A novel algorithm for real-time adaptive signal detection and identification

Gerard E. Sleefe, Mark D. Ladd, Daniel E. Gallegos, Carl W. Sicking, and Ireena A. Erteza

Sandia National Laboratories, MS 0843, Albuquerque, NM 87185-0843

ABSTRACT

This paper describes a novel digital signal processing algorithm for adaptively detecting and identifying signals buried in noise. The algorithm continually computes and updates the long-term statistics and spectral characteristics of the background noise. Using this noise model, a set of adaptive thresholds and matched digital filters are implemented to enhance and detect signals that are buried in the noise. The algorithm furthermore automatically suppresses coherent noise sources and adapts to time-varying signal conditions. Signal detection is performed in both the time-domain and the frequency-domain, thereby permitting the detection of both broad-band transients and narrow-band signals. The detection algorithm also provides for the computation of important signal features such as amplitude, timing, and phase information. Signal identification is achieved through a combination of frequency-domain template matching and spectral peak picking. The algorithm described herein is well suited for real-time implementation on digital signal processing hardware. This paper presents the theory of the adaptive algorithm, provides an algorithmic block diagram, and demonstrate its implementation and performance with real-world data. The computational efficiency of the algorithm is demonstrated through benchmarks on specific DSP hardware. The applications for this algorithm, which range from vibration analysis to real-time image processing, are also discussed.

Keywords: detection, identification, signal processing, adaptive, algorithm

1. INTRODUCTION

The detection and identification of signals buried in noise is of widespread importance [1]. Traditional detection and identification methods have focused on detecting/estimating known signals buried in stationary noise. The problem is further complicated by noise environments which are non-stationary [2]. Non-stationary noise environments require the use of adaptive detection and estimation schemes to compensate for the time-varying noise statistics. In recent years, the need for adaptive detection and identification algorithms that can operate in real-time on dedicated digital signal processing (dsp) hardware has become critical for many applications. Applications for real-time adaptive signal detection and identification include; vibration control, sonar target analysis, image processing, and automatic radar target recognition. Hence, this paper focuses on an adaptive detection/identification algorithm suited for such real-time applications.

In this paper, we describe the details of a digital signal processing algorithm for detecting and identifying signals buried in noise. The algorithm described herein is most appropriate for the following two cases: 1.) when the signal is a relatively broadband transient waveform, and the background noise consists of random noise and several interfering sinusoidal tones; and 2.) when the signal of interest consists of one or more sinusoidal tones that are obscured by both broadband random noise and other interfering sinusoidal tones. The technique is intended for efficient implementation on digital signal processing hardware.

This paper begins with a description of the algorithmic approach. The approach includes signal processing in the time-domain for detecting transient signals, and signal processing in the frequency-domain for detecting tones. The paper then provides details on the adaptive nature of both the time-domain and frequency-domain techniques. The adaptive behavior of the algorithm is achieved through both matched filtering and adaptive statistical thresholding. The paper continues with a discussion of the real-time implementation of the algorithm, including example benchmark performance on the Texas Instrument's TMS320C3x DSP. In the last section of the paper, examples of the algorithm are provided using typical noise-

corrupted data. The paper concludes with a brief summary of the algorithm, its nominal performance, and intended applications.

2. PROBLEM DEFINITION AND ALGORITHMIC APPROACH

In this paper, we consider a system whose input is a discrete-time series and represented by y[n]. The objectives of this digital signal processing system are to 1.) detect and identify transient signals buried in time-varying noise; and 2.) detect and identify the switching on or off of distinct sinusoidal tones. In certain applications, only one of these two objectives may be required. For generality, we consider the case where both objectives are of interest. The digital signal, y[n], can then be represented as:

$$y[n] = x[n - T_0] + \sum_i A_i \sin(\omega_i n + \phi_i) w[n, T_{1i}, T_{2i}] + \sum_j B_j \sin(\Omega_j n + \Psi_j) w[n, \tau_{1j}, \tau_{2j}] + z[n] \tag{1}$$

where x[] represents a transient signal starting at time T_0. The term, $A_i \sin(\omega_i n + \phi_i)$, represents sinusoidal signal tones with amplitude A_i, frequency ω_i and phase ϕ_i, The term $B_j \sin(\Omega_j n + \psi_j)$ represents interfering sinusoidal noise tones with amplitude B_j, frequency Ω_j, and phase ψ_j. The function, z[n], represents random non-stationary noise. The function w[n] is a digital window function defined as:

$$w[n, T_1, T_2] = \begin{cases} 0; & n < T_1, n > T_2 \\ 1; & T_1 \le n \le T_2 \end{cases} \tag{2}$$

Thus w[] is used to indicate that the signal tones and interfering tones start and stop at arbitrary times.

Based on Eq. 1, the objectives of the proposed digital signal processing algorithm are to 1.) detect and identify the transient signal x[] in the presence of known tones, interfering tones, and random non-stationary noise; and 2.) to detect and identify the occurrence of sinusoidal signal tones in the presence of interfering tones and random non-stationary noise.

The general algorithmic approach is depicted in Figure 1. The first step in the algorithm is to capture a frame of digital data. The digital data is then calibrated depending on the needs of the signal processing application. At this point, the processing is dividing into two paths: transient signal detection and identification; and periodic signal detection and identification. These two signal processing paths are done in separate domains: transient signal processing is performed in the time domain; and periodic signal processing is performed in the frequency domain. Once in their respective domains, the processing paths utilize similar functions to perform adaptive signal detection and template matching (identification). The next sections of this paper describe these signal processing functions in detail.

3. TRANSIENT SIGNAL DETECTION AND IDENTIFICATION

3.1 Transient signal enhancement

Transient signal enhancement is performed through optimized digital filtering of the received signal, y[n]. The objectives of the signal enhancement are to enhance the transient component, x[n], by removing interfering sinusoidal tones, and by removing broadband noise that is outside the bandwidth of the signal of interest. Three different techniques have been implemented for transient signal enhancement. The enhancement technique utilized will depend on the specific application and the *a-priori* knowledge of the signal and noise characteristics. The three enhancement techniques are: 1.) non-adaptive digital filtering; 2.) partially adaptive digital filtering; and 3.) adaptive predictive filtering.
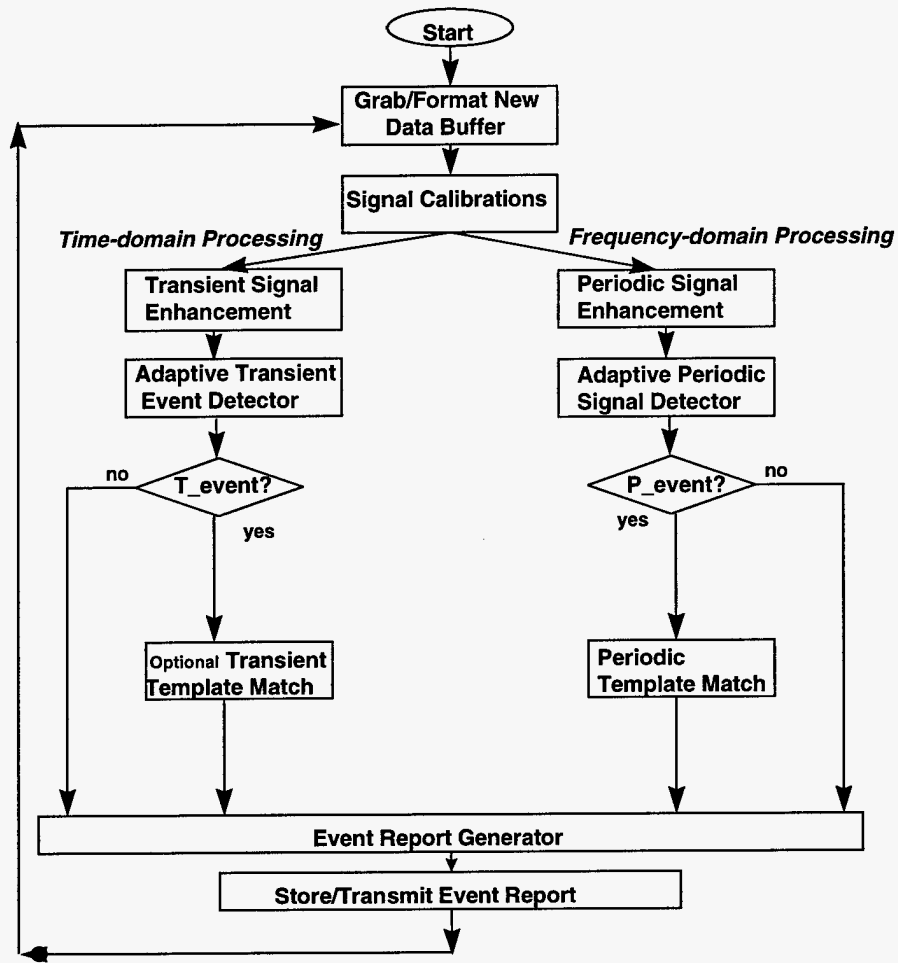
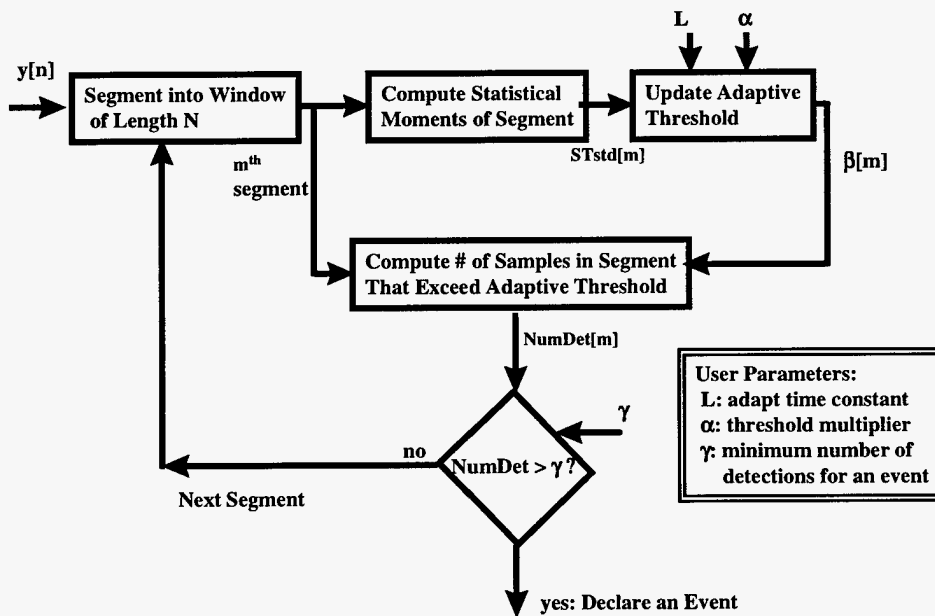**Figure 1: Real-time Signal Detection and Identification Algorithm**

Figure 2: Adaptive Transient Event Detection Algorithm

The non-adaptive digital filtering technique is most applicable when there is considerable *a-priori* information on the signal and/or noise characteristics. Optimally, the non-adaptive filtering should include a pre-whitening filter to equalize the background noise and a digital matched filter to enhance the signal components. In many applications, however, the transient signal characteristics are unknown, and hence matched filtering is not suitable. If the characteristics of the noise are well known, then simple digital band-stop filters can be used to enhance a variety of transient signals. Non-adaptive band-stop digital filters can be designed in advance using standard digital filter design techniques [3]. Suitable non-adaptive digital filters include both Infinite Impulse Response (IIR) and Finite Impulse Response (FIR) filters. For example, IIR notch filters are computationally efficient and are extremely effective for removing interfering tones at known frequencies. Partially adaptive digital filtering is accomplished by incorporating the digital filter design technique into the real-time algorithm. For example, the algorithm would occasionally estimate the frequency of the interfering tone(s), and then automatically design an IIR filter to notch the interfering tones.

In many applications, there is only limited *a-priori* information available on the background noise and interfering tones. Furthermore, the background noise may have significant time-varying characteristics. In this case, an adaptive signal enhancement technique is required. A computationally efficient and popular technique is the Least Mean Squares (LMS) adaptive filter. The LMS algorithm adaptively builds an FIR filter that cancels background noise. Detailed information on the LMS algorithm can be found in Reference [4].

3.2 Adaptive transient event detector

An adaptive statistically-based event detection algorithm has been developed and implemented. The algorithm continually computes the statistics of the background noise and determines an adaptive threshold. The technique is depicted in Figure 2. The algorithm segments the incoming signal, y[n] into buffers of length N. At this point the first and second moments of y[n] are computed. For simplicity, assume that the incoming signal has a zero mean. Then a 'Short Term Standard Deviation', or STstd can be computed as:

$$STstd = \sqrt{\frac{1}{N} \sum_{k=1}^{N} y^2[k]} \tag{3}$$

A 'Long-Term Standard Deviation', or LTstd can be computed iteratively from successive segments of y[n]:

$$LTstd[m] = \frac{1}{L} STstd[m] + \frac{L-1}{L} LTstd[m-1] \tag{4}$$

where m is an index on successive segments of y[n], and L is the time duration over which the algorithm will adapt. An adaptive threshold is updated for each segment of y[n]. The adaptive threshold, $\beta[m]$, is determined as follows:

$$\beta[m] = \alpha * LTstd[m-1] \tag{5}$$

Where α represents the number of standard deviations required to declare a detection. If the background noise is Gaussian distributed, then the user can judiciously select α to achieve a desired probability of detection with a desired probability of false alarm. The next step in the adaptive threshold algorithm is to determine the number of samples in the current window that exceed the adaptive threshold. The number of detected samples in the window, NumDet, is computed as:

$$NumDet = \sum_{k=1}^{N} BinSig[k] \tag{6}$$

where

$$BinSig[k] = \begin{cases} 1; & |y[k]| \geq \beta[m] \\ 0; & |y[k]| < \beta[m] \end{cases} \tag{7}$$

The last step in the algorithm is to determine whether or not to declare in an event based on the computed value of NumDet:

If NumDet $\geq \gamma$ Then Declare a Transient Event Detection (8)

Where γ is the minimum number of detections in a segment required to declare an event. The transient event detection algorithm described above is both computationally efficient and robust. There are three user-selectable parameters in the algorithm: the time frame over which the algorithm adapts, L; the primary threshold-multiplier relative to the background noise standard deviation, α; and the secondary threshold for the minimum number of detections required in a window before declaring an event, γ.

3.3 Transient Signal Identification

Identification of a transient signal is achieved using the maximum likelihood method (MLM). With this method, the received signal is passed through a bank of digital matched filters. Each matched filter represents a transient signature that we are attempting to identify. The outputs of the bank of matched filters are compared to determine which transient signature provides the best match. In many applications, however, deterministic transient signatures are not available. In this case, signal identification is best performed in the frequency domain, as described in subsequent sections.

4. PERIODIC SIGNAL DETECTION AND IDENTIFICATION

Periodic signal detection can be accomplished by using the fast Fourier transform (FFT) and a simple signal to noise estimator. Figure 3 below shows a block diagram of an algorithm that can detect a change in the power spectral estimate for an FFT bin, and as such, determine when a periodic signal turns on or off.
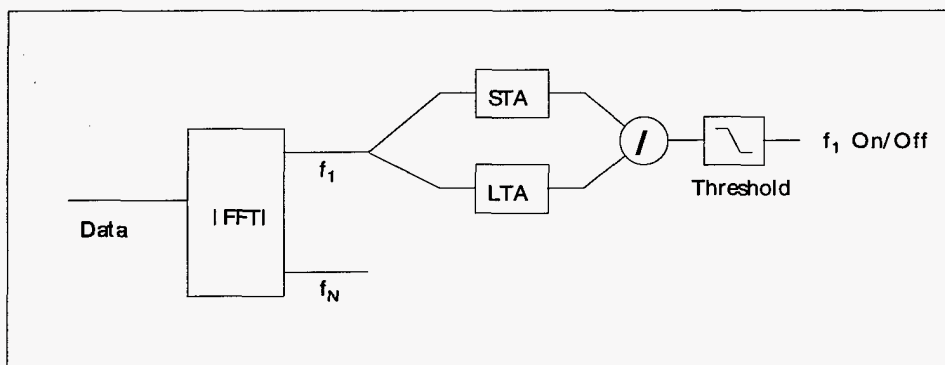

Figure 3: Tone Detector.

The Bartlett method for estimating the power spectrum of a data segment is used. Given a data buffer, the data is segmented into J*K non overlapping partitions of length N:

$$x_{jk}(t_n) = x(t_n + jb_0 + kb_1),$$
$$n = 0,1,...,N-1,$$
$$j = 0,1,...,J-1$$
$$k = 0,1,...,K-1.$$
(9)

The periodogram estimate for a segment is formed using a windowing function

$$\hat{X}_{j,k}(f_l) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x_{j,k}(t_n)W(t_n)exp\left(\frac{-i2\pi f_l t_n}{N}\right)\right|^2 \tag{10}$$

and the short term average (STA), or the power spectrum estimate, is formed by averaging the periodogram estimates

$$STA_k(f_l) = \hat{X}_k(f_l) = \frac{1}{J}\left|\sum_{j=0}^{J-1}\hat{X}_{j,k}(f_l)\right|. \tag{11}$$

The long term average (LTA) is formed by using a recursive method

$$LTA_k(f_l) = \frac{1}{M}STA_k(f_l) + \frac{M-1}{M}LTA_{k-1}(f_l) \tag{12}$$

where the first LTA value is set to the first STA value.

The ratio of the STA to the LTA is taken to form the detection statistic for each frequency f_l. If the ratio exceeds a fixed threshold (greater than one) a turn on is declared since there is a significant increase in power for that FFT bin. One can also detect when a tone turns off by assessing when the ratio goes below a threshold less than one. Since the techniques described for the transient processing do not work as well for signal and noise statistics which are not zero mean we are exploring other techniques to make the tone detector's threshold adaptive.

For system identification, a similar technique to the tone detector is used. However, a template of the tones associated with a particular system is used to determine whether or not that system turned on or off. Figure 4 below shows a block diagram of an algorithm that uses the dot product of a template with the LTA of the signal's power spectrum to determine the identification.
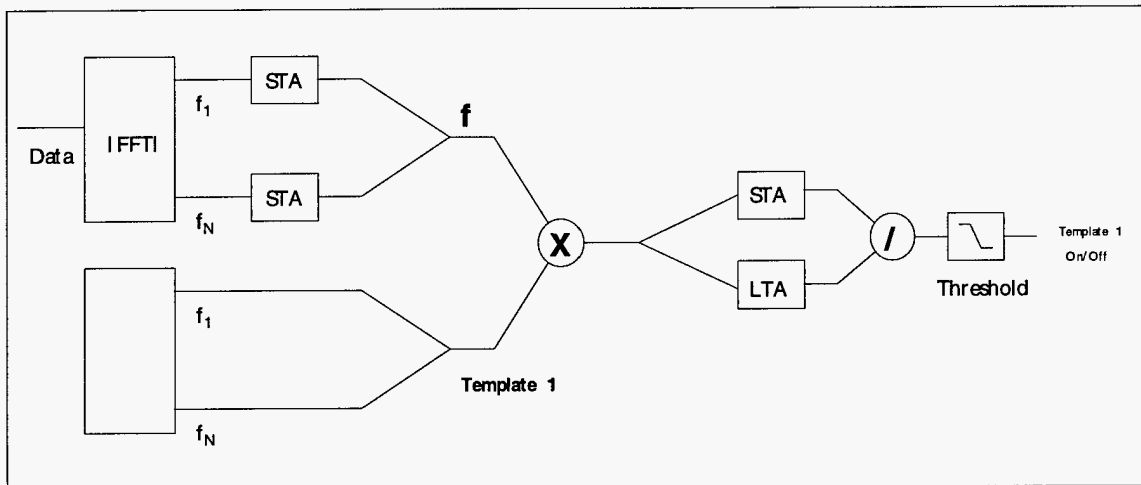

Figure 4: Signal Identification Using a Template.

The dot product is formed by

$$Dot(k) = \sum_{l=0}^{N-1} STA_k(f_l) * T_l(f_l) \tag{13}$$

where $T_1(f_l)$ is the template value at f_l. The template is a mask that identifies the frequencies associated for a particular system. If a system produces tones at certain frequencies f_{10} and f_{40}, a weight of one is given in the template at those frequencies. Neighboring frequencies can be assigned a smaller value if desired. The template is then normalized by the template's magnitude as indicated by

$$T_l(f_9, f_{10}, f_{11}) = \{0.7, 1.0, 0.7\} \qquad T_l(f_l) = \frac{T_l(f_l)}{\sqrt{\sum_{l=0}^{N-1} T_l(f_l)^2}} \tag{14}$$
$$T_l(f_{39}, f_{40}, f_{41}) = \{0.7, 1.0, 0.7\}$$

An STA and LTA of the dot product is formed as done in Section 4.0. The STA/LTA ratio is calculated and then compared to a threshold, and similar to the tone detector, turn ons and turn offs can be detected for a particular template. If there are a collection of systems to identify, one uses a bank of detectors where each detector uses a template designed for the system to be identified. However, it is important to make the templates as orthogonal as possible to each other; that is, use frequencies that don't overlap if at all possible. Section 6.2 below shows an example of this detection and identification algorithm.

5. REAL-TIME ALGORITHM IMPLEMENTATION

The algorithm described above is particularly well-suited for real-time applications. This algorithm has been coded for the Texas Instruments TMS320C3x floating point digital signal processor. The coding has been performed primarily using standard C-language code. Certain standard digital signal processing functions, such as IIR filters, FIR filters and the Fast Fourier Transform have been written in assembly language. Example benchmarks utilizing 512-sample segments and a 33 MHz TMS320C31 processor are provided in Table 1.

Table 1: Example Benchmarks of Algorithm on 33 MHz TMS320C31: 512-sample buffers

Function	Execution Time, microseconds
Transient Signal Enhancement: 4-pole IIR filter	990
Transient Signal Enhancement: 16th order LMS Filter	1850
Transient Event Detection:	115
Periodic Signal Enhancement	1850
Periodic Signal Detection	370
Periodic Template Match (per template)	65

6. RESULTS

6.1 Transient Signal Detection Results

In this section, we demonstrate the performance of the transient detection algorithm. In Figure 5a, a transient signal is shown. The transient signal is corrupted by both broadband random noise and an interfering sinusoidal tone. Figure 5b depicts the output of the transient signal enhancement algorithm. Note that the signal enhancement algorithm successfully removes the interfering sinusoidal tone. In Figure 5c, the STstd function is plotted. Note the smoothing of the random component of the background noise. Finally, Figure 5d depicts the output of the transient signal detector, BinSig. Note that the transient signal has been successfully detected.
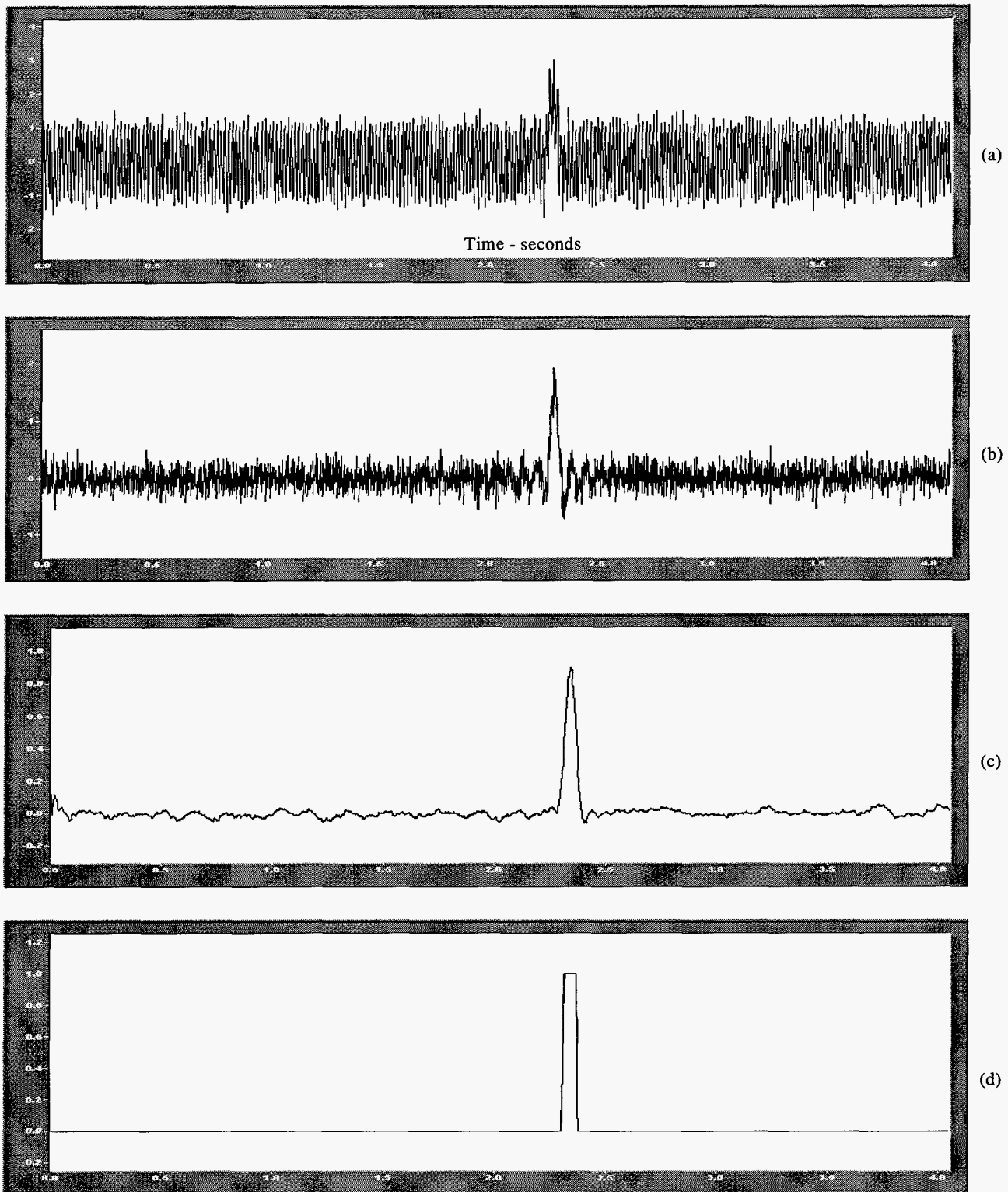
Figure 5. a) Transient signal corrupted by both broadband random noise and interfering sinusoidal tone, b) Signal after transient signal enhancement algorithm, c) Short-term average result, and d) Detection algorithm output.

6.2 Periodic Signal Detection and Identification Results

Two sine waves at 25 and 60 Hz are added to normal random white noise to create a test signal. The sine waves turn on after 7.68 seconds into the data record and turn off after 15.36 seconds (on an FFT processing boundary). The signal to noise ratio in the time domain is 12 dB in power (factor of 4). Figures 6 and 7 show the test signal in the time and frequency domain. A sampling rate of 200 Hz was used, N was set at 256, J at 2, and M at 5 for the processing.
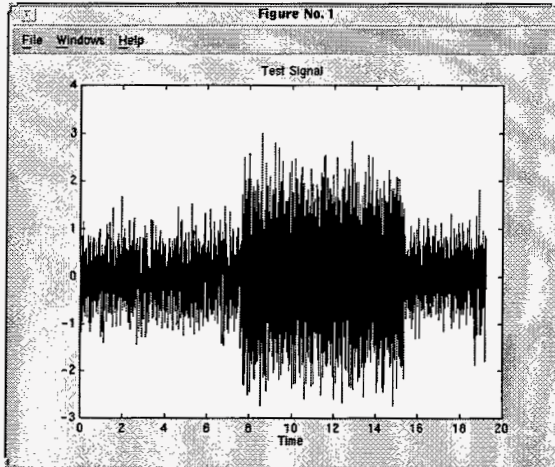
Figure 6: Test Signal, Time Domain
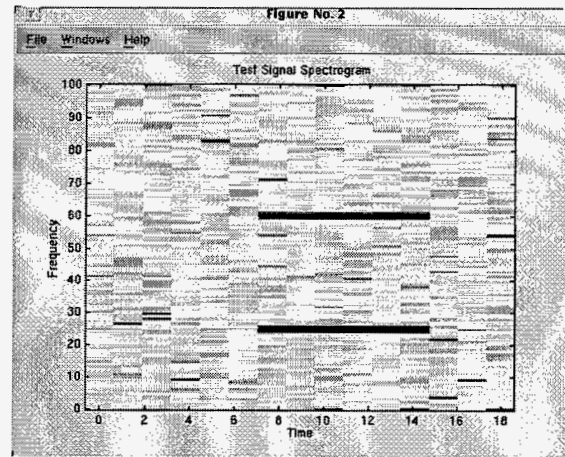
Figure 7: Test Signal, Spectrogram

A turn on threshold of 3.0 and a turn off threshold of 0.15 was selected for the tone detector. The template detector used 2.7 and 0.20 respectively. The thresholds were selected after running the simulation with different noise sequences. Requiring *a priori* knowledge of the signal and noise characteristics is an inherent disadvantage of using a fixed threshold instead of an adaptive threshold as done with the transient detector. The tone turn-on detections are tabulated in Table 2 and the tone turn-off detections are provided in Table 3.. Note that both the 25 Hz and 60 Hz tones were properly detected at turn-on and turn-off. Only one instance of a false alarm (at a frequency of 100 Hz) was observed.. Furthermore, note that in several instances, multiple detections occurred at slightly different times. Additional logic can be added to the detector that would require the detector to reach a value close to one before declaring another detection. This would eliminate the multiple turn-on and turn-off detections observed for the same event.

Table 4 provides the results for the template matching algorithm. Note that the template matching algorithm properly detects the turn-on and turn-off of the simultaneous tones. No false alarms or multiple detections were observed using the template matching algorithm.

Table 2: Tone Turn-On Detections.

Event Number	SNR value at detection	Frequency	Time
1	3.81	25.0	7.68
2	3.26	25.0	8.96
3	3.04	60.2	8.96

Table 3: Tone Turn-Off Detections

Event Number	SNR value at detection	Frequency	Time
1	0.0714	25.0	15.36
2	0.0367	25.0	16.64
3	0.1118	60.2	15.36
4	0.1197	60.2	16.64
5	0.1480	60.9	16.64
6	0.0899	100	2.56

Table 4: Template Detections.

Event Number	SNR value at detection	On/Off	Time
1	2.92	On	7.68
2	0.163	Off	15.36

7. CONCLUSIONS

In this paper, we have described a novel algorithm for adaptively detecting and identifying signals buried in noise. The algorithm continuously computes the statistics of the background noise and adapts its detection parameters. Detection and identification is performed in both the time-domain and the frequency-domain. This allows both broadband transients and narrow band signals to be detected and identified. Results presented in this paper demonstrate that the method performs well even under low signal-to-noise conditions. The algorithm has been implemented for real-time applications, and example benchmarks demonstrate the computational efficiency of the method. Recommendations for future work involve providing additional adaptive capabilities to the frequency-domain detection algorithm.

8. REFERENCES

1. H. L. Van Trees, *Detection, Estimation, and Modulation Theory,* J. Wiley and Sons, New York, 1968.
2. B. Widrow, S. Stearns, *Adaptive Signal Processing,* Prentice Hall, New Jersey, 1985.
3. A. Oppenheim, R. Schafer, *Digital Signal Processing,* Prentice Hall, New Jersey, 1975
4. J. Treichler, et al, *Theory and Design of Adaptive Filters,* J. Wiley and Sons, New York, 1987.

ACKOWLEDGEMENTS

DISCLAIMER

Report Number (14) _SAND--98-0840C_
CONF-980412--

Publ. Date (11) _199804_
Sponsor Code (18) _DoE/CR , XF_
UC Category (19) _UC-900 , DOE/ER_

DOE