Alternative Forms of Albersheim’s Equation

Mark A. Richards
June 2014

1 Albersheim’s Equation

In radar detection theory, the detection probability \( P_D \), false alarm probability \( P_{FA} \), number of samples \( N \) noncoherently integrated for a single detection test, and single-sample average signal-to-noise ratio (SNR) \( \chi \) are closely interrelated. A common problem is to find one of these quantities given the other three. This can be a difficult calculation. For example, in the nonfluctuating case with a linear detector, the target+noise probability density function (PDF) is Rician, a distribution which involves Bessel functions and for which the \( P_D \) calculation requires “Marcum’s Q function” [1]. While these equations can be solved using modern computational tools such as MATLAB®, an excellent empirical approximation requiring only a scientific calculator known as Albersheim’s equation is available for this case. Albersheim’s equation relates \( P_D \), \( P_{FA} \), \( N \), and the single-sample SNR in decibels \( \chi_{dB} \). As usually presented, Albersheim’s “equation” is the series of calculations [2],[3]

\[
A = \ln \left( \frac{0.62}{P_{FA}} \right) \quad B = \ln \left( \frac{P_D}{1 - P_D} \right)
\]

\[
\chi_{dB} = -5 \log_{10} N + 6.2 + 4.54 \log_{10} \left( A + 0.12AB + 1.7B \right)
\]

(1)

The error in the estimated value of \( \chi_{dB} \) required to obtain specified values of \( P_D \) and \( P_{FA} \) for a specified \( N \) is stated to be less than 0.2 dB for \( P_{FA} \) from \( 10^{-3} \) to \( 10^{-7} \), \( P_D \) from 0.1 to 0.9, and \( N \) from 1 to 8096. Albersheim’s equation is very useful not only for direct calculations of required SNR but also for such tasks as estimating noncoherent integration gain for the nonfluctuating case [4].

2 Solving Albersheim’s Equation for \( P_D \) or \( P_{FA} \)

As stated, Albersheim’s equation determines the average single-sample SNR corresponding to a specified \( P_D, P_{FA}, \) and \( N \). However, it can be rearranged to obtain a solution for \( P_D \) in terms of the other parameters using the following sequence of calculations:

---

1 “Shnidman’s equation” plays the same role for the Swerling target models as does Albersheim’s equation for the nonfluctuating case, but is somewhat more complicated; see [1].
M. A. Richards, “Alternative Forms of Albersheim’s Equation”

June 20, 2014

\[
A = \ln\left(\frac{0.62}{P_{FA}}\right), \quad Z = \frac{X_{dB} + 5 \log_{10} N}{6.2 + \frac{4.54}{\sqrt{N+0.44}}}, \quad B = \frac{10^Z - A}{1.7 + 0.12 A}
\]

(2)

\[
P_{D} = \frac{1}{1 + e^{-B}}
\]

A similar rearrangement gives \( P_{FA} \) in terms of the other parameters:

\[
B = \ln\left(\frac{P_{D}}{1 - P_{D}}\right), \quad Z = \frac{X_{dB} + 5 \log_{10} N}{6.2 + \frac{4.54}{\sqrt{N+0.44}}}, \quad A = \frac{10^Z - 1.7 B}{1 + 0.12 B}
\]

(3)

\[
P_{FA} = 0.62 e^{-A}
\]

Unfortunately, solving the standard form of Albersheim’s equation for \( N \) in terms of \( P_{D}, P_{FA} \) and \( X_{dB} \) does not appear to be possible because \( N \) appears in both logarithm and square root form.

## 3 Solving Albersheim’s Equation for \( N \)

A method for solving Albersheim’s equation for \( N \) can be developed by replacing the term \( 6.2 + 4.54/\sqrt{N+0.44} \) with an approximation that uses \( \log_{10} N \), so that \( N \) appears in only one functional form and can be isolated. Define \( \alpha = \log_{10} N \). Then it can be seen that

\[
6.2 + \frac{4.54}{\sqrt{N+0.44}} \approx -\frac{0.4125}{\alpha^2} + \frac{2.4194}{\alpha} + 5.5606 \quad (2 \leq N \leq 100)
\]

(4)

This second-order least-squares approximation was found using the `polyfit` function in MATLAB®. Slightly different approximation coefficients result for different ranges of \( N \). For example, using the entire range \( 2 \leq N \leq 8096 \) over which Albersheim’s equation is applied, the coefficients of \( \alpha \) are \((-0.2289, 1.8402, 5.7777)\). Here we concentrate on the range \( 2 \leq N \leq 100 \), believing that is a range of more practical interest. Figure 1 illustrates the close fit of the approximation to the Albersheim equation term in Eq. (4).

Using Eq. (4) in Eq. (1) and defining \( Z = \log_{10} \left( A + 0.12 AB + 1.7 B \right) \) gives

\[
X_{dB} = -5\alpha + Z\left( -\frac{0.4125}{\alpha^2} + \frac{2.4194}{\alpha} + 5.5606 \right)
\]

(5)

This equation can be rearranged to give a cubic equation in \( \alpha \):

\[
5\alpha^3 + \left( X_{dB} - 5.5606 \cdot Z \right) \alpha^2 - 2.4194 \cdot Z \alpha + 0.4125 \cdot Z = 0
\]

(6)

---

2 Since the expansion is in terms of \( 1/\log_{10} N \), the \( N = 1 \) term must be excluded from the least-squares fit.

3 This is the same \( Z \) as in Eq. (2) or (3).
Specifying the values of $P_D$, $P_{FA}$, and $\chi_{\text{dB}}$ fixes the value of $Z$. Equation (6) can then be solved for $\alpha$. Since Eq. (6) is cubic, it is not trivially solvable with a calculator for the value of $\alpha$ and thus of $\hat{N}$. However, it can be solved numerically (for instance, using the roots function in MATLAB®). The largest real root is chosen as the desired value of $\alpha$. This choice is entirely ad hoc but appears to produce good results. $N$ must be a positive integer, therefore the final estimate is

$$\hat{N} = \text{round} \left( \max \left( 10^\alpha, 1 \right) \right) \quad (7)$$

The estimate $\hat{N}$ given in Eq. (7) can be tested by varying values of $P_D$, $P_{FA}$, and $N$ over a wide range within the applicability of Albersheim’s equation. For each choice of these three parameters, $\chi_{\text{dB}}$ is computed using Albersheim’s equation. Using $P_D$, $P_{FA}$, and that computed value of $\chi_{\text{dB}}$, Eqs. (6) and (7) are then used to compute $\hat{N}$ and compare it to the true value of $N$ for that case. Figure 2, for example, shows that when $P_D = 0.9$ and $P_{FA} = 10^{-3}$, the estimated value of $N$ is correct for $3 \leq N \leq 84$. The error is $-1$ for $N = 2$ (i.e., $\hat{N} = 1$ in this case) and $+1$ for $85 \leq N \leq 100$. In fact, the magnitude of the error does not exceed 1 until $N > 560$. The maximum percentage error in $\hat{N}$ for this case never exceeds 4.8% for $N$ up to the limit of 8096, even though the approximation coefficients for only the range $2 \leq N \leq 100$ are used in this sample.
Figure 2. Error in estimate of $N$ when $P_D = 0.9$ and $P_{FA} = 10^{-3}$.

For a more complete test, this procedure was carried out as the value of $N$ was varied from 1 to 100; $P_D$ was varied from 0.1 to 0.9 in steps of 0.01; and $P_{FA}$ was varied from $10^{-7}$ to $10^{-3}$ in 100 logarithmically-spaced steps. Over this range, the absolute value of the error in $\hat{N}$ never exceeds 1. For $N > 2$, the percentage error in $\hat{N}$ never exceeds 1.266%, which occurs when $P_D = 0.88$, $P_{FA} = 10^{-7}$, and $N = 79$. (For $N = 2$ the estimated value is often $\hat{N} = 1$, giving a 50% error.) Figure 3 shows the variation of the absolute error in $\hat{N}$ versus $P_D$ and $P_{FA}$ for $N = 2$ and $N = 90$. For all of the similar plots for $3 \leq N \leq 78$, the error is zero for the full range of $P_D$ and $P_{FA}$ tested.

Figure 3. Absolute value of error in estimate of $N$ vs. $P_D$ and $P_{FA}$: (a) $N = 2$, (b) $N = 90$. The error is zero for all values of $P_D$ and $P_{FA}$ considered and $N$ between 3 and 78.
4 The Largest Root

Equation (6) has three roots, possibly leading to three different estimates $\hat{N}$ in Eq. (7). Which should be used? In general, a cubic equation with real coefficients such as Eq. (6) has either three real roots, or one real root and a pair of complex conjugate roots. In the discussion so far, the largest real root has been used in Eq. (7) to compute the estimate $\hat{N}$. This choice was *ad hoc* but produces good results.

A numerical search across the same dense grid of $P_D$, $P_{FA}$, and $N$ values used to obtain Fig. 3 showed that in practice, both cases occur: three real roots, and one real root plus two complex roots. For values of $N$ of 3 or more, complex roots appear to occur only at the corner case of $P_D = 0.1$ and $P_{FA} = 10^{-3}$. For $N = 2$ the complex case occurs more frequently.

Explicit solutions for the roots of a cubic equation are given in [5]. These formulas could be used to compute the roots of Eq. (6) without the use of a numerical solver such as MATLAB®’s `roots`, and then the maximum real root used to compute $\hat{N}$. Such a solution could be implemented, if a bit tediously, on a programmable calculator. If it was possible to predict which of the three roots would be the largest, then the formula for that root would complete a straightforward, deterministic algorithm for estimating $N$. It is not, unfortunately, obvious how to identify *a priori* which of the three roots will be the largest positive root. This last step remains a topic for additional research. Lacking that result, either a root-finding algorithm such as `roots` or explicit calculation of all three roots using the closed-form equations in [5] must be used.

5 References