Communications Research Centre

IMPLEMENTING A MATRIX-INVERSION ALGORITHM IN A LIMITED-PRECISION ADAPTIVE ANTENNA ARRAY PROCESSOR

by

Robert W. Jenkins

This work was sponsored by the Department of National Defence, Research Development Branch, under Project No. 32B75.

CRC TECHNICAL NOTE NO. 725
OTTAWA, DECEMBER 1984

Government of Canada
Department of Communications

Gouvernement du Canada
Ministère des Communications

Canada

COMMUNICATIONS RESEARCH CENTRE

DEPARTMENT OF COMMUNICATIONS

CANADA

IMPLEMENTING A MATRIX-INVERSION ALGORITHM IN A LIMITED-PRECISION ADAPTIVE ANTENNA ARRAY PROCESSOR

by

Robert W. Jenkins

Directorate of Military Communications

CRC TECHNICAL NOTE NO. 725

December 1984
OTTAWA

This work was sponsored by the Department of National Defence, Research Development Branch, under Project No. 32B75.

Implementing a Matrix-Inversion Algorithm in a Limited-Precision Adaptive Antenna Array Processor

Robert W. Jenkins

Abstract

The effect of computation accuracy on the performance of a small adaptive antenna system using a digitally-implemented matrix inversion algorithm is examined. A single-board array processor with 16-bit mantissa floating-point accuracy is compared with a VAX-11/750 computer having 24-bit accuracy in the single-precision mode, and 56-bit accuracy in the double-precision mode. The effects of limited precision on array performance are worse when the signal covariance matrix to be inverted is ill-conditioned (i.e., nearly singular), as is the case with an n-element array in the presence of less than n unrelated signals and a low system noise level. Using a simulated signal environment consisting of a wanted communications signal, a single jamming signal, and a variable system noise level, the performance of the array is evaluated. A need for about -20 dB of artificially-injected system noise (relative to the total signal) is demonstrated in the case of the single-board array processor, in order that the covariance matrix be sufficiently well-conditioned for adequate array performance.

Implementing a Matrix-Inversion Algorithm in a Limited-Precision Adaptive Antenna Array Processor

Robert W. Jenkins

1.0 INTRODUCTION

In adaptive antenna arrays designed to cancel unwanted interference while receiving communications, the signals from the array elements are weighted in both amplitude and phase and combined, so as to minimize the array response to the interfering signals and maintain the response to the communications at the desired level. One approach is to choose the weights so as to minimize the power in the error signal, which can be defined as the mismatch between the array output and a reference signal representing the desired communications. The least mean square error solution for the weights is given by the Wiener-Hopf equation:

$$\vec{w} = (C_{xx})^{-1} \vec{c}_{rx}$$

where $\vec{w}$ is the (complex) vector of the weights, C_{xx} is the covariance matrix of the input signals $\vec{x}$,

$$C_{xx} = \langle \vec{x}\vec{x}^H \rangle$$

and $\vec{c}_{rx}$ is the correlation vector between the reference signal r and the input signals vector $\vec{x}$,

$$\vec{c}_{rx} = \langle r\vec{x}^* \rangle$$

where $\langle\rangle$ denotes a statistical average, $\rightarrow$ denotes a vector quantity, $*$, the complex conjugate, and ^{H} the conjugate transpose.

Figure 1 provides a block diagram of the architecture of an adaptive array system. The signals from the individual antennas in the array are multiplied by weights and combined to produce the output signal. The weights are chosen using information from the input signals themselves, and a reference signal representing the communications signal that the system wants to see, and possibly the output signal. In many systems, the reference signal is derived from the output. The present paper, however, is restricted to the problem of finding the weights from the input signals and reference via the Wiener-Hopf equation, and it is assumed that an adequate reference is available.

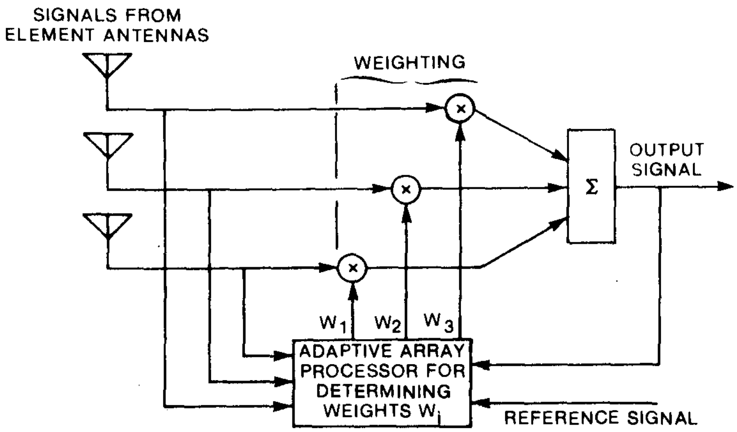

Figure 1. Adaptive antenna system architecture

The Wiener-Hopf equation may be solved in an iterative manner, using the method of steepest descent, or LMS algorithm (Widrow et al, 1967) and many of the arrays built to date have used this technique. The equation can be implemented with analogue components, as a closed loop system which does not require precise knowledge or control of the values of the analogue weights. In its digital form, it involves only a small amount of computation and so can be handled by a relatively simple signal processing system. The drawbacks are that it does not make optimum use of the available signal information, and it involves a finite adaptation time which can be very long in the presence of several interfering signals of unequal strength.

A more direct approach is to replace the statistical averages in the Wiener-Hopf equation with time averages and solve the resulting equation directly. Such an approach, making use of an output-derived reference signal, has been successfully modelled in non-real time (Jenkins, 1983). A similar approach, the sample matrix inversion (SMI) algorithm, which does not involve a reference signal, but makes use of either a known signal direction or the power-inversion properties of adaptive arrays, has been analyzed (Reed et al., 1974) and implemented (Horowitz, 1979). The use of a matrix inversion procedure makes more optimal use of the available signal information, and permits a much faster adaptation which is not degraded in the presence of several unequal interfering sources. However, it requires precise control of the weights, being an open-loop solution, and represents a much greater computational load than the LMS algorithm. Recent advances in digital electronics and the accompanying greater processing power available have made this approach technically feasible, at least for arrays of a few elements.

One area of concern in implementing a matrix inversion solution for the weights is the precision with which the covariance matrix can be inverted. When the signal environment consists of only several signals incident on the array, less than the number of antennas, and no noise, the resultant covariance matrix is singular and its inverse does not exist. This never actually occurs in practice, because the uncorrelated system noise at each of the antennas, plus the digital "sampling" noise, add terms to each of the diagonal elements of the covariance matrix, making it nonsingular. However, when these noise components are small relative to the several large signals, the covariance matrix remains "close-to-singular", or ill-conditioned. The matrix inversion calculation for such a matrix involves finding small differences between very large numbers, and so, for cases where the background noise level is low, the matrix inversion process will be affected adversely by a limited computational precision.

A method, of permitting the matrix inversion technique to be implemented when the computational precision is limited, is to add diagonal terms to the covariance matrix. These diagonal terms represent, in effect, an artificial noise component or "noise injection", which improves the conditioning of the covariance matrix, thereby permitting its accurate inversion without requiring an increase in the precision of computation. The artificial noise, by virtue of the way it is introduced, does not correlate with any of the input signals and so the weights found will still cancel the undesired interference.

The present paper looks at an implementation of the matrix inversion algorithm (Jenkins, 1983), using a commercially-available Multibus-controlled single-board array processor of limited digital accuracy. Its performance in a simulated four-element adaptive antenna array is determined, and compared with that of a VAX-11/750 computer with both single-precision and double-precision modes of calculation. The need for noise injection in the case of the single-board array processor is demonstrated, and conclusions are drawn regarding the performances achievable and appropriate levels of injected noise.

2. DESCRIPTION OF SYSTEM

A Marinco APB3024M single-board array processor was used for the processor, and a floating-point routine was used to implement the matrix inversion procedure. The floating-point accuracy of the APB3024M is 16 bits (in the mantissa), but accumulated products have an accuracy of 32 bits so long as they remain in the processor register. The LU decomposition technique (Williams, 1972), which takes advantage of the greater precision in the product accumulation process, was used for matrix inversion in the APB3024M processor.

The VAX-11/750 computer, has a floating-point accuracy of 24 mantissa bits in its single-precision mode, and 56 mantissa bits in its double-precision mode of calculation. The matrix inversion routines used with this computer were part of an already-existing subroutine package, and used the standard Gauss-Jordon method of matrix inversion.

3. SIGNAL SIMULATION AND TEST PROCEDURE

A simple procedure was devised for simulating the signals in an antenna array. The input signal environment was assumed to consist of a desired signal $\vec{s}$, (denoted as a complex vector, with its components representing its instantaneous values at each of the array elements), a single input jamming signal $\vec{t}$, and white noise $\vec{n}$ uncorrelated between array elements. It was assumed that the desired signal and jamming signal were uncorrelated, and that the noise was also uncorrelated with either signal.

The Wiener-Hopf equation for the weights is

$$\vec{w} = (C_{xx})^{-1}\vec{c}_{rx} \tag{1}$$

where C_{xx} is found from

$$C_{xx} = \overline{\vec{x}\vec{x}^H}$$

where $\overline{}$ refers to a time average. Since $\vec{x} = \vec{s}+\vec{t}+\vec{n}$, we can write

$$C_{xx} = \overline{(\vec{s}+\vec{t}+\vec{n})(\vec{s}^H+\vec{t}^H+\vec{n}^H)}$$

$$= \overline{\vec{s}\vec{s}^H} + \overline{\vec{t}\vec{t}^H} + \overline{\vec{n}\vec{n}^H} + \overline{\vec{s}\vec{t}^H} + \overline{\vec{t}\vec{s}^H} + \overline{\vec{s}\vec{n}^H} + \overline{\vec{n}\vec{s}^H} + \overline{\vec{t}\vec{n}^H} + \overline{\vec{n}\vec{t}^H}$$

The last six terms are zero, since the wanted, jamming, and noise signals do not correlate. The k,jth element of the matrix C_{xx} thus has the form

$$C_{xx}(k,j) = \overline{s(k)s^*(j)} + \overline{t(k)t^*(j)} + \overline{n(k)n^*(j)}$$

For identical antenna elements and narrowband plane wave signals, the time averages for the wanted signal contribution can be reduced to the form

$$\overline{s(k)s^*(j)} = S^2 e^{i\theta_s(k,j)}$$

where S is the common wanted signal amplitude at the elements, and $\theta_s(k,j)$ is the phase difference between the kth and jth elements. Likewise, for the jamming contribution,

$$\overline{t(k)t^*(j)} = T^2 e^{i\theta_t(k,j)} \quad .$$

The noise contribution becomes

$$\overline{n(k)n^*(j)} = N^2 \delta(k,j)$$

where $\delta(k,j)$ = (1 for k=j, 0 otherwise). Thus we have

$$C_{xx}(k,j) = S^2 e^{i\theta_s(k,j)} + T^2 e^{i\theta_t(k,j)} + N^2 \delta(k,j) \qquad (2)$$

The reference signal is assumed to be of unit amplitude and to have the same phase dependence as the desired signal. Then the correlation vector $\vec{c}_{rx}$ is given by

$$\vec{c}_{rx} = \overline{r\vec{x}^*} = \vec{S}'^* \qquad (3)$$

where ' denotes that the time dependence has been removed.

The test procedure was to start with a pair of time-independent wanted and jamming signals $\vec{s}'$, $\vec{t}'$, and a noise level N, then to calculate the covariance matrix according to equation (2), and to find the weights by means of equations (3) and (1). The weights were then applied to the input signal vectors $\vec{s}'$, $\vec{t}'$ and the uncorrelated noise N, to determine the output signal levels.

Note in equation (2) that the uncorrelated noise contributes only to the diagonal elements of C_{xx}. It can also be shown that the matrix $C_{xx}(N=0)$ formed from the two signals $\vec{s}'$ and $\vec{t}'$ without any uncorrelated noise is singular. From equation (2), we can write C_{xx} in the form

$$C_{xx} = C_{xx}(N=0) + N^2 I \qquad (4)$$

where I is the identity matrix. The sum of C_{xx} and $N^2 I$ can be shown to be nonsingular and so the effect of the uncorrelated noise is to make the covariance matrix nonsingular. The larger the value of N^2, the better the conditioning of C_{xx}, and the less susceptible will computations of its inverse be to round-off errors in the matrix itself or in the calculation. Noise injection, as previously mentioned, consists of improving the conditioning of C_{xx} by adding a constant term to the diagonal elements, i.e.,

$$C_{xx}' = C_{xx} + aI, \quad (5)$$

A comparison of equations (4) and (5) reveals the similarity between noise injection and uncorrelated input noise.

Equation (5) was used in the tests, to provide noise injection where it was required.

4.0 TEST DETAILS

The array in these tests was assumed to consist of four linearly-placed identical antennas, spaced one-half wavelength apart. The desired signal $\vec{s}$ was assumed to be incident on the array at +10° relative to the normal. The jamming signal $\vec{t}$ was given various amplitudes relative to the desired signal and was incident at +30° unless otherwise stated. The uncorrelated noise was varied in steps of 10 dB, downwards from 0 dB relative to the desired signal. Initial tests, in which the performances of the APB3024M and VAX-11/750 (in both single- and double-precision modes) were assessed in the adaptive array system, used no noise injection. Later tests, in which the effectiveness of noise injection in overcoming the effects of limited digital accuracy was assessed, used the APB3024M only. For these tests, injected noise levels of -10, -20, and -30 dB relative to the total input signal power (wanted signal plus jamming signal plus noise) were used.

5.0 RESULTS

5.1 Results without noise injection

The exact solution to the Wiener-Hopf equation is a set of weights which minimize the error power. The error power P_E is given in terms of the output jamming signal J(out), the output desired communications signal S(out), and the output noise power P_n(out), by

$$P_E = \left|\overline{J(out)}\right|^2 + P_n(out) + \left|\overline{S(out) - r}\right|^2$$

where r is the reference signal. Rewriting this in terms of the weights and input signals, we have

$$P = \left|\vec{w}^H \vec{t}'\right|^2 + \left|\vec{w}\right|^2 N + \left|\vec{w}^H \vec{s}' - r'\right|^2 \quad (6)$$

The minimum-error solution for the weights thus minimizes the sum of the three terms on the right-hand side of equation (6), which reflect the error contributions from the jamming, system noise, and comms signal/reference mismatch, respectively.

The first term of equation (6) implies that the minimum-error weights correspond to a null in the jammer direction. The second term implies that the amplitude of the minimum-error weights vector be zero, in the absence of any other requirement, and as small as possible otherwise. Since the weights vector amplitude does not affect the direction or existence of nulls in the array pattern, the first and second terms do not present conflicting requirements. The third term implies that the array match the output desired signal to the reference as closely as possible, which restricts the weights vector amplitude to a fixed, non-zero value. Thus the second and third terms conflict in that they cannot be simultaneously minimized.

The relative importance of the second and third terms in equation (6) depends on the relative magnitudes of the noise N and the desired signal S. If N is the smaller, the emphasis is to match the desired signal to the reference before reducing the weights vector amplitude (as the second term is already small). If however the noise N is larger than S, then the weights vector is reduced so that the output noise contribution to error power is the same approximately as the signal/reference mismatch contribution. When the noise is much larger, this implies that the output noise power is maintained equal to the reference power. Thus, when the input noise power is significant, the exact-solution weights represent a compromise between reducing the output noise-plus-jamming power to a minimum and matching the output desired signal as closely as possible to the reference signal.

In Figure 2, the array performance in terms of output signal to noise-plus-interference ratio SNIR(out) is plotted against the input signal to noise ratio SNR(in) for various input jamming levels. In this figure, the exact solution is represented by a straight line, with SNIR(out) being proportional to SNR(in). This solution corresponds to the jammer being nulled almost completely and the array directivity made optimum in terms of permitting maximum gain in the direction of the desired signal (up to the level of the reference signal) while maintaining the least possible mean gain over all directions (thereby minimizing the output noise).

The results are shown in Figure 2 for the three means of calculation (APB3024M, VAX-11/750 single precision, and VAX-11/750 double precision). At relatively high noise levels, i.e., SNR(in) ≈ 0dB, the results for the three processors lie on the exact (straight-line) curve. However, as the input noise level is reduced, and the covariance matrix becomes less well-conditioned (closer to being singular), the actual weights solution is perturbed from the exact solution, and the performance in terms of SNIR(out) drops below the exact-solution result. The point of departure depends on the jammer strength,

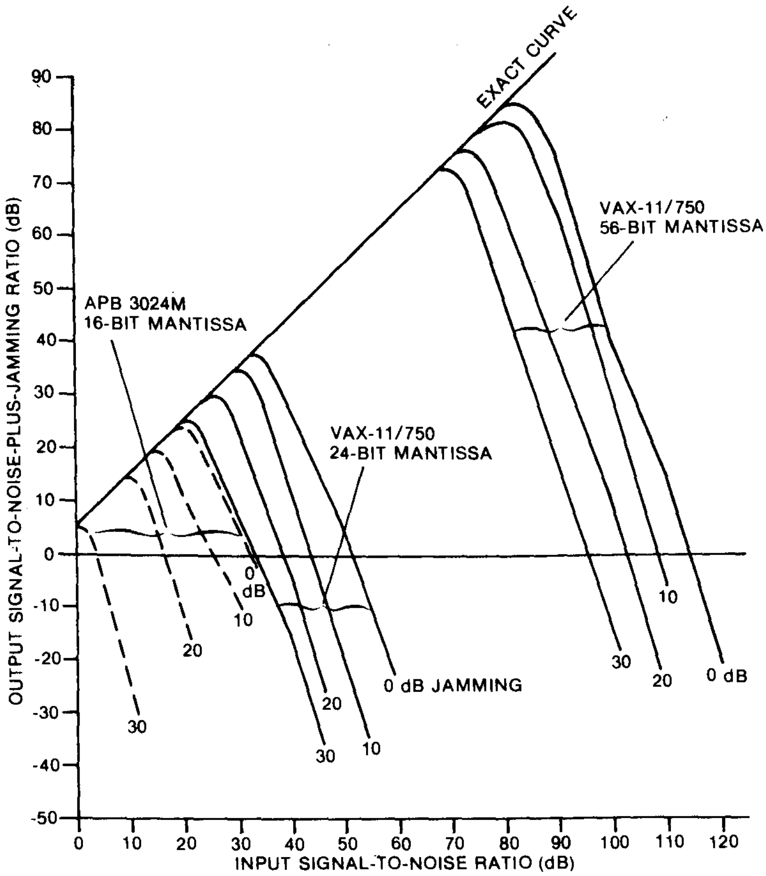

Figure 2: Output signal-to-noise-plus-jamming ratio SNIR(out) as a function of the input signal to noise ratio SNR(in), for various input jamming powers expressed in dB relative to the desired signal, for the simulated adaptive array with each of the three matrix inversion processors: APB3024M, VAX-11/750 single precision, and VAX-11/750 double precision. No injected noise.

but more importantly, on the precision of the calculations. For 20 dB jamming (relative to the desired signal at the input), the results depart from the exact curve above SNR(in) = 70 dB for the (most precise) VAX-11/750 double-precision processor, above SNR(in) = 25 dB for the VAX-11/750 single-precision processor, and above SNR(in) = 10 dB for the (least precise) APB3024M processor.

A detailed look at the results shows that, for the VAX-11/750 single- and double-precision matrix inversion processors, the first effect of the limited computational accuracy as the input noise level is dropped, is a dramatic increase in the weight vector amplitude, which causes the output noise level to increase sharply. The desired signal at that point continues to be matched to the reference and the jamming to be reduced well below the desired signal at the output for both processors. Thus the deterioration in performance can be described mainly as a loss in array directivity.

For the less-accurate APB3024M single-board array processor (which as previously noted used a different matrix inversion algorithm), the departure from the exact solution occurred at quite low input signal-to-noise ratios. In fact, with a 30 dB jamming signal, the matrix inversion process was not adequately accurate unless the input signal-to-noise ratio remained at 0 dB or lower, which does not usually occur and in fact is not acceptable for most applications. For this processor, unlike the VAX-11/750 processors, the deterioration manifested itself more in terms of a lack of jammer nulling than a reduction in array directivity towards the desired signal, although the latter effect was observed to a lesser degree.

Figures 3 and 4 show the separate effects of the minimum-error requirement and limited processor-accuracy on the radiation patterns for the array. The array patterns found for various levels, using the limited-precision APB3024M processor, are given in these figures. The gains in the directions of the desired and jamming signals are also indicated.

In Figure 3, the jamming level is 10 dB above the desired signal. From Figure 2 it can be concluded that for this jamming situation, the input noise level has to be higher than -20 dB for the matrix inversion to be sufficiently accurate to produce the theoretically expected result. Both the curves for 0 dB and -10 dB noise levels in Figure 3 have strong nulls in the jammer direction. Further, they tend to have their maximum gain in the desired signal direction, as would be suggested by the requirement to minimize the output noise level while maintaining the output desired signal close to the reference signal. For the -10 dB noise case, where the input noise is below the signal level, the weights can remain large enough to maintain a near-zero dB gain in the direction of the desired signal (needed to match it to the reference at the output). When the noise is increased to 0 dB, the noise contribution to output power begins to constrain the weights vector amplitude to values sufficiently low that the gain in the desired-signal direction is reduced below 0 db (-2 dB).

As the noise levels are reduced to -20 dB and lower for the conditions of Figure 3, the covariance matrix becomes sufficiently ill-conditioned for the effects of the limited precision to be felt. For an input noise level of -20 dB, the array pattern null toward the jammer is not only reduced in depth, but offset in direction. The general shape of the array pattern and average gain remain close to that for -10 dB noise. For a noise level of -30 dB, the null in the jammer direction has effectively dissappeared, and the array gain has increased dramatically.

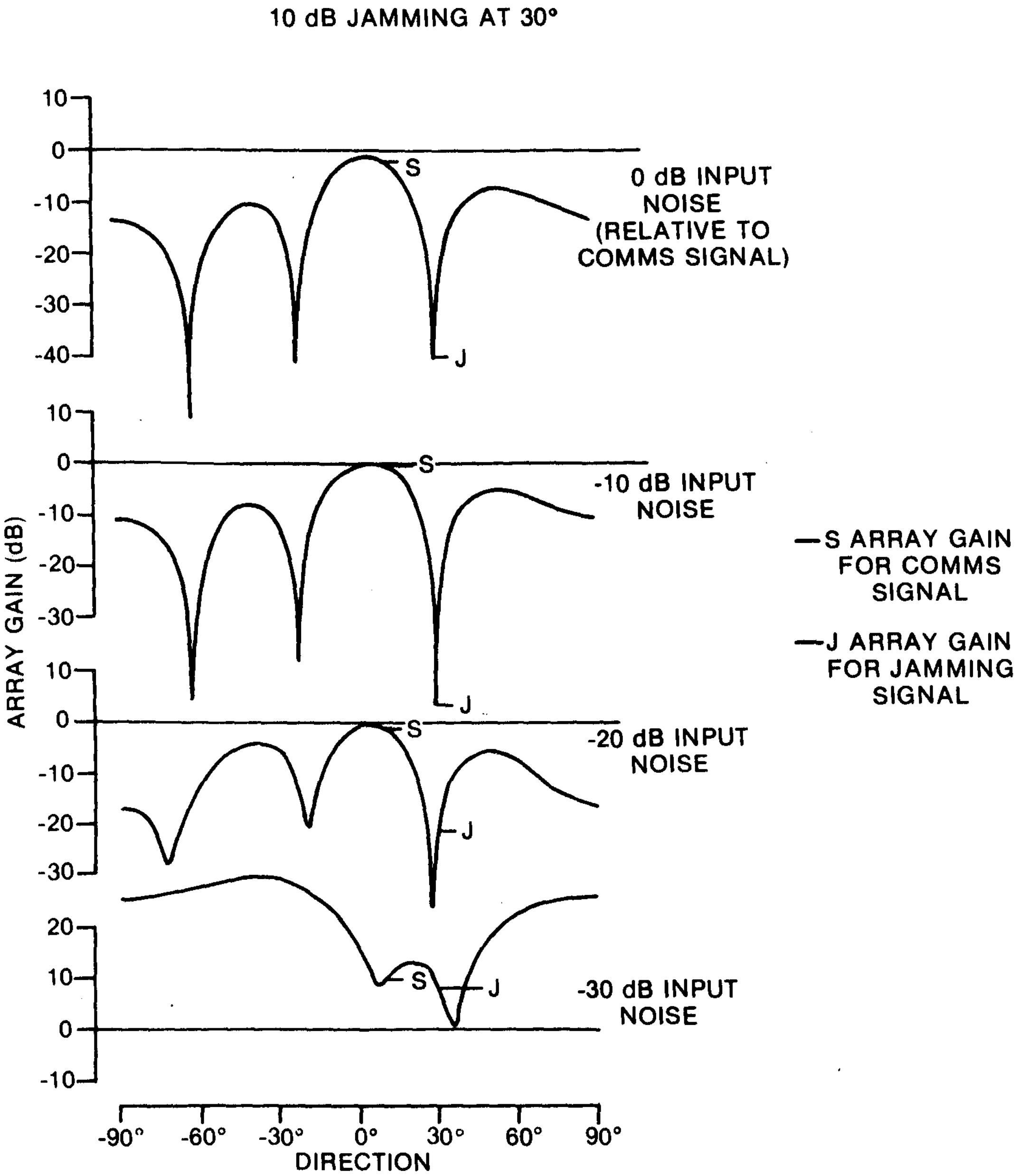

Figure 3: Adaptive array patterns using the APB3024M processor, for 10 dB jamming and various levels of input noise.

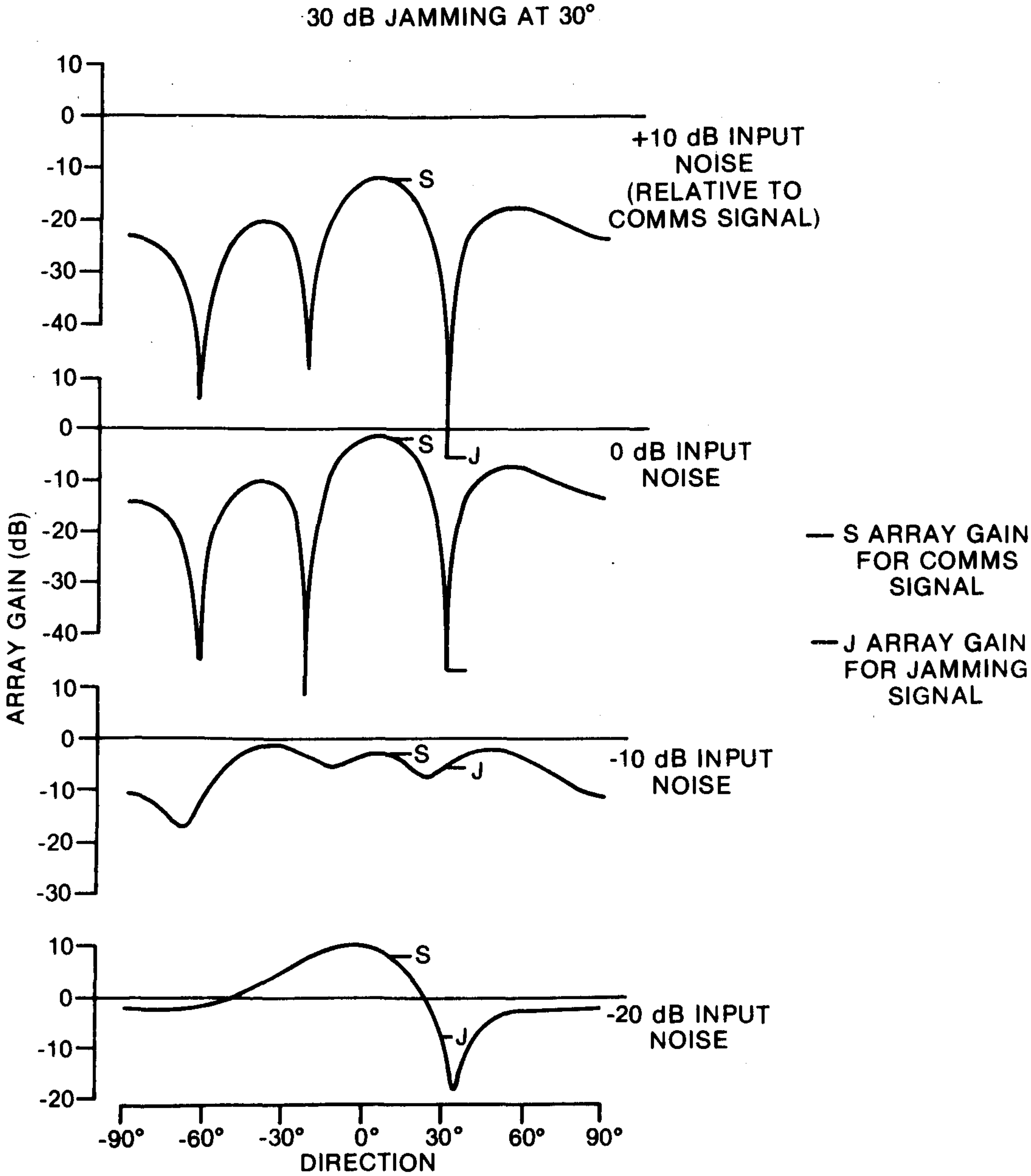

Figure 4. Adaptive array patterns using the APB3024M processor, for 30 dB jamming and various levels of input noise.

The array patterns in Figure 4, calculated for an input jamming level of 30 dB, show similar effects. Noise levels of +10 and 0 dB are sufficiently high for the covariance matrix conditioning to permit adequate matrix inversion. The 0 dB noise level requires only a slightly reduced gain in the direction of the desired signal. For smaller noise levels (-10 and -20 dB), the effects of limited precision are strongly evident. Drastic changes are seen in the array pattern: the null in the jammer direction is gone, and the average gain begins to increase.

4.2 Results with noise injection

The poor results shown in Figure 2, 3 and 4 for the limited-accuracy APB3024M indicate that some modification of the matrix inversion process is necessary for this implementation to be successful. It was noted that, when the noise level was sufficiently high, the covariance matrix was well-conditioned enough to permit adequate jammer nulling, but that such high noise levels cannot be expected to always occur naturally, nor are they acceptable for most applications. The technique of noise injection, however, adds an artificially high noise component to the diagonal elements of the covariance matrix, thereby improving its conditioning, without adding any noise to the original input signal upon which the derived weights act to produce an output.

The noise injection technique was tested using the APB3024M processor. The artificial noise level to be added to the diagonal elements of the covariance matrix was chosen relative to the total input power (i.e. the diagonal elements) so that the change in conditioning of the covariance matrix remained independent of the signal level.

Figure 5 shows the results found. The output ratio SNIR(out) is shown as a function of the input jamming to signal ratio, for various amounts of injected noise.

Separate graphs are shown for different input-signal-to-(real) noise ratios SNR(in). The most straightforward case to interpret is that of the lowest input noise level (SNR(in) = 40 dB). With no injected noise, the results for this case were seen in Figure 2 to be very poor (SNIR(out) less than 0 dB). However, as the injected noise is added, and increased from -30 to -20 dB, the performance improves as can be seen in Figure 5. When the injected noise is increased further, beyond the input level of the desired signal, the performance is reduced. For example, a reduction in performance occurs for an input jamming level of 10 dB or greater when the injected noise is increased to -10 dB relative to the total signal, near the desired signal level. An examination of the output signal levels reveals that increasing the injected noise beyond this

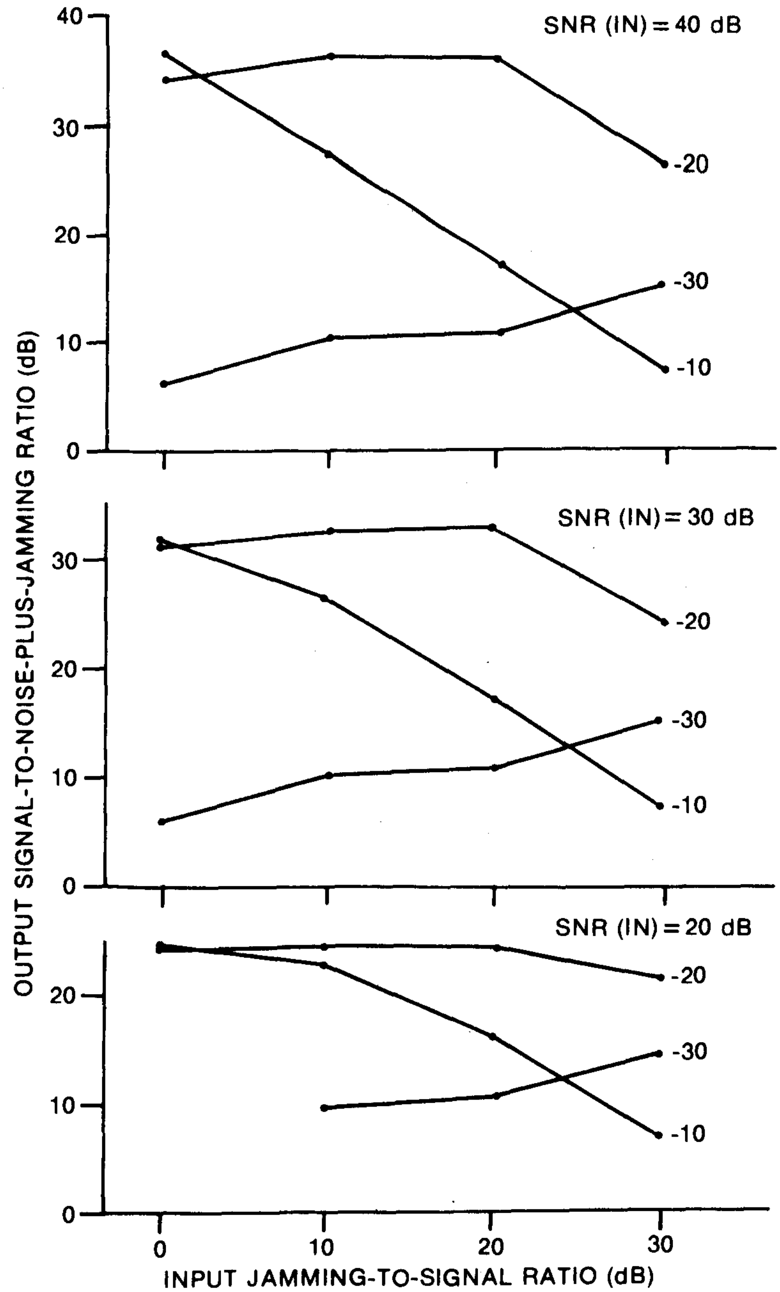

Figure 5: Output signal-to-noise-plus-jamming ratio SNIR(out) as a function of the input jamming power in dB relative to the desired signal, for the simulated adaptive array with the APB3024M processor, for three levels of injected noise. Separate graphs are shown for input signal-to-noise ratios SNR(in) = 20, 30, and 40 dB.

point causes a proportionate decrease in the output wanted signal without necessarily reducing the output jamming signal.

Increasing the real noise from the low level (-40 dB) of the upper graph in Figure 5 does little beyond limiting the maximum output ratio (SNIR(out) achievable. The values of SNIR(out) that are below the noise-defined limit are almost unaffected, as can be seen by comparing the graphs for SNR(in) = 20 and 30 dB with those for 40 dB.

An examination of the corresponding array patterns provides additional insight into the effects of artificial noise injection. Figure 6 shows the array patterns found for injected noise levels of -30, -20, and -10 dB relative to the total input signal, when the total input signal consists of a communications signal (at 0 dB) and a 20 dB jamming signal.

All three array patterns in Figure 6 are somewhat similar in shape and have a null in the jammer direction. However there are significant differences. The array pattern for the -30 dB injected noise level is relatively limited in its null depth, suggesting that the covariance matrix is not quite well-conditioned enough to overcome the effects of limited-precision processing. The gain in the communications signal direction is close to 0 dB, since the injected noise level is not high enough to force a mismatch between the output communications signal and reference. When the injected noise is increased to -20dB relative to the total signal (0 dB relative to the communications signal), the matrix conditioning improves and a deep null results in the direction of the jammer. The gain in the communications signal direction is reduced to -2 dB, since now the injected noise is strong enough to start to restrict the weights vector to values that are too small to permit complete communications signal/reference matching. When the injected noise is increased still further, to -10 dB, there is no further improvement in null depth, but the gain in the communications signal direction is reduced correspondingly, to -12 dB.

From the results of Figure 5 and 6, an injected noise level of -20 dB (relative to the total signal) appears to be the most appropriate over a range of input jamming signals 0 to -30 dB (relative to the desired signal). The array performance achieved with such noise injection appears to be quite satisfactory.

The results discussed so far have dealt with one fixed signal geometry and have used the full 16-bit digital range available in the APB3024M for specifying the covariance matrix. In Figure 7, the effect of noise injection is shown for three different situations as follows:

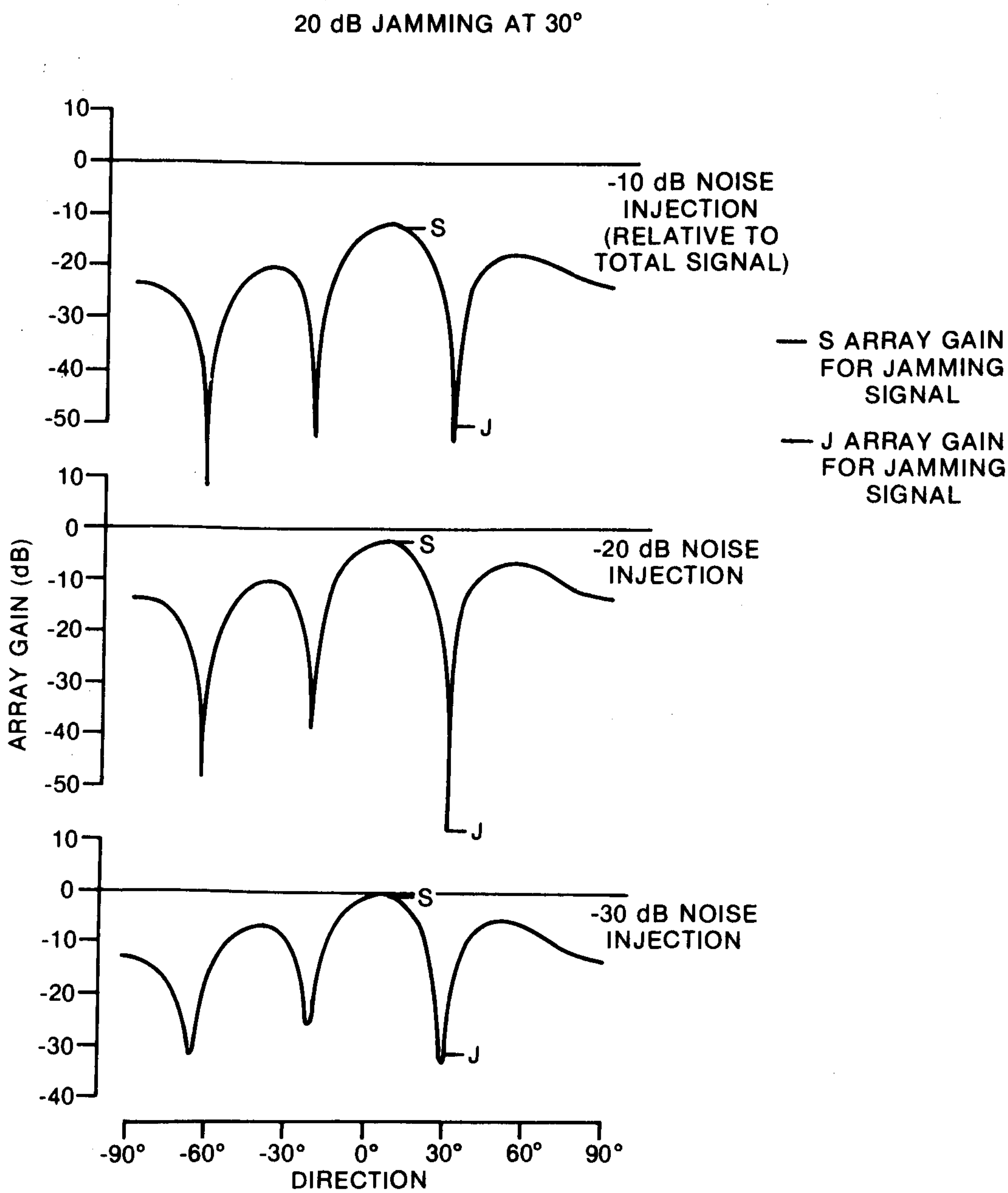

Figure 6: Adaptive array patterns using the APB3024M processor, for 20 dB jamming and various levels of injected noise.

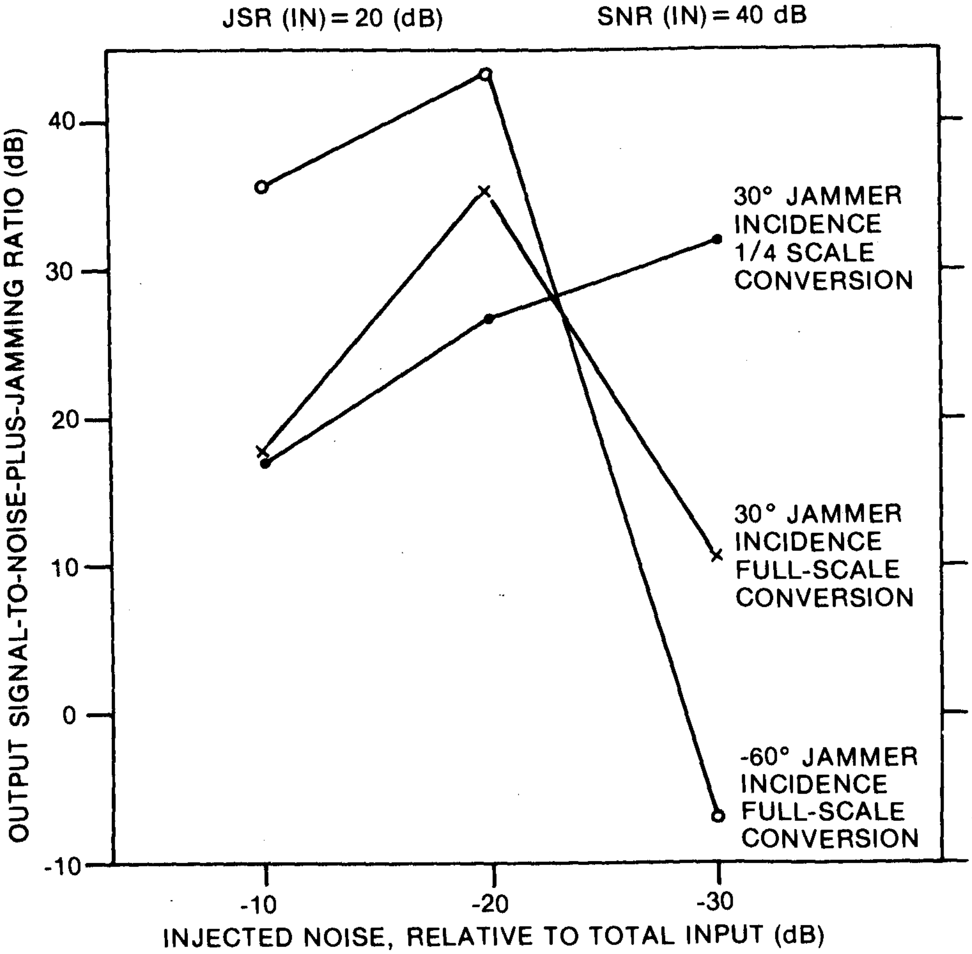

Figure 7: Output signal-to-noise-plus-jamming ratio SNIR(out) as a function of the level of noise-injection, for the simulated adaptive array with the APB3024M processor, for three different signal situations.

1. A 20 dB jamming signal incident at 30° (as previously), but with the digital conversion such that the initial covariance matrix used only one-quarter of the available digital range, i.e., 14 of the 16 bits available. (Note that the actual inversion computation still proceeded with the same 16-bit mantissa accuracy).

2. A 20 dB jamming signal incident at 30°, the covariance matrix using the full digital range available (as previously).

3. A 20 dB jamming signal incident at -60°, with the initial covariance matrix using the full digital range available.

The output ratio SNIR(out) is shown as a function of the injected noise level, for the three situations.

The results of Figure 7 support the previous findings, that a -20 dB injected noise level is most appropriate for the presently-considered processor. The -30 dB noise injection, although an improvement over no noise injection, was normally not sufficient for good performance, although the results for the three cases differed widely (see comment following this paragraph*). The values of SNIR(out) at the higher (-10 dB) injected-noise level were down by about 10 dB from those seen for the -20 dB noise injection, and reflect the proportionately lower signal levels that occurred when the injected-noise level was raised above the wanted-signal level at the input. As before, the array performance for all cases considered was entirely satisfactory with the -20 dB injected-noise level.

---

*Note:

The surprisingly good results in Figure 3 for the quarter-scale conversion for -30 dB noise injection probably reflect the element of chance that exists in the process of determining null depth and exact position. It is at first glance tempting to attribute this result to the extra "contribution to injected noise" from the more limited digital accuracy in the original covariance matrix. However, this contribution was at least an order of magnitude below the -30 dB injected-noise level itself, and so could not have had much effect. It should also be noted that truncation errors at the 14-bit level may well be masked by computation errors. This is suggested by the observation (see Figure 2) that noise levels of the order of 2^{-14} of the input signal (-42 dB) do not produce a sufficiently well-conditioned matrix for inversion by the APB3024M.

6.0 CONCLUSION

The effects of limited computation accuracy on the performance of a small adaptive array system using a matrix inversion algorithm have been examined. The system in question implemented the matrix inversion in floating-point form in a APB3024M single-board array processor for which the digital accuracy is 16 bits in the mantissa. A simulated signal environment was used to evaluate the adequacy of the implemented matrix inversion process. This environment consisted of a wanted communications signal matched by an identical reference signal in the processor, a single jamming signal of various amplitudes and directions relative to the communications signal, and a variable background noise level, as seen by a set of four identical omnidirectional antennas spaced one-half wavelength apart in a linear array.

A need is demonstrated for approximately -20 dB (relative to the total signal) of additional system noise, to be added to the covariance matrix prior to inversion, in order that the covariance matrix be sufficiently well-conditioned for accurate inversion. With this level of injected noise the adaptive array system was found to perform adequately, for jamming signals of strength up to 30 db greater than that of the desired signal.

ACKNOWLEDGEMENT

The contribution of Dr. S. Dutta and Mr. B. Carroll of Miller Communications systems, Ltd., in selecting, adapting, and implementing the matrix inversion algorithm on the APB3024M, is gratefully acknowledged.

REFERENCES

Horowitz, L.L., H. Blatt, W. Brodsky, and K.D. Senne, "Controlling Adaptive Antenna Arrays with the Sample Matrix Inversion Algorithm", IEEE Trans. Aerospace and Electronics, Vol. AES-15, pp 840-847,1979.

Jenkins, R.W., "Computer-Modelling Studies of Small Adaptive Array Systems for HF Communications", CRC Rep. 1364, 1983.

Reed, I.S., J.D. Mallett, and L.E. Brennan, "Rapid Convergence Rate in Adaptive Arrays", IEEE Trans. Aerospace and Electronics, Vol. AES-10, pp 853-863, 1974.

Widrow, B., P.E. Mantey, L.J. Griffiths, and B.B. Goode, "Adaptive Antenna Systems", Proc. IEEE, Vol. 55, pp 2143-2159, 1967.

Williams, P.W., "Numerical Computation", Ch. 4, Nelson, 1972.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA – R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATING ACTIVITY COMMUNICATIONS RESEARCH CENTRE Department of Communications Ottawa, Ontario K2H 8S2	2a. DOCUMENT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. DOCUMENT TITLE IMPLEMENTING A MATRIX-INVERSION ALGORITHM IN A LIMITED-PRECISION ADAPTIVE ANTENNA ARRAY PROCESSOR	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) CRC Technical Note No. 725	
5. AUTHOR(S) (Last name, first name, middle initial) JENKINS, Robert W.	

6. DOCUMENT DATE December, 1984	7a. TOTAL NO. OF PAGES 18	7b. NO. OF REFS 5
8a. PROJECT OR GRANT NO. 0885	9a. ORIGINATOR'S DOCUMENT NUMBER(S) TN 725	
8b. CONTRACT NO.	9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)	
10. DISTRIBUTION STATEMENT Unlimited		
11. SUPPLEMENTARY NOTES	12. SPONSORING ACTIVITY Defence Research Establishment Ottawa	

13. ABSTRACT

The effect of computation accuracy on the performance of a small adaptive antenna system using a digitally-implemented matrix inversion algorithm is examined. A single-board array processor with 16-bit mantissa floating-point accuracy is compared with a VAX-11/750 computer having 24-bit accuracy in the single-precision mode, and 56-bit accuracy in the double-precision mode. The effects of limited precision on array performance are worse when the signal covariance matrix to be inverted is ill-conditioned(i.e., nearly singular), as is the case with an n-element array in the presence of less than n unrelated signals and a low system noise level. Using a simulated signal environment consisting of a wanted communications signal, a single jamming signal, and a variable system noise level, the performance of the array is evaluated. A need for about -20 dB of artificially-injected system noise (relative to the total signal) is demonstrated in the case of the single-board array processor, in order that the covariance matrix be sufficiently well-conditioned for adequate array performance.

DSIS

77-065

UNCLASSIFIED

Security Classification

KEY WORDS

Adaptive Antenna
Matrix Inversion Algorithm

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP: Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER: If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S): Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this document from their defence documentation center."

(2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

The length of the abstract should be limited to 20 single-spaced standard typewritten lines; 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.