



P.O. Box 627
Ellicott City, MD 21041
<http://CausaSci.com>

Analysis and Design of Sliding *m-of-n* Detectors

Report Number 2011-01

October 10, 2011

D. A. Abraham

This work was sponsored by the Office of Naval Research
under contract number **N00014-09-C-0097**.

UNCLASSIFIED

approved for public release

Report Documentation Page				Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.						
1. REPORT DATE OCT 2011		2. REPORT TYPE N/A		3. DATES COVERED -		
4. TITLE AND SUBTITLE Analysis and Design of Sliding m-of-n Detectors				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) CausaSci LLC Ellicott City, MD 21041				8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited						
13. SUPPLEMENTARY NOTES Report Number 2011-01., The original document contains color images.						
14. ABSTRACT Quickest detection of the onset of a signal is a common problem in many applications. For example, consider the detection of a sonar contact as it enters the sonar's detection range. While Page's test is known to optimally provide the lowest average delay before detection (i.e., D or detection latency) for a constrained average time between false alarms, it does not necessarily maximize the probability of detecting (Pd) an ephemeral signal (e.g., a sonar contact passing through a convergence zone). In such cases a common alternative is the sliding m-of-n detector where a detection is declared when m successes are observed within n of the most recent trials (e.g., 3 detections during the 5 most recent pings). Techniques for evaluating or approximating the performance measures of the sliding m-of-n detector are developed and used to optimally design the detector. As expected, Page's test outperforms the sliding m-of-n detector with respect to D , except under certain cases of signi cant mismatch between the assumed and actual single-trial success probability. However, for nite-duration signals, the sliding m-of-n detector outperforms Page's test with respect to Pd or robustness in false-alarm rate to mismatch in design assumptions. Unfortunately, optimization requires dierent (m; n) pairs as a function of signal length. Thus, while Page's test remains the most desirable detector to minimize D or if the signal length is unknown, the gains in Pd achievable by a properly designed sliding m-of-n detector make it the best choice for nite-duration signals of known length.						
15. SUBJECT TERMS						
16. SECURITY CLASSIFICATION OF:				17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 64	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified				

Abstract

Quickest detection of the onset of a signal is a common problem in many applications. For example, consider the detection of a sonar contact as it enters the sonar's detection range. While Page's test is known to optimally provide the lowest average delay before detection (i.e., $\bar{D}$ or detection latency) for a constrained average time between false alarms, it does not necessarily maximize the probability of detecting (P_d) an ephemeral signal (e.g., a sonar contact passing through a convergence zone). In such cases a common alternative is the sliding m -of- n detector where a detection is declared when m successes are observed within n of the most recent trials (e.g., 3 detections during the 5 most recent pings). Techniques for evaluating or approximating the performance measures of the sliding m -of- n detector are developed and used to optimally design the detector. As expected, Page's test outperforms the sliding m -of- n detector with respect to $\bar{D}$, except under certain cases of significant mismatch between the assumed and actual single-trial success probability. However, for finite-duration signals, the sliding m -of- n detector outperforms Page's test with respect to P_d or robustness in false-alarm rate to mismatch in design assumptions. Unfortunately, optimization requires different (m, n) pairs as a function of signal length. Thus, while Page's test remains the most desirable detector to minimize $\bar{D}$ or if the signal length is unknown, the gains in P_d achievable by a properly designed sliding m -of- n detector make it the best choice for finite-duration signals of known length.

UNCLASSIFIED

approved for public release

Contents

1	Introduction	1
2	Quickest detection and Page's test	2
2.1	Performance measures	3
2.2	Exponentially- and Bernoulli-distributed data examples	4
2.2.1	Exponentially-distributed data	4
2.2.2	Bernoulli-distributed data	5
2.2.3	Performance comparison	6
2.3	Optimal quantization	8
3	Sliding m-of-n detectors	12
3.1	m -of- n detection as a finite-state Markov process	12
3.2	State and stopping time probabilities for stationary data	14
3.3	Average stopping time	17
3.3.1	Small- p approximation	17
3.3.2	Lower bound	19
3.3.3	Alternative computational approximation	23
3.4	Choosing an optimal m -of- n detector	24
3.4.1	The choice of m dominates the FAR	24

UNCLASSIFIED

approved for public release

3.4.2	Minimizing $\bar{D}$	25
3.4.3	Approximately maximizing P_d	29
4	Detection performance comparison	33
4.1	Average delay to detection	33
4.2	Probability of detection	36
5	Conclusions	41
A	MGF unity roots	44
A-1	Exponential data	44
A-2	Bernoulli data	44
B	Matlab code for m-of-n detector performance evaluation and design	46
B-1	Subroutines to represent the sliding m -of- n detector as an FSMP	46
B-2	Subroutines for performance measures from FSMP	49
B-3	Subroutines for performance measure approximations	51
B-4	Subroutines for m -of- n detector design	55

UNCLASSIFIED

approved for public release

1 Introduction

Several detection problems in sonar signal and information processing can be characterized as detecting the onset of a signal as quickly as possible. A detection example is found in detecting the presence of an active sonar contact over multiple pings as it enters within the sonar's detection range. Tracking algorithms provide examples both in track initialization, which occurs when energy is first observed consistently along paths representative of constant-velocity target motion, and also when tracks are ended. Page's test [1] is the most common choice for the quickest detection of signal onset, owing to its ease of implementation and optimality in providing the minimum average delay to detection ($\bar{D}$) for a constrained average time between false alarms ($\bar{T}$). A common alternative, however, is the sliding m -of- n detector [2, 3] which declares a detection when m of the previous n trials are successes (e.g., a signal observed on 3 of the 5 most recent pings). Noting that the optimality of Page's test does not extend to the probability of detecting a finite duration signal, it is possible that the sliding m -of- n detector is a desirable alternative.

The intuitiveness of the m -of- n detection criteria makes it popular, despite the limited availability of performance assessment techniques when it is applied sequentially to detect the onset of a signal. The m -of- n detector is most often analyzed as it is applied in data fusion [4, Sect. 3.4]; that is, in the context of m successes within a *single* sample of n trials. When sliding the m -of- n detector along a data sequence, it can be described as a finite-state Markov process (FSMP) [2, 3]. Williams [3] developed an algorithm to compute the probability of detection when the input data had a time-varying single-trial probability of success. Bar-Shalom [2, Sect. 3.3] presents results for small values of m and n . Unfortunately, the FSMP characterization encounters computational limitations as n increases. The sliding m -of- n detector can also be described as a scan statistic [5]. Approximations for the average stopping time of the test can be found in [5, Sect. 4.2]; however, they assume a minimum decision time of n samples and are therefore most appropriate for use under false alarm conditions. The focus of this report is on techniques for evaluating performance measures for the sliding m -of- n detector, designing the detector to meet a false-alarm rate (FAR) specification and minimize $\bar{D}$ or maximize the probability of detection (P_d) of a finite-duration signal, and to compare the performance of the sliding m -of- n detector to that of Page's test.

The m -of- n detector operates on binary data which are typically obtained through an intermediate thresholding process. In Sect. 2, the quickest detection performance measures and their approximations for Page's test are reviewed and applied to exponentially-distributed data and the Bernoulli-distributed binary data they produce upon thresholding. Analysis of the quickest detection receiver operating characteristic (ROC) curves illustrates the performance lost from using thresholded data rather than pre-thresholded data. Approximations are presented for the data threshold maximizing the asymptotic efficiency of Page's test for several common signal and noise scenarios.

UNCLASSIFIED

approved for public release

In Sect. 3, the standard quickest detection performance measures ($\bar{D}$ and $\bar{T}$) are derived for the sliding m -of- n detector using Williams' FSMP description and stationary data statistics. Unfortunately, the number of states in the FSMP characterization of the sliding m -of- n detector grows exponentially with n , which makes analysis difficult for $n > 10$. Approximations are therefore developed that are applicable to $\bar{T}$ and $\bar{D}$, amenable to evaluation for much larger values of n than the FSMP exact solutions, and used to optimally choose the (m, n) pair meeting a FAR specification and minimizing $\bar{D}$. A technique is also presented to choose the (m, n) pair meeting the FAR specification and approximately maximizing the probability of detection (P_d) of a finite-duration signal using an easily evaluated lower bound on P_d for the m -of- n detectors.

Finally, the performance of Page's test and the sliding m -of- n detector is compared in Sect. 4 in terms of average delay to detection and probability of detection. Conclusions are presented in Sect. 5 and Matlab code for some of the m -of- n subroutines is found in Appendix B.

2 Quickest detection and Page's test

Let the input data be the sequence of independent samples

$$\dots X_{k-1}, X_k, X_{k+1}, \dots \quad (1)$$

where for $k < k_0$ the data follow probability density function (PDF) $f_0(x)$ and PDF $f_1(x)$ for $k \geq k_0$. The time of the change or signal onset, k_0 , is unknown. The quickest detection problem is to declare that the change has occurred using the fewest number of samples beyond k_0 . Page's test [1] is one of the most common detection algorithms for the quickest detection of the onset of a signal. Its detection statistic is obtained through the simple update

$$Y_k = \max \{0, Y_{k-1} + g(X_k)\} \quad (2)$$

with the initial value taking on $Y_0 = 0$. When the function $g(x)$ is the log-likelihood ratio (LLR),

$$g(x) = \log \left[\frac{f_1(x)}{f_0(x)} \right], \quad (3)$$

no other detection algorithm provides a smaller average delay before detection while simultaneously meeting a constraint on the average time between false alarms [6, 7].

For the purposes of comparing Page's test with a sliding m -of- n detector, performance measures for quickest detection are reviewed and evaluated for Page's test. An exponentially-distributed data example is used to produce the Bernoulli-distributed (i.e., binary) data on which the m -of- n

UNCLASSIFIED

approved for public release

detector operates. The binary, decision-level, data are produced by the exponentially-distributed data through an intermediate thresholding process (e.g., as in de-centralized detection where only local decisions are transmitted to a fusion agent).

2.1 Performance measures

While the probability of detection (P_d) and probability of false alarm (P_{fa}) are the standard performance measures for hypothesis testing with fixed sample sizes, the quickest detection problem requires different measures. Under the null hypothesis, the average time between false alarms ($\bar{T}$) is used to represent how false-alarm prone a detector is while the average delay to detection ($\bar{D}$) quantifies detection performance under the alternative hypothesis of signal presence. $\bar{T}$ and $\bar{D}$ are sometimes called the average stopping times (ASTs), average sample numbers (ASNs) or average record lengths (ARLs). In terms of performance measures of interest to sonar signal processing, $\bar{T}$ is the inverse of the false alarm rate while $\bar{D}$ is detection latency. These measures suffice when the duration of the signal is infinite once it starts (e.g., as in machinery failure) or at least much longer than $\bar{D}$. However, they can be supplemented with P_d when the signal is ephemeral; see [8] for several techniques to evaluate the probability of detecting finite-duration signals using Page's test.

The detector data mapping function $g(x)$ is often not the LLR which requires some knowledge about the signal (e.g., its strength). When the LLR is used, it satisfies the inequalities

$$E_0[g(X)] < 0 < E_1[g(X)] \quad (4)$$

where $E_0[\cdot]$ and $E_1[\cdot]$ represent taking the expectation over, respectively, the null and alternative hypotheses. Thus, when no signal is present Page's test is often reset to zero, but then tends to rise when signal is present. When $g(x)$ is not the LLR, good performance, as defined by an exponential relationship between the threshold h (to which Y_k is compared in declaring a detection) and $\bar{T}$ and a linear relationship between h and $\bar{D}$, requires the inequalities in (4) be satisfied. When they are, approximations to the ASNs are [9, Ch. 5]

$$\bar{T} \approx \frac{1 + ht_0 - e^{ht_0}}{t_0 E_0[g(X)]} \quad (5)$$

and

$$\bar{D} \approx \frac{1 + ht_1 - e^{ht_1}}{t_1 E_1[g(X)]} \quad (6)$$

where t_0 and t_1 are the non-zero unity roots of the moment generating function (MGF) of $g(X)$. That is, $t_i \neq 0$ such that

$$E_i[e^{t_i g(X)}] = 1 \quad (7)$$

UNCLASSIFIED

approved for public release

for $i = 0$ and 1 .

In most practical situations, $\bar{T}$ will be large, which will require a large threshold h . In this regime, and under the conditions described in (4), the asymptotic efficiency (η) of Page's test [10] describes the log-linear relationship between $\bar{T}$ and $\bar{D}$,

$$\begin{aligned} \eta &= \lim_{h \rightarrow \infty} \frac{\log \bar{T}}{\bar{D}} \\ &= t_0 E_1 [g(X)]. \end{aligned} \tag{8}$$

2.2 Exponentially- and Bernoulli-distributed data examples

The primary focus of this paper is on quickest detection of Bernoulli-distributed data. However, it is important to recall the inherent loss in performance incurred by working with such a harsh quantization. To illustrate the loss in performance, an exponentially-distributed-data example is used to derive the Bernoulli-distributed data through the standard thresholding process. In order to evaluate performance, the data mapping function $g(x)$ must be derived along with the MGF unity roots and means under the null and alternative hypotheses. While $g(x)$ and its mean values are evaluated here, the procedure for obtaining the MGF unity roots is described in Appendix A.

2.2.1 Exponentially-distributed data

Suppose the data follow an exponential distribution with mean λ under the null hypothesis and mean $\lambda(1+s)$ under the alternative. Thus, λ is the average noise power and s is the signal-to-noise power ratio (SNR). The LLR is

$$l(x) = \frac{s}{1+s} \left[\frac{x}{\lambda} - \left(1 + \frac{1}{s} \right) \log(1+s) \right] \tag{10}$$

and clearly depends on both s and λ . In this paper λ is assumed to be known, although it is typically estimated in a normalization procedure. The detector is implemented with a design SNR, $\tilde{s}$, which will most likely differ from the observed SNR s . The optimality of the LLR in Page's test is not altered by a change in scale, so the data mapping function

$$g(x) = \frac{x}{\lambda} - b(\tilde{s}) \tag{11}$$

is used where the bias

$$b(\tilde{s}) = \left(1 + \frac{1}{\tilde{s}} \right) \log(1+\tilde{s}) \tag{12}$$

UNCLASSIFIED

approved for public release

only depends on the design SNR.

The mean of the data mapping function is

$$E_0 [g(X)] = 1 - b(\tilde{s}) \quad (13)$$

under the null hypothesis and

$$E_1 [g(X)] = 1 + s - b(\tilde{s}) \quad (14)$$

under the alternative hypothesis.

2.2.2 Bernoulli-distributed data

If the exponentially-distributed data, normalized by the noise power, are compared with a threshold (h_x) prior to application to Page's test, the binary data sequence can be described as

$$U_i = I\left(\frac{X_i}{\lambda} \geq h_x\right) \quad (15)$$

where $I(\cdot)$ is an indicator function returning one when the argument is true and zero otherwise. The datum U_i is a Bernoulli random variable with probability

$$\begin{aligned} p_0 &= \Pr\left\{\frac{X_i}{\lambda} \geq h_x | H_0\right\} \\ &= e^{-h_x} \end{aligned} \quad (16)$$

under the null hypothesis (H_0) and

$$\begin{aligned} p_1 &= \Pr\left\{\frac{X_i}{\lambda} \geq h_x | H_1\right\} \\ &= e^{-\frac{h_x}{1+s}} = p_0^{\frac{1}{1+s}} \end{aligned} \quad (17)$$

under the alternative hypothesis (H_1). Note that these are the well known false-alarm and detection probabilities for envelope detection of a Gaussian target in Gaussian noise.

Scaling the LLR of the Bernoulli data to isolate the dependence on p_0 and p_1 into the bias results in the data mapping function

$$g(u) = u - b(p_0, p_1) \quad (18)$$

where the bias is

$$b(p_0, p_1) = \left[1 + \frac{\log(p_1/p_0)}{\log[(1-p_0)/(1-p_1)]}\right]^{-1}. \quad (19)$$

UNCLASSIFIED

approved for public release

If the detection process producing the Bernoulli data is unbiased, then $p_1 > p_0$ and $b(p_0, p_1) \in (0, 1)$. While it is reasonable to expect p_0 to be known, it is unlikely that p_1 will be known with any accuracy. As was done for the exponentially-distributed data, design probabilities $\tilde{p}_0$ and $\tilde{p}_1$ can be used to form the bias

$$\tilde{b} = b(\tilde{p}_0, \tilde{p}_1) \quad (20)$$

or $\tilde{b}$ may be chosen in $(0, 1)$.

The mean value of the data mapping function is

$$E_0[g(U)] = p_0 - \tilde{b} \quad (21)$$

under the null hypothesis and

$$E_1[g(U)] = p_1 - \tilde{b} \quad (22)$$

under the alternative hypothesis.

2.2.3 Performance comparison

The receiver operating characteristic (ROC) curves for Page's test operating on the exponentially- and Bernoulli-distributed data are compared in Fig. 1 for the case of $p_0 = 10^{-4}$ and $p_1 = 0.5$, which equates to $s \approx 10.9$ dB for the exponentially-distributed data model. The solid lines in the figure are $\bar{D}$ as a function of $\log_{10} \bar{T}$ when the bias is chosen correctly according to s or p_1 . The asymptotic efficiencies (as computed via (9)) for these cases are

$$\eta_B = 3.9 \text{ and } \eta_E = 9.7 \quad (23)$$

for the Bernoulli- and exponentially-distributed data, respectively. Their ratio $\eta_E/\eta_B = 2.5$ is representative of the factor by which $\bar{D}$ for the exponentially-distributed data is below that for the Bernoulli-distributed data; that is, the signal is detected approximately 2.5 times faster using the original data compared to using the quantized, binary data.

The dashed lines in Fig. 1 represent the performance when the bias in the data mapping function is chosen incorrectly. For the exponential data, the bias was chosen assuming an SNR of 15, 5, 2, and 0 dB, with performance increasingly worse in this order. Although not always the case, this illustrates that overestimating the SNR is often not as bad as overestimating it. The dashed curves for the Bernoulli data are formed with a design $\tilde{p}_1$ formed from the design SNR according to

$$\tilde{p}_1 = p_0^{\frac{1}{1+s}} \quad (24)$$

and illustrate similar performance degradation relative to the correctly chosen bias, which optimizes performance.

UNCLASSIFIED

approved for public release

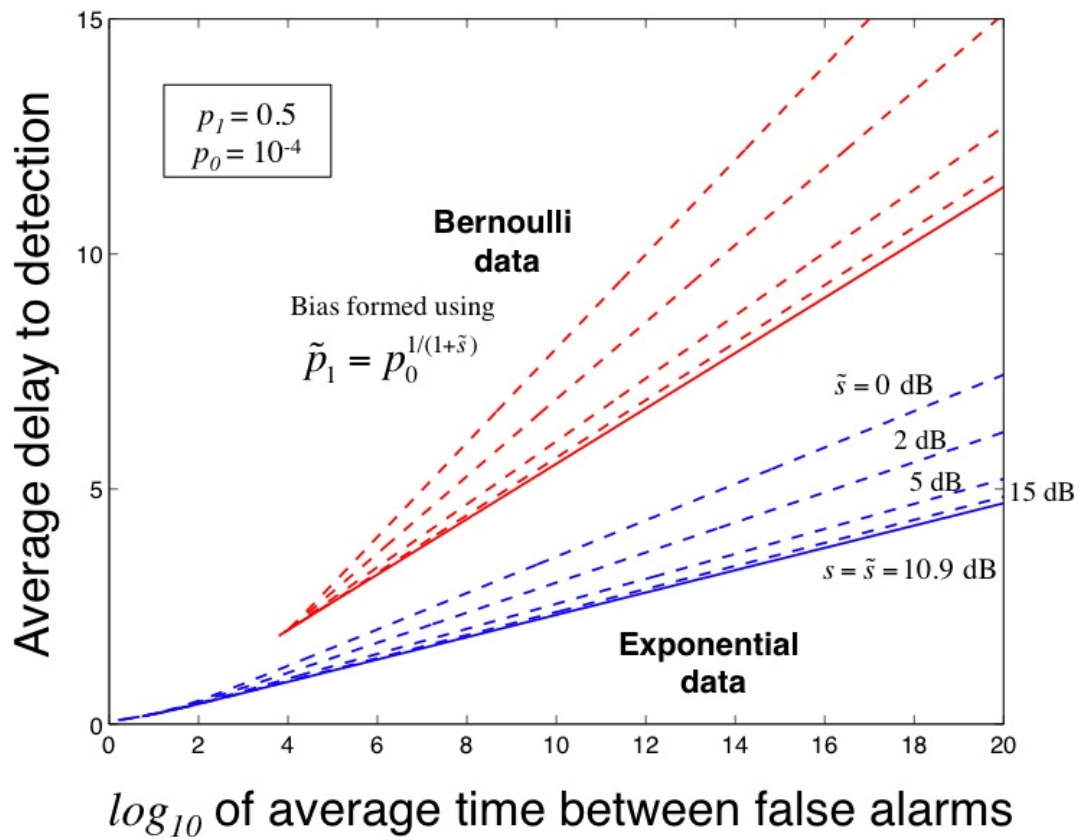


Figure 1: ROC curves for Page's test operating on exponential and Bernoulli data with correct bias (solid lines) and a mismatched bias (dashed lines). The advantage of working with data prior to quantization is evident in the lower values of $\bar{D}$ for the exponential data.

UNCLASSIFIED

approved for public release

2.3 Optimal quantization

When the pre-thresholded data are accessible or the quantization process is included in the detector design, the quantization threshold h_x can be chosen to maximize the asymptotic efficiency of Page's test formed from the binary data of (15). When the FAR specification results in a high threshold in Page's test, then optimizing η will also minimize $\bar{D}$ for a given $\bar{T}$.

Using the definition of η in (9), the optimal bias from (19), and the approximation for t_0 for Bernoulli-distributed data found in Appendix A,

$$t_0 \approx \frac{-\log(p_0)}{1 - \tilde{b}}, \quad (25)$$

yields the approximation

$$\eta \approx -p_1 \log(p_0) + \frac{(1 - p_1) \log\left(\frac{1-p_0}{1-p_1}\right)}{\frac{\log(p_1)}{\log(p_0)} - 1}. \quad (26)$$

For the fluctuating signal described in Sect. 2.2.1, the first term in (26) dominates the second, especially as SNR increases (where $p_1 \rightarrow 1$). Using (16) and (17) for p_0 and p_1 , the first term in (26) is seen to be maximized when

$$h_x = s + 1. \quad (27)$$

Starting with this threshold and implementing one step of a Newton-Raphson update to find the maximum of (26) results in the refined approximation to the optimal threshold (in dB),

$$H_x^* = 10 \log_{10} h_x^* \approx 10 \log_{10}(s + 1) + 10 \log_{10} \left[1 + \left(1 + \frac{s - (e - 1)^{-1}}{1 + \log(1 - 1/e)} \right)^{-1} \right] \quad (28)$$

where $e = e^1 \approx 2.7183$.

The approximation of (28) is compared in Fig. 2 with the optimum threshold for the fluctuating signal as obtained through a numerical search using (9). As seen in the figure, the approximation is quite good, especially for larger values of SNR. The average absolute error is only 0.026 dB for SNR ranging from 2 to 16 dB. Although the maximum error is more than one-tenth of a dB, it occurs for the smallest SNR. Three other common signal and noise scenarios are also considered: a non-fluctuating signal in complex Gaussian noise, a Gaussian shift-in-mean signal, and a Gaussian change-in-variance signal. The statistical characterizations of the detection statistic under the null and alternative hypotheses along with equations for p_0 and p_1 for the various scenarios considered are found in Table 1.

UNCLASSIFIED

approved for public release

Table 1: Data statistical characterizations under the null and alternative hypotheses and the associated success probabilities for various signal and noise scenarios as a function of the data threshold h_x and SNR. An exponential distribution with mean λ is denoted by $\mathcal{E}(\lambda)$, a non-central chi-squared distribution with two degrees of freedom and non-centrality parameter δ by $\chi_2^2(\delta)$, and a Gaussian distribution with mean μ and variance σ^2 by $\mathcal{N}(\mu, \sigma^2)$. The cumulative distribution functions (CDFs) of the non-central chi-squared and Gaussian distributions are, respectively, $F_{\chi_2^2}(x; \delta)$ and $\Phi((x - \mu)/\sigma)$.

Signal type	H_0	p_0	H_1	p_1
Fluctuating	$X \sim \mathcal{E}(1)$	e^{-h_x}	$X \sim \mathcal{E}(1 + s)$	$e^{-h_x/(1+s)}$
Non-fluctuating	$X \sim \mathcal{E}(1)$	e^{-h_x}	$2X \sim \chi_2^2(2s)$	$1 - F_{\chi_2^2}(2h_x; 2s)$
Gaussian shift-in-mean	$X \sim \mathcal{N}(0, 1)$	$1 - \Phi(h_x)$	$X \sim \mathcal{N}(\sqrt{s}, 1)$	$1 - \Phi(h_x - \sqrt{s})$
Gaussian change-in-variance	$X \sim \mathcal{N}(0, 1)$	$1 - \Phi(h_x)$	$X \sim \mathcal{N}(0, 1 + s)$	$1 - \Phi(h_x/\sqrt{1 + s})$

The complicated forms of the threshold/exceedance-probability relationships for these latter signal types hinders derivation of an approximation similar to (28). However, noting the smooth functional forms of the optimal threshold seen in Fig. 2 on logarithmic scales, the functions were fit via least-squared-error optimization using half-powers of $S_{\text{dB}} = 10 \log_{10} s$ (i.e., $\sqrt{S_{\text{dB}}}$, S_{dB} , and $S_{\text{dB}}^{1.5}$). The functional forms of the approximations to the optimal threshold are shown in Table 2 for each signal and noise scenario along with the average, median and maximum errors over the range of SNR evaluated. A least-squared-error fit to the optimal threshold for the fluctuating signal (not shown in Fig. 2) provided a better overall fit than (28) at the expense of an increase in error for large SNR. While the Gaussian signals only required two terms to keep the maximum absolute error less than one-twentieth of a decibel over the range of SNR evaluated, the fluctuating and non-fluctuating signals required four terms. As seen in Fig. 2, the fit is quite good within the range of evaluation. Other than (28), these functions are not expected to work beyond $S_{\text{dB}} \in [2, 16]$.

The general trend for the optimal threshold is one that increases with SNR, indicating that η is maximized primarily by reducing p_0 , although p_1 was also slightly increasing with SNR. The errors in the approximations found in Table 2 are small enough that they can be used to design the intermediate thresholding when it is desired to apply Page's test to the binary, decision-level data. As will be seen in the following section, this is likely a good choice when an m -of- n detector is used instead of Page's test.

UNCLASSIFIED

approved for public release

Table 2: Approximations to the data threshold maximizing the asymptotic efficiency of Page's test for the various signal types as a function of SNR and the error over the evaluation range ($S_{\text{dB}} \in [2, 16]$).

Signal type	$H_x^* = 10 \log_{10} h_x^* \approx$	Absolute error (dB)		
		Average	Median	Maximum
Fluctuating	$10 \log_{10}(s+1) + 10 \log_{10} \left[1 + \left(1 + \frac{s-(e-1)^{-1}}{1+\log(1-1/e)} \right)^{-1} \right]$	0.026	0.013	0.187
Fluctuating	$5.1714 - 0.8084\sqrt{S_{\text{dB}}} + 0.4758S_{\text{dB}} + 0.1039S_{\text{dB}}^{1.5}$	0.013	0.014	0.032
Non-fluctuating	$4.6982 - 2.0715\sqrt{S_{\text{dB}}} + 0.7814S_{\text{dB}} + 0.0939S_{\text{dB}}^{1.5}$	0.016	0.017	0.050
Gaussian shift-in-mean	$-1.0719 + 0.5190S_{\text{dB}}$	0.018	0.018	0.041
Gaussian change-in-variance	$3.2602 + 0.0867S_{\text{dB}}^{1.5}$	0.011	0.008	0.048

UNCLASSIFIED

approved for public release

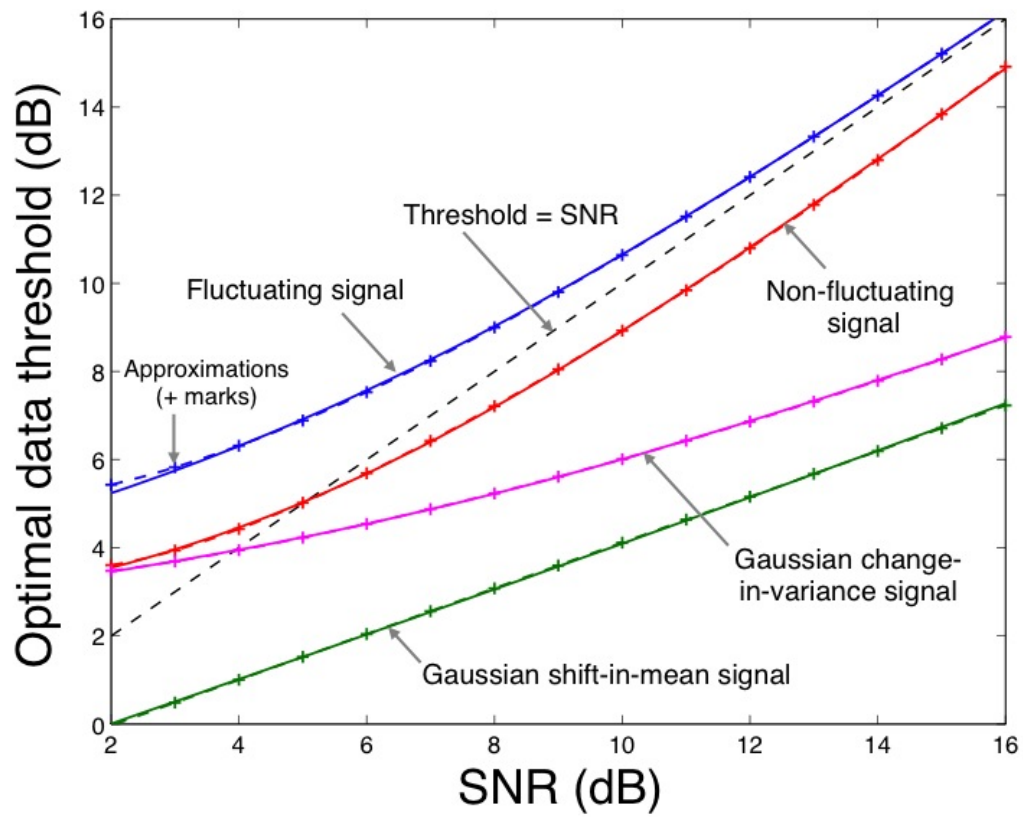


Figure 2: Data threshold (h_x) optimizing the asymptotic efficiency of Page's test for various signal and noise scenarios (solid lines). The general trend observed is for an increasing threshold with SNR. Approximations to the optimal threshold are shown as dashed lines or '+' marks.

UNCLASSIFIED

approved for public release

3 Sliding m -of- n detectors

A common technique for fusing information is to apply an m -of- n rule where a detection declaration is made whenever m successes occur within n associated trials. The situation is common in distributed detection [4, Sect. 3.4] and known as binary integration in the radar community [11, Sect. 6.4] owing to its equivalence to thresholding a rolling sum of binary data representing the individual trials. Analysis of the m -of- n detector is usually limited to the static case where only one set of n inputs is evaluated. Sliding an m -of- n detector along a data sequence, which is often how it is used to reduce false alarms, results in what is known in the statistics literature as a *scan statistic* [5]. It can also be described as a discrete-time, finite-state Markov process (FSMP) [2, 3]. In [3] a technique was presented for numerically evaluating the probability of detection for an input sequence of Bernoulli random variables with time-varying probabilities—the example considered in [3] was an active sonar contact transiting through an area presenting a limited detection opportunity. BarShalom [2, Sect. 3.3] presents results for small values of m and n .

In this section, the quickest-detection performance measures ($\bar{D}$ and $\bar{T}$) are developed along with a more simply evaluated alternative to Williams' algorithm for P_d and a lower bound on P_d for the stationary-data case in preparation for comparing the performance of the sliding m -of- n detector with Page's test. While the characterization of the sliding m -of- n detector as an FSMP allows for exact evaluation of the average stopping times and P_d , it is limited by exponential growth in the number of states as n increases. Approximations to the average stopping time of the scan statistic [5, Sect. 4.2] surmount this limitation, but assume a minimum decision time of n rather than m and have numerical evaluation issues for small p . To surmount this limitation, a small- p approximation to $\bar{T}$ is presented along with an alternative computational method approximating the average stopping times that is useful for a much wider range of n . Finally, techniques are presented for choosing m and n to meet a false alarm specification and to approximately maximize the probability of detecting a finite duration signal and minimizing the average delay to detection.

3.1 m -of- n detection as a finite-state Markov process

The key to Williams' analysis in [3] was to describe the m -of- n detector as an FSMP with the states defined by the previous $n - 1$ trials, what he termed the detector's history. The detection event, observing m successes in n consecutive trials, was characterized by an 'accepting' state while all other states were defined as non-accepting. The complete set of states was enumerated by the integer value of the binary number defined by the history. Consider, for example, $m = 3$ and $n = 5$. If the previous 4 trials were the sequence *failure*, *success*, *success*, *failure* then the history is the binary number 0110 with a state enumeration of the integer 6. If the next trial is a success,

UNCLASSIFIED

approved for public release

then this state transitions to the accepting state having observed 3 success in 5 consecutive trials. However, a failure in the next trial would transition to state 1100 (integer value 12). A technique for obtaining an equivalent FSMP with a reduced set of states was also presented in [3], enabling a more efficient numerical analysis for larger values of n . Matlab code for creating the structures defining the Markov process is presented in Appendix B.

The Markovian nature of the process implies that the vector $\mathbf{p}(k)$ of probabilities of being in each of the states at time k can be determined from the probabilities at time $k - 1$ according to [12, Sect. 4.3]

$$\mathbf{p}^T(k) = \mathbf{p}^T(k - 1)\mathbf{P}(k) \quad (29)$$

where $\mathbf{P}(k)$ is the (potentially) time-varying state transition matrix and the superscript T represents the transpose operation. Williams presented in [3] an algorithmic update accomplishing (29); however, the matrix representation is maintained here owing to its simpler implementation and the corresponding ability to evaluate $\bar{D}$ and $\bar{T}$ with a simple summation over the FSMP states for stationary data.

Suppose there are n_s states, including the accepting or absorbing state. In ordering the states within $\mathbf{p}(k)$, let the first state be a history comprising all zeros (i.e., none of the past $n - 1$ trials were successful) and let the last state be the accepting state (i.e., the m -of- n success state). Assuming that once the accepting state is entered, the process remains there, the state transition matrix can be described by the decomposition

$$\mathbf{P}(k) = \begin{bmatrix} \mathbf{P}_{cc}(k) & \mathbf{p}_{cs}(k) \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (30)$$

where $\mathbf{P}_{cc}(k)$ is the transition matrix for the $n_s - 1$ continuing states, $\mathbf{p}_{cs}(k)$ is a vector of the transition probabilities from the continuing states to the accepting state, and $\mathbf{0}$ is a vector of zeros.

The probability of being in one of the continuing states in the k th trial can then be described by the recursion

$$\mathbf{p}_c^T(k) = \mathbf{p}_c^T(k - 1)\mathbf{P}_{cc}(k) \quad (31)$$

where $\mathbf{p}_c(0)$ is a vector of the initial probabilities of being in a continuing state.

The performance of the sliding m -of- n detector is described by defining the probability mass function (PMF) of the stopping time K , which is defined as the first time the process enters the accepting state. The PMF for K is the probability of transitioning to the accepting state at time k from one of the continuing states at time $k - 1$,

$$f_K(k) = \mathbf{p}_c^T(k - 1)\mathbf{p}_{cs}(k) \quad (32)$$

UNCLASSIFIED

approved for public release

for $k = 1, 2, \dots$.

The probability of detecting a signal within k trials using a sliding m -of- n processor is the CDF of K ,

$$P_d(k) = F_K(k) = \sum_{l=1}^k f_K(l). \quad (33)$$

This also approximates the probability of detecting a finite-duration signal of length k , only ignoring any detections occurring after time k but before time $k + n$ (i.e., while the sliding window is still affected by the signal). For Page's test, these were termed latent detections [8].

The average sample numbers ($\bar{T}$ and $\bar{D}$) are the mean of K under the null and alternative hypotheses, respectively. These are typically evaluated when the process is stationary (i.e., p_k is constant) and therefore discussed after the simplifications of the following section.

The time-varying part of the transition matrix and vector is limited to the success probability of the k th trial. Defining this as p_k , the continuing state transition matrix can be described as

$$\mathbf{P}_{cc}(k) = p_k \mathbf{J}_+ + (1 - p_k) \mathbf{J}_- \quad (34)$$

where $\mathbf{J}_+$ and $\mathbf{J}_-$ are indicator matrices (the elements containing either a zero or a one) representing which state is achieved at the next time step given a success ($\mathbf{J}_+$) or a failure ($\mathbf{J}_-$). Similarly, the probability of transitioning from the continuing to accepting states can be described by

$$\mathbf{p}_{cs}(k) = p_k \mathbf{j}_+ \quad (35)$$

where $\mathbf{j}_+$ is an indicator vector. Using (34) and (35) in (31) and (32) is more efficient in Matlab than the algorithm developed in [3]. Matlab code for producing the indicator matrices from Williams' Markov-process structure definition is found in Appendix B.

3.2 State and stopping time probabilities for stationary data

When the state transition probabilities do not change with time, simplifications allow further analysis of the m -of- n processor. Dropping the dependence on k of $\mathbf{P}_{cc}$, $\mathbf{p}_{cs}$, and p , the update equation (31) for the continuing state probabilities becomes

$$\begin{aligned} \mathbf{p}_c^T(k) &= \mathbf{p}_c^T(0) \mathbf{P}_{cc}^k \\ &= \mathbf{p}_c^T(0) \mathbf{V} \mathbf{\Lambda}^k \mathbf{V}^{-1} \end{aligned} \quad (36)$$

where $\mathbf{V}$ and $\mathbf{\Lambda}$ are the eigenvector and eigenvalue matrices of $\mathbf{P}_{cc}$. Noting that $\mathbf{\Lambda}$ is diagonal illustrates the ease with which the state probabilities may be calculated.

UNCLASSIFIED

approved for public release

Similarly, the PMF for K is simplified to

$$\begin{aligned}
 f_K(k) &= \mathbf{p}_c^T(0) \mathbf{P}_{cc}^{k-1} \mathbf{p}_{cs} \\
 &= \mathbf{p}_c^T(0) \mathbf{V} \mathbf{\Lambda}^{k-1} \mathbf{V}^{-1} \mathbf{p}_{cs} \\
 &= \sum_{i=1}^{n_s-1} a_i b_i \lambda_i^{k-1}
 \end{aligned} \tag{37}$$

where λ_i are the eigenvalues of $\mathbf{P}_{cc}$ (i.e., the diagonal elements of $\mathbf{\Lambda}$) and a_i and b_i are elements of the vectors

$$\mathbf{a} = \mathbf{V}^T \mathbf{p}_c(0) \tag{38}$$

and

$$\mathbf{b} = \mathbf{V}^{-1} \mathbf{p}_{cs}. \tag{39}$$

The CDF of K (yielding $P_d(k)$) can also be evaluated without resorting to the cumulative summation of (33),

$$\begin{aligned}
 F_K(k) &= \sum_{l=1}^k f_K(l) \\
 &= \sum_{i=1}^{n_s-1} a_i b_i \sum_{l=1}^k \lambda_i^{l-1} \\
 &= \sum_{i=1}^{n_s-1} a_i b_i \frac{(1 - \lambda_i^k)}{(1 - \lambda_i)}.
 \end{aligned} \tag{40}$$

Note that the PMF of the stopping time for times $k = m, \dots, n$ is simply the probability of observing $m - 1$ successes in $k - 1$ trials followed by a success,

$$f_K(k) = \binom{k-1}{m-1} p^m (1-p)^{k-m} \tag{41}$$

where ‘ i choose j ’,

$$\binom{i}{j} = \frac{i!}{j!(i-j)!} = \frac{\Gamma(i+1)}{\Gamma(j+1)\Gamma(i-j+1)}, \tag{42}$$

is the number of different combinations of size j from a population of size i .

UNCLASSIFIED

approved for public release

The CDF of the stopping time is simply the summation

$$F_K(k) = \sum_{j=m}^k \binom{j-1}{m-1} p^m (1-p)^{j-m} \quad (43)$$

and provides an easily evaluated alternative to the FSMP result of (40) that is extremely useful when n is large.

The initial state probability vector for the continuing states ($\mathbf{p}_c(0)$) should be chosen assuming the sliding m -of- n detector has been operating for some time under the null hypothesis, but not resulted in a false alarm. Following [8], the update equation for the continuing state probabilities given no false alarms is

$$\mathbf{p}_{c|c}^T(k) = \frac{\mathbf{p}_{c|c}^T(k-1) \mathbf{P}_{cc}}{\mathbf{p}_{c|c}^T(k-1) \mathbf{P}_{cc} \mathbf{1}} \quad (44)$$

where $\mathbf{1}$ is a vector of ones. Using the eigen-decomposition of (36), it is straightforward to show that the limiting continuing state probability vector is the left eigenvector of $\mathbf{P}_{cc}$ associated with its largest eigenvalue

$$\begin{aligned} \mathbf{p}_{c|c}^T(\infty) &= \lim_{k \rightarrow \infty} \mathbf{p}_{c|c}^T(k) \\ &= \frac{\mathbf{e}_1^T \mathbf{V}^{-1}}{\mathbf{e}_1^T \mathbf{V}^{-1} \mathbf{1}} \end{aligned} \quad (45)$$

where $\mathbf{e}_1$ is a unit vector pointing in the direction of the first element and it is assumed that the largest eigenvalue is in the first position.

For small values of p , an accurate approximation can be obtained from the probability of observing the number of successes in the history of each continuing state. Defining $\beta(i)$ as the number of successes in the length $n-1$ history of the i th state, the asymptotic probability of observing the i th continuing state is approximately

$$\mathbf{p}_{c|c}(\infty) \approx \frac{1}{\sum_{j=1}^{n_s-1} p^{\beta(j)}} \begin{bmatrix} p^{\beta(1)} & p^{\beta(2)} & \dots & p^{\beta(n_s-1)} \end{bmatrix}^T. \quad (46)$$

For values of $n \leq 10$, the approximation of (46) is quite good when $p < 0.1$. Define the error as the maximum over all values of m for a given n of the sum of the total absolute error between the FSMP exact solution of (45) and the approximation of (46). The error was worst for small values of $m > 1$ and larger values of n , and ranged from p^2 for the smaller values of n to p for the larger values. Noting that the first state ($i = 1$) is the only state with no successes in the history (i.e., $\beta(1) = 0$), most of the weight is placed on this state when p is small. Therefore, using

$$\mathbf{p}_{c|c}(\infty) \approx \mathbf{e}_1 \quad (47)$$

is a reasonable approximation when p is sufficiently small.

UNCLASSIFIED

approved for public release

3.3 Average stopping time

Under the stationary conditions described in the previous section, the average stopping time of the sliding m -of- n detector is

$$E[K] = \sum_{k=1}^{\infty} k f_K(k) \quad (48)$$

$$= \sum_{i=1}^{n_s-1} a_i b_i \sum_{k=1}^{\infty} k \lambda_i^{k-1} \quad (49)$$

$$= \sum_{i=1}^{n_s-1} \frac{a_i b_i}{(1 - \lambda_i)^2} \quad (49)$$

$$= \mathbf{p}_c^T(0) (\mathbf{I} - \mathbf{P}_{cc})^{-2} \mathbf{p}_{cs}. \quad (50)$$

The average time between false alarms ($\bar{T}$) is obtained by using p_0 in forming the state transition probabilities, while $\bar{D}$ requires use of p_1 .

When $m = 1$, the stopping time of the sliding m -of- n detector is simply the time of the first success on an individual trial, which is a geometric random variable. The PMF of the stopping time, regardless of the value of n , is then

$$f_K(k) = p(1 - p)^{k-1} \quad (51)$$

with the associated CDF

$$F_K(k) = 1 - (1 - p)^k \quad (52)$$

for $k = 1, \dots$. The average stopping time is

$$E[K] = \frac{1}{p} \quad (53)$$

and standard deviation

$$\text{Std}[K] = \frac{\sqrt{1-p}}{p}. \quad (54)$$

3.3.1 Small- p approximation

When the probability of success on an individual trial is sufficiently small, the sliding m -of- n detector can be approximated as a sequence of independent Bernoulli trials where the success

UNCLASSIFIED

approved for public release

probability is the probability of observing a failure, followed by $m - 1$ successes in the next $n - 1$ trials, followed by one success,

$$\begin{aligned}\tilde{p} &\approx (1 - p) \cdot \left[\binom{n-1}{m-1} p^{m-1} (1-p)^{n-m} \right] \cdot p \\ &= \frac{(n-1)! p^m (1-p)^{n-m+1}}{(m-1)! (n-m)!}.\end{aligned}\tag{55}$$

Note that (55) is slightly more accurate than using the probability of observing m successes in n trials, which does not require a success on the final trial or a failure in the trial immediately preceding the set of n trials leading to a detection. Similar to the $m = 1$ case, the stopping time is then approximately a geometric random variable with statistics characterized by (51)–(54) using $\tilde{p}$ from (55) in place of p . The average stopping time is approximated as

$$E[K] \approx \frac{(m-1)!(n-m)!}{(n-1)! p^m (1-p)^{n-m+1}},\tag{56}$$

which is significantly easier to evaluate and less prone to computational errors than the exact FSMP solution of (50). Support for the geometric-distribution assumption comes from the form of the stopping time PMF seen in (37), which is the weighted sum of the eigenvalues of $\mathbf{P}_{cc}$ raised to the power $k - 1$. As k increases, the PMF increasingly becomes dominated by the contributions of the maximum eigenvalue,

$$f_K(k) \approx a_1 b_1 \lambda_1^{k-1}\tag{57}$$

where it is assumed that λ_1 is the largest eigenvalue. When p is small, the largest eigenvalue of $\mathbf{P}_{cc}$ is significantly larger than the next largest one, additionally contributing to the accuracy of this approximation as seen in the following analysis.

The small- p approximation of (56) is compared with the FSMP exact solution of (50) in Figs. 3 and 4 for $n = 7$ and all possible values of m . Note that the exact result of (53) as a function of p is shown for $m = 1$. The FSMP exact solution is shown as asterisks and encounters computational errors when p is too small. The approximations from [5] had numerical evaluation issues in the same places as the FSMP exact solution and converge to n as $p \rightarrow 1$ rather than m . Figure 4 expands the region $p \in [0.01, 1]$ where the poor fit of the small- p approximation as p increases is revealed. Defining the relative error as

$$\epsilon_R = \frac{|\bar{K} - \bar{K}_a|}{\bar{K}}\tag{58}$$

where $\bar{K}$ is the average stopping time from (50) and $\bar{K}_a$ is the small- p approximation of (56), the latter (dashed lines in Figs. 3 and 4) is quite accurate for p less than about 0.1 for $n = 7$ with ϵ_R below 0.1 (10% error) and decreasing proportionately with p (i.e., 1% error for $p < 10^{-2}$). Note

UNCLASSIFIED

approved for public release

that the average stopping time is shown on a logarithmic scale in the figures. The relative error observed or calculated on a logarithmic scale is less than that computed via (58) and reported herein. Although not shown, the regions for which the relative error is below 10% for $n = 3$ are $p < 0.25$, $p < 0.18$ for $n = 5$, and $p < 0.06$ for $n = 10$. This is representative of a constant np effect where for larger n the approximation requires p to be commensurately smaller for accuracy. The limited results presented here indicate the small- p approximation has maximum errors around 10% when $np \leq 0.7$ or 1% errors when $np \leq 0.07$. As can be seen in Fig. 4, the errors are worst for $m = 2$ and decrease in severity as m increases (e.g., the error is less than 10% for $p < 0.7$ when $m = n = 7$).

3.3.2 Lower bound

Recall the stopping time PMF from (41) for times $k \leq n$. If the average stopping time is significantly less than n , then (41) might yield an accurate approximation. More specifically, by letting $n \rightarrow \infty$, the stopping time PMF of (41) represents a lower-bound on the stopping time. That is, the test stopping after the first m successes will stop as soon or sooner than the test stopping the first time m successes are observed in n consecutive trials. Using (41) in (48), the lower bound is seen to be

$$\begin{aligned} E[K] &\geq \sum_{k=m}^{\infty} k \binom{k-1}{m-1} p^m (1-p)^{k-m} \\ &= mp^m \sum_{j=0}^{\infty} \frac{\Gamma(j+m+1)}{\Gamma(m+1)} \frac{(1-p)^j}{j!} \end{aligned} \tag{59}$$

$$= \frac{m}{p} \tag{60}$$

where the infinite summation in (59) is recognized as a hypergeometric function simplifying to $p^{-(m+1)}$ [13, Table 18-1].

The bound is compared with the exact FSMP solution in Fig. 5. As might be expected, the bound is tightest for $m \ll n$ and as $p \rightarrow 1$, but otherwise a very loose bound. While it is better than the small- p approximation near $p = 1$, its primary utility is as a lower bound rather than an accurate approximation to the average stopping time.

UNCLASSIFIED

approved for public release

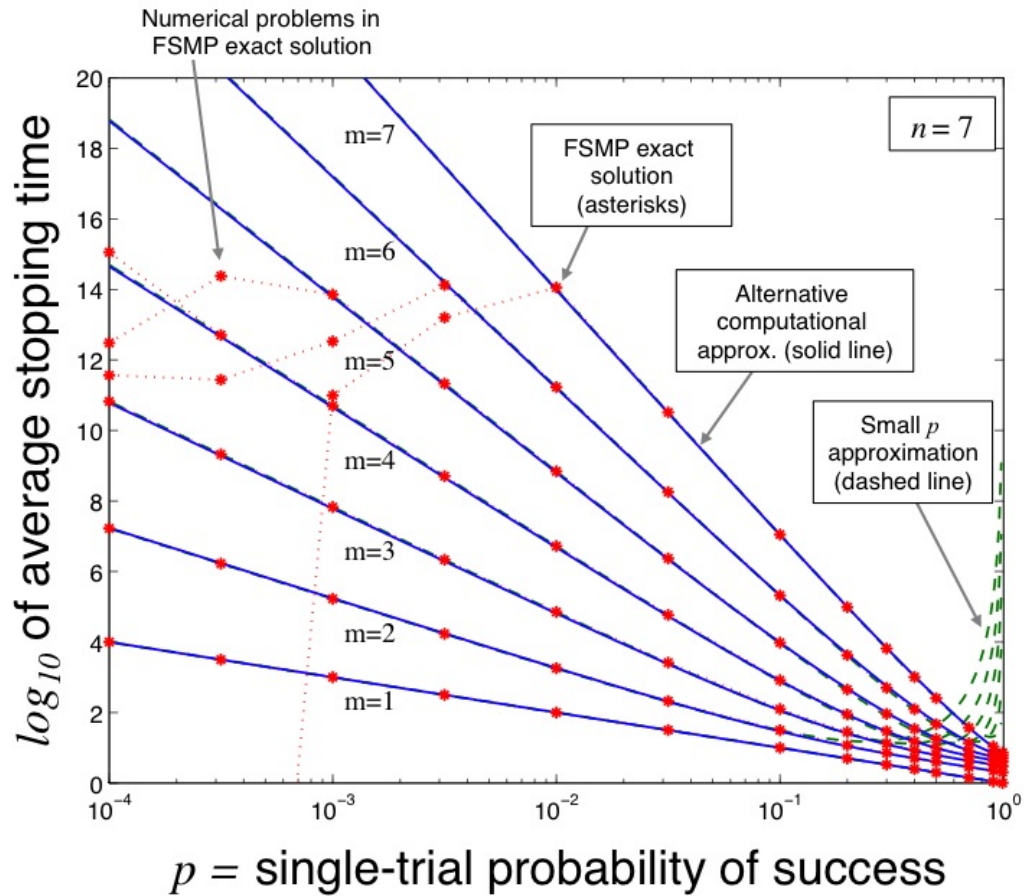


Figure 3: Comparison of the FSMP exact solution to the average stopping time to small- p and alternative computational approximations for $n = 7$ and all possible values of m . The FSMP exact solution encounters numerical issues when p is small enough while the small- p approximation degrades as p increases.

UNCLASSIFIED

approved for public release

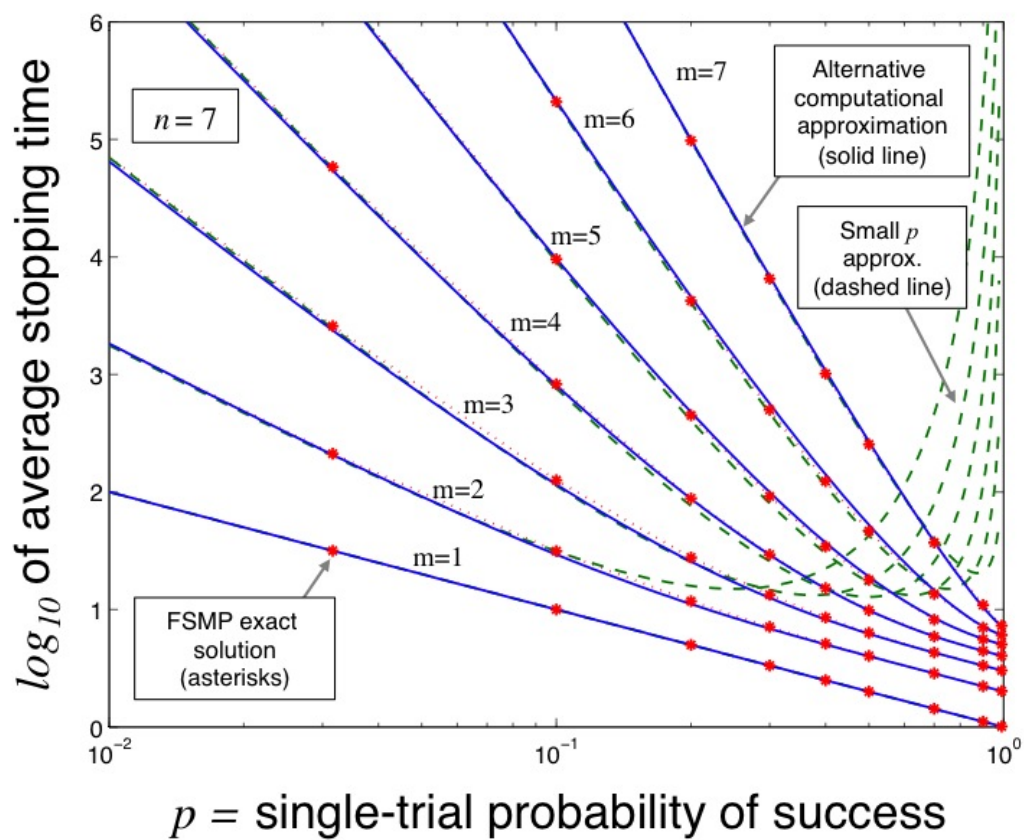


Figure 4: Expanded view of Fig. 3 in the region $p \in [0.01, 1]$ illustrating the failure of the small- p approximation and the accuracy of the alternative computational approximation as p increases.

UNCLASSIFIED

approved for public release

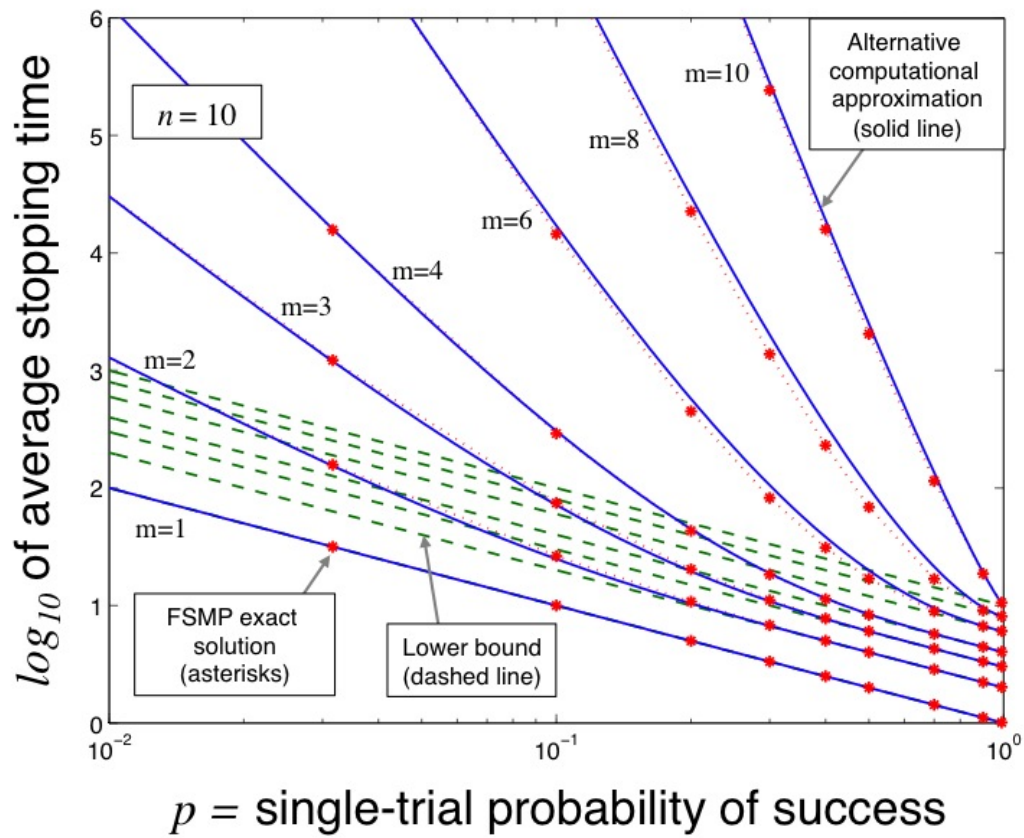


Figure 5: Average stopping time as a function of p for $n = 10$ comparing the exact solution, the alternative computational approximation, and the lower bound for various values of m .

UNCLASSIFIED

approved for public release

3.3.3 Alternative computational approximation

As seen in Fig. 3, the FSMP exact solution of (50) encounters numerical issues when p is small. It can also be computationally intensive when n is large owing to the substantial number of states, which for m near $n/2$ grows exponentially with n . For example, there are up to 253 states when $n = 10$, up to 6,436 when $n = 15$ and 184,757 when $n = 20$. The approximations from the scan-statistics literature [5] similarly have problems for small values of p and then are derived to converge on n rather than m as $p \rightarrow 1$, so are primarily useful for evaluating $\bar{T}$ when p is not too small. An alternative computational method approximating the average stopping time is presented in this section avoiding the numerical issues of the scan-statistic approximations and the FSMP exact solution both in defining the states and evaluating $E[K]$. Extensive evaluation of (50) revealed that the average stopping time could be described approximately as

$$E_p[K] \approx \frac{m}{p} e^{(m-1)G(p, c(m, n))} \quad (61)$$

where the function

$$G(p, c) = \int_0^{-\log p} (1 - e^{-u})^c du \quad (62)$$

and $c(m, n)$ is a term that requires numerical evaluation. Noting that $G(1, c) = 0$, it can be seen that the average stopping time in (61) tends to m as $p \rightarrow 1$. It was also observed that $c(m, n)$ increases with n , but decreases with m , and that $c(2, n) \approx n - 1$ for large n . While there does not appear to be any simple closed-form solutions to the integral in (62), the argument

$$g_c(u) = (1 - e^{-u})^c \quad (63)$$

is suitable for numerical integration. The procedure implemented in the Matlab code found in Appendix B exploits a second-order Taylor-series expansion of $g_c(u)$ about $u_0 \in (0, -\log p)$

$$\begin{aligned} g_c(u) &\approx g_c(u_0) + (u - u_0)g'_c(u_0) + \frac{1}{2}(u - u_0)^2 g''_c(u_0) \\ &= (1 - e^{-u_0})^c \left[1 + c \left(\frac{u - u_0}{e^{u_0} - 1} \right) + \frac{c^2 (u - u_0)^2 (c - e^{u_0})}{2 (e^{u_0} - 1)^2} \right] \end{aligned} \quad (64)$$

for a set of points equally spaced within the integration limits. The integral in (62) is then approximated as the sum of the integrals over the piece-wise quadratic functions.

The value of $c(m, n)$ is chosen to minimize the error between the numerical evaluation of (61) and the small- p approximation of (56) for some sufficiently small value of p . Although the minimization is not a computationally burdensome evaluation, $c(m, n)$ only needs to be evaluated once for each (m, n) pair and then stored for subsequent use in evaluating (61). The value of p at which the two

UNCLASSIFIED

approved for public release

curves are forced to intersect is chosen small enough that the small- p approximation is accurate, $p = 0.1/n$. Evaluating the average stopping time in this manner forces accuracy as $p \rightarrow 1$ and also for $p \leq 0.1/n$, but does not guarantee accuracy in the intervening region. For example, the relative errors for the $n = 7$ case shown in Figs. 3 and 4 have a maximum error of $\epsilon_R = 0.18$ when $p = 0.4$ for $m = 6$. While an 18% error is not small, the error is often less and there is little alternative when n is large and the FSMP exact solution is not easily computed. For $n \in [6, 11]$, the errors were generally worst for $m = n - 1$ (excepting $n = 10$ for which $m = 8$ had the worst error) and occurred for some $p \in (0.3, 0.5)$. For $n \leq 5$, the worst error was less than 10% and occurred for smaller values of m ($m = 2$ for $n = 4$ and 5 and $m = 3$ for $n = 3$). Of the cases evaluated ($n \in [3, 11]$), the errors increased with n .

Of the techniques available for approximating the average stopping time, the alternative computational approximation provides the best overall accuracy, but especially so for values of $p > 0.1$ where $\bar{D}$ is typically evaluated. For evaluating $\bar{T}$ when p is small, the small- p approximation should be adequate; however, care must be taken if n is large. While the trend toward increasing errors with n is troubling, there is little alternative for evaluating $\bar{D}$ when $n > 10$.

3.4 Choosing an optimal m -of- n detector

The standard technique for detector design is to first set the false-alarm rate (FAR) and then optimize for detection performance. However, the Neyman-Pearson lemma, which dictates that a likelihood ratio yields the optimal detector under these conditions, does not apply to the sliding m -of- n detector. Optimality will still be defined as the (m, n) pair that maximizes detection performance while constraining FAR. Detection performance is quantified by the $\bar{D}$ for signals that start and do not stop or have very long durations and by P_d for finite-duration signals.

3.4.1 The choice of m dominates the FAR

Typically, p will be small under the null hypothesis, so using the approximation of (56) will normally be adequate. Consider the logarithm of $\bar{T}$ using (56) with $p = p_0$,

$$\log \bar{T} \approx -m \log p_0 - (n - m + 1) \log (1 - p_0) - \log \left[\binom{n-1}{m-1} \right]. \quad (65)$$

If p_0 is small enough, the first term dominates the second. Then, by noting that the last term is negative, a lower bound can be constructed for the values of m meeting the FAR specification

$$m \geq m_0 = \left\lceil \frac{\log \bar{T}}{-\log p_0} \right\rceil = \left\lceil \frac{\log_{10} \bar{T}}{-\log_{10} p_0} \right\rceil \quad (66)$$

UNCLASSIFIED

approved for public release

where $\lceil \cdot \rceil$ is the ceiling function. For a fixed value of m , and assuming p_0 is small enough that the third term in (65) dominates the second, increasing n decreases $\bar{T}$. When a signal is present, increasing n (as long as the signal duration exceeds n) increases the opportunity of observing m successes and therefore increases P_d . As previously argued, increasing n also reduces $\bar{D}$. Thus, the optimal value of n for a fixed m is the largest value still meeting the false-alarm specification. Define this value as $\bar{n}(m)$. Noting the inequality in (66), there are multiple (m, n) pairs that will meet the FAR specification and provide the largest P_d and smallest $\bar{D}$ as a function of n for each associated value of m . The following two sections describe how to choose the (m, n) pair optimizing $\bar{D}$ or P_d .

3.4.2 Minimizing $\bar{D}$

Consider the example FAR specification of $\bar{T} = 10^{20}$ with $p_0 = 10^{-3}$. The minimum value of m satisfying the false alarm constraint is 7, easily computed from $\lceil \log_{10}(\bar{T}) / \log_{10}(p_0) \rceil$. As seen in Table 3, the largest value of n meeting the FAR specification for $m = 7$ is $\bar{n}(7) = 8$. Increasing m results in successively larger values of the optimal n ; for example, $\bar{n}(10) = 58$ and $\bar{n}(14) = 292$. The larger values of n generally result in less overshoot of the FAR specification and therefore might be expected to provide better detection performance; however, the value of p_1 appears to dominate the choice, as seen from the average delay to detection shown in Table 3. When p_1 is small, (m, n) pairs with larger values provide the best performance (e.g., when $p_1 = 0.3$, $\bar{D}$ is minimized by $m = 10$ and $n = 58$). However, as p_1 approaches one, smaller values of m provide the best performance. In this case, the optimality of the (m, n) pair with the smallest value of m satisfying the FAR specification is evident by noting that the minimum number of samples to a decision is m .

Table 3: Pairs of (m, n) meeting the false alarm specification $\bar{T} = 10^{20}$ for $p_0 = 10^{-3}$, the FAR achieved as computed through the alternative computational approximation, and $\bar{D}$ for several values of p_1 . The bolded entries denote the (m, n) pair minimizing $\bar{D}(p_1)$.

m	$\bar{n}(m)$	$\log_{10} \bar{T}$	$\bar{D}(0.30)$	$\bar{D}(0.50)$	$\bar{D}(0.70)$	$\bar{D}(0.95)$
7	8	20.128	1713.7	100.2	20.6	7.7
8	16	20.150	136.5	24.0	12.2	8.4
9	32	20.065	45.5	18.7	12.9	9.5
10	58	20.022	35.9	20.0	14.3	10.5
11	96	19.996	37.0	22.0	15.7	11.6
12	147	19.995	40.0	24.0	17.1	12.6
13	212	20.005	43.3	26.0	18.6	13.7
14	292	20.013	46.7	28.0	20.0	14.7

Values of m minimizing $\bar{D}$ as a function of $\bar{T}$ and p_1 , denoted as m^* , are listed in Tables 4–6 for,

UNCLASSIFIED

approved for public release

respectively, $p_0 = 10^{-2}$, 10^{-3} , and 10^{-4} . The associated value of n is simply $\bar{n}(m^*)$, the largest value of n meeting the FAR specification for m^* . As seen in Tables 4–6, m^* tends to be m_0 for smaller values of $\bar{T}$ or larger values of p_1 . For example, with $p_0 = 10^{-3}$, setting $\bar{T} \leq 10^5$ resulted in choosing m_0 for $p_1 \geq 0.1$.

Table 4: Value of m minimizing $\bar{D}$ for $p_0 = 10^{-2}$ and various $\bar{T}$ and p_1 .

$\log_{10} \bar{T}$	m_0	m^* for $p_1 =$									
		0.95	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
5	3	3	3	3	3	3	3	3	4	4	5
10	5	5	5	6	7	7	7	8	9	11	15
15	8	8	9	9	10	11	11	12	14	16	23
20	10	11	12	12	13	14	15	16	18	21	31
25	13	14	15	15	16	17	18	20	22	27	39

Table 5: Value of m minimizing $\bar{D}$ for $p_0 = 10^{-3}$ and various $\bar{T}$ and p_1 .

$\log_{10} \bar{T}$	m_0	m^* for $p_1 =$									
		0.95	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
5	2	2	2	2	2	2	2	2	2	2	2
10	4	4	4	4	4	4	4	4	5	5	6
15	5	5	5	6	6	6	7	7	7	8	10
20	7	7	7	8	8	9	9	10	10	11	13
25	9	9	10	10	10	11	11	12	13	14	16

Table 6: Value of m minimizing $\bar{D}$ for $p_0 = 10^{-4}$ and various $\bar{T}$ and p_1 .

$\log_{10} \bar{T}$	m_0	m^* for $p_1 =$									
		0.95	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
5	2	2	2	2	2	2	2	2	2	2	2
10	3	3	3	3	3	3	3	3	3	3	4
15	4	4	4	4	4	5	5	5	5	5	6
20	5	5	5	6	6	6	6	6	7	7	8
25	7	7	7	7	7	8	8	8	9	9	10

While the optimal value of m may be taken from the data in Tables 4–6 if the desired values of p_0 , p_1 , and $\bar{T}$ have been evaluated, computation of $\bar{D}$ for each value of $m \geq m_0$ via (50) or (61) (the latter when $n > 10$) until a minimum is found is cumbersome and needlessly time consuming. An alternative approximate method may be found by exploiting the technique proposed in the following section for choosing m to maximize P_d for a signal of a specific length. By assuming the signal length is equal to the average delay to detection achieved by Page’s test operating on Bernoulli data with no mismatch in the choice of bias, an approximate value can be obtained as

UNCLASSIFIED

approved for public release

a starting point to find m^* . The average delay to detection for Page's test can be approximated using the efficacy (η) from (8) and the FAR specification,

$$\bar{D}_P \approx \frac{\log \bar{T}}{\eta}. \quad (67)$$

The value of m approximately maximizing P_d for a signal of length $\bar{D}_P$ for the example of Table 3 is shown in Fig. 6 as a red circle on the curves of $\bar{D}$ against m with the optimal value of m denoted by a triangle. For $p_1 = 0.3$, this method yielded $m^* \approx 9$ where the optimum value was 10. There was a similar error for $p_1 = 0.5$, but no error for 0.7 or 0.95. The error counts over all cases considered in Tables 4–6 are reported in Table 7 where there appears to be a skew toward underestimating m^* by one. Noting that the initialization was always within one of the optimal value, there are clear reductions in computational effort compared with the brute-force search when $m^* \gg m_0$. As seen in Fig. 6, it is still important to find the optimal value of m definitively as erring toward a smaller value can come at a significant penalty in $\bar{D}$.

Table 7: Counts of occurrences of errors in the initialization approach for m^* over all the cases shown in Tables 4–6.

p_0	$m^* - m_{\text{init}} =$		
	-1	0	1
10^{-2}	2	13	35
10^{-3}	1	31	18
10^{-4}	0	43	7

Noting the rapid rate at which $\bar{n}(m)$ increases with m (e.g., see Table 3), only the technique of (61) is amenable to extensive evaluation of $\bar{D}$. However, it is also important to recall that (65) is an approximation, requiring that p_0 be small. When $\bar{n}(m)$ is large, the approximation can degrade to the point where it is not accurate. In these situations, the numerical technique described by (61) should be used to evaluate $\bar{T}$ as well as $\bar{D}$. For example, the column in Table 3 listing the FAR specification is computed using the alternative computational approximation of (61) for the (m, n) pair obtained using the small- p approximation. While the inaccuracy of the latter is small in this case, it is still evident for $m = 11$ and 12 and can be worse for larger values of p_0 or n . To speed up the evaluation of $\bar{n}(m)$, a Newton-Raphson iteration can be applied to (65) to invert the equation for n by writing the third term using gamma functions,

$$\log \bar{T} \approx -m \log p_0 - (n - m + 1) \log (1 - p_0) - \log \Gamma(m) + \log \Gamma(n - m + 1) - \log \Gamma(n) \quad (68)$$

and noting that the derivative of $\log \Gamma(x)$ is the digamma function $\psi(x)$.

UNCLASSIFIED

approved for public release

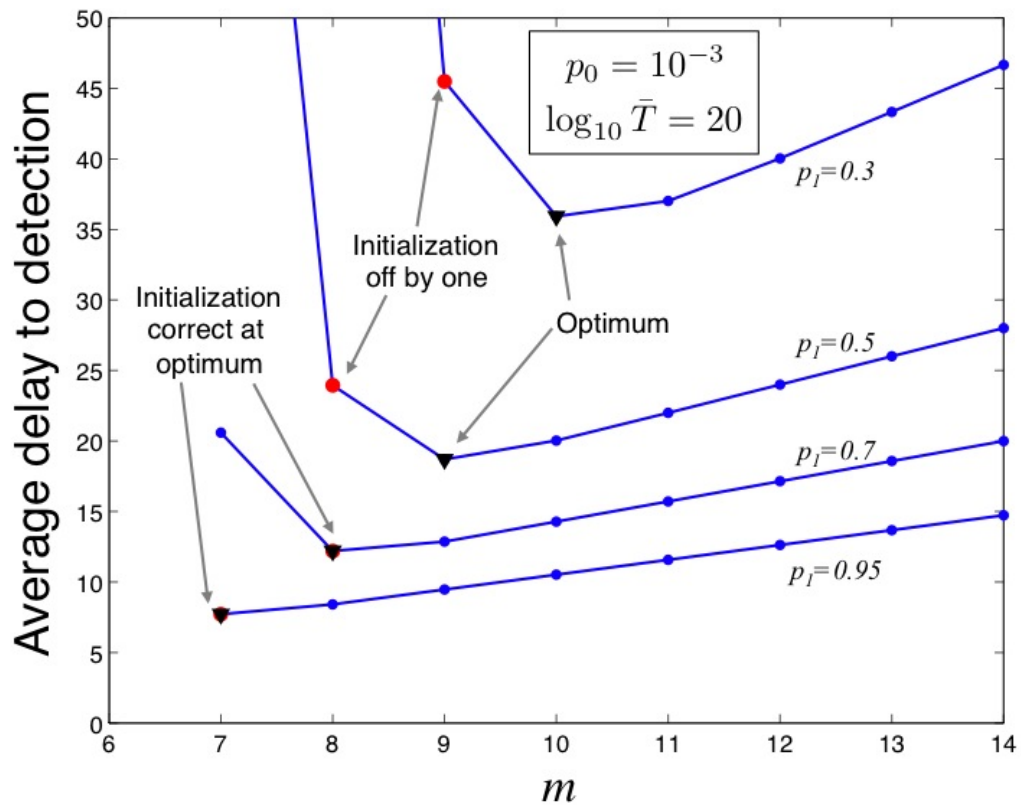


Figure 6: Average delay before detection as a function of m for $p_0 = 10^{-3}$, $\log_{10} \bar{T} = 20$, and $p_I = 0.3, 0.5, 0.7$, and 0.95 . The optimum value of m increases as p_I decreases.

UNCLASSIFIED

approved for public release

3.4.3 Approximately maximizing P_d

Often signals persist only for a short period of time, in which case the probability of detecting the signal is a more appropriate performance measure than the average delay to detection. Consider the example of the previous section (recall $p_0 = 10^{-3}$ and $\log_{10} \bar{T} = 20$) with an individual-trial success probability of $p_1 = 0.75$. As noted in Table 3, the smallest value of m meeting the FAR specification is $m_0 = 7$. The probability of detecting a signal of length k is shown in Fig. 7 for the 7-of-7 and 7-of-8 detectors (solid lines). The dashed curve labeled ‘7-of- k envelope’ is P_d when n equals the signal length and represents an upper bound on the performance of all detectors with $m = 7$ and $n \leq k$. Similarly, the curve labeled ‘8-of- k envelope’ is an upper bound when $m = 8$ and $n \leq k$. While P_d for m -of- n detectors where $k > n$ must be evaluated through (40), the m -of- k performance envelope can be obtained through (43), without encountering the numerical issues of the FSMP approach for large n . This is exploited in developing a technique for choosing the (m, n) pair approximately maximizing P_d .

In Fig. 7, it can be seen that the 7-of-8 detector provides the highest P_d for signals with lengths $k \leq 9$. While a 7-of-9 detector would have higher P_d when $k = 9$, $\bar{n}(7) = 8$ (from Table 3) indicating that the 7-of-9 detector does not meet the FAR specification. When $k = 10$, the 8-of- k envelope shows that the 8-of-10 detector outperforms the 7-of-8 detector. Noting that $\bar{n}(8) = 16$, the 8-of- k detector would then maximize P_d for signals of length $10 \leq k \leq 16$.

Notice that when the signal has length $k = 7$, both the 7-of-7 and 7-of-8 detectors yield the same P_d performance; however, they have different FAR. As previously noted, increasing n when m is fixed reduces $\bar{T}$. As such, any equality in P_d for detectors with the same value of m can be resolved by choosing the smallest value of n ; that is, the smaller of the signal length or $\bar{n}(m)$.

The structure of the P_d curves in Fig. 7 provides a means for approximately choosing the optimal (m, n) pair to maximize P_d while constraining $\bar{T}$ without resorting to evaluating P_d through the FSMP solution found in (40). Divide the signal-length space (the natural numbers) into consecutive, non-overlapping intervals $\mathcal{L}_1, \mathcal{L}_2, \dots$, such that within any interval a fixed value of m approximately maximizes P_d . Call this value m_i^* for the i th interval and let the associated value of n be

$$n^*(m, k) = \min \{k, \bar{n}(m)\}. \quad (69)$$

The first interval is $\mathcal{L}_1 = [1, k_1^*]$ with $m_1^* = m_0$ from (66) as the optimal value of m . The next interval starts at $k_1^* + 1$ and is the point at which a larger value of m has a performance envelope exceeding that achieved with the current value. In Fig. 7, the first interval would be $\mathcal{L}_1 = [1, 9]$ with $m_1^* = 7$ and $n^*(7, k)$. The second starts at $k_1^* + 1 = 10$ with $m_2^* = 8$ and $n^*(8, k)$ and, although not shown, ends at $k_2^* = 17$. The example is continued in Fig. 8, but with a smaller single-trial success probability $p_1 = 0.4$ to further illustrate the optimization process. The smaller value of p_1

UNCLASSIFIED

approved for public release

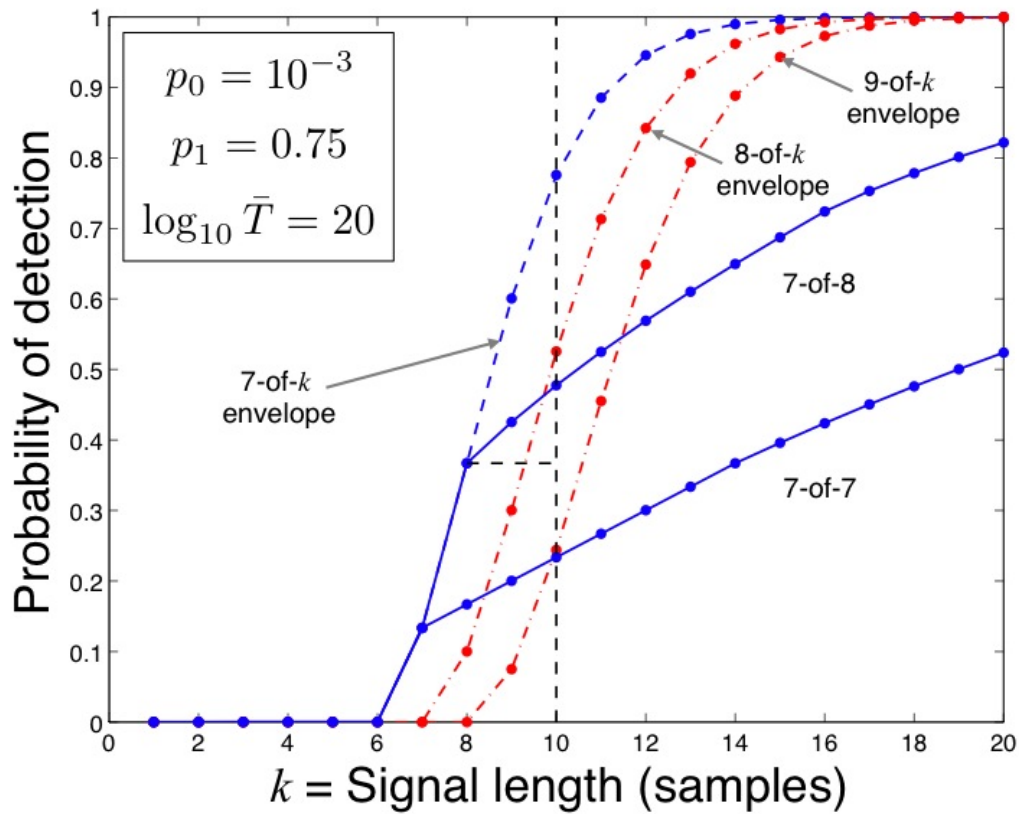


Figure 7: P_d as a function of signal length for various (m, n) pairs. The figure illustrates that a 7-of- $n^*(7, k)$ rule works best when the signal length k is less than 10 where an 8-of- $n^*(8, k)$ rule then works best.

UNCLASSIFIED

approved for public release

doesn't change the first interval; however, the second interval is slightly larger at $\mathcal{L}_2 = [10, 18]$.

Generalizing the description of the signal-length intervals, the i th interval ends at the shortest signal length k_i^* such that a larger value of m in the next region $m_{i+1}^* > m_i^*$ has a P_d envelope greater than that achieved by m_i^* at $k_i^* + 1$ and where the associated value of n meets the FAR specification (i.e., $k_i^* + 1 < \bar{n}(m_{i+1}^*)$). The break-point between the i th and $(i + 1)$ st interval can be described analytically as

$$k_i^* = \arg \min_k \left\{ k : \bar{P}_d(m_{i+1}^*, n^*(m_{i+1}^*, k)) > \bar{P}_d(m_i^*, n^*(m_i^*, k)) \right\} - 1 \quad (70)$$

where the P_d envelope is (following (43)) defined as

$$\bar{P}_d(m, k) = \sum_{j=m}^k \binom{j-1}{m-1} p_1^m (1-p_1)^{j-m} \quad (71)$$

for $k \geq m$ and zero otherwise.

The approximation comes from using the P_d envelope (which is easily computed) held constant at $\bar{n}(m)$ rather than the exact P_d for cases where $k > \bar{n}(m)$. In the example of Fig. 7, the interval is correctly chosen resulting in an optimal selection; however, if the 7-of-8 detector performed slightly better at $k = 10$ than the 8-of- k envelope, the rule of (70) would have incorrectly opted for the 8-of-10 detector. As seen in Fig. 7, the P_d envelope has a higher rate of ascent than the exact P_d for cases where $k > \bar{n}(m)$, so the sub-optimality should be limited to the interval edges. For applications where n is always small (e.g., $n \leq 10$), the exact P_d can be computed and used in place of the P_d envelope on the right-hand-side of the inequality in (70). When n is larger, there is no clear alternative to the proposed technique.

UNCLASSIFIED

approved for public release

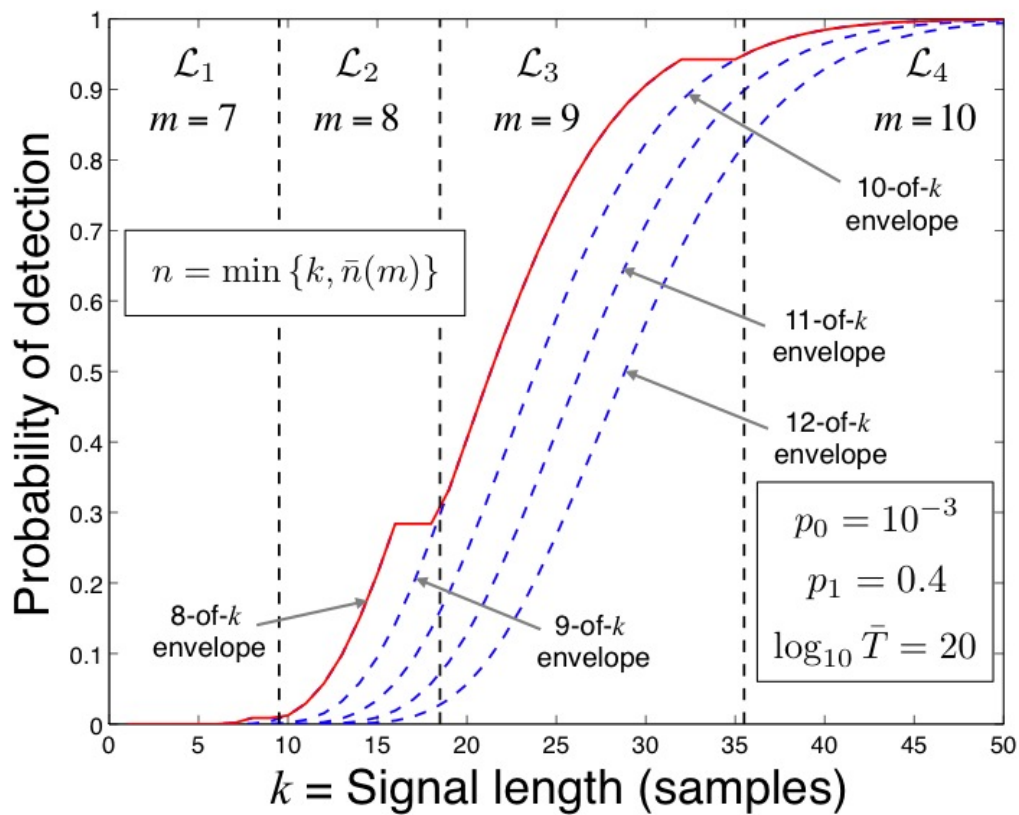


Figure 8: P_d performance envelopes for m -of- n detection as a function of signal length for $p_0 = 10^{-3}$, $p_1 = 0.4$, and $\log_{10} \bar{T} = 20$. As signal length increases, the optimal value of m increases with n the lesser of the signal length and the largest value of n meeting the FAR specification for the given m .

UNCLASSIFIED

approved for public release

4 Detection performance comparison

The detection performance of m -of- n detectors is compared with Page's test both in terms of $\bar{D}$ and P_d for a given FAR specification. It has been proven [6, 7] that Page's test is optimal in terms of providing the lowest $\bar{D}$ for a given $\bar{T}$, so the primary issue is how poorly the m -of- n detectors perform in comparison. As there is no such optimality with respect to P_d , an analysis of performance is required to determine which detector performs best in any given situation.

4.1 Average delay to detection

ROC curves similar to those found in Fig. 1 are shown in Fig. 9 comparing the average delay before detection as a function of the average time between false alarms for the optimal (in the sense of minimizing $\bar{D}$ for a constrained $\bar{T}$) m -of- n detector and Page's test. The case of $p_0 = 10^{-3}$ is considered for various values of p_1 . As expected based on its optimality, Page's test is seen to provide lower $\bar{D}$ than the best m -of- n detector for all cases considered, with an increasing disparity in performance as p_1 decreases.

Both detectors require an assumption on signal characteristics (i.e., p_1) in order to choose the optimal m -of- n rule or the bias in Page's test. The consequence of mismatch as a function of the assumed p_1 is shown in Fig. 10 for $p_0 = 10^{-3}$, $p_1 = 0.5$ and various FAR specifications. When the design value of p_1 equals the true value (0.5 in this example), the performance is optimized; however, mismatch can or will result in a longer average delay to detection. The m -of- n detectors resulted in zero loss in many of the cases considered, reflecting the quantization involved in the design; that is, the same values of m and n will be optimal for a range of conditions rather than a single point condition. Page's test exhibited an immediate performance loss upon any level of mismatch, but still outperformed the m -of- n detectors for the majority of cases considered. For both detectors, the loss arising from mismatch appears exacerbated by larger values of $\bar{T}$.

From the perspective of the average delay before detection, Page's test is clearly a more desirable detector than the sliding m -of- n detector. It enjoys optimality providing the lowest $\bar{D}$ of all tests meeting the FAR specification, but also exhibited less overall sensitivity to mismatch compared with the best m -of- n detector. Additionally, it is easier to implement, only requiring a simple recursion, and to optimize. Changing the FAR specification or the signal-presence probability only requires a change in, respectively, the detector threshold or bias. Changing either for the m -of- n detector requires a completely new evaluation of the optimum values of m and n .

UNCLASSIFIED

approved for public release

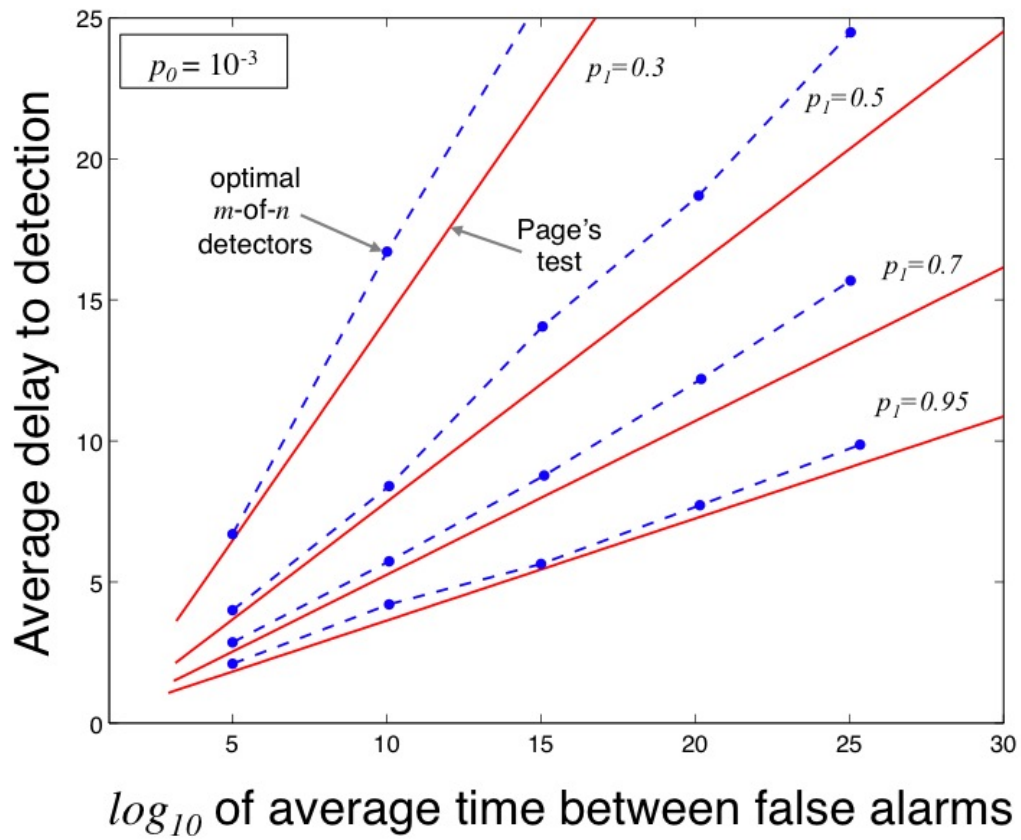


Figure 9: ROC curves of $\bar{D}$ as a function of $\log_{10} \bar{T}$ for the best m -of- n detector and Page's test for $p_0 = 10^{-3}$ and various values of p_1 .

UNCLASSIFIED

approved for public release

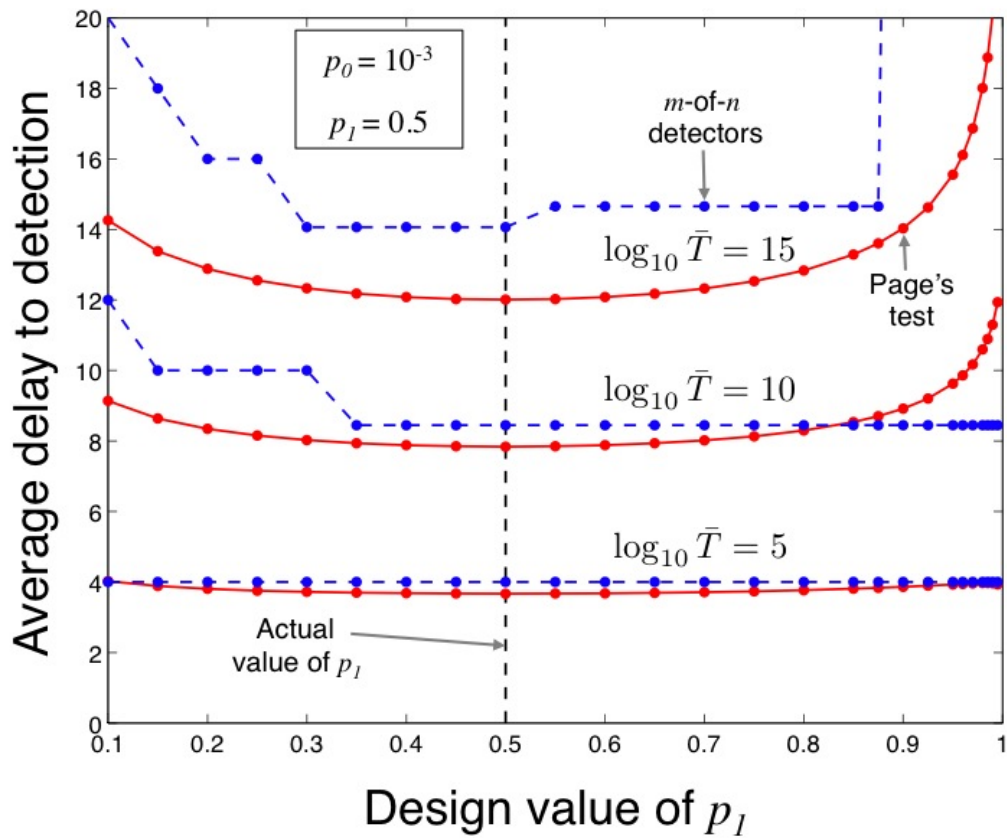


Figure 10: $\bar{D}$ under a mismatch between assumed and actual values of p_I for the best m -of- n detector and Page's test for $p_0 = 10^{-3}$, various values of $\log_{10} \bar{T}$ and a true value of $p_I = 0.5$.

UNCLASSIFIED

approved for public release

4.2 Probability of detection

Based on the optimality of Page's test for minimizing $\bar{D}$ and the Neyman-Pearson optimality of the likelihood ratio test for detection with fixed sample sizes, it is easy to assume that the likelihood ratio would be at the heart of the detector optimizing P_d for a finite-duration signal in a sequential detection setting where signal onset is unknown. Interestingly, the m -of- n detectors appear to provide comparable or better performance than Page's test.

As described in Sect. 3.4.3, evaluating P_d for m -of- n detectors is difficult when n is above about 10, which forces the use of the P_d envelope described by (70) with the breakpoints in (71) dictating when to change m and n chosen from (69). The resulting function is a lower bound on P_d as a function of signal length. In a practical sense, P_d above this lower bound can only be obtained when the signal length is small enough to allow exact evaluation of P_d through the FSMP approach or if simulation analysis is undertaken to approximate P_d .

For Page's test, P_d is evaluated using the FSMP approach described in [8] applied to the binary data and data mapping function of (18). The approach requires quantizing the Page's test detection statistic (Y_k in (2)) and data mapping function in order to characterize Page's test as an FSMP, similar to what was done for the m -of- n detector in Sect. 3.1. The data mapping function of (18) can only produce values of $1 - b(p_0, p_1)$ and $-b(p_0, p_1)$, so the transition matrix for the continuing states is easily constructed. However, accuracy in the result requires a fine enough quantization for there to be many (at least 10) steps between either value and zero. This limits the practical utility of the method for cases where $b(p_0, p_1)$ is near one or zero. In evaluating P_d for both detectors, so called 'latent' detections (those occurring after the signal has stopped, but while the detection statistic is still influenced by signal presence) are ignored.

Consider the case of $p_0 = 10^{-3}$ and $\log_{10} \bar{T} = 10$ shown in Fig. 11 where P_d or its lower bound are shown as a function of signal length for several values of p_1 . Generally, the m -of- n detectors provide equivalent or better detection performance than Page's test. P_d for Page's test appears to initially track the P_d envelope for the m -of- n detectors where m is chosen as the smallest number of Page-test updates allowing a detection declaration,

$$m_p = \left\lceil \frac{h}{1 - b(p_0, p_1)} \right\rceil. \quad (72)$$

However, it shifts to follow the P_d envelope for the next higher value of m before the breakpoint technique described in Sect. 3.4.3 for the m -of- n detectors. As seen in Fig. 11, the m -of- n detector can provide a significant performance improvement over Page's test for signals with lengths in these regions. The P_d curves for $p_1 = 0.5$ are expanded in Fig. 12 and shown along with the 4-of- k detector curves up to $k = 10$. This figure illustrates that the P_d envelope is a lower bound for

UNCLASSIFIED

approved for public release

performance, with the 4-of-10 detector providing better performance than the 5-of- k (which would be chosen by the P_d envelope approach) performance when $k = 12$ through 15. Thus, the range of signal lengths over which the m -of- n detectors outperform Page's test is greater than that indicated by the P_d envelope, but not necessarily attainable if it requires evaluating P_d through the FSMP approach.

It can be seen in Fig. 12 that if the signal has length 6, Page's test and the 4-of- k detectors for $k = 4, \dots, 10$ all provide the same P_d . As noted in Sect. 3.4.3, choosing the 4-of-4 detector results in the best achieved FAR. This can be seen in Fig. 13 where it clearly has the highest average time between false alarms not only for the design value of $p_0 = 10^{-3}$, but also if the observed value of p_0 differs from the design value. In fact, the 4-of-4 detector is robust enough to meet the FAR specification even when the observed p_0 is three times greater than the design value. As seen in Fig. 13, the robustness decreases as $n \rightarrow \bar{n}(m)$, the maximum value meeting the FAR specification for the given value of m . However, from Figs. 11 and 12, the m -of- n detectors have higher P_d than Page's test in this region of signal length.

The primary disadvantage of the m -of- n detectors with respect to maximizing P_d is that different (m, n) pairs are required to optimize the performance for each signal length. The performance results for Page's test seen in Figs. 11–13 are obtained with just one design (i.e., bias and threshold) for each value of p_1 , irrespective of the signal length. Noting the significant loss in performance seen in Fig. 12 when n is less than the signal length, it is better to overestimate the signal length if it is not known precisely or to use Page's test. However, it is important to maintain the optimal value of m ; that is, overestimating to the point where m increases will result in a lower P_d while merely increasing n such that $n \leq \bar{n}(m)$ will maintain P_d and trade robustness in FAR for P_d optimality over a broader range of signals (e.g., using the 4-of-10 detector optimizes P_d for signals of length $k \leq 10$ and just meets the FAR specification). Thus, there are clear advantages in using the m -of- n detectors rather than Page's test when the signal length is known.

UNCLASSIFIED

approved for public release

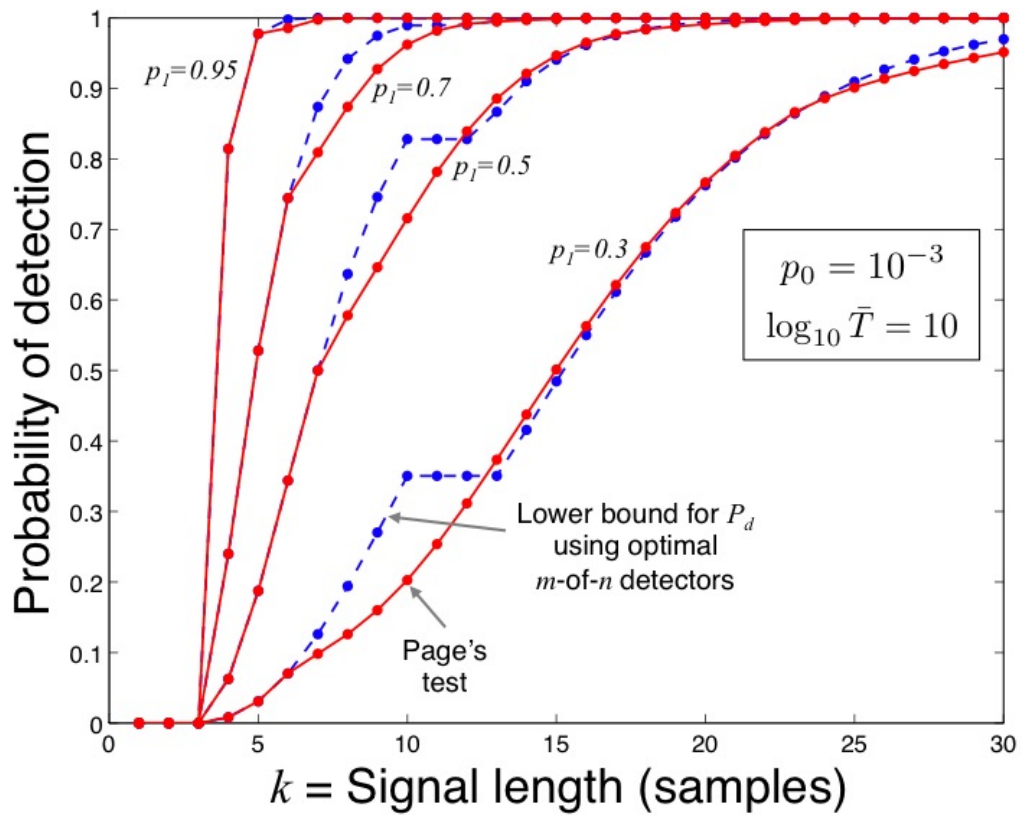


Figure 11: P_d as a function of signal length for Page's test and the P_d envelope (lower bound) for the optimal m -of- n detectors for $p_0 = 10^{-3}$, $\log_{10} \bar{T} = 10$ and various values of p_1 .

UNCLASSIFIED

approved for public release

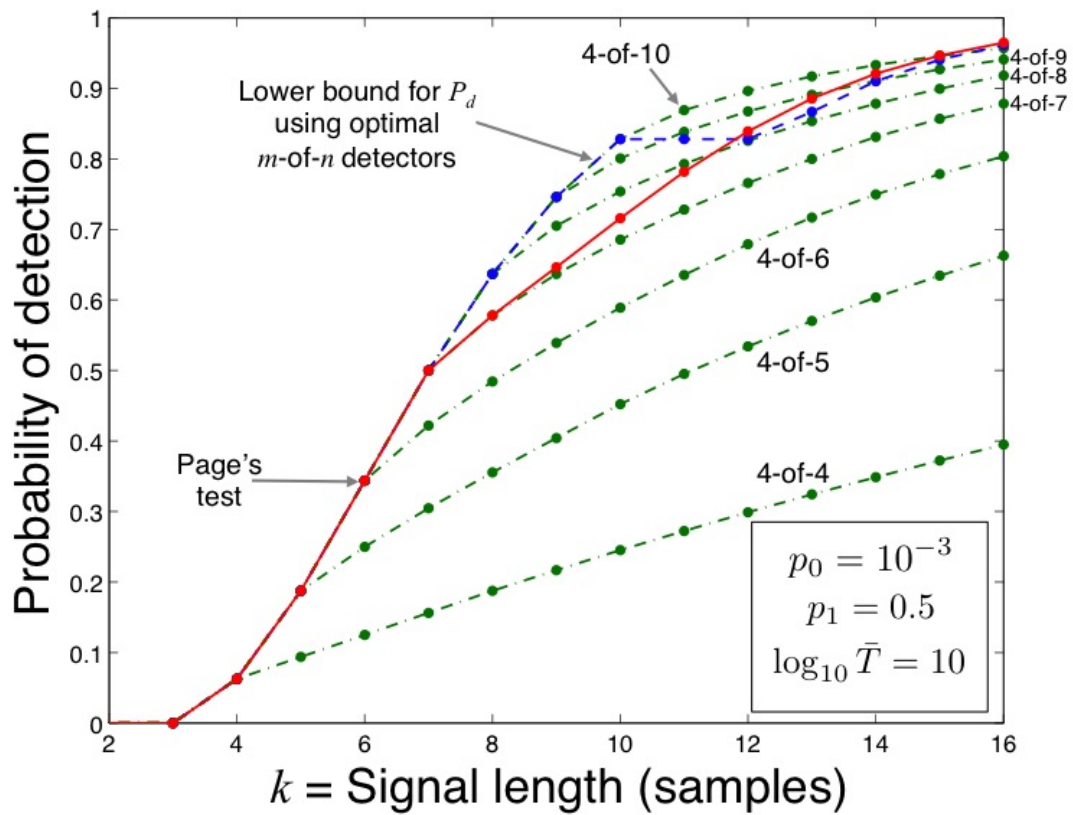


Figure 12: P_d as a function of signal length for Page's test and various m -of- n detectors for $p_0 = 10^{-3}$, $\log_{10} \bar{T} = 10$ and $p_1 = 0.5$.

UNCLASSIFIED

approved for public release

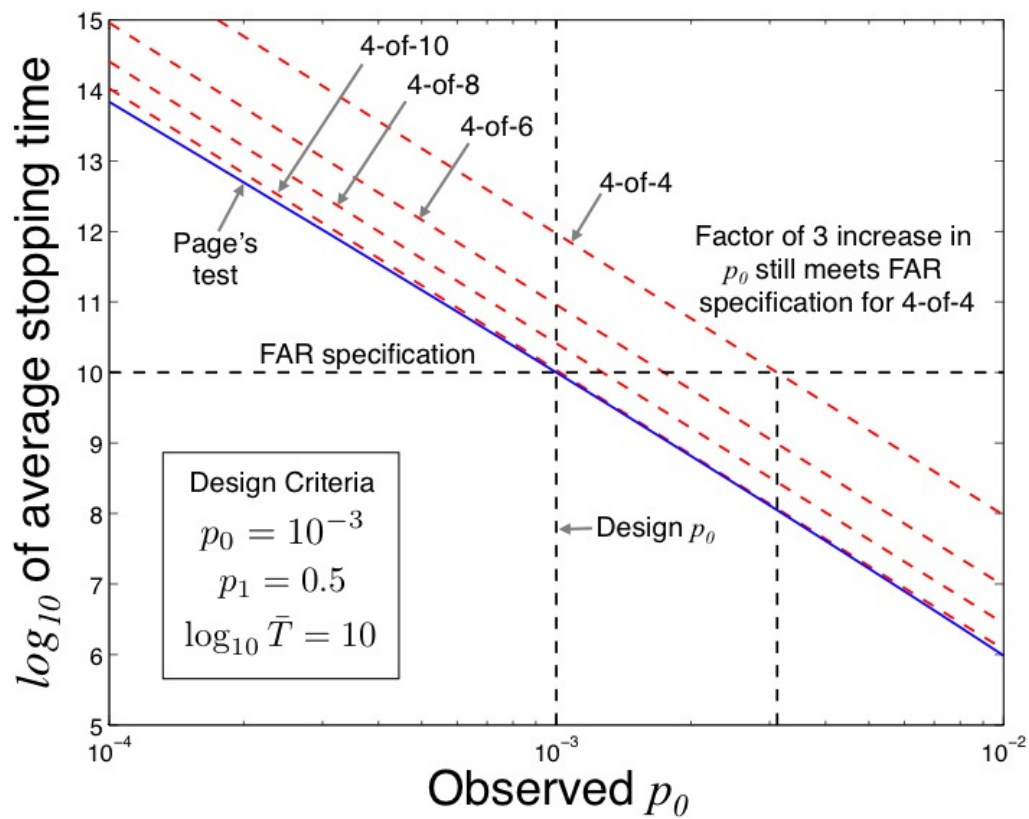


Figure 13: FAR achieved with detectors designed for $p_0 = 10^{-3}$, $\log_{10} \bar{T} = 10$ and $p_1 = 0.5$ when the observed value of p_0 differs by up to an order of magnitude. Some of the m -of- n detectors are robust enough to meet the FAR specification even when the observed p_0 is over three times larger than the design value.

UNCLASSIFIED

approved for public release

5 Conclusions

This report has presented analysis techniques for the sliding m -of- n detector to enable design meeting a FAR specification and approximately optimizing detection performance. Exact computational techniques were presented for evaluation of $\bar{D}$, $\bar{T}$, and P_d through the FSMP description of the sliding m -of- n detector presented in [3] under stationary data conditions. As the number of states grows exponentially with n , these techniques become computationally cumbersome for $n > 10$ and impractical shortly thereafter. Approximations from scan-statistics literature [5] may be useful for larger values of n , but are still limited at smaller values of p and assume a minimum detection time of n rather than m , restricting their utility to evaluation of $\bar{T}$ for moderate values of p , unless extensions exist to accommodate this alternative stopping criteria. To counter the numerical limitations of these approaches, approximations applicable for a much larger range of n were developed, including a small- p approximation useful for evaluating $\bar{T}$ and an alternative computational approximation for $\bar{D}$.

The value of m was seen to dominate the FAR of the sliding m -of- n detector, although multiple (m, n) pairs with m above a minimum value were found to meet the specification. Of these, a unique pair can be chosen to either minimize $\bar{D}$ or approximately maximize P_d . The latter was done by choosing the (m, n) pair maximizing an easily evaluated lower bound on the probability of detecting a finite-duration signal using m -of- n detectors. Given the optimal m , any value of n from the signal length up to $\bar{n}(m)$ (the maximum value meeting the FAR specification) maximizes the P_d lower bound, with the lower values having greater FAR robustness but a smaller range of signal lengths over which they are optimal. Designing the detectors to minimize $\bar{D}$ was less complicated with the optimal value of m approximated by that optimizing P_d for a signal of length equal to the average delay to detection achieved by Page's test. The value of n minimizing $\bar{D}$ was then the largest one meeting the FAR specification.

While the m -of- n detectors operate on binary or decision-level data, Page's test can be applied to binary data or the pre-thresholded data. An example illustrated the significant loss in performance when decision-level data are used unnecessarily. The performance comparison between Page's test operating on binary data and the sliding m -of- n detectors showed that the latter can come close to the optimal performance of Page's test with respect to the average delay to detection, but not exceed it except under extreme mismatch. Although the sliding m -of- n detector was robust to minor mismatch, often yielding no loss, it could result in significant loss at larger levels of mismatch. The sliding m -of- n detectors clearly outperformed Page's test with respect to the probability of detecting finite-duration signals. In cases where the detection performance was similar, the sliding m -of- n detectors exhibited significant FAR robustness to varying individual trial false-alarm probability (p_0) where Page's test did not. Although not considered extensively in this report, design of the threshold used to produce decision-level data should be incorporated into the optimization process.

UNCLASSIFIED

approved for public release

An initial approximation was obtained by approximately optimizing the asymptotic efficiency of Page's test.

The primary disadvantage of the sliding m -of- n detectors is that optimal design requires different configurations (i.e., (m, n) pairs) that depend on the FAR specification, the individual-trial success probabilities and signal length. With Page's test's optimality for minimizing $\bar{D}$ and simpler design, it remains the most desirable detector for signals of unknown or infinite length. However, the performance improvement attained by a properly designed m -of- n detector over Page's test for detecting finite-duration signals was significant enough that they are the clear choice when signal length is known or known within a range where one optimal sliding m -of- n detector can be implemented.

References

- [1] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, pp. 100–114, 1954.
- [2] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*. YBS Publishing, 2011.
- [3] P. Williams, "Evaluating the state probabilities of m out of n sliding window detectors," Maritime Operations Division Aeronautical and Maritime Research Laboratory, Technical Report DSTO-TN-0132, February 1998. [Online]. Available: http://www.dsto.defence.gov.au/publications/scientific_record.php?record=3652
- [4] P. K. Varshney, "Multisensor data fusion," *Electronics & Communication Engineering Journal*, vol. 9, no. 6, pp. 245–253, 1997.
- [5] J. Glaz, J. Naus, and S. Wallenstein, *Scan Statistics*. New York: Springer-Verlag, 2001.
- [6] G. Lorden, "Procedures for reacting to a change in distribution," *The Annals of Mathematical Statistics*, vol. 42, no. 6, pp. 1897–1908, 1971.
- [7] G. V. Moustakides, "Optimal stopping times for detecting changes in distributions," *The Annals of Statistics*, vol. 14, no. 4, pp. 1379–1387, 1986.
- [8] C. Han, P. K. Willett, and D. A. Abraham, "Some methods to evaluate the performance of Page's test as used to detect transient signals," *IEEE Transactions on Signal Processing*, vol. 47, no. 8, pp. 2112–2127, August 1999.
- [9] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, 1993.

UNCLASSIFIED

approved for public release

- [10] B. Broder, “Quickest detection procedures and transient signal detection,” Ph.D. dissertation, Princeton University, Princeton, NJ, 1990.
- [11] M. A. Richards, *Fundamentals of Radar Signal Processing*. New York: McGraw-Hill, 2005.
- [12] R. G. Gallager, *Discrete Stochastic Processes*. Norwell, Massachusetts: Kluwer Academic Publishers, 1996.
- [13] K. Oldham, J. Myland, and J. Spanier, *An Atlas of Functions*, 2nd ed. New York: Springer Science, 2009.

UNCLASSIFIED

approved for public release

A MGF unity roots

A-1 Exponential data

Finding the unity root of the MGF is equivalent to finding the root (zero) of the logarithm of the MGF, which is the cumulant generating function (CGF). For the exponentially distributed data described in Sect. 2, the CGF of $g(X)$ under the signal-present hypothesis is

$$\Psi_s(t) = -b(\tilde{s})t - \log[1 - \lambda(1 + s)t] \quad (73)$$

for $t < [\lambda(1 + s)]^{-1}$. The CGF under the null hypothesis is simply $\Psi_0(t)$, which still depends on the design SNR $\tilde{s}$. Under the null hypothesis and the condition stated in (4), $E_0[g(X)] < 0$, it can be shown that $t_0 > 0$. Similarly, $t_1 < 0$ under the alternative hypothesis when $E_1[g(X)] > 0$. Note that some values of $s < \tilde{s}$ may violate the condition stated in (4) yielding $t_1 < 0$, and very poor detection performance.

The procedure for obtaining t_1 is described below; t_0 may be obtained by setting $s = 0$ without altering $\tilde{s}$. The root t_1 ($\Psi_s(t_1) = 0$) is easily calculated through a Newton-Raphson iteration (NRI),

$$t_1 \leftarrow t_1 - \frac{\Psi_s(t_1)}{\Psi'_s(t_1)}, \quad (74)$$

where the the derivative of the CGF is

$$\Psi'_s(t) = -b(\tilde{s}) + \frac{1}{[\lambda(1 + s)]^{-1} - t}. \quad (75)$$

The NRI must be initialized far enough away from the origin that it converges on the non-zero root as opposed to the root at the origin. This is accomplished by starting beyond the minimum point of the CGF occurring at

$$t_* = \frac{1}{\lambda(1 + s)} - \frac{1}{b(\tilde{s})} \quad (76)$$

which sets the derivative in (75) to zero. When the root is negative, the NRI can be started at $2t_*$; however, when it is positive, it must be started between t_* and the upper limit on the range of t for which the MGF is defined.

A-2 Bernoulli data

The CGF of the Bernoulli data with probability p is

$$\Psi_p(t) = \log \left[pe^{t(1-\tilde{b})} + (1-p)e^{-t\tilde{b}} \right] \quad (77)$$

UNCLASSIFIED

approved for public release

and its derivative is

$$\Psi'_p(t) = \frac{p(1 - \tilde{b})e^{t(1-\tilde{b})} - (1 - p)\tilde{b}e^{-t\tilde{b}}}{pe^{t(1-\tilde{b})} + (1 - p)e^{-t\tilde{b}}}. \quad (78)$$

The non-zero roots are again found using a NRI as described in (74). When the root $t_p > 0$ (i.e., $p < \tilde{b}$), the iteration can be initialized using the approximation

$$\Psi_p(t_p) \approx \log \left[pe^{t_p(1-\tilde{b})} \right]. \quad (79)$$

Equating (79) to zero and solving results in

$$t_p = \frac{-\log(p)}{1 - \tilde{b}}. \quad (80)$$

Alternatively, when $t_p < 0$ (i.e., $p > \tilde{b}$) the approximation

$$\Psi_p(t_p) \approx \log \left[(1 - p)e^{-t_p\tilde{b}} \right] \quad (81)$$

yields the NRI initialization

$$t_p = \frac{\log(1 - p)}{\tilde{b}}. \quad (82)$$

UNCLASSIFIED

approved for public release

B Matlab code for m -of- n detector performance evaluation and design

B-1 Subroutines to represent the sliding m -of- n detector as an FSMP

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S=mofn_setup_opt(m,n)
% S=mofn_setup_opt(m,n)
% Returns a structure S of states characterizing a sliding m-of-n detector.
% S.ns = number of states
% S.H = integer representation of the state
% S.Hp = transitioning state upon a success
% S.iHp = index of state in S.Hp
% S.Hm = transitioning state upon a failure
% S.iHm = index of state in S.Hm
% S.P = temporary storage for Williams (DSTO-TN-0132) algorithm for Pd

% Data are stored in a .mat file so only have to do this once
fname=['mofn_struct_n' int2str(n)];
if (exist([fname '.mat'],'file')==2),
    load(fname)
else
    % Set structures to be empty for each m on first use
    for i=1:n,
        Sm{i}=[];
    end;
end;
% Calculate structure if not already stored
if isempty(Sm{m})==1,
    disp('---- Stand by, calculating structures ----');
    fprintf(' m=%i n=%i ',m,n);
    % Add the accepting state
    i=1;
    S.ns=i;
    S.P(i)=0;
    S.H(i)=2^n;
    S.Hp(i)=2^n;
    S.Hm(i)=2^n;
    % Number of non-accepting states (eq 2 in DSTO-TN-0132)
    % Easier to enumerate and find number of non-accepting states
    hiall=0:(2^(n-1)-1); % integer rep of each possible history
    ibin=binnum(hiall); % sum of 1's of binary rep of each state
end;

```

UNCLASSIFIED

approved for public release

```

hna=hiall(ibin<m); % integer equivalents of non-accepting states
% Loop through all possible non-accepting-states
nnas=length(hna);
fprintf('# states = %i \n',nnas);
for i=1:nnas,
    if (nnas>500)&&(rem(i,round(nnas/10))==1),
        fprintf('.');
    end;
    % Reduce to smallest equivalent state
    hna0=reduce_hexad(hna(i),m,n);
    % If it hasn't been added, then add it
    if ~any(S.H==hna0),
        S=addhexad_opt(S,hna0,m,n);
    end;
end;
% Sort hexads by H
[~,isrt]=sort(S.H);
S.H=S.H(isrt);
S.Hp=S.Hp(isrt);
S.Hm=S.Hm(isrt);
S.P=S.P(isrt);
% Add indices for each
S.iH=1:S.ns;
for i=1:S.ns,
    S.iHp(i)=find(S.Hp(i)==S.H);
    S.iHm(i)=find(S.Hm(i)==S.H);
end;
% Save data to .mat file for later retrieval
Sm{m}=S;
eval(['save ' fname ' Sm']);
% If structure is stored, then return it
else
    S=Sm{m};
end;
[S.H' S.Hm' S.Hp' S.iHm' S.iHp'];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function S=addhexad_opt(S,h,m,n)
% S=addhexad_opt(S,h,m,n)
% Function internal to mofn_setup_opt(m,n) to add a hexad to structure
%
i=S.ns+1; % Find index for this hexad
S.ns=i; % Increment number of hexads
S.P(i)=0;
S.H(i)=h; % Current hexad's integer state representation

```

UNCLASSIFIED

approved for public release

```

% If next trial is a failure
hm=mod(2*h,2^(n-1)); % Integer state
hm0=reduce_hexad(hm,m,n); % Reduce to smallest equivalent state
S.Hm(i)=hm0;
% If next trial is a success
% If meeting m-of-n criteria
if binnum(h)==(m-1),
    S.Hp(i)=2^n; % Accepting state integer ID
% Otherwise into a continuing state
else
    hp=hm+1; % Integer state
    S.Hp(i)=reduce_hexad(hp,m,n); % Reduce to smallest equivalent state
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function h0=reduce_hexad(h,m,n)
% h0=reduce_hexad(h,m,n)
% Function internal to mofn_setup_opt(m,n) to reduce a state to the
% "lowest common state", which minimizes the number of unique states
% in the m-of-n FSMP
h0=h;
if binnum(h)~=(m-1),
    hp=mod(2*h,2^(n-1))+1;
    np=reduce_hexad(hp,m,n);
    if np<h,
        h0=floor(np/2);
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function b=binnum(i)
% b=binnum(i)
% Returns the number of 1's in the binary
% representation of integer i
%   binnum(2)=1
%   binnum(3)=2
%   binnum(4)=2
% Input i can be a vector (output is a column vector)
x=dec2bin(i(:));
[nr,nc]=size(x);
b=sum(reshape(str2num(reshape(x,nr*nc,1)),nr,nc)')';

```

UNCLASSIFIED

approved for public release

B-2 Subroutines for performance measures from FSMP

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [uK,sK,err]=mofn_stats(pv,S,P0v)
% [uK,sK]=mofn_stats(pv,S,P0v)
% Returns the mean and standard deviation of the stopping time
% for the m-of-n process defined by the structure S.
%   pv = vector of Bernoulli probabilities to be evaluated
%   S = structure describing m-of-n process (see mofn_setup)
%   P0v = vector of initial starting probabilities
%         = [1 0 ... 0]' by default
%
% uK = average stopping time (latency) for each value in pv
% sK = standard deviation of stopping time
%
[Pcp,Pcm,pcp]=mofn_states(S);
ns=length(pcp)+1;
if nargin<3,
    P0v=[1;zeros(ns-2,1)];
else
    P0v=P0v(1:(ns-1));
end;
uK=zeros(size(pv));
sK=uK;
for i=1:length(pv),
    % Construct continuing state trans matrix
    Pcc=pv(i)*Pcp+(1-pv(i))*Pcm;
    pcs=pv(i)*pcp;
    % Eigen decomposition
    [Vc,Dc]=eig(Pcc);
    dc=diag(Dc);
    % If eigenvectors are not linearly independent use pseudoinverse
    if rcond(Vc)<1e-10,
        cc=pinv(Vc)*pcs;
    else
        cc=Vc\pcs;
    end;
    % Check to make sure pseudoinverse worked
    err1=max(abs(pcs-Vc*cc));
    if err1>sqrt(eps), disp('**** Error ****'); end;
    err(i)=rcond(Vc);
    % Form coefficients
    c=(P0v'*Vc).'*(cc);
    % Form mean, power & standard deviation

```

UNCLASSIFIED

approved for public release

```

    uK(i)=real(sum(c./(1-dc).^2));
    pK=real(sum(c.*(1+dc)./(1-dc).^3));
    sK(i)=sqrt(pK-uK(i)^2);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Pd=mofn_pd(pv,Lv,m,n)
% Pd=mofn_pd(pv,Lv,m,n)
% Returns Pd for set of success probabilities in pv for the time
% indexes in the vector Lv
% Output Pd is a matrix with length(pv) rows and length(Lv) columns
%
S=mofn_setup_opt(m,n);
% Don't do this if there are too many states
if S.ns>1010,
    Pd=nan*ones(size(Lv));
else
    [Pcp,Pcm,pcp]=mofn_states(S);
    ns=length(pcp)+1;
    P0v=[1;zeros(ns-2,1)];
    nL=length(Lv);
    np=length(pv);
    Pd=zeros(np,nL);
    % Construct continuing state trans matrix
    for i=1:np,
        Pcc=pv(i)*Pcp+(1-pv(i))*Pcm;
        pcs=pv(i)*pcp;
        % Eigen decomposition
        [Vc,Dc]=eig(Pcc);
        dc=diag(Dc);
        c=conj(Vc'*P0v).*(Vc\pcs);
        Pd(i,:)=abs(sum(repmat(c./(1-dc),1,nL).*(1-repmat(dc,1,nL).^repmat(Lv(:)',ns-1,1))));
    end;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [Pcp,Pcm,pcp]=mofn_states(S)
% [Pcp,Pcm,pcp]=mofn_states(S)
% Returns the indicator matrices and vector required to form the
% FSMP transition probability matrix.
ns=S.ns;
Pcp=zeros(ns-1);
Pcm=Pcp;
pcp=zeros(ns-1,1);

```

UNCLASSIFIED

approved for public release


```

for i=1:(ns-1),
    if S.iHp(i)==ns,
        pcP(i)=1;
    else
        PcP(i,S.iHp(i))=1;
    end;
    Pcm(i,S.iHm(i))=1;
end;

```

B-3 Subroutines for performance measure approximations

```

function [uK,sK]=mofn_stats_asy(p,m,n)
% [uK,sK]=mofn_stats_asy(p,m,n)
% Returns the mean and standard deviation of the stopping time
% for an m-of-n process when the probability of success is small
% (i.e., pv -> zero)
%
% Primarily useful in evaluating average time between false alarms.
%
% p = Bernoulli probability of success
% m = number of successes required for stopping
% n = window over which successes are counted
% Any input can be a vector
%
% uK = average stopping time (latency) for each value in pv
% sK = standard deviation of stopping time
%

% Evaluate (n-1)-choose-(m-1) using gamma functions to allow vector inputs
if m==1,
    pa=p;
else
    pa=exp(gammln(n)-gammln(m)-gammln(n-m+1)).*(p.^m).*((1-p).^(n-m+1));
end;

uK=1./pa;
sK=sqrt(1-pa)./pa;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [uK,sK]=mofn_stats_cfcn(p,m,n,delmin)
% [uK,sK]=mofn_stats_cfcn(p,m,n)
% Returns the mean and standard deviation of the stopping time
% for an m-of-n process when the probability of success is small

```

UNCLASSIFIED

approved for public release

```

% (i.e., pv -> zero)
% p = Bernoulli probability of success
% m = number of successes required for stopping
% n = window over which successes are counted
% delmin = minimum step size to evaluate integral
%
% uK = average stopping time (latency) for each value in p
% sK = standard deviation of stopping time
%
if nargin<4,
    delmin=[];
end;
if m==1,
    pa=exp((gammaln(n)-gammaln(m)-gammaln(n-m+1))).*(p.^m);
else
    pa=(1/m)*p.*exp(-(m-1)*mofn_cfcn_int(-log(p),mofn_cfcn(m,n),delmin));
end;
uK=1./pa;
sK=sqrt(1-pa)./pa;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function c=mofn_cfcn(mv,nv)
% c=mofn_cfcn(mv,nv)
% Returns the value of c(m,n) for each element in the vectors mv & nv

% Make vectors the same size if one is a scalar
nmax=max(nv);
if (length(nv)==1)&&(length(mv)>1)
    nv=nv*ones(size(mv));
elseif (length(mv)==1)&&(length(nv)>1),
    mv=mv*ones(size(nv));
end;
c=zeros(size(nv));
% Load file see if data for this case is present
fname='mofn_cfcn_data';
if (exist([fname '.mat'],'file')~=2),
    for i=1:nmax,
        Cs{i}=[0; nan*ones(i-1,1)];
    end;
else
    load(fname);
end;
% Populate additional cells w/ nans up to n
nCs=length(Cs);
for i=(nCs+1):nmax,

```

UNCLASSIFIED

approved for public release

```

    Cs{i}=[0; nan*ones(i-1,1)];
end;
opts=optimset('TolX',1e-8);
log10e=log10(exp(1));
delmin=0.01;
qnew=0;
for i=1:length(nv),
    n=nv(i);
    m=mv(i);
    % See if data for present case exists
    if ~isnan(Cs{n}(m)),
        c(i)=Cs{n}(m);
    else
        qnew=1;
        % Get best fit exponent
        p0v=0.1/n;
        u0v=log(p0v);
        % Use asymptotic approximation as barometer
        Ft=-log10e*(gammaln(n)-gammaln(m)-gammaln(n-m+1))-m*log10(p0v)-(n-m+1)*log10(1-p0v);
        % Find optimal exponent
        c(i)=fminbnd(@(c)((log10(m)-log10(p0v)+...
            (m-1)*mofn_cfcn_int(-u0v,c,delmin)*log10e-Ft).^2),0,n-1,opts);
        Cs{n}(m)=c(i);
    end;
end;
% Save the data if any new ones added
if qnew,
    eval(['save ' fname ' Cs']);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function F=mofn_cfcn_int(xu,c,delmin)
% F=mofn_cfcn_int(xu,c,delmin)
% Subroutine used by c=mofn_cfcn(mv,nv)

% Spacing of approximations in -log(p)
if nargin<3,
    delmin=[];
end;
if ~any(delmin), delmin=0.25; end;
del=min([delmin c/2]);
% Center points for quadratic Taylor approximation
xc=(del/2):del:(max(xu)+del/2);
[fc,fp1c,fp2c]=mofn_cfcn_deriv(xc,c);
% Coefficients of quadratic Taylor approximation

```

UNCLASSIFIED

approved for public release

```

b0=(fp2c/6);
b1=0.5*(fp1c-xc.*fp2c);
b2=fc-xc.*fp1c+fp2c.*(xc.^2)/2;
% Evaluation points for integral at region edges
xim=xc-del/2;
xip=xc+del/2;
% Integral over each region
Fpm=b0.*(xip.^3-xim.^3)+b1.*(xip.^2-xim.^2)+b2.*(xip-xim);
% Find nearest point for each element of xu
F=zeros(size(xu));
for i=1:length(xu),
    [~,iu]=min(abs(xip-xu(i)));
    % Integral from last region edge to desired point
    Fu=b0(iu)*(xu(i)^3-xip(iu)^3)+b1(iu).*(xu(i)^2-xip(iu)^2)+b2(iu).*(xu(i)-xip(iu));
    % Final result
    F(i)=sum(Fpm(1:iu))+Fu;
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [f,fp1,fp2]=mofn_cfcn_deriv(x,c)
% [f,fp1,fp2]=mofn_cfcn_deriv(x,c)
% Function used by mofn_cfcn_int
ex=exp(x);
f=(1-exp(-x)).^c;
fp1=c*f./(ex-1);
fp2=fp1.*(c-ex)./(ex-1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [pd,k]=mofn_pd_k_less_n(p,m,n)
% [pd,k]=mofn_pd_k_less_n(p,m,n)
% Returns Pd for the case of k<=n using exact formula
%   p = scalar probability of success
%   k = 1:n
k=1:n;
pd=zeros(size(k));
k1=m:n;
pd(k1)=cumsum(exp(gammain(k1)-gammain(m)-gammain(k1-m+1)+m*log(p)+(k1-m)*log(1-p)));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function POv=mofn_steadystate(p0f,h)
% POv=mofn_steadystate(p0f,h)
% Returns the steady-state probabilities of each state
% given a long-term probability of succes of p0f (e.g., a
% false alarm probability) and assuming no successes.
%
```

UNCLASSIFIED

approved for public release

```
% Note that when p0f is small, P0v is approximately
%   P0v ~ [1 0 ... 0]
%
if p0f>0,
    P0v=p0f.^binnum(h);
    P0v(end)=0; % Set accepting state probability to zero
    P0v=P0v/sum(P0v); % Condition on being in non-accepting state
else
    P0v=[1; zeros(length(h)-1,1)];
end;
```

B-4 Subroutines for m -of- n detector design

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [mv,nv,log10Tv]=mofn_far_spec(p0,log10T,nsol)
% [mv,nv,log10Tv]=mofn_far_spec(p0,log10T,nsol)
% Returns nsol pairs of (m,n) meeting the FAR spec
% and the value of log10(Tbar) achieved by the pair
icmax=20;
if nargin<3,
    nsol=5;
end;
m0=ceil(log10T/-log10(p0));
log10e=log10(exp(1));
for i=1:nsol,
    m=m0+i-1;
    mv(i)=m;
    n0=mv(i);
    err=2;
    icv=0;
    while (err>0.6)&&(icv<icmax)&&(n0<1000),
        icv=icv+1;
        fa=log10e*(-gamma(m,n0)+gamma(m,n0-m+1)+gamma(m)-m*log(p0)-(n0-m+1)*log(1-p0));
        fpa=log10e*(-psi(n0)+psi(n0-m+1)-log(1-p0));
        deln=(fa-log10T)/fpa;
        n0=n0-deln;
        err=abs(deln);
    end;
    nv(i)=floor(n0);
end;
log10Tv=log10e*(gamma(mv)+gamma(nv-mv+1)-gamma(nv)-mv*log(p0)-(nv-mv+1)*log(1-p0));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

UNCLASSIFIED

approved for public release

```

function [B,pd,k]=mofn_pd_breakpoint(p0,p1,log10T,Lmax)
%[B,pd,k]=mofn_pd_breakpoint(p0,p1,log10T,Lmax)
% Returns the breakpoints for m-of-n detector design to maximize
% the lower bound on Pd formed from the m-of-k Pd envelopes
%
% p0 = noise-only success probability
% p1 = noise+signal success probability
% log10T = log10(Tbar) FAR specification
% Lmax = maximum signal length of interest
%
% B.mv = value of m in each region
% B.nv = maximum value of n meeting FAR spec for each m
% B.L0 = beginning of each region
% B.L1 = end of each region
% pd = Pd lower bound
% k = vector of signal lengths associated with pd

% Get (m,n) pairs meeting FAR spec
nsol=5;
[mv,nv,log10Tv]=mofn_far_spec(p0,log10T,nsol);
while max(nv)<Lmax+1,
    nsol=nsol+1;
    [mv,nv,log10Tv]=mofn_far_spec(p0,log10T,nsol);
end;
nsol=find(nv>Lmax,1,'first');
mv=mv(1:nsol);
nv=nv(1:nsol);
pd=zeros(1,Lmax);
[pd0,k0]=mofn_pd_k_less_n(p1,mv(1),nv(1));
B.mv(1)=mv(1); % currently active value of m
B.nv(1)=nv(1);
B.L0(1)=1; % first signal length for this region
B.L1c(1)=nv(1);
ir=1;
pd(k0)=pd0;
k_new=[];
for i=1:(nsol-1),
    [pd1,k1]=mofn_pd_k_less_n(p1,mv(i+1),nv(i+1));
    if pd1(end)>pd0(end),
        ir=ir+1;
        B.mv(ir)=mv(i+1);
        B.nv(ir)=nv(i+1);
        B.L0(ir)=find(pd1>pd0(end),1,'first');
        B.L1c(ir)=nv(i+1);
        B.L1(ir-1)=B.L0(ir)-1;
    end
end

```

UNCLASSIFIED

approved for public release

```

    % Interim region is flat at previous Pd (lower bound)
    k_int=B.L1c(ir-1):min([Lmax B.L1(ir-1)]);
    pd(k_int)=pd0(end)*ones(size(k_int));
    % Fill in up to end of new value of m
    k_new=B.L0(ir):min(Lmax,nv(i+1));
    pd(k_new)=pd1(k_new);
    pd0=pd1;
end;
end;
npd=length(pd);
if any(k_new),
    pd(k_new(end):npd)=pd1(k_new(end):npd);
end;
B.L1(ir)=Lmax;
k=1:length(pd);%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [m0,n0,T0,D0,I0]=mofn_opt_mn_T_and_D(p0,p1,log10T)
% [m0,n0,T0,D0,I0]=mofn_opt_mn_T_and_D(p0,p1,log10T)
% p0 = noise-only success probability
% p1 = noise+signal success probability
% log10T = log10(Tbar) FAR specification (can be a vector)
%
% m0 = optimal value of m
% n0 = optimal value of n
% T0 = achieved average time between false alarms
% D0 = achieved latency
% I0 = structure containing initialization

% Initial starting point
[ms0,ns0]=mofn_opt_mn_T_and_D_init(p0,p1,log10T);
I0.ms=ms0;
I0.ns=ns0;
for i=1:length(log10T),
    fprintf(' ');
    nsol=ms0(i)-ceil(log10T(i)/log10(p0))+5;
    [ms,ns,log10Ts]=mofn_far_spec(p0,log10T(i),nsol);
    Dmin=inf;
    isol=find(ms==ms0(i));
    Ds1=mofn_stats_cfcn(p1,ms(isol),ns(isol));
    I0.D(i)=Ds1;
    I0.T(i)=10^log10Ts(isol);
    % Back up until increasing
    Ds0=Ds1-1;
    i0=isol-1;

```

UNCLASSIFIED

approved for public release

```

Ds2=[];
while (i0>=1)&&(Ds0<Ds1),
    Ds0=mofn_stats_cfcn(p1,ms(i0),ns(i0));
    % Initialization beyond minimum, move down again
    if Ds0<Ds1,
        isol=i0;
        i0=i0-1;
        Ds2=Ds1;
        Ds1=Ds0;
    end;
end;
% Increase until at minimum if haven't already found it
if ~any(Ds2),
    Ds2=Ds1-1;
    while (Ds2<Ds1)&&(isol<nsol),
        % Evaluate D
        Ds2=mofn_stats_cfcn(p1,ms(isol+1),ns(isol+1));
        % Not at minimum yet, keep going
        if Ds2<Ds1,
            isol=isol+1;
            Ds1=Ds2;
        end;
    end;
end;
m0(i)=ms(isol);
n0(i)=ns(isol);
D0(i)=Ds1;
T0(i)=10.^log10Ts(isol);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [ms,ns]=mofn_opt_mn_T_and_D_init(p0,p1,log10T)
% [ms,ns]=mofn_opt_mn_T_and_D_init(p0,p1,log10T)
% Returns the starting point design (ms,ns) to minimize
% Dbar given p0, p1, and log10T

% Get efficiency from Page's test & Dbar
b1=1/(1+log(p1/p0)/log((1-p0)/(1-p1)));
[~,~,eta]=page_stats_bino(p0,p1,b1,1);
Dp=log10T/(eta*log10(exp(1)));
for i=1:length(log10T),
    % Get Pd breakpoints
    [B,~,~]=mofn_pd_breakpoint(p0,p1,log10T(i),5*ceil(Dp(i)));
    % Find m maximizing Pd for signal of length Dbar from Page's test
    iopt=find(B.L0<=Dp(i),1,'last');

```

UNCLASSIFIED

approved for public release


```

ms(i)=B.mv(iopt);
ns(i)=B.nv(iopt);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [T,D,eta]=page_stats_bino(p0,p1,b,h)
% [T,D,eta]=page_stats_bino(p0,p1,b,h)
% Returns performance measures for Page's test
% operating on binary data
% p0 = noise-only success probability
% p1 = noise+signal success probability
% b = bias
% h = threshold
TOL=1e-4;
% Get unity roots of MGF
t0=mgf_root_bino(p0,b,TOL);
t1=mgf_root_bino(p1,b,TOL);
% Approximations
ug0=p0-b;
ug1=p1-b;
T=nan*ones(size(h));
D=T;
ib=find(h>1-b);
T(ib)=(1+h(ib)*t0-exp(h(ib)*t0))./(t0*ug0);
D(ib)=(1+h(ib)*t1-exp(h(ib)*t1))./(t1*ug1);
eta=t0*ug1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function t=mgf_root_bino(p,b,TOL)
% t=mgf_root_bino(p,b,TOL)
% Returns the unity root of the moment generating function (MGF)
% for binary data with success probability p and bias b
if nargin<3,
    TOL=1e-6;
end;
if (p-b)<0,
    t=-log(p)/(1-b);
else
    t=log(1-p)/b;
end;
err=2*TOL;
while err>TOL,
    f=log(p*exp((1-b)*t)+(1-p)*exp(-b*t));
    dfdt=(p*(1-b)*exp(t*(1-b))-b*(1-p)*exp(-t*b))/(p*exp((1-b)*t)+(1-p)*exp(-b*t));
    t=t-f/dfdt;

```

UNCLASSIFIED

approved for public release

```
    err=abs(f/dfdt);  
end;
```

UNCLASSIFIED

approved for public release