MERGING TELEMETRY DATA FROM MULTIPLE RECEIVERS

Michael J. Wilson
Weapons and Materials Research Directorate
US Army Research Laboratory
Aberdeen Proving Ground, MD

ABSTRACT

Multiple receiver telemetry systems are common in the aeroballistics test and evaluation community. These systems typically record telemetry data independently, requiring post-flight data processing to produce the most accurate combination of the available data. This paper addresses the issues of time synchronization between multiple data sources and determination of the best choice for each data word. Additional filtering is also developed for the case when all available data are corrupted. The performance of the proposed algorithms is presented.

KEY WORDS

Merging Telemetry Data, Telemetry Data Processing, Multiple Receivers

INTRODUCTION

The United States military is a major employer of telemetry systems for test and evaluation purposes. Such systems typically aquire data from various sensors, such as accelerometers, magnetometers, and GPS, and then transmit that data from the projectile or moving body to a ground station. The sensor readings are sampled and quantized for digital transmission. In practice, the telemetry community relies on legacy PCM/FM systems [1,2], usually without any channel coding or equalization. As a result of such a system, telemetry data are prone to bit errors that must be dealt with.

To add redundancy, multiple receivers are used, which connect to antennas that cover different portions of the projectile flight. This helps to compensate for problems an individual receiver may experience. Multiple receivers result in multiple sources of the telemetry data. It is therefore desirable to combine the multiple data files into one master file that represents the best possible combination of the data. This results in many challenging signal processing problems which this paper addresses.

The first problem associated with combining telemetry data from multiple sources is identifying the same data across different sources. A receiver will use a time source to stamp each frame with the time when it was received. Because of the ambiguity associated with these times, a periodic counter is included within a frame to help identify that frame across multiple sources. Section 2 discusses how this identification is made.

Once multiple sources are identified as having the same frame of data, it must be decided which source of data to use for each word in the master. Since oversampling is largely employed, the correlation between successive samples is utilized in determining which source contains the best data. This method is explained in section 3.

Even with multiple sources for a word, the final choice for the master word may still be corrupted by noise. Bit errors introduced by wireless channels result in unusual discrete noise densities that depend on the number of bit differences between the transmitted and received symbol. An error correcting methodology using a Bayesian estimate for this type of noise in conjunction with an oversampling assumption is derived in section 4. The Bayes estimate allows a priori information to improve the estimate.

TIMING

Frame counters or identifiers can be included with each from of data. These frame counters are used to identify the same frame across different receivers. Let c_{frame} be an N_c bit counter(usually the size of a word) that wraps to zero when it overflows. At a sampling frequency of F_s, the counter repeats every $2^{N_c}(1/F_s)$ seconds. For typical sampling frequencies are $\sim$3 kHz with flight times of $\sim$15 seconds, the frame counter is expected to overflow and wrap many times. Each set of consecutive frames with $c_{frame} \in [0, 2^{N_c} - 1]$ is called a major frame. Because several major frames are transmitted, absolute time values cannot be determined with c_{frame}. A typical plot of c_{frame} is shown in Fig. 1(a). Notice the received frame counter includes both dropouts and erroneous values.

The first steps in using the frame counter to determine the time associated with a frame of data is to identify and eliminate errors with the frame counter. To begin, assume the data is received in the order it was transmitted. In other words, no frames are out of order. Therefore, the correct form of the frame counter is assumed to be monotonically increasing except when it wraps. The increase between successive samples may be greater than one if a dropout occurs. Note that dropouts are capable of producing any c_{frame} time series as well as the type assumed here. However, high signal-to-noise ratio (SNR) implies a low probability of deviation from the assumed form of c_{frame}. Also, received data that do not reflect the assumptions made here are

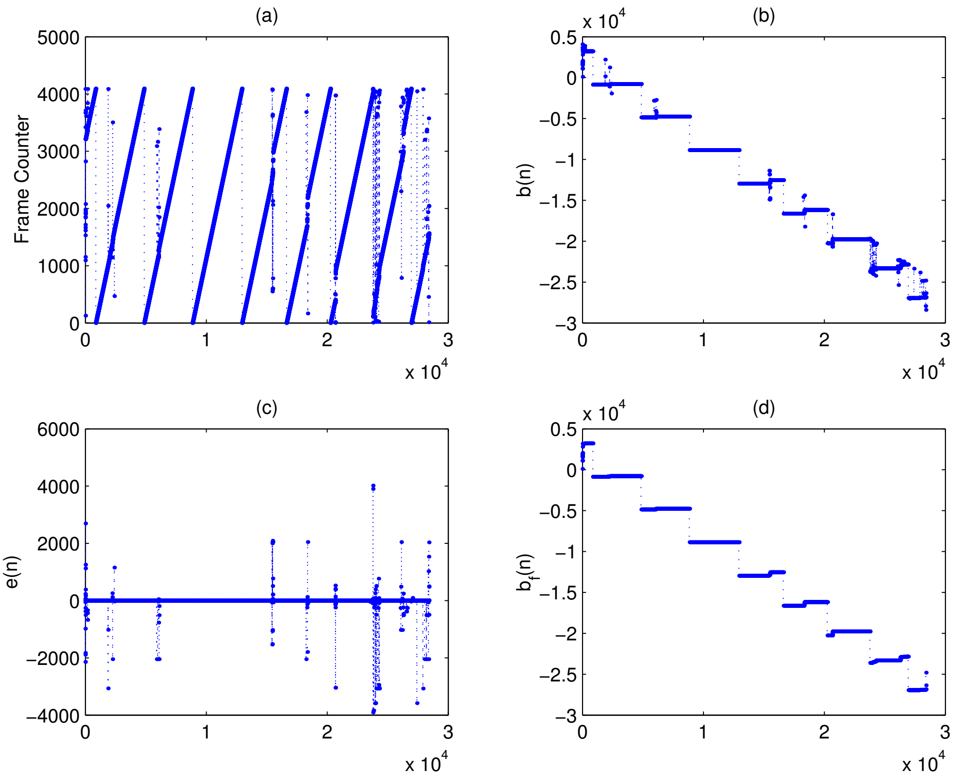

Fig. 1. Frame counter example: (a) c_{frame}, (b) $b(n)$, (c), $e(n)$, and (d) $b_f(n)$

fundamentally useless. In order to detect errors in c_{frame}, a simple transform of the data and a median smoother are employed.

Let n be a discrete time index for the received frames with 0 corresponding to the first frame, and let $b(n)$ equal the difference between $c_{frame}(n)$ and n, therefore

$$b(n) = c_{frame}(n) - n. \tag{1}$$

The structure of b can help identify errors. To see how, consider a plot of b (Fig. 1(b)) for the frame counter in Fig. 1(a). Notice that single errors in the frame counter are transformed into impulses in b. Dropouts produce a positive step whereas wraps produces a negative step. Because of the local monotonicity of the steps, b is invariant to a median smoother with a properly chosen window length. To see this, consider the median smoother defined as

$$y(n) = \text{median}[x(n - N_1), ..., x(n), ..., x(n + N_1)]. \tag{2}$$

The median operation sorts the samples by magnitude in the window of length N_m, where N_m is any positive odd integer, and selects the middle sample. The selection property is desirable since the only possible outputs of the smoother are unsigned integer values. This eliminates the need for any requantization of the outputs. If the samples in the window are monotonically increasing or decreasing, the center sample in the window is the output of the smoother. Therefore, a window is selected that is larger than an error run and smaller than time between local steps, a median smoother will change the error points without changing the correct points.

The following algorithm is used to detect errors in the frame counter:

1) Compute $b(n)$, using Eq. (1).
2) Smooth $b(n)$ using a median smoother with a properly chosen window length.
3) Repeat step 2 multiple times, depending on the noise level.
4) Let $b_f(n)$ be the output of the last smoother (see Fig. 1(d)) and compute the difference $e(n) = b(n) - b_f(n)$.
5) An error is expected to occur in $c_{frame}(n_0)$ if $e(n_0) \neq 0$.

The output after each smoother stage will be invariant to additional smoothing if it consists only of step functions that change slowly with respect to the window length of the smoother. Figure 1(c) shows the error $e(n)$ in the example.

Because of the ambiguity due to multiple wraps in the frame counter, additional information must be utilized in order to identify the same frame in different sources. Fortunately, most telemetry receivers will use a reference time base such as GPS to stamp each frame with the time it was received, t_{RX}. Now consider how t_{RX} relates to the actual time the data in the frame was acquired, t_{TX}. Define the time difference,

$$D_s(n) = t_{RX,s}(n) - t_{TX,s}(n), \tag{3}$$

for each source, s, which is, in general, a function of the discrete time index n. $D_s(n)$ depends on the relative positions of the receiver and the projectile. Therefore, different times are assigned to the same frame across multiple receivers. However, if the time errors that result satisfy

$$D_s(n) \ll \frac{2^{N_c}(1/F_s)}{2} \tag{4}$$

for all s, T_{RX} can be used in conjunction with the frame counter to identify the same frame across multiple sources. Reference times for all frames created in the master are then generated with two parameters, the time of the first frame, $t(0)$ and the sampling period, T_s, as

$$t(n) = t(0) + nT_s. \tag{5}$$

The parameters are computed as

$$T_s = \frac{1}{N-1} \sum_{n \in S_1} [t_{RX}(n+1) - t_{RX}(n)], \tag{6}$$

where S_1 is the set of all indexes such that $T_{RX}(n+1)$ and $T_{RX}(n)$ exist, and

$$t(0) = \frac{1}{N} \sum_{n \in S_2} [t_{RX}(n) - nT_s]. \tag{7}$$

where S_2 is the set of all indexes such that $T_{RX}(n)$ exists.

CREATING THE MASTER WORD TABLE

After a master time base is created and the frames from each source that correspond to the time points are identified, the master word table assembly can begin. When a frame of data is transmitted, it is possible that all or most of the receivers receive the frame. With the timing discussed in the previous section, specific data from each source as representing the same frame can be identified.

Suppose there are N_S sources with word tables $\Phi_{s_1}, \Phi_{s_2}, ..., \Phi_{s_{N_S}}$, and let $\Phi_s(n, w)$ denote the wth word in frame n from source s. An empty master word table, Φ_m, is now created with frame times that span all the sources' times. The elements in Φ_m must be generated for each word in each frame as a function of all the data available:

$$\Phi_m(n, w) = f(\{\Phi_{s_1}, \Phi_{s_2}, ..., \Phi_{s_{N_S}}\}) \tag{8}$$

First, note that is it highly improbable for the transmission errors to change in exactly the same way the same word in different sources. Therefore, the occurrences of the same word across different sources can be exploited. Let $Q(n, w)$ denote the mode (element that occurs the most often) or set of modes (if a unique mode does not exist) defined as

$$Q(n, w) = \mathrm{mode}(\Phi_{s_1}(n, w), \Phi_{s_2}(n, w), ..., \Phi_{s_{N_S}}(n, w)) \tag{9}$$

If there is only one mode, set $\Phi_m(n, w) = Q(n, w)$, otherwise, an additional criterion must be used to choose from the set.

It will be shown is section 4 that the transmission noise is impulsive. As such, erroneous data tends to be far away in magnitude from its true value. Also, sensor outputs are generally oversampled, which produces highly correlated data. Thus, an erroneous datum tends to be far away in magnitude from the words in neighboring frames. Therefore, the distance between the word of interest and its neighbors in different frames can be exploited as a criterion for choosing the master word.

Let d_q be a distance metric for each $q \in Q(n, w)$. $\Phi_m(n, w)$ is then defined as

$$\Phi_m(n, w) = \underset{q}{\arg\min} \; \{d_q : q \in Q(n, k)\}. \tag{10}$$

The square differences in magnitudes of neighboring points are used to calculate each distance as

$$d_q = \frac{1}{N_{S_q}} \sum_{s \in S_q} \sum_{k=n-N_p}^{n+N_p} \frac{1}{|n-k|} [\Phi_s(n, w) - \Phi_s(k, w)]^2 \tag{11}$$

where S_q is the set of sources in which q occurs, N_{S_q} is the number of sources in the set, S_q, and N_p is the number of neighbors to each side of the current word. $N_p = 1$ provides good results in high SNR. This step completes the building of the master word table with timing information. Next, the individual words are filtered to eliminate any remaining errors.

DE-NOISING WITH BAYESIAN ESTIMATION

Even with multiple sources of word values, errors can still be present in the master. Therefore, a filtering approach to denoise the word values is derived by analyzing the channel and methods that generated the errors. This method is applicable when additional information about the data is known a priori. The example considered here is when the signals are oversampled.

To begin, consider a binary symmetric channel (BSC) such that the output $z(n)$ is defined as

$$z(n) = x(n) \oplus y(n), \tag{12}$$

where $x(n) \in \{0, 1\}$ is the input sequence to the BSC, and $y(n) \in \{0, 1\}$ is an independent and identically distributed sequence of Bernoulli random variables with parameter $p = p_e$, the bit error rate. The BSC models the effect of the wireless channel. The BSC is used to transmit unsigned integer words or symbols that are M bits long. The words, $a(n)$, are converted directly into their binary representation, sent through the BSC, and the output of the BSC is converted back to create the new output sequence, $b(n)$. The combined system is therefore a discrete memoryless channel (DMC) [3] that is completely described by a set of conditional probability functions. The bit error rate of the DMC causes the symbols to change. The probability of receiving any b when a symbol a is transmitted depends on the hamming distance of a and b defined as

$$d(a, b) = \sum_{i=1}^{M} \alpha_i \oplus \beta_i \tag{13}$$

α_i and β_i are the ith bits of a and b, respectively, and $\oplus$ is the Boolean XOR or modulo-2 addition. It is possible to minimize errors using error correcting codes; however, many telemetry systems do not employ coding because of hardware complexity. Bit Errors produced from this model result in a very unusual conditional output probability density, which is now derived.

Given the input symbol to the previously described channel is a, the probability density function (pdf) of the output, B, is

$$f(b|a) \equiv P(B = b|A = a) = p_e^{d(a,b)}(1 - p_e)^{M-d(a,b)} \tag{14}$$

where $d(a, b)$ is the hamming distance between a and b. Since each of the bits received are assumed independent, Eq. (14) follows from that the fact that $d(a, b)$ bits change, and $M - d(a, b)$ bits remain the same in order for the channel to change a into b. Equation (14) is similar to a binomial distribution, but no counting term is needed since only one possible combination is accounted for. Figure 2 shows Eq. (14) for an a of 2047, 2048, and 2730, $p_e = 0.3$, and the number of bits per word equal to 12. Notice the densities are not symmetric or semi-monotonic. The peaks or high points correspond to errors in only a few bits. That error may cause a large change in the output if the bit is significant. Likewise, small changes in the output may require more bit errors and are therefore not as probable.

If the data source was assumed memoryless, that is, each symbol generated is independent of all the other symbols, a maximum likelihood estimate would be optimal [3]. However, such an estimate would not change the received data since no a priori information is utilized. Therefore,

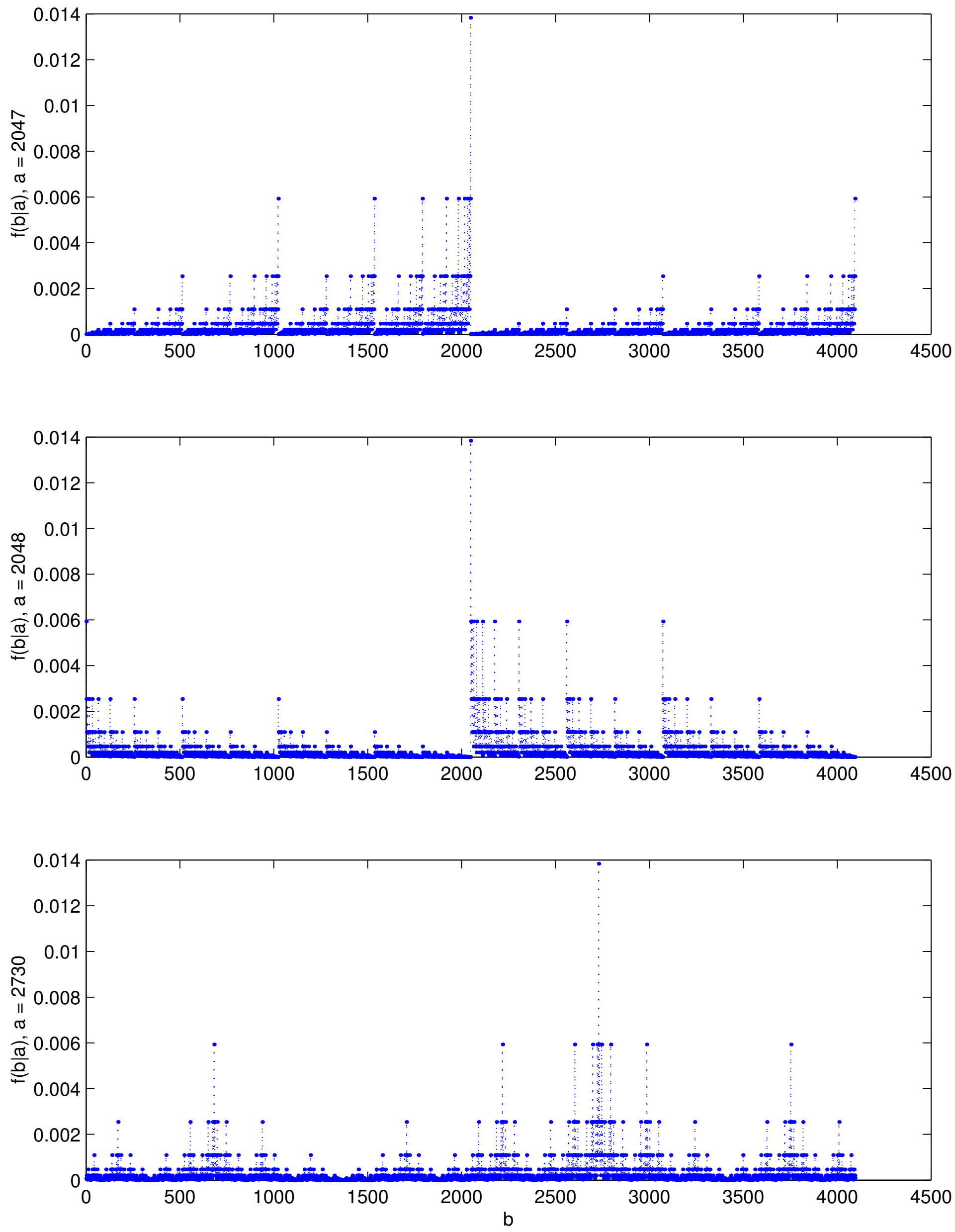

Fig. 2. $f(b|a)$ for different a when the number of bits per word is 12

a Bayesian maximum a posteriori (map) estimate approach is considered. In order to compute the joint pdf of a and b, the a priori pdf of a must be determined. If the data are oversampled, the current sample value should be very close to the preceding and succeeding sample values. Therefore, the pdf of the input sample can be estimated as a Gaussian distribution with mean,

$$\hat{\mu} = \text{median}[b(n-N_p), ..., b(n-1), b(n), b(n+1), b(n+2), ..., b(n+N_p)] \tag{15}$$

where N_p is the number of points to the left and right, and variance, $\hat{\sigma}^2$ equal to a constant that

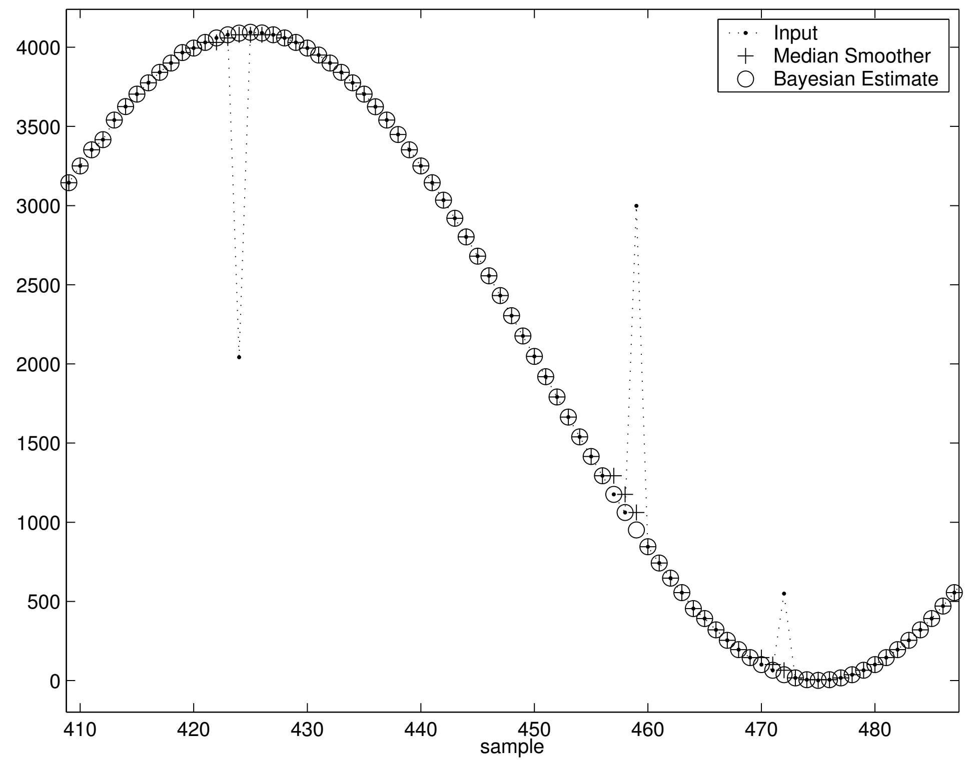

Fig. 3. Bayesian estimation example of a sine wave corrupted by noise

depends on the amount of oversampling. The map estimate is defined as

$$\hat{a} = \underset{a}{\arg\max}\ f(a|b). \tag{16}$$

In words, since the output, b, is known, the best estimate of a is the one that is most probable to occur given b. Since,

$$f(a|b) = \frac{f(a,b)}{f(b)}, \tag{17}$$

and $f(b)$ is not a function of a, maximizing the numerator of Eq. (17) is equivalent to Eq. (16). Now,

$$f(a,b) = f(b|a)f(a). \tag{18}$$

Substituting Eq. (14) into Eq. (17) and setting $f(a)$ equal to a Gaussian density with parameters $(\hat{\mu}, \hat{\sigma}^2)$ yields

$$f(a,b) = p_e^{d(a,b)}(1-p_e)^{M-d(a,b)} \frac{1}{\sqrt{2\pi\hat{\sigma}^2}} \exp\left[\frac{(a-\hat{\mu})^2}{\hat{\sigma}^2}\right]. \tag{19}$$

The estimate is therefore,

$$\hat{a} = \underset{a}{\arg\max}\ p_e^{d(a,b)}(1-p_e)^{M-d(a,b)} \exp\left[\frac{(a-\hat{\mu})^2}{\hat{\sigma}^2}\right]. \tag{20}$$

For implementation purposes, the first two factors in Eq. (20) are computed and stored for all possible values while the exponential factor is computed at runtime. The estimate is computed for each $\Phi_m(n, w)$. Figure (3) shows the performance of the estimate for a sine wave contaminated by the previously described channel with $p_e = 0.01$. Notice that the median smoother is successful in removing the impulsive noise but does not correct to the most probable values like the Bayesian estimate.

PERFORMANCE

To demonstrate the effectiveness of the algorithms presented, a merge of three source files is shown. Figure 4 plots an individual word generated by a sensor for each source and the master. Notice source 1 has clean data but contains large gaps in the beginning and the middle. Source 2 does not have any large gaps but the noise increases with time. The data from source 3 have many small gaps and a substantial amount of noise but contains some frames not present in sources 1 or 2. The algorithm successfully merges all three sources to produce a continuous set of data with no gaps and practically no impulse noise.

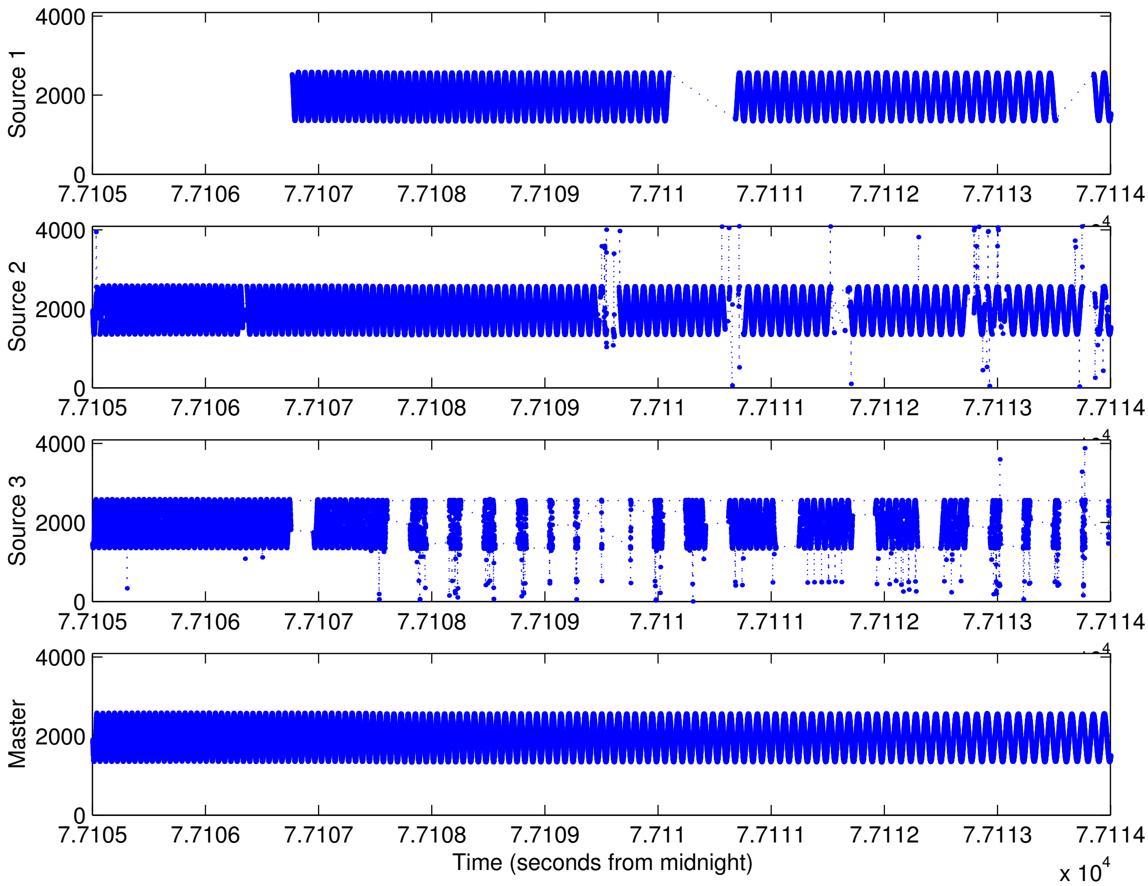

Fig. 4. Word value plot for source merging example: 3 sources

CONCLUSION AND IMPACT

The problems associated with combining multiple telemetry data sources have been presented. The previous algorithms are currently employed with the DFuze [4] telemetry system in several US Army and Navy programs. The implementation of the algorithms replace several days of “manual” processing with an program that runs in a few seconds. The proven methods result in virtually error free master files despite substantial noise presence in the received data. The algorithms presented here are completely general and can be applied to other applications with multiple telemetry receivers.

REFERENCES

[1] Secretariat, Range Commanders Council, *Telemetry Standards*, White Sands Missile Range, NM: RCC, IRIG 106-00.
[2] F. Carden, R. Jedlicka, R. Henry, *Telemetry Systems Engineering*, Artech House, 2002.
[3] J. G. Proakis, *Digital Communications*, 4th Ed., McGraw Hill, 2001.
[4] D. J. Hepner et. al., “Aeroballistic diagnostic fuze,” U.S. Patent No. 6349652.