilization of the same strategy for FSM.

The maximum correlation (MC) method uses the principle of finding maximum correlation between the FSC mask and the random data within one frame length. If several values are found to be equal, the first one is chosen as the frame sync. If this were indeed the correct location of frame sync, then exactly L bits away one would expect another maximum correlation, etc. Though this procedure is simple enough, it does not by itself produce low P_{FS} . In practice, this procedure is augmented by a "verification strategy" which requires the examination of the bit errors in the expected frame sync location L bits away from that of the initial maximum correlation. This strategy considerably lowers P_{FS}, but also increases FSA.

The variable threshold (VT) method attempts to vary the allowed bit errors in the search mode, E_s, and those in the verification and maintenance modes, E_L, on the basis of a priori knowledge of the data BER, and the number of bit errors observed in the frame sync location. Again, it is seen that a certain strategy is involved, and this strategy depends on the BER. If the BER is accurately estimated, the performance of this method is about the same as that of the MC method.

The fixed threshold method (FT), as the name implies, employs fixed numbers of allowed bit errors with which frame sync may be acquired and maintained. Depending on the expected data BER and FSC length, K, used in the data format, it might be necessary to require one verification in each mode. The value of E_s (bit errors allowed in search) is determined by $E_s \leq$ (BER)K, and E_L (bit errors allowed in lock) is always set to approximately 0.3 K. For example, if BER $\leq$ 5%, and K = 30 bits, then $E_s \leq 0.05 = x\ 30 = 1.5$ or 1 bit, and $E_L = 0.3 \times 30 = 9$ bits. In general, it has been determined experimentally that for K > 24 and BER < 5% no verification is required in either mode. Otherwise, only one verification is needed in both modes to assure best performance in terms of FSA and FSM. By verification is meant the "successive occurrence of hits or misses" in acquisition (search) or maintenance (lock) mode, respectively. For example, one verification implies the occurrence of two successive hits (or misses), i.e., the initial and a second hit (or miss) separated by exactly L bits (one frame length).

The rationale for setting E_s and E_L to the values mentioned above (which was verified experimentally) is that, in order to minimize FSA, the allowed number of bit errors in the frame sync correlator should be equal to the expected average bit error rate. Since, in practice, the BER seldom exceeds 10%, E_s may always be set to $\leq$0.1 K, whichever amounts to integral bits. Indeed, since bit synchronization is seldom maintained for BER > 10% (E/N_o < OdB) anyway, it is unreasonable to expect frame synchronization under such conditions. To maximize FSM, the strategy must be such as to permit transfer to search only when there is no data or when there is bit slippage (loss of bit sync). Thus, infrequent bursts of bit errors due to the noise distribution or sudden degradation of the communication channel, should not cause loss of frame sync as long as bit sync has been maintained. Therefore, E_L is chosen on the basis of the probability that the expected BER would exceed 3 times (or less) its mean value is negligible. For normally distributed noise, hence normal BER distribution, P(BER > 3 times its mean) is extremely small, and that this would occur twice in a frame of 4000 bits is, for all practical purposes, “zero.” Therefore, by setting $0.3\ K < E_L < 0.4K$, FSM is assured to be maximum. It should be observed that, theoretically, E_L could be set to 0.49 K, since only bit slippage would cause 50% bit errors in the FSC, as well as in the data, of course. The danger of setting $E_L > 0.4\ K$ (40%) is due to the possibility of falsely remaining in lock. This is why the FT method requires the P_{FS} to be negligibly small.

Performance of Frame Sync Methods Experimental tests of the MC, VT, and FT methods indicate that the overall performance of each one is approximately the same for BER < 10% and K > 16 bits. Each method is highly dependent on the particular strategy used for given BER and K. However, the FT method, in addition to being the simplest to implement, is the least sensitive to these conditions. Actually, the same FSM strategy should be used with all 3 methods, viz., using $E_L = 0.3\ K$ with one or no verification.

Since a complete analysis of FSA and FSM of the above methods has not been made, nor has it been available in the literature, the experimental results given herein can provide some idea as to what might be expected of frame syncs. In a limited sense, however, Ref. 2 does give some analytical values which do agree with the experimental data obtained in this investigation. Emphasis should be placed on total performance, and on the way in which the system is to operate. In this sense, it can be stated that the FT method’s performance is as given in Table 5.

Generally, best performance is obtained when K is large (> 27 bits), since this obviates the need for verification. Thus, without verification, frame sync is acquired on the first hit, and returned to search on the first miss. For BER < 5%, FSA $\leq$ 1, and FSM = 100%.

Experimental Tests Performance characteristics of the methods described above have been obtained by subjecting both hardware and software frame synchronizers to the various data conditions and FSC that might be expected in operational applications. Simulated random data was formatted in various frame lengths up to 4096 bits, including some fixed words for reference checks. BER was controlled by adding white Gaussian noise to the data which was put through a calibrated bit synchronizer. Alternating data formats (one of which was proper) were continuously applied to the bit synchronizer in order to maintain bit sync all the time. By alternating these formats, acquisition events were created and counted. False acquisitions were automatically detected and counted versus the total number of events. Thus, probability of false sync acquisition, P_{FS}, was computed from false sync, FS, and acquisition events N_A so that $P_{FS} = \frac{FS}{N_A}$.

The average acquisition time, FSA, was determined by simply counting the number of data frames that were searched until proper frame sync was established. Again, by applying alternate data formats it was possible to create acquisition events so that

$FSA = \frac{1}{N} \sum_{i}^{N} F_i$, where F = number of frames searched per acquisition event, and N = the number of events (samples) subjected to the test.

Selection of Method Having at least four possible methods to select from, it is naturally desirable to employ the most suitable approach (including strategy and FSC) for a given application. To do this, it has been found very helpful to know the characteristics of the data and expected performance of the communication link, especially the bit synchronizer. In addition, the data recovery requirement should be carefully considered in light of the expected bit slippage rate and amount of available data. For instance, if the expected bit slippage rate (BSR) is low, say less than 1%, and the telemetered data is voluminous, say upwards of 10 Kbps, it might be acceptable to lose 3%-4% of data as a result of somewhat longer FSA.

Of the four methods described herein, the ML is, almost by definition, the best yet the least used, if at all. The FT method has been found to approach the ML when K >27, and BER < 5%. It has been used very extensively at GSFC for the last 3-4 years with great success. Its simple and inexpensive hardware implementation lends itself to operating at very high data rates (> 25 Mbps), limited only by the speed of available logic circuits.

Conclusion Frame synchronization can be achieved almost optimally by means of the fixed threshold, unique and simple strategy method in most telemetry applications. Frame sync acquisition can be minimized to less than one, on the average, by using longer frame sync codes in the range of 27 to 32 bits. The frame sync code pattern does not appreciably affect the performance as long as it is pseudo-random.

References

1. Jesse L. Maury and Frederick J. Styles, “Development of Optimum Frame Synchronization Codes for GSFC PCM Telemetry Standards.” Proceedings of NTC, 1964.
2. Peter E. Mallory, “Methods of Frame Sync Analysis,” Proceedings of NTC, 1971.
3. R. H. Barker, “Group Synchronizing of Binary Digital Systems,” Signals Research and Development Establishment, Christchurch, England.
4. D. D. McRae and F. A. Perkins, 11PCM Synchronization and Demultiplexing Study,” Final Report, prepared by Radiation, Inc. for NASA, NAS 8-5111.
5. “Formulation of Design Rules for PCM Telemetry Frame Synchronization,” Final Report, Oct. 1970, prepared by Computer Transmission Corp. for NASA-GSFC, NAS 5-21196.

Table 1. -Measured P_{FS} vs P_{FS} in Ref. 1

FSC	K	E_S	P_{FS}	P_{FS} (Ref. 1)
$(746500)_8$	18	2	80%	0.08%
$(765514600)_8$	27	2	2%	0.00016%
$(7657146400)_8$	30	2	1%	0.00002%
BER = 10%				

Table 2. -FSC vs. P_{FS}

FSC	K	P_{FS}
1. $(70516614)_8$	24	4%
2. $(76551460)_8$	24	3%
3. $(76571440)_8$	24	3%
4. $(07315701)_8$	24	4%
5. $(77007700)_8^*$	24	10%

*This code is not pseudo-random.
Frame sync strategy: First hit.
BER = 4%

Table 3. -P_{FS} vs. FSC Length

P_{FS} (%)	K (code length)	E_S
10	18	1
7	20	2
4	22	2
2.5	24	2
2	27	2
1	30	2
BER = 10%		
Frame sync accepted on first hit.		

Table 4. -P_{FS} vs Frame Length

P_{FS} (%)	L (bits)	K (bits)
1	1000	30
1	2000	30
2	4000	30
4	1000	21
15	4000	21
2	1000	24
5	4000	24

Table 5. -Frame Sync Performance

BER	K	V_S	V_L	FSA	FSM
$< 4\%$	> 24	0	0	1	100%
$< 10\%$	> 27	0	0	< 2	99.9%
$< 10\%$	> 16	1	1	< 3	100%
$< 5\%$	> 16	1	1	< 2	100%

Conditions:

1. FSC is pseudo-random
2. Data is random
3. Frame Length up to 4000 bits
4. $E_S \leq (BER)K$
5. $E_L < 0.4K$

Notes:

1. V_S - verification in search
2. V_L - verification in lock