INCORPORATION OF FALSE ALARMS IN SIMULATIONS OF ELECTRONIC RECEIVERS

V. P. Sobczynski
C. J. Pearson

SYCOM, Inc.

Abstract

A simple method for simulating false alarms in simulations of electronic receivers is developed. False alarms are an important part of electronic receivers and previously their generation was computationally complex because of their random nature and very small probability of occurrence. The approach derives the probability density function of the time between false alarms and shows how, for typical values of false alarm probability, the density function may be approximated by a discrete uniform density. Procedures are outlined for simulating false alarms both in the absence and presence of jamming. Also, a method is shown for representing the false alarm as having a finite width commensurate with the bandwidth of the receiver. A rapid computer algorithm for obtaining the amplitude of a false alarm is derived.

In digital time-ordered simulation of electronic receivers, it may be desirable to simulate detection threshold crossings due to false alarms randomly throughout the entire run. If a receiver has an ideal video bandwidth B, then the rate of occurrence of independent noise samples is 2B, corresponding to average time intervals of 1/(2B). An immediately obvious way to generate false alarms is to pick a random number from an appropriate distribution every 1/(2B) seconds. In practice, this can lead to an excessive number of computations and an unacceptably inefficient code. For example, if 2B = 20MHz, a draw must be made every

50 nanoseconds (simulation time). The purpose of this paper is to describe an alternate procedure which is generally much more efficient.

Time of Occurrence of False Alarm

The approach proposed here employs the probability density function of the time between false alarms. Starting at time zero, a draw is made from an appropriate density function to determine the time of the first false alarm. At the time determined in the first draw, a false alarm is generated and a draw is made. This procedure is continued through the entire run. The derivation of the procedure is as follows:

Let p = probability of false alarm during single time increment

and $$q = 1 - p. \tag{1}$$

Then if a trial is considered to have occurred at each time slot (every 50 nanoseconds if 2B = 20MHz), the probability of occurrence of a false alarm on the n^{th} trial is the following for several values of n:

$$\begin{aligned} pr\{n = 1\} &= p, \\ pr\{n = 2\} &= pq, \\ pr\{n = 3\} &= pq^2, \\ pr\{n = 4\} &= pq^3, \end{aligned} \tag{2}$$

and, in general,

$$pr\{n = N\} = pq^{N-1}. \tag{3}$$

That is, the probability that the next false alarm will be generated at the n^{th} time slot is pq^{n-1}. It may be easily checked that this is a probability density function by noting that

$$\sum_{i=0}^{\infty} pq^i = 1, \tag{4}$$

by the formula for the sum of an infinite geometric progression.

It can be shown that if one specifies a percentage of the total number of time slots, then the geometric distribution may be approximated by a uniform density from zero to approximately twice the average value. Since the density function actually extends over an infinite number of time slots, it is not possible to include all of them. If it is assumed that the last 10% of the density function may be disregarded, then the point in question becomes how many time slots to include to get 90% of all possible values of n. The formula for the sum of N terms of a geometric progression is:

$$S_n = a\left(\frac{r^n - 1}{r - 1}\right), \tag{5}$$

where a = first term, and r = ratio between successive terms. Substituting values, we find

$$\sum_{n=0}^{N} pq_n = p\left(\frac{q^N - 1}{q - 1}\right). \tag{6}$$

The average value $\overline{n}$ of the random number n may be obtained without difficulty as follows. Dividing the above by p yields:

$$\sum_{n=0}^{N} q^n = \frac{q^N - 1}{q - 1}. \tag{7}$$

Taking the limit as $N \rightarrow \infty$ gives

$$\sum_{n=0}^{\infty} q^n = \frac{1}{1 - q}, \tag{8}$$

which yields, upon differentiating with respect to q and multiplying by n and q,

$$\sum_{n=0}^{\infty} npq^n = \bar{n} = q/p \, . \qquad (9)$$

The value of N, which includes 90% of the range of n, may be found very easily. Noting that

$$p + q = 1, \qquad (10)$$

we obtain

$$\sum_{n=0}^{N_{90}} pq^n = .9 = 1 - q^{N_{90}} \, . \qquad (11)$$

This gives

$$.1 = q^{N_{90}} \, . \qquad (12)$$

Solving by logarithms yields

$$N_{90} = \frac{\ln(.1)}{\ln(q)} \, . \qquad (13)$$

In general

$$N_F = \frac{\ln(1 - F)}{\ln q} \, , \qquad (14)$$

where F = the fractional part of the density which is retained. We can now obtain the ratio $N_{90}/\bar{n}$:

$$\frac{N_{90}}{\bar{n}} = \frac{p}{q} \, \frac{\ln(.1)}{\ln(q)} = -2.3 \frac{p}{q} \, \frac{1}{\ln(1 - p)} \, . \qquad (15)$$

Using the Taylor series expansion for natural log and considering that p is small yields

$$\approx -2.3 \frac{p}{q} \, \frac{1}{(-p)} = \frac{2.3}{q} \approx 2.3 \, . \qquad (16)$$

Note that this result is independent of the value of probability of false alarm p so long as p is small.

The quantity $\bar{n}$ may be easily computed from p, the probability of false alarm. As an example, consider the following. Let the video bandwidth B = 10MHz and $P_{fa} = p = 10^{-6}$. Then the average time between false alarms will be

$$T_{fa} = \frac{1}{2BP_{fa}} = .05 \text{ sec} \, . \qquad (17)$$

The time per slot is equal to 1/(2B). The time to the next false alarm is equal to n x (time per slot), and

$$\bar{n} = N_{av} = \frac{T_{fa}}{\text{Time per Slot}}$$

$$= \frac{.05}{50 \times 10^{-9}} = 1 \times 10^6 \, . \qquad (18)$$

For these particular parameters, false alarms occur on the average every 10^6 time slots.

Clearly, 90% of all values of n is about twice the average value. The procedure can be very simple. To compute the time to a false alarm, pick a number out of a uniform distribution from zero to 2.3 times the average time between false alarms. The actual distribution is geometric. However, because p is small, the density function is very flat and 90% of all possible values are between zero and 2.3 times the average value. Therefore, it is possible to approximate the actual distribution by a uniform distribution from zero to 2.3 times the average value (for small values of probability of false alarm).

Alternately, a more accurate but less rapid method for generating deviates with a geometric distribution is given by Naylor, et al.[6] as follows:

$$g = (\ln u)/\ln q \, , \qquad (19)$$

where g and u are geometric and uniform deviates, respectively, and

$$q = 1 - p \qquad (20)$$

as before.

Threshold Detection

The method of setting the threshold in the absence of jamming noise will be described next. The receiver consists of hardware, active and passive devices such as filters, a preamplifier, IF amplifiers followed by a detector, and possibly an amplifier. The noise generated by the receiver (given by the receiver noise figure) can be modeled as though it were generated by a resistive element in the amplifier front end of the receiver. The overall noise voltage generated by the receiver may be considered to be Gaussian with negligible error if the gain of the preamplifier is sufficiently high (e.g., 20db or greater). The variance of this Gaussian density is

$$\sigma_T^2 = kT_eB_e \tag{21}$$

where k = Boltzmann's constant, T_e = effective noise temperature of the receiver, and B_e = effective two-sided video bandwidth. The effective noise temperature of the receiver is (F-1) 290°K, where F is the noise figure (not in db). Once the average thermal noise power is known, the threshold SNR can be determined from the required probability of detection, average time between false alarms and effective video bandwidth of the receiver. Then the threshold power can be determined by adding the required signal to noise ratio (db) to the average thermal noise power (dbm). Reference to the actual system configuration enables one to determine a threshold voltage V_T valid for thermal noise alone. V_T is an input constant for the entire run.

This voltage V_T corresponds to a threshold power P_T into the receiver. There exists an input voltage V_T' which corresponds to the output threshold voltage V_T. (V_T' is computed in the program and remains constant). Consider the Gaussian probability density of the amplitude of the noise generated by the receiver. V_T' is the value of the amplitude at the input that corresponds to the threshold voltage V_T. If P_{fa} is the predetermined probability of false alarm in presence of receiver noise only, then:

$$P_{fa} = 2\int_{V_T'}^{\infty} \frac{1}{\sqrt{2\pi\sigma_T^2}} \exp\left[-\frac{x^2}{2\sigma_T^2}\right] dx . \tag{22}$$

It is necessary to solve for V_T'. If one makes the substitution

$$t^2 = \frac{x^2}{2\sigma_T^2} , \tag{23}$$

then

$$dx = \sqrt{2}\ \sigma_T dt , \tag{24}$$

and

$$P_{fa} = \frac{2}{\sqrt{\pi}} \int_{\frac{V_T'}{\sqrt{2}\,\sigma_T}}^{\infty} e^{-t^2} dt , \tag{25}$$

or

$$P_{fa} = \operatorname{erfc}\left(\frac{V_T'}{\sqrt{2}\,\sigma_T}\right) \tag{26}$$

and

$$V_T' = \sqrt{2}\ \sigma_T \text{ erfcin } P_{fa} , \tag{27}$$

where erfc represents the familiar error function complement

$$\operatorname{erfc}(x) = \left(\frac{2}{\sqrt{\pi}}\right) \int_x^{\infty} e^{-t^2} dt \tag{28}$$

and erfcin is its inverse such that if

$$\mathrm{erfc}(x) = w , \tag{29}$$

then

$$\mathrm{erfcin}(w) = x . \tag{30}$$

This expression for V_T' is programmed to find V_T' from P_{fa} using a stored table of error function complement inverse, which will be discussed below.

False Alarm Probability with Varying External Noise Level

If there is jamming noise with power σ_j^2, the probability of false alarm p changes because the total noise power

$$\sigma^2 = \sigma_T^2 + \sigma_j^2 \tag{31}$$

is different. The new probability of false alarm is given by:

$$p_1 = 2 \int_{V_T'}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2}{2\sigma^2}\right] dx = \mathrm{erfc}\left(\frac{V_T'}{2\,\sigma}\right). \tag{32}$$

It is this quantity which must be used in the geometric distribution of the number of time slots to the next false alarm. Since p_1 may not be extremely small, the actual distribution rather than the approximation may be preferable.

Time of Occurrence of False Alarms with Noise Jamming

The procedure to be used for generating false alarms in the event of jamming is as follows. When the jamming signal is received, the false alarm which has been scheduled to occur is dropped. A new probability of false alarm (p_1) calculation is made. This procedure for calculating false alarms is continued for the duration of the jamming. When the jamming ends, any predicted false alarm due to jamming that has not yet occurred is dropped and the nonjamming value of p is used again. System operation then reverts to normal.

The procedure being described here normally lends itself very well to event-controlled as well as fixed time frame dynamic simulation designs. However, one must deal with the case in which changes in noise level occur between false alarms. It is suggested merely that a significant change in noise level be cause for the recomputation of the next false alarm time. Another uniform random draw may also be used to account for the fact that the previous false alarm probably did not occur at the time of the recomputation of the time interval. If changes in received noise level occur extremely rapidly or continuously in the simulation due to narrow moving beams or particular jamming techniques or noise sources such that recomputation would be too frequent, some major modification of this technique may be necessary.

Amplitude of False Alarms

The probability density function of the amplitude of the false alarm, given that a false alarm has occurred, is Gaussian, having a variance equal to the total input noise power and having a domain such that the allowed input power corresponds to a voltage amplitude output greater than V_T, where V_T if the threshold voltage. The threshold signal to noise ratio (SNR) is determined from P_{fa}, which is a simulation parameter, and the probability of detection by readily available graphs[1,2].

If it is assumed that all noise pulse amplitudes are normally distributed, then the distribution of false alarm amplitudes $f_y(y)$ will have the shape of the normal distribution in the region $|y| \geq V_T'$, where V_T' is again the input voltage amplitude corresponding to the threshold voltage V_T, and will vanish in the region $|y| < V_T'$:

$$f_y(y)dy = \begin{cases} \frac{a}{\sqrt{2\pi}\,\sigma} \exp[-y^2/(2\sigma^2)]dy, & |y| \geq V_T', \\ 0, & |y| < V_T', \end{cases} \quad (33)$$

where σ^2 is variance of the noise amplitude and a is a normalization factor such that

$$\int_{-\infty}^{\infty} f_y(y)dy = 1 \,. \quad (34)$$

It is readily discovered that

$$\frac{1}{a} = \text{erfc}\left(\frac{V_T'}{a\sqrt{\pi}}\right). \quad (35)$$

The generation of pseudorandom deviates with this distribution may be accomplished, in principle, merely by using any of the usual methods for generating Gaussian deviates[4] and rejecting those which do not exceed the detection threshold. For cases in which the ratio of noise level to detection threshold is even moderately large, however, this will result in the rejection of a very large fraction of the deviates which are generated and a correspondingly large expense in computation time. We therefore suggest that the inverse method[5] be used with tabulated functions. This method, to be described below, has the principal advantage that for every uniform deviate drawn an acceptable normal deviate may be computed by a fixed-length algorithm. If equal-interval function tables may be used the method is also quite rapid. Accuracy depends on the statistical accuracy of the uniform pseudorandom deviate generator and on the granularity and accuracy of the functional tables employed.

The inverse method of generating pseudorandom deviates y with the density function $f_y(y)dy$ consists essentially in the statement that if u is a uniform pseudorandom deviate on the interval [0, 1], then y will have the required density if y and u are related according to

$$u = F(y), \quad (36)$$

where F represents the cumulative probability distribution function of the variable y:

$$F(y) = \int_{-\infty}^{y} f_y(t)dt \,. \quad (37)$$

In other words, the technique consists of drawing u from the uniform density and then solving for y, the random pulse amplitude.

In particular, for the case of interest here,

$$F(y) = \begin{cases} \frac{a}{2}\,\text{erfc}\left(\frac{|y|}{\sigma\sqrt{2}}\right), & y \leq V_T', \\ 1/2, & -V_T' \leq y \leq V_T', \\ 1 - \frac{a}{2}\,\text{erfc}\left(\frac{y}{\sigma\sqrt{2}}\right), & y > V_T'. \end{cases} \quad (38)$$

This may be inverted, in principle, in the form:

$$y = \begin{cases} -\sqrt{2}\ \sigma\ \mathrm{erfcin}(2u/a), & 0 \leq u \leq 1/2, \\ \sqrt{2}\ \sigma\ \mathrm{erfcin}[2(1-u)/a], & 1/2 \leq u \leq 1, \end{cases} \quad (39)$$

where erfcin(x) again represents the inverse error function complement, a single valued, monotonic function for real arguments.

There remains the problem of developing a rapid computational algorithm for obtaining the error function complement (required for the computation of the normalization factor a) and its inverse. (Library routines are assumed to be available for generating the uniform pseudo-random deviates u.)

The methods generally used for computing the error function complement employ the Taylor series expansion or a rational polynomial approximation for this function in the central region and use the asymptotic form for arguments larger in magnitude than about 3. Such expansions have been obtained for the inverse function, or values of the inverse function may be obtained from the direct function by means of Newton's or some similar method of successive approximations. Since speed is presumably more significant than accuracy in simulating false alarms, it is suggested that the two required functions be tabulated and a table look-up routine used.

The value of a (note: $a = p_1$) must be recomputed only when the receiver noise level or the detection threshold changes (cf. Equation 32). This generally occurs somewhat less frequently than false alarms, which require the computation of the inverse function. Therefore, a table of erfcin(x) at equal intervals of x is recommended. This same table then may be used to find erfc(w) by means of a table look-up routine for unequal intervals. It is only necessary to interchange the roles of dependent and independent tabular variables. These table look-up techniques will not be discussed here.

Because detection threshold are usually well out in the tails of the noise pulse amplitude distribution, values of the inverse error function complement for arguments near 1.0 (i.e., in the central region) are not used. In particular, it may readily be shown that arguments greater than the receiver-noise-only false alarm probability P_{fa} are not required. It is then convenient to tabulate erfcin(x) versus ln x (or even $\log_{10} x$) to obtain the proper granularity with a table of equal intervals.

Table I is an example developed for just this purpose. To accomodate both central and asymptotic regions, Table I is divided into two overlapping portions, both tabulated at equal intervals of the independent variable. If $x \geq 0.34$ the central portion is used, while if $x < 0.34$ the asymptotic portion must be used. In simulating false alarms under conditions of varying input noise levels both erfcin and erfc functions are required. Nevertheless, in a large simulation where core locations are dear, careful programming of table look-up procedures makes it possible to avoid storing tabular values of the independent variable (x). Only three of the following four parameters are required to specify each part of the table of independent variables: first value, last value, interval size, number of intervals. At the same time, further computation of erfc functions can be avoided. Linear interpolation is normally entirely adequate for the purpose of generating these random numbers.

It may be of interest to consider the range of false alarm amplitudes which the table shown will allow. If the largest value of erfcin tabulated is $1/0.1898 \approx 5.27$, it follows that the largest possible pulse amplitude which will be generated is $(5.27)\sqrt{2}\ \sigma_T = 7.4\ \sigma_T$. (If the limiting values (0 or 1) of the uniform random number are drawn, they must be rejected, since pulse amplitudes of $\pm\infty$ would result.)

Effective Pulse Width of False Alarms

The next question that arises is, "What pulse width should be assigned to a false alarm?" A false alarm is generated by random thermal fluctuations in the front end of the receiver. Before it gets to the output amplifier, a false alarm of power A may be represented by an impulse function $\sqrt{A}\ \delta(t)$, then the output of the filter $Y(t)$ may be determined by the convolution integral:

$$Y(t) = \int_{-\infty}^{\infty} \sqrt{A}\ \delta(t - \tau) h(\tau)\, d\tau\ , \qquad (40)$$

where $h(t)$ = the time response of the linear filter. If the filter is a single pole low pass filter, then

$$y(t) = \frac{\sqrt{A}}{RC} \int_{-\infty}^{\infty} \delta(t - \tau)\, e^{-\tau/RC} d\tau \qquad (41)$$

$$= \frac{\sqrt{A}}{RC}\, e^{-t/RC} \qquad (42)$$

is the voltage output of the filter.

Because of the finite bandwidth of the receiver being simulated, it is desired to represent the false alarm as a rectangular pulse with an "effective" pulse width. The procedure used will be to calculate the energy in the exponential pulse which is the output of the filter and assume that the false alarm is a rectangular pulse of the same energy. The energy in the pulse $y(t)$ is

$$\int_0^{\infty} y^2(t)dt\ . \qquad (43)$$

The energy in the output pulse is then

$$\int_0^{\infty} y^2(t)dt = \frac{A}{(RC)^2} \int_0^{\infty} \exp[-2t/RC]dt \quad (44)$$

$$= \frac{A}{2RC}\ . \qquad (45)$$

An equal energy rectangular pulse of width T would have a voltage amplitude of $\sqrt{A/(2RCT)}$. If a time $T = RC$ is chosen, then the amplitude of the pulse is $(\sqrt{A/2})/(RC)$. The false alarm is then modeled as a pulse of amplitude $(\sqrt{A/2})/(RC)$ and width RC, where RC is equal to $1/(2\pi B_d)$, and B_d is the 3db bandwidth of the detector. If at the time of occurrence of a false alarm an input signal is present, then the amplitudes of the signal input and false alarm sum together.

Table I. Inverse Error Function Complement (erfcin)

ln x	1/erfcin(x)	ln x	1/erfcin(x)	ln x	1/erfcin(x)
-1.0	1.5705586	-12.5	0.30570521	-23.5	0.21629837
-1.5	1.1608596	-13.0	0.29907443	-24.0	0.21386504
-2.0	0.94698245	-13.5	0.29284743	-24.5	0.21151096
-2.5	0.81336734	-14.0	0.28698569	-25.0	0.20923195
-3.0	0.72087959	-14.5	0.28145508	-25.5	0.20702413
-3.5	0.65246551	-15.0	0.27622580	-26.0	0.20488390
-4.0	0.59945271	-15.5	0.27127180	-26.5	0.20280790
-4.5	0.55694114	-16.0	0.26657003	-27.0	0.20079300
-5.0	0.52194136	-16.5	0.26210000	-27.5	0.19883629
-5.5	0.49251962	-17.0	0.25784345	-28.0	0.19693503
-6.0	0.46736597	-17.5	0.25378410	-28.5	0.19508668
-6.5	0.44555928	-18.0	0.24990731	-29.0	0.19328883
-7.0	0.42643151	-18.5	0.24619993	-29.5	0.19153924
-7.5	0.40948547	-19.0	0.24265009	-30.0	0.18983581
-8.0	0.39434286	-19.5	0.23924703		
-8.5	0.38071033	-20.0	0.23598100		
-9.0	0.36835655	-20.5	0.23284312		
-9.5	0.35709646	-21.0	0.22982530		
-10.0	0.34678015	-21.5	0.22692014		
-10.5	0.33728465	-22.0	0.22412085		
-11.0	0.32850803	-22.5	0.22142120		
-11.5	0.32036523	-23.0	0.21991547		
-12.0	0.31278465				

REFERENCES

1) Skolnik, M. I., Introduction to Radar Systems, McGraw-Hill Book Co., New York, 1962.

2) D'Franco, J. V., and W. L. Rubin, Radar Detection, Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

3) Abramowitz, M. and I. A. Stegun (Editors), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards, Washington, D.C., 1964, Sec. 7.1.2.

4) Ibid., pp. 952 et seq.

5) Ibid, p. 950.

6) Naylor, Thomas H., Joseph L. Balintfy, Donald S. Burdick, and Kong Chu, Computer Simulation Techniques, John Wiley and Sons, Inc., New York, 1968.