Signal-to-Noise Ratio and Combiner Weight Estimation for Symbol Stream Combining

Q. D. Vo
Communications Systems Research Section

A method is presented for signal-to-noise ratio (SNR) and symbol stream combiner weight estimation. The SNR estimator employs absolute value moments as in an earlier method. The main contribution of this work is that a new algorithm is derived for the combiner weight estimator to remove the large bias at low SNRs. The new algorithm is simulated to combine two independent symbol streams at various SNRs. As an example, in combining two symbol streams at SNRs of -1 dB and -7 dB, combiner weight estimates using 1000 samples for the -1 dB stream and 10,000 samples for the -7 dB stream achieve an output SNR of -0.039 dB, which is just 0.012 dB below the theoretical limit achievable with perfect knowledge of the SNRs.

I. Introduction

Signals received from interplanetary telecommunications are usually very weak. An improvement in signal-to-noise ratio (SNR) can be obtained by properly combining the outputs from several receiving stations. The optimal combining strategy depends on the SNR levels of the received signals. In this report, we consider a technique of SNR and combiner weight estimation for possible use in symbol stream combining. This is as opposed to baseband combining wherein combining is done prior to subcarrier demodulation and symbol synchronization. The SNR estimators, both biased and unbiased, are considered by Layland (Ref. 1). We follow the same procedure to define the biased and unbiased combiner weight estimators. The SNR loss of the combined output stream is analyzed and compared for both biased and unbiased combiner weight estimates. The combining process has negligible loss using either method for two symbol streams with the same SNRs. However, with the biased method, the SNR loss grows significantly as the difference in the two SNRs gets larger due to large bias introduced by the use of absolute moment as an approximation to the mean at low SNRs. For the unbiased method, simulation results show that the bias is practically removed with a sufficiently large number of samples. For example, with 1000 samples for the -1 dB stream and 10,000 samples for the -7 dB stream, the obtained output SNR is just 0.012 dB less than the maximum achievable limit (-0.027 dB) which assumes perfect knowledge of the input SNRs.

II. Background

As shown in Fig. 1, a spacecraft signal is received simultaneously at two tracking stations. This signal is demodulated and integrated to produce two sequences of symbols, $\{x_k\}$ and $\{y_k\}$. Due to differences in receiving systems, the two symbol streams have different levels of SNR. The symbol stream combiner is designed to maximize the overall SNR. Let

$$E(x_k) = \pm m_1, \quad \mathrm{var}(x_k) = \sigma_1^2$$

and

$$E(y_k) = \pm m_2, \quad \mathrm{var}(y_k) = \sigma_2^2$$

Then

$$\mathrm{SNR}_{x_k} = \frac{m_1^2}{2\sigma_1^2}, \quad \mathrm{SNR}_{y_k} = \frac{m_2^2}{2\sigma_2^2}$$

(The factor of two in the denominator is used so that the definition of SNR is the same as symbol energy to noise spectral density.) The combining strategy is

$$z_k = \alpha_1 x_k + \alpha_2 y_k$$

where α_1 and α_2 are chosen to maximize

$$\mathrm{SNR}_{z_k} = \frac{(\alpha_1 m_1 + \alpha_2 m_2)^2}{2\alpha_1^2\sigma_1^2 + 2\alpha_2^2\sigma_2^2} \leqslant \mathrm{SNR}_{x_k} + \mathrm{SNR}_{y_k}$$

Taking derivatives with respect to α_1, α_2 and setting them to zero, it is found that the output SNR is maximized when

$$\frac{\alpha_1}{\alpha_2} = \frac{m_1}{\sigma_1^2}\frac{\sigma_2^2}{m_2} = \frac{\sqrt{\mathrm{SNR}_{x_k}}}{\sqrt{\mathrm{SNR}_{y_k}}} \times \frac{\sigma_2}{\sigma_1}$$

Since m_k and σ_k^2 ($k = 1,2$) are not known we must have some way to estimate them. The conventional method which deals with absolute moment is considered first, and then improvement by bias removal is investigated.

III. Conventional SNR Estimator

Let the input signal $x(t)$ be either $+V$ or $-V$ in the intervals t_k to t_{k+1}, $k = 1,2,\ldots$. This signal is corrupted by additive white Gaussian noise $n(t)$ having zero mean and two-sided spectral density $N_0/2$. Then received signal is integrated over the symbol time to produce the symbol stream $\{x_k\}$,

$$x_k = \pm VT + n_k$$

where n_k is a zero-mean Gaussian random variable with variance $N_0T/2$. Thus,

$$m = E(x_k) = \pm VT, \quad \sigma^2 = \mathrm{Var}(x_k) = \frac{N_0 T}{2}$$

and

$$R = \mathrm{SNR}_{x_k} = \frac{m^2}{2\sigma^2} = \frac{V^2 T}{N_0}$$

The conventional SNR estimator uses the sample mean and variance in conjunction with absolute moment to estimate m and σ^2

$$\widetilde{m} = \frac{1}{N}\sum_{k=1}^{N} |x_k|$$

$$\widetilde{\sigma}^2 = \frac{1}{N-1}\sum_{k=1}^{N}(|x_k| - \widetilde{m})^2 = \frac{N}{N-1}\left[\frac{1}{N}\sum_{k=1}^{N} x_k^2 - \widetilde{m}^2\right]$$

So the estimate $\widetilde{R}$ for the SNR is

$$\widetilde{R} = \frac{\left(\frac{1}{N}\sum_{k=1}^{N}|x_k|\right)^2}{\frac{2}{N-1}\sum_{k=1}^{N}\left(|x_k| - \frac{1}{N}\sum_{i=1}^{N}|x_i|\right)^2}$$

The performance of this estimator was analyzed by Layland (Ref. 1), and then by Lesh (Ref. 2). The mean of the estimator is found to be dependent heavily on the input noise but relatively insensitive to changes in the sample size. For large enough N (N>500) we have

$$E(\widetilde{R}) \simeq \frac{[E(|x_k|)]^2}{2\,\mathrm{Var}(|x_k|)} = f(R)$$

where

$$E(|x_k|) = \sqrt{N_0 T}\left[\frac{\exp^{-R}}{\sqrt{\pi}} + \sqrt{R}\,\mathrm{erf}(\sqrt{R})\right]$$

$$\mathrm{Var}(|x_k|) = E(x_k^2) - E(|x_k|)^2$$

$$E(x_k^2) = N_0 T(R + \tfrac{1}{2})$$

and

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp^{-t^2} dt$$

The estimator mean is plotted in Fig. 2 over the interested range of input SNRs. We see that there is a significant bias at low SNRs due to the nonlinearity of the absolute moment. This problem can be solved by defining another estimator (Ref. 1) $\hat{R} = f^{-1}(\tilde{R})$; then for large N we have

$$E(\hat{R}) \simeq R$$

and

$$\operatorname{Var}(\hat{R}) \simeq \operatorname{Var}(\tilde{R}) \cdot \left[\left. \frac{df(x)}{dx} \right|_{x=R} \right]^{-2}$$

In real time implementation, the inverse of the function f could be evaluated using table lookup.

The variance of $\tilde{R}$ is found to be strongly dependent on the sample size and relatively insensitive with respect to noise. A computer program is written to evaluate this variance based on equations provided by Lesh (Ref. 2). Figure 3 shows the dependence of the standard deviation on the sample size for $R = -1$ dB and $R = -7$ dB. To be consistent with the previous paper (Ref. 2), the standard deviation in dB is defined as

$$\sigma_{\mathrm{dB}}(\tilde{R}) = 10 \log_{10} \left(1 + \frac{\sigma(\tilde{R})}{R}\right)$$

In Fig. 4 we plot the number of symbols (i.e., sample size) required to achieve a standard deviation of 0.1 dB versus input SNRs.

IV. Implementation for the Symbol Stream Combiner

The combining strategy is

$$z_k = \alpha_1 x_k + \alpha_2 y_k$$

Before we consider the algorithms for estimating α's, we would like to see how sensitive the output SNR is with respect to these coefficients. Since the optimum weights result in an SNR equal to the sum of the input SNRs, the SNR loss is

$$\text{SNR loss (dB)} = 10 \log_{10}(R_1 + R_2) - 10 \log_{10} R$$

where

$$R_1 = \mathrm{SNR}_{x_k}$$

$$R_2 = \mathrm{SNR}_{y_k}$$

$$R = \mathrm{SNR}_{z_k}$$

Then

$$E(z_k) = \alpha_1 E(x_k) + \alpha_2 E(y_k) = \alpha_1 m_1 + \alpha_2 m_2$$

$$\operatorname{Var}(z_k) = \alpha_1^2 \operatorname{Var}(x_k) + \alpha_2^2 \operatorname{Var}(y_k) = \alpha_1^2 \sigma_1^2 + \alpha_2^2 \sigma_2^2$$

So the output SNR will be

$$R = \mathrm{SNR}_{z_k} = \frac{(\alpha_1 m_1 + \alpha_2 m_2)^2}{2\left[\alpha_1^2 \sigma_1^2 + \alpha_2^2 \sigma_2^2\right]}$$

Figure 5 shows the SNR loss as a function of R_2 and α_2 for $R_1 = -1$ dB and $\alpha_1 = \sqrt{2R_1}$ (we consider a normalized case where $\sigma_1^2 = \sigma_2^2 = 1$). We see that highly accurate estimates of α's (hence of input SNRs) are unnecessary since the SNR loss does not depend very critically on these coefficients around their optimal values. (For example with a change of 0.1 from the optimal value of α_2, the SNR loss increases by less than 0.02 dB). However, if large biases are present, the SNR loss grows very fast and becomes very sensitive to changes in α's. So we need to derive an algorithm with negligible bias. We first consider the biased combining method and then try to remove the bias with appropriate adjustment.

A. Biased Combining Algorithm

The biased estimates of α's are just the sample means divided by the sample variances:

$$\tilde{\alpha}_1 = \frac{\frac{1}{N}\sum_{i=1}^{N} |x_i|}{\frac{N}{N-1}\left[\frac{1}{N}\sum_{i=1}^{N} x_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} |x_i|\right)^2\right]}$$

$$\tilde{\alpha}_2 = \frac{\frac{1}{N}\sum_{i=1}^{N} |y_i|}{\frac{N}{N-1}\left[\frac{1}{N}\sum_{i=1}^{N} y_i^2 - \left(\frac{1}{N}\sum_{i=1}^{N} |y_i|\right)^2\right]}$$

Since the optimal strategy just depends on the ratio between α_1 and α_2, for simple implementation we could define

$$\widetilde{\alpha}_1 = \frac{\sum_{i=1}^{N} |x_i|}{\sum_{i=1}^{N} x_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} |x_i|\right)^2}$$

$$\widetilde{\alpha}_2 = \frac{\sum_{i=1}^{N} |y_i|}{\sum_{i=1}^{N} y_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} |y_i|\right)^2}$$

The combined output stream is

$$z_k = \widetilde{\alpha}_1 x_k + \widetilde{\alpha}_2 y_k$$

The SNR loss can be evaluated by computing the output SNR:

$$E(z_k) = E(\widetilde{\alpha}_1)m_1 + E(\widetilde{\alpha}_2)m_2$$

$$\begin{aligned}\mathrm{Var}(z_k) &= E(z_k^2) - E(z_k)^2 \\ &= E\left[(\widetilde{\alpha}_1 x + \widetilde{\alpha}_2 y)^2\right] - \left(E(\widetilde{\alpha}_1)m_1 + E(\widetilde{\alpha}_2)m_2\right)^2 \\ &= E(\widetilde{\alpha}_1^2)E(x^2) + E(\widetilde{\alpha}_2^2)E(y^2) \\ &\quad - \left(E(\widetilde{\alpha}_1)m_1\right)^2 - \left(E(\widetilde{\alpha}_2)m_2\right)^2\end{aligned}$$

From Appendix A, we can find

$$C_1 = E(\widetilde{\alpha}_1)m_1,\ C_2 = E(\widetilde{\alpha}_2)m_2$$

$$D_1 = E(\widetilde{\alpha}_1^2)E(x^2),\ D_2 = E(\widetilde{\alpha}_2^2)E(y^2)$$

Then

$$R = \mathrm{SNR}_{z_k} = \frac{[E(z_k)]^2}{2\,\mathrm{VAR}(z_k)} = \frac{(C_1 + C_2)^2}{2[D_1 + D_2 - C_1^2 - C_2^2]}$$

which is a function of R_1, R_2 and N_1, N_2 where N_1, N_2 denote the number of samples of the sequences $\{x_k\}$ and $\{y_k\}$ respectively.

Figure 6 shows the SNR loss as a function of R_1 and N_1 for $R_2 = -7$ dB and for $N_2 = 1000$ and 10,000. Figure 7 plots the SNR loss versus R_2 and N_2 for $R_1 = -1$ dB and for $N_1 = 1000$ and 10,000. We see that the combining loss is negligible for two symbol streams with same SNRs. The SNR loss grows significantly as the difference in the two SNRs gets larger. This is due to the large bias at low SNRs introduced by the use of absolute moment as an approximation to the mean.

B. Unbiased Combining Algorithm

Since we will often be dealing with low and unequal SNRs (from -12 dB to -1 dB), we must find some way to remove the combiner bias. For convenience let us define

$$a_1 = \sum_{i=1}^{N_1} |x_i|$$

$$s_1 = \sum_{i=1}^{N_1} x_i^2$$

Let $\widetilde{\alpha}_1$ be the biased estimate of α_1 and $\widetilde{R}_1$ be the biased estimate of R_1, then

$$\widetilde{\alpha}_1 = (N_1 - 1)\frac{a_1}{N_1 s_1 - a_1^2}$$

$$\widetilde{R}_1 = \frac{\widetilde{\alpha}_1 a_1}{2N_1}$$

From Appendix A, for large N we have

$$E[\widetilde{\alpha}_1] = \frac{\mu}{\sigma_v^2}(1 + B) \simeq \frac{\mu}{\sigma_v^2}$$

$$= \frac{m_1}{2\sigma_1^2}\,\frac{\dfrac{\exp -R_1}{\sqrt{\pi R_1}} + \mathrm{erf}(\sqrt{R_1})}{\left\{R_1 + \dfrac{1}{2} - \left[\dfrac{\exp -R_1}{\sqrt{\pi}} + \sqrt{R_1}\,\mathrm{erf}(\sqrt{R_1})\right]^2\right\}}$$

But $\alpha_1 = m_1/\sigma_1^2$, so

$$\alpha_1 = \frac{2\left[R_1 + \frac{1}{2} - \left(\frac{\exp - R_1}{\sqrt{\pi}} + \sqrt{R_1}\ \mathrm{erf}\,(\sqrt{R_1})\right)^2\right]}{\frac{\exp - R_1}{\sqrt{\pi R_1}} + \mathrm{erf}\,(\sqrt{R_1})} E(\tilde{\alpha}_1)$$

$$= g(R_1)\, E(\tilde{\alpha}_1) \qquad (1)$$

Since we don't know R_1, we also have to estimate it. From Section III,

$$E(\tilde{R}_1) = f(R_1)$$

$$= \frac{\left[\frac{\exp - R_1}{\sqrt{\pi}} + \sqrt{R_1}\ \mathrm{erf}\ \left(\sqrt{R_1}\right)\right]^2}{2\left[R_1 + \frac{1}{2} - \left(\frac{\exp - R_1}{\sqrt{\pi}} + \sqrt{R_1}\ \mathrm{erf}\,(\sqrt{R_1})\right)^2\right]}$$

Layland (Ref. 1) defines the asymtotically unbiased estimate of R_1 to be

$$\hat{R}_1 = f^{-1}(\tilde{R}_1)$$

Using $\hat{R}_1$ in place of R_1 in Eq. (1), we define the asymtotically unbiased estimate of α_1 to be

$$\hat{\alpha}_1 = g(\hat{R}_1)\,\tilde{\alpha}_1 = gof^{-1}(\tilde{R}_1)\,\tilde{\alpha}_1 = h(\tilde{R}_1)\,\tilde{\alpha}_1$$

The function $h(\tilde{R})$ is shown in Fig. 8 for $\tilde{R}$ from 0.8796 to 1.3893 (for R from -13 dB to 0 dB). Using second-order polynominal interpolation, this function can be approximated with an error less than 0.002 by

for $0.8796 \leqslant \tilde{R} \leqslant 0.9000$

$$h(\tilde{R}) = -121.49\tilde{R}^2 + 221.5\tilde{R} - 100.6772$$

for $0.9000 < \tilde{R} \leqslant 0.9600$

$$h(\tilde{R}) = -14.6\tilde{R}^2 + 29.5263\tilde{R} - 14.4791$$

for $0.9600 < \tilde{R} \leqslant 1.0500$

$$h(\tilde{R}) = -3.5596\tilde{R}^2 + 8.4951\tilde{R} - 4.4639$$

for $1.0500 < \tilde{R} \leqslant 1.3893$

$$h(\tilde{R}) = -1.0004\tilde{R}^2 + 3.0811\tilde{R} - 1.5990$$

Note that, for $\tilde{R} \leqslant 0.9000$, the polynominal coefficients are quite large compared to the function values. So the function values are *very* sensitive to small changes in coefficients, and one must be careful in using this approximation in finite arithmetic applications.

Since it is extremely complicated to analyze the performance of this estimator, we simulate the algorithm (see Appendix B) to find the SNR loss for different R_1, R_2 and N_1, N_2. Figure 9 shows the SNR loss as a function of R_1 and N_1 for $R_2 = -7$ dB and $\log_{10} N_2 = 3{,}4$. Figure 10 plots the SNR loss versus R_2 and N_2 for $R_1 = -1$ dB and $\log_{10} N_1 = 3.4$. We see that as N_1, N_2 get larger, the SNR loss goes to zero. In Fig. 9, the $R_1 = -1$ dB curves cross because with a very small number of samples ($N_1 = 100$) the simulation results are not very reliable, especially for low SNR sequences. The 100-sample block may fall into either a very low noise section or a very noisy one. The same thing happens for the $R_2 = -3$ dB curves in Fig. 10. The combining loss goes down to approximately 0.012 dB with $N_1 = 1000$ for the -1 dB stream and $N_2 = 10{,}000$ for the -7 dB stream. In general larger numbers of samples are required for lower SNR sequences.

V. Conclusion

The conventional SNR and combiner weight estimator has large bias at low SNRs. As a result, the output SNR loss grows significantly as the difference in SNR of the two symbol streams gets larger. When the algorithm is modified to remove the bias, the bias is essentially removed with a sufficiently large number of samples. Sample sizes of 1000 for the -1 dB stream and 10,000 for the -7 dB stream are sufficient to achieve an output SNR which is just 0.012 dB less than the maximum achievable limit.

References

1. Layland, J. W., "On S/N Estimation," *Space Programs Summary 37-48*, Vol. III, pp. 209–212, Jet Propulsion Laboratory, Pasadena, Calif., Dec. 31, 1967.

2. Lesh, J. R., "Accuracy of the Signal to Noise Ratio Estimator," *Technical Report 32-1526*, Vol. X, pp. 217–235, Jet Propulsion Laboratory, Pasadena, Calif., Aug. 15, 1972.

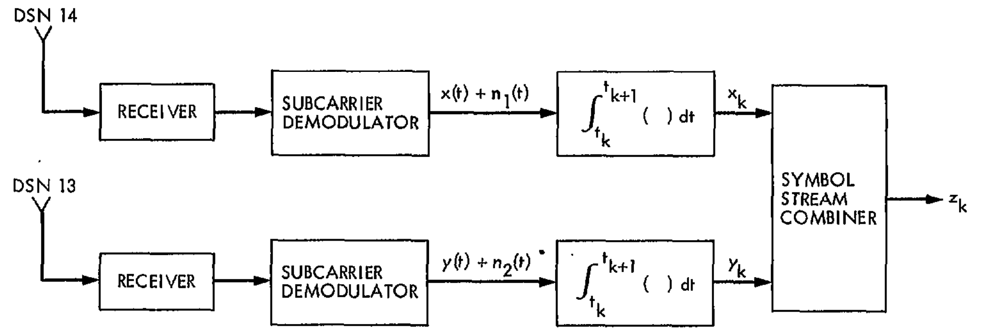

Fig. 1. Model of the symbol stream combining process

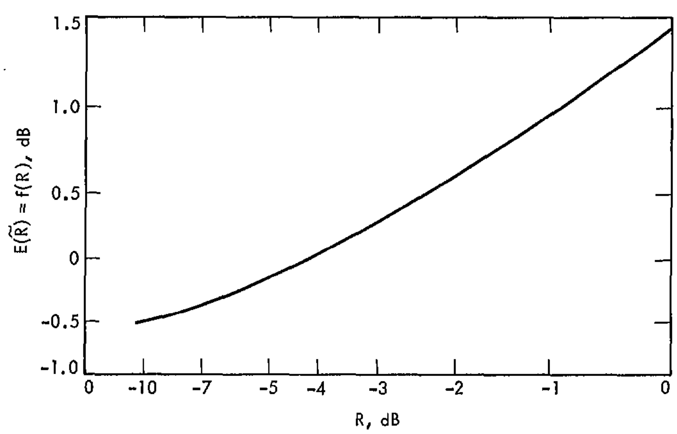

Fig. 2. Biased SNR estimator mean vs actual SNR

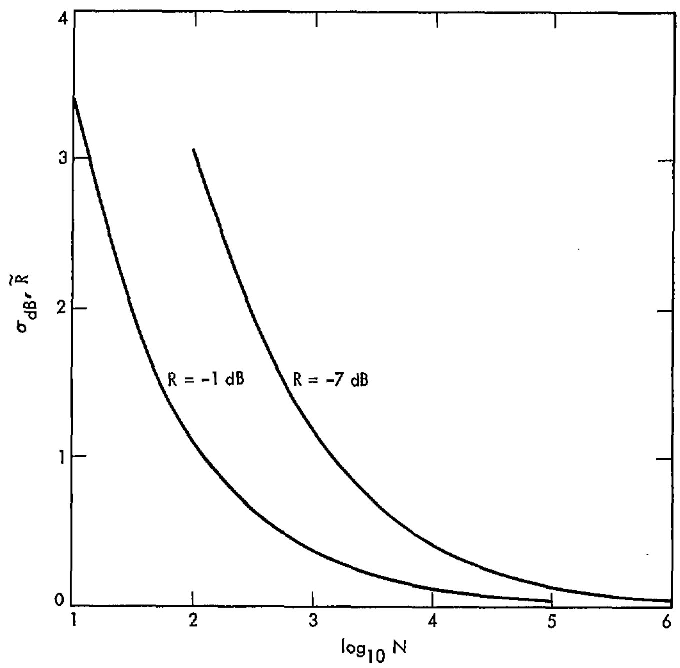

Fig. 3. Dependence of standard deviation on sample size

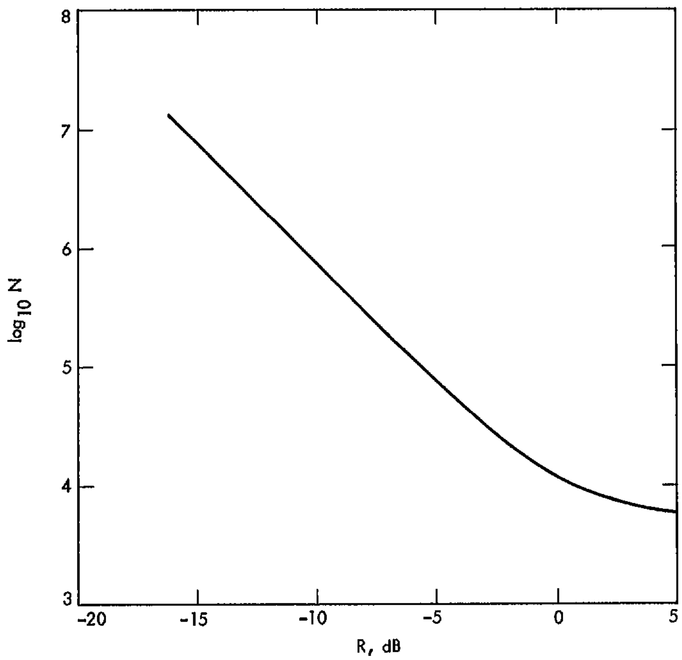

Fig. 4. Number of symbols required to achieve $\sigma dB(\hat{R}) = 0.1$ dB

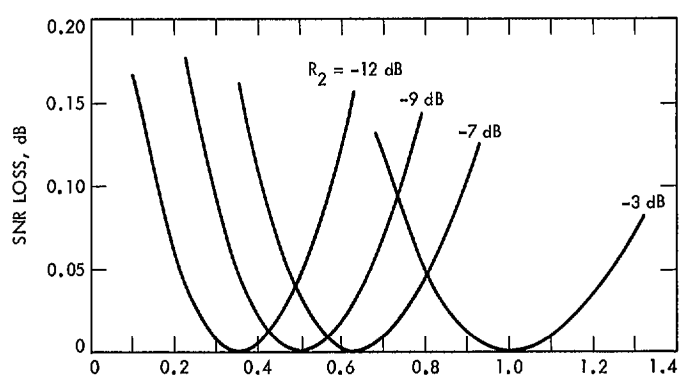

Fig. 5. SNR loss vs R_2 and α_2 for R_1 = −1 dB

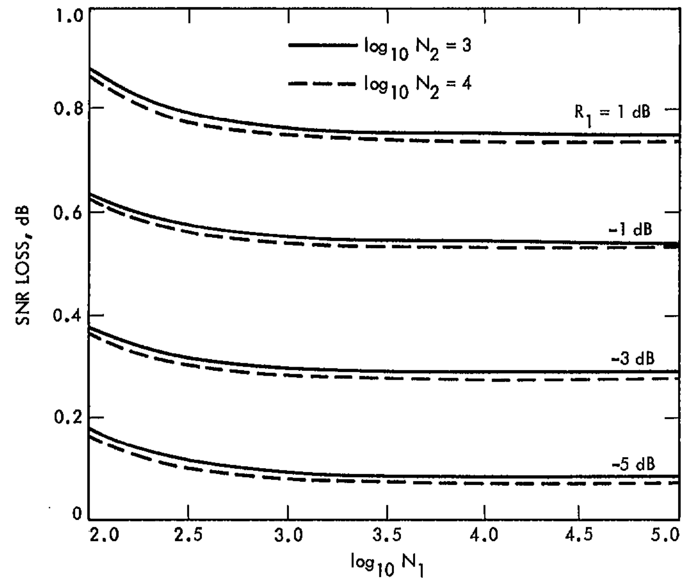

Fig. 6. SNR loss vs R_1 and N_1 for R_2 = −7 dB (conventional method, without bias removed)

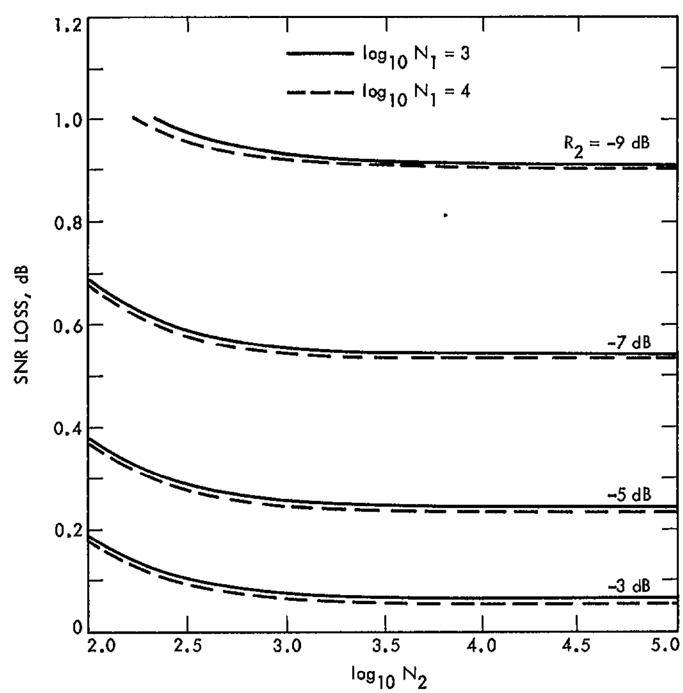

Fig. 7. SNR loss vs R_2 and N_2 for R_1 = −1 dB (conventional method, without bias removed)

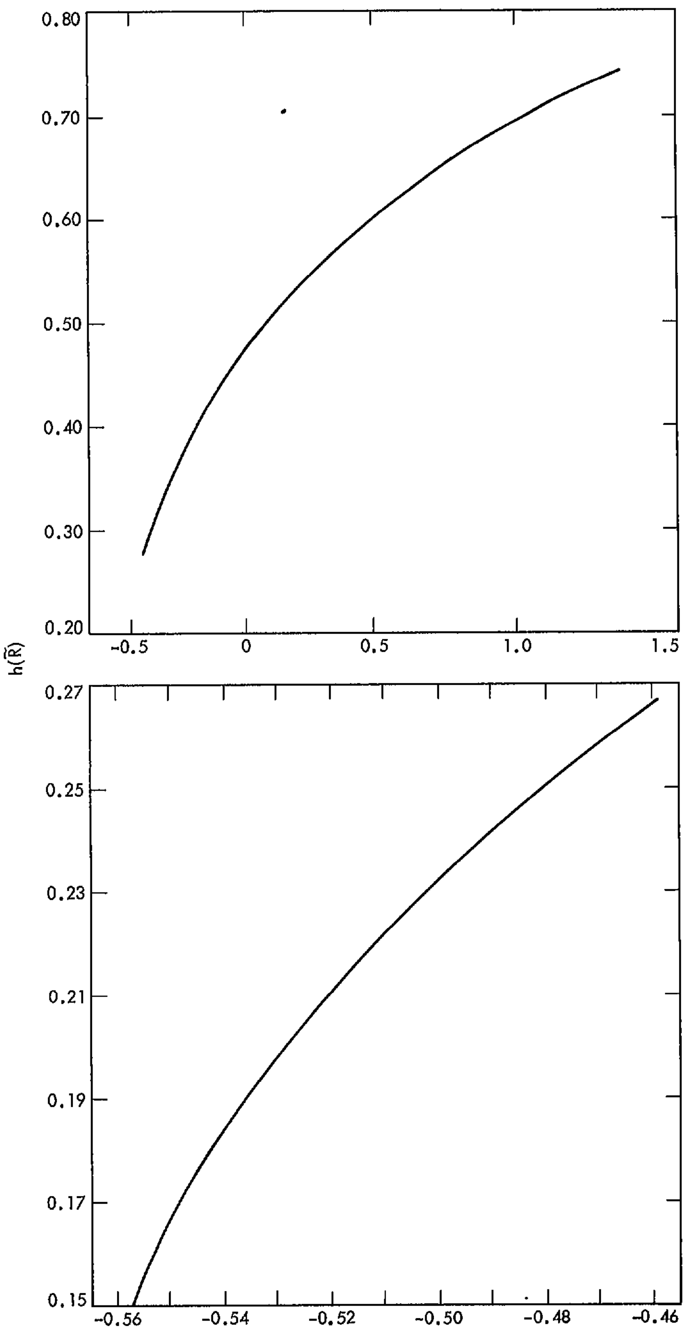

Fig. 8. Bias-removed adjustment function h(R̃)

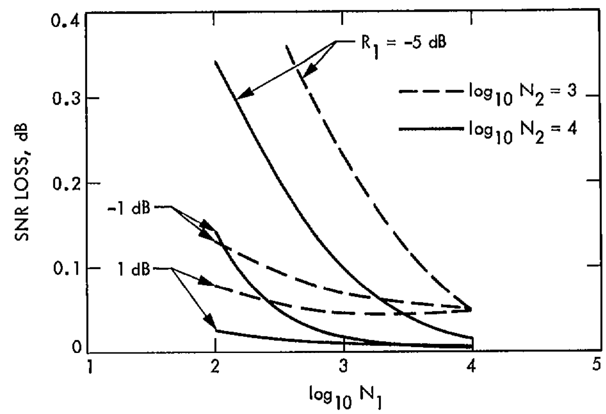

Fig. 9. Simulation of SNR loss vs R_1 and N_1 for $R_2 = -7$ dB (conventional method, with bias removed)

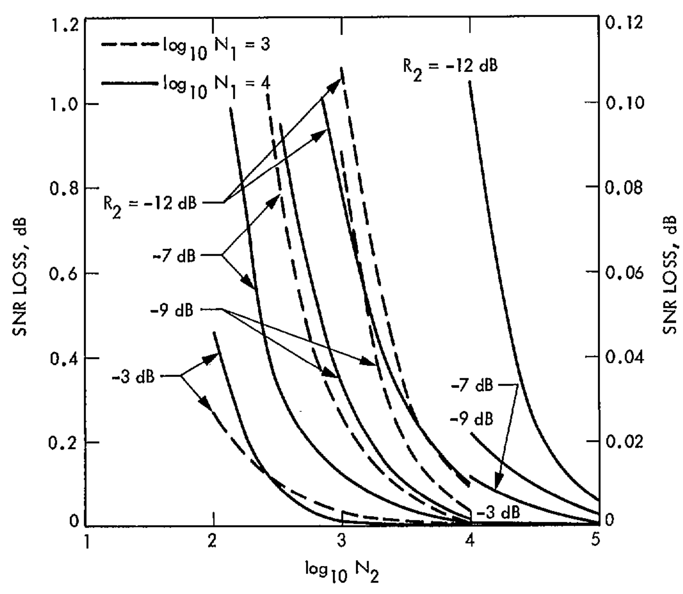

Fig. 10. Simulation of SNR loss vs R_2 and N_2 for $R_1 = -1$ dB (conventional method, with bias removed)

Appendix A

Evaluation of First and Second Moments for the Biased Combiner Weight Estimator

Let

$$x_k = \pm m + n_k$$

where n_k are i.i.d. zero mean Gaussian random variables with variance σ^2.

We want to find $E[\widetilde{\alpha}]$ and $E[\widetilde{\alpha}^2]$ where

$$\widetilde{\alpha} = \frac{\text{sample mean of } |x_k|}{\text{sample variance of } |x_k|}$$

$$= \frac{\frac{1}{N}\sum_{k=1}^{N} |x_k|}{\frac{1}{N-1}\sum_{k=1}^{N}\left(|x_k| - \frac{1}{N}\sum_{i=1}^{N}|x_i|\right)^2}$$

If the sample size is large enough ($N \geqslant 20$ should suffice), we can use the central limit theorem to express α by

$$\widetilde{\alpha} = \frac{\mu + \sqrt{\epsilon_1^2}\,\xi}{\sigma_\nu^2 + \sqrt{\epsilon_2^2}\,\psi}$$

where ξ and ψ are independent zero mean Gaussian random variables with unit variances and

$$\mu = E\{|x_k|\}$$

$$\sigma_\nu^2 = \text{var}\,|x_k|$$

$$\epsilon_1^2 = \frac{1}{N}\sigma_\nu^2$$

From Appendix A of Ref. 2, the variance of a sample variance is

$$\epsilon_2^2 = \frac{1}{N}\left[E(x_k^4) - 4E(|x_k|^3)\,\mu + 3\left(E(x_k^2)\right)^2 - \frac{4N-6}{N-1}\sigma_\nu^4\right]$$

For $\epsilon_2/\sigma_\nu^2 << 1$ (which is usually the case for large N) we have

$$\widetilde{\alpha} = \frac{\mu}{\sigma_\nu^2}\left(1 + \frac{\epsilon_1}{\mu}\xi\right)\left(1 - \frac{\epsilon_2}{\sigma_\nu^2}\psi + \frac{\epsilon_2^2}{\sigma_\nu^4}\psi^2\right)$$

So

$$E[\widetilde{\alpha}] = \frac{\mu}{\sigma_\nu^2}(1+B)$$

where

$$B = \frac{\epsilon_2^2}{\sigma_\nu^4}$$

From Lesh (Ref. 2), we know

$$E\{|x_k|\} = \sqrt{NoT}\left\{\frac{\exp -R}{\sqrt{\pi}} + \sqrt{R}\,\text{erf}(\sqrt{R})\right\}$$

$$E(x_k^2) = NoT\left(R + \frac{1}{2}\right)$$

$$E\{|x_k|^3\} = (NoT)^{3/2}$$

$$\left\{\frac{R+1}{\sqrt{\pi}}\exp -R + \left(R + \frac{3}{2}\right)\sqrt{R}\,\text{erf}(\sqrt{R})\right\}$$

$$E\{x_k^4\} = No^2T^2\left(R^2 + 3R + \frac{3}{4}\right)$$

Since $m = \sqrt{R}\,\sqrt{NoT}$, we have

$$E[\tilde{\alpha}]\, m = \frac{\sqrt{\frac{R}{\pi}}\, \exp -R + R\ \mathrm{erf}(\sqrt{R})}{R + \frac{1}{2} - \left[\frac{\exp -R}{\sqrt{\pi}} + \sqrt{R}\ \mathrm{erf}(\sqrt{R})\right]^2} (1 + B)$$

$$E(\tilde{\alpha}^2) = \frac{\mu^2}{\sigma_\nu^4} E\left(1 + \frac{2\epsilon_1}{\mu}\xi + \frac{\epsilon_1^2}{\mu^2}\xi^2\right)$$

$$\times E\left(1 - 2\sqrt{B}\,\psi + 3B\,\psi^2 - 2B^{3/2}\,\psi^3 + B^2\,\psi^4\right)$$

$$= \frac{\mu^2}{\sigma_\nu^4} (1 + A)(1 + 3B + 3B^2)$$

where

$$A = \frac{\epsilon_1^2}{\mu^2}$$

Thus

$$E(\alpha^2)\, E(x_k^2) = \frac{\left(R + \frac{1}{2}\right)\left[\frac{\exp -R}{\sqrt{\pi}} + \sqrt{R}\ \mathrm{erf}(\sqrt{R})\right]^2}{\left[R + \frac{1}{2} - \left[\frac{\exp -R}{\sqrt{\pi}} + \sqrt{R}\ \mathrm{erf}(\sqrt{R})\right]^2\right]^2}$$

$$\times (1 + A)(1 + 3B + 3B^2)$$

Appendix B

Simulation of the Unbiased Symbol Stream Combining Algorithm

The combining algorithm is summarized as follows:

Step 1: Compute

$$a_1 = \sum |x_k| \ , \quad s_1 = \sum x_k^2$$

$$a_2 = \sum |y_k| \ , \quad s_2 = \sum y_k^2$$

Step 2: Compute the intermediate estimates

$$\widetilde{\alpha}_1 = \frac{(N_1 - 1)a_1}{N_1 s_1 - a_1^2} \ , \quad \widetilde{R}_1 = \frac{\widetilde{\alpha}_1 a_1}{2N_1}$$

$$\widetilde{\alpha}_2 = \frac{(N_2 - 1)a_2}{N_2 s_2 - a_2^2} \ , \quad \widetilde{R}_2 = \frac{\widetilde{\alpha}_2 a_2}{2N_2}$$

Step 3: Compute the final estimates

$$\widehat{\alpha}_1 = h(\widetilde{R}_1)\widetilde{\alpha}_1, \quad \widehat{\alpha}_2 = h(\widetilde{R}_2)\widetilde{\alpha}_2$$

where the function $h(\cdot)$ is approximated by second-order polynominals given in Section IV. The combined symbol stream will be $z_k = \widehat{\alpha}_1 x_k + \widehat{\alpha}_2 y_k$.

By means of subroutine NOISE, we generate two independent sequences of symbols,

$$x_k = \pm\sqrt{2R_1} + n_k^1, \ y_k = \pm\sqrt{2R_2} + n_k^2$$

where n_k^1, n_k^2 are independent, identically distributed normal random variables. Thus,

$$\mathrm{SNR}_{x_k} = \frac{m_1^2}{2\sigma_1^2} = R_1$$

and

$$\mathrm{SNR}_{y_k} = \frac{m_2^2}{2\sigma_2^2} = R_2$$

Each time subroutine NOISE is called, it generates two symbols, either x_k, x_{k+1} or y_k, y_{k+1}. In the main program, we gather all the x_k's and y_k's and compute $\widehat{\alpha}_1$ and $\widehat{\alpha}_2$ according to the above algorithm. To find the output SNR we note that

$$E(z_k) = \widehat{\alpha}_1 E(x_k) + \widehat{\alpha}_2 E(y_k) = \widehat{\alpha}_1 \sqrt{2R_1} + \widehat{\alpha}_2 \sqrt{2R_2}$$

$$\mathrm{Var}(z_k) = \widehat{\alpha}_1^2 \mathrm{Var}(x_k) + \widehat{\alpha}_2^2 \mathrm{Var}(y_k) = \widehat{\alpha}_1^2 + \widehat{\alpha}_2^2$$

So,

$$\mathrm{SNR}_{z_k} = R = \frac{\left(\widehat{\alpha}_1 \sqrt{2R_1} + \widehat{\alpha}_2 \sqrt{2R_2}\right)^2}{2(\widehat{\alpha}_1^2 + \widehat{\alpha}_2^2)}$$

We run the same routine 25 times with different sets of samples (i.e., $\{x_k\}$ and $\{y_k\}$). The final SNR loss is computed based on the averaged output SNR:

$$\text{SNR loss} = 10 \log \left(\frac{R_1 + R_2}{R}\right)$$

The simulation is run for different values of R_1, N_1 and R_2, N_2. The results are plotted in Figs. 8 and 9. Listings of the main program, the function $h(\cdot)$, and the subroutine NOISE are also provided for reference in Figs. B-1, B-2, and B-3.

```
SNR.FOR

      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(100000),Y(100000)
      COMMON /COM1/B1,BB,CC,VV
      B1=2.**31
      BB=B1-1.
      CC=7.**5
      VV=1099.
      READ(5,10)R1,N1
   10 FORMAT(F4.0,I6)
      PRINT 15,R1,N1
   15 FORMAT(2X,F4.0,2X,I6)
      R1=10.**(.1*R1)
      AR1=DSQRT(2.*R1)
      XN1=DFLOAT(N1)
      DO 70 K=1,2
      R2=DFLOAT(4*K-11)
      R2=10.**(.1*R2)
      AR2=DSQRT(2.*R2)
      DO 70 L=2,5
      N2=10**L
      XN2=DFLOAT(N2)
      R=0.
      DO 50 I=1,25
      A1=0.
      S1=0.
      DO 20 N=1,N1,2
      CALL NOISE(AR1,X(N),X(N+1))
      A1=A1+DABS(X(N))+DABS(X(N+1))
   20 S1=S1+X(N)*X(N)+X(N+1)*X(N+1)
      AT1=(XN1-1.)*A1/(XN1*S1-A1*A1)
      RT1=.5*A1*AT1/XN1
      AT1=H(RT1)*AT1
      A2=0.
      S2=0.
      DO 30 N=1,N2,2
      CALL NOISE(AR2,Y(N),Y(N+1))
      A2=A2+DABS(Y(N))+DABS(Y(N+1))
   30 S2=S2+Y(N)*Y(N)+Y(N+1)*Y(N+1)
      AT2=(XN2-1.)*A2/(XN2*S2-A2*A2)
      RT2=.5*A2*AT2/XN2
      AT2=H(RT2)*AT2
      RR=.5*(AT1*AR1+AT2*AR2)**2/(AT1*AT1+AT2*AT2)
   50 R=R+RR
      R=R/25.
      RLOSS=10.*DLOG10((R1+R2)/R)
      PRINT 60,N2,R1,R2,R,RLOSS
   60 FORMAT(2X,I8,4(2X,F8.4))
   70 CONTINUE
      STOP
      END
```

Fig. B-1. Main program

```
      FUNCTION H(X)
      IF(X.GT.0.9000) GO TO 10
      H=-121.49*X*X+221.5*X-100.6772
      RETURN
   10 IF(X.GT.0.9600) GO TO 20
      H=-14.6*X*X+29.5263*X-14.4791
      RETURN
   20 IF(X.GT.1.0500) GO TO 30
      H=-3.5596*X*X+8.4951*X-4.4639
      RETURN
   30 H=-1.0004*X*X+3.0811*X-1.5990
      RETURN
      END
```

Fig. B-2. Function h(·)

```
      SUBROUTINE NOISE(X,E1,E2)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /COM1/B1,BB,CC,VV
   10 VV=VV*CC
      VV=DMOD(VV,BB)
      V1=2.*VV/B1-1.
      VV=VV*CC
      VV=DMOD(VV,BB)
      V2=2.*VV/B1-1.
      S=V1*V1+V2*V2
      IF(S-1.) 20,10,10
   20 V=DSQRT(-2.*DLOG(S)/S)
      E1=V1*V+X
      E2=V2*V-X
      RETURN
      END
```

Fig. B-3. Subroutine NOISE