NASA-CR-192887

The Telecommunications and Data Acquisition Progress Report 42-112

October–December 1992

E. C. Posner
Editor

February 15, 1993

NASA
National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The Telecommunications and Data Acquisition Progress Report 42-112

October–December 1992

E. C. Posner
Editor

February 15, 1993

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

PRIMARY

Preface

This quarterly publication provides archival reports on developments in programs managed by JPL's Office of Telecommunications and Data Acquisition (TDA). In space communications, radio navigation, radio science, and ground-based radio and radar astronomy, it reports on activities of the Deep Space Network (DSN) in planning, supporting research and technology, implementation, and operations. Also included are standards activity at JPL for space data and information systems and reimbursable DSN work performed for other space agencies through NASA. The preceding work is all performed for NASA's Office of Space Communications (OSC). The TDA Office also peforms work funded by two other NASA program offices through and with the cooperation of OSC. These are the Orbital Debris Radar Program (with the Office of Space Systems Development) and 21st Century Communication Studies (with the Office of Exploration).

For the High-Resolution Microwave Survey (HRMS), *The TDA Progress Report* reports on implementation and operations for searching the microwave spectrum. In solar system radar, it reports on the uses of the Goldstone Solar System Radar for scientific exploration of the planets, their rings and satellites, asteroids, and comets. In radio astronomy, the areas of support include spectroscopy, very long baseline interferometry, and astrometry. These three programs are performed for NASA's Office of Space Science and Applications (OSSA), with OSC funding DSN operational support.

Finally, tasks funded under the JPL Director's Discretionary Fund and the Caltech President's Fund that involve the TDA Office are included.

This and each succeeding issue of *The TDA Progress Report* will present material in some, but not necessarily all, of the following categories:

OSC Tasks:
- DSN Advanced Systems
  - Tracking and Ground-Based Navigation
  - Communications, Spacecraft–Ground
  - Station Control and System Technology
  - Network Data Processing and Productivity
- DSN Systems Implementation
  - Capabilities for Existing Projects
  - Capabilities for New Projects
  - New Initiatives
  - Network Upgrade and Sustaining
- DSN Operations
  - Network Operations and Operations Support
  - Mission Interface and Support
  - TDA Program Management and Analysis
- Ground Communications Implementation and Operations
- Data and Information Systems
- Flight–Ground Advanced Engineering
- Long-Range Program Planning

OSC Cooperative Tasks:
- Orbital Debris Radar Program
- 21st Century Communication Studies

Contents

DSN Operations
FLIGHT-GROUND ADVANCED ENGINEERING

TDA PROGRAM MANAGEMENT AND ANALYSIS

Feasibility Study of a Ka-/Ka-Band Dichroic Plate With Stepped Rectangular Apertures

J. C. Chen
Ground Antennas and Facilities Engineering Section

For the Cassini spacecraft mission, a dichroic plate is needed to pass Ka-band uplink (34.2 to 34.7 GHz) and to reflect Ka-band downlink (31.8 to 32.3 GHz) for dual-frequency operation in the Deep Space Network. The special characteristic of the Ka-/Ka-band dichroic plate is that the passband and the reflective band are only 1.9 GHz (5.7 percent) apart. A thick dichroic plate with stepped rectangular apertures that function as resonator filters was chosen for the Ka-/Ka-band dichroic plate design. The results of the feasibility study are presented in this article.

I. Introduction

For the existing dichroic plates in the Deep Space Network (DSN), the passband frequency is approximately four times higher than the reflecting-band frequency. For example, the S-/X-band (2.2-GHz/8.45-GHz) dichroic plate reflects 2.2 GHz and passes 8.45 GHz, and the X-/Ka-band (8.45-GHz/32.0-GHz) dichroic plate reflects 8.45 GHz and passes 32.0 GHz. In these cases the sizes of the apertures are large enough to pass the higher frequency band, yet small relative to the wavelength of the reflective frequency, resulting in efficient reflection. However, if the passband and reflective band are close [as they are in the Ka-/Ka-band (31.8- to 32.3-GHz/34.2- to 34.7-GHz) dichroic plate], the apertures on the dichroic plate may be large enough to pass the higher frequency passband, but may not be small enough to cut off the lower frequency adequately. A dichroic plate with straight rectangular apertures may not satisfy this requirement. Therefore, a dichroic plate with stepped rectangular apertures, which act as filters, was chosen for the Ka-/Ka-band dichroic plate feasibility study.

II. Analysis

It is assumed that the dichroic plate is illuminated by linearly polarized plane waves with oblique incidence, in this case 30 deg from normal. The analysis of the dichroic plate with stepped rectangular apertures is based on a modal matching method. The electromagnetic field in the free-space region is represented by Floquet modes and the field in the waveguide region is represented by rectangular waveguide modes. By applying the boundary condition at the junctions and using the modal matching method, the scattering matrix of the dichroic plate is then achieved [1]. Two computer programs, the Thick Frequency Selective Surface With Rectangular Apertures

program[1] and the Rectangular Waveguide program,[2] were integrated in order to analyze the dichroic plate with stepped rectangular apertures. The Thick Frequency Selective Surface With Rectangular Aperture program calculates the scattering matrix of the free space and waveguide junction, and the rectangular waveguide program computes the scattering matrix of the stepped waveguide region. These matrices are then combined to form the scattering matrix of the dichroic plate with stepped rectangular apertures. The computer code, the Thick Frequency Selective Surface With Stepped Rectangular Apertures program, was used to design a Ka-/Ka-band dichroic plate. Because multiple design parameters were involved, an optimization program was utilized to speed up the design process.

The number of modes representing the electromagnetic fields on either side of the junction has to be sufficient to ensure the convergence in the modal matching method [2]. There is an equation in the program in which the number of Floquet modes used increases as the number of waveguide modes increases. The speed of the convergence versus the number of waveguide modes may vary depending on the structure of the stepped waveguide. For a straight rectangular aperture, a total of 40 rectangular waveguide modes is sufficient [3], but in this particular design, a total of 180 rectangular waveguide modes was necessary. Figure 1 shows how the transmission coefficient at 34.7 GHz fluctuates as the number of waveguide modes varies. The highest waveguide number indicates that the waveguide numbers M and N are not greater than that number. For example, if the highest waveguide number is equal to one, the TE_{01}, TE_{10}, TE_{11}, and TM_{11} rectangular waveguide modes are used to expand the electromagnetic field in the waveguide region. Figure 1 indicates that the convergence is achieved when the highest waveguide numbers reach nine (180 waveguide modes) or higher.

III. Design and Theoretical Performance

The Ka-/Ka-band dichroic plate design is an array of stepped rectangular apertures on a 5.6388-mm grid with a 60-deg skew angle. Each stepped aperture consists of five steps (Fig. 2). The stepped aperture is a length of rectangular waveguide ($A_1 \times B_1 = 4.6126$ mm $\times$ 4.6228 mm) with two thin irises ($A_2 \times B_2 = 3.81$ mm $\times$ 3.9268 mm, thickness $I = 0.508$ mm) inside. The irises divide the waveguide into three sections with lengths of 9.8044 mm (L_1), 9.144 mm (L_2), and 9.8044 mm. The first and the last sections act as resonators with low quality factor Q, while the center section acts as a resonator with high Q. The size of the main waveguide aperture is smaller than the cutoff size of the highest frequency at the downlink and larger than the cutoff size of the lowest frequency of the uplink. The stepped aperture is basically a resonator filter to pass the uplink and reflect the downlink. The advantage of having only two different waveguide sizes is to reduce the fabrication costs. Since multiple metal sheets can be stacked together to be wire-electrical-discharge machined for an identical pattern, only two sets of sheets need to be run through the machine.

Theoretical performance of the Ka-/Ka-band dichroic plate design was calculated and is shown in Table 1. The transmission of the Ka-band downlink is about -36 to -48 dB over the bandwidth. The downlink is almost totally reflected, but there is a relative phase shift of 7.26 $\pm$0.77 deg over the bandwidth between two orthogonal linear polarizations (TE and TM polarizations). The phase shift between two polarizations can be compensated for by the polarizer, which is connected to the horn in the microwave system. The transmission of the passband (uplink) is better than -0.19 dB over the bandwidth for both TE polarization and TM polarization. The relative phase shift of the passband is 9.45 $\pm$1.00 deg over the Ka-band uplink. The curve of the magnitude of the transmission coefficients for TE and TM polarizations from 31 to 36 GHz shows nice roll-off from reflective band to passband (Fig. 3). The transmission coefficient over the passband is shown in more detail in Fig. 4. The relative phase shift between TE and TM polarizations varies less than $\pm$1.00 deg in both downlink and uplink (Fig. 5). Since the incident wave in the Deep Space Network is circularly polarized (CP), the transmitted CP power is about 0.01 to 0.15 dB with ellipticity of 1.3 to 1.65 dB over the passband, and total reflected power is below -37 dB with ellipticity of 2.09 dB (Figs. 6 and 7).

A tolerance study was also done over the downlink and uplink bandwidths (Table 2). The large aperture dimensions (A_1 and B_1), small aperture dimensions (A_2 and B_2), length of each step (L_1 and L_2), and thickness of the iris-like step were examined in the tolerance study. One dimension was varied by $\pm$0.0127 mm (or $\pm$0.0254 mm) at a time, while the rest of the dimensions were kept at the design size. The last row in Table 2 is the performance at the design dimensions without any variation. For example, the transmission coefficient for TM polarization

[1] The Thick Frequency Selective Surface With Rectangular Aperture program was written by J. C. Chen, Jet Propulsion Laboratory, Pasadena, California, and was submitted to COSMIC, University of Georgia, 382 E. Broad St., Athens, Georgia 30602.

[2] The Rectangular Waveguide Program was written by D. J. Hoppe, Radio Frequency and Microwave Subsystems Section, and F. Manshadi, Ground Antennas and Facilities Engineering Section, Jet Propulsion Laboratory, Pasadena, California.

changes from 0.10 ±0.09 (tenth row, seventh column) to 0.21 ±0.2 dB (fourth row, seventh column) over the bandwidth when B_1 is decreased by 0.0127 mm. The performance of the dichroic plate is more sensitive to changes in the dimensions of the large apertures than the small apertures. Changing the length of the apertures has little effect on the performance of both uplink and downlink. Overall, variation in the dimensions of the dichroic plate with stepped apertures has less impact on the performance of the downlink than the uplink.

IV. Conclusion

The analysis of the performance of a dichroic plate with stepped apertures indicates that it is a feasible choice for the Ka-/Ka-band dichroic plate. The stepped aperture is a new technique in dichroic plate design, therefore the fabrication technique and experiment method require further study. Also, the accuracy of the new program needs to be verified. Fabrication of a test Ka-/Ka-band dichroic plate is the next step in understanding its performance.

Acknowledgment

The author thanks P. Stanton for the suggestion of using stepped apertures in the Ka-/Ka-band dichroic plate design.

References

[1] J. C. Chen, "Analysis of a Thick Dichroic Plate With Rectangular Holes at Arbitrary Angle of Incidence," *The Telecommunications and Data Acquisition Progress Report 42-104, vol. October–December 1990*, Jet Propulsion Laboratory, Pasadena, California, pp. 9–16, February 15, 1991.

[2] R. Mittra, T. Itoh, and T. S. Li, "Analytical and Numerical Studies of the Relative Convergence Phenomenon Arising in the Solution of an Integral Equation by the Moment Method," *IEEE Transactions on Microwave Theory and Technique*, vol. MTT-20, no. 2, pp. 96–104, February 1972.

[3] J. C. Chen, "X-/Ka-Band Dichroic Plate Design and Grating Lobe Study," *The Telecommunications and Data Acquisition Progress Report 42-105, vol. January–March 1991*, Jet Propulsion Laboratory, Pasadena, California, pp. 21–30, May 15, 1991.

Table 1. Transmission coefficients for *TE* and *TM* polarizations and relative phase shift between *TE* and *TM* polarizations for a Ka-/Ka-band dichroic plate with stepped rectangular waveguide.

Ka-band	Frequency, GHz	Transmission coefficients, dB		Relative phase shift, deg
		TE	*TM*	
Downlink	31.8	−47.94	−46.24	8.033
	32.0	−44.27	−42.56	7.505
	32.3	−38.28	−36.46	6.496
Uplink	34.2	−0.102	−0.193	10.468
	34.5	−0.013	−0.062	10.495
	34.7	−0.025	−0.168	8.450

Table 2. Tolerance study for Ka-/Ka-band dichroic plate with stepped rectangular apertures.

Dimensions of stepped aperture, mm	Variation,[a] mm	Ka-band downlink (31.8 to 32.3 GHz) reflection[b]			Ka-band uplink (34.2 to 34.7 GHz) transmission[c]		
		TE, dB	*TM*, dB	Δ phase, deg	*TE*, dB	*TM*, dB	Δ phase, deg
A_1, 4.6126	+0.0127	0.00 ±0.00	0.00 ±0.00	4.98 ±1.19	0.03 ±0.02	0.10 ±0.09	1.92 ±0.05
	−0.0127	0.00 ±0.00	0.00 ±0.00	9.43 ±0.39	0.09 ±0.08	0.06 ±0.05	21.12 ±2.05
B_1, 4.6228	+0.0127	0.00 ±0.00	0.00 ±0.00	9.35 ±0.41	0.06 ±0.05	0.08 ±0.08	20.38 ±2.54
	−0.0127	0.00 ±0.00	0.00 ±0.00	5.27 ±1.08	0.06 ±0.04	0.21 ±0.20	0.80 ±0.44
A_2, 3.8100	+0.0127	0.00 ±0.00	0.00 ±0.00	7.23 ±0.78	0.06 ±0.05	0.10 ±0.09	7.99 ±0.06
	−0.0127	0.00 ±0.00	0.00 ±0.00	7.30 ±0.75	0.06 ±0.04	0.10 ±0.09	1.07 ±1.23
B_2, 3.9268	+0.0127	0.00 ±0.00	0.00 ±0.00	7.34 ±0.73	0.06 ±0.03	0.11 ±0.10	11.21 ±1.14
	−0.0127	0.00 ±0.00	0.00 ±0.00	7.22 ±0.79	0.06 ±0.04	0.11 ±0.11	8.19 ±0.81
I, 0.5080	+0.0127	0.00 ±0.00	0.00 ±0.00	7.26 ±0.77	0.06 ±0.04	0.10 ±0.10	9.52 ±1.04
	−0.0127	0.00 ±0.00	0.00 ±0.00	7.27 ±0.77	0.06 ±0.05	0.10 ±0.09	9.40 ±0.98
L_1, 9.8044	+0.0254	0.00 ±0.00	0.00 ±0.00	7.26 ±0.77	0.05 ±0.04	0.09 ±0.08	9.55 ±1.06
	−0.0254	0.00 ±0.00	0.00 ±0.00	7.27 ±0.77	0.06 ±0.05	0.11 ±0.10	9.42 ±0.98
L_2, 9.1440	+0.0254	0.00 ±0.00	0.00 ±0.00	7.27 ±0.77	0.06 ±0.05	0.10 ±0.09	9.39 ±1.01
	−0.0254	0.00 ±0.00	0.00 ±0.00	7.26 ±0.77	0.05 ±0.04	0.10 ±0.09	9.52 ±1.00
	0.000[d]	0.00 ±0.00	0.00 ±0.00	7.26 ±0.77	0.06 ±0.04	0.10 ±0.09	9.45 ±1.00

[a] One dimension of the aperture is varied by ±0.0127 mm (or 0.0254 mm), while the other dimensions of the aperture remain the same size.

[b] The third and fourth columns indicate the magnitude of the reflection coefficients over the downlink bandwidth for TE and TM polarizations, respectively, and the fifth column indicates the relative phase shift between these two polarizations over the downlink bandwidth.

[c] The sixth and seventh columns indicate the magnitude of the transmission coefficients over the uplink bandwidth for TE and TM polarizations respectively, and the eighth column indicates the relative phase shift between these two polarizations over the uplink bandwidth.

[d] This row indicates the performances of the Ka-/Ka-band dichroic plate at the designed dimensions with no variations.

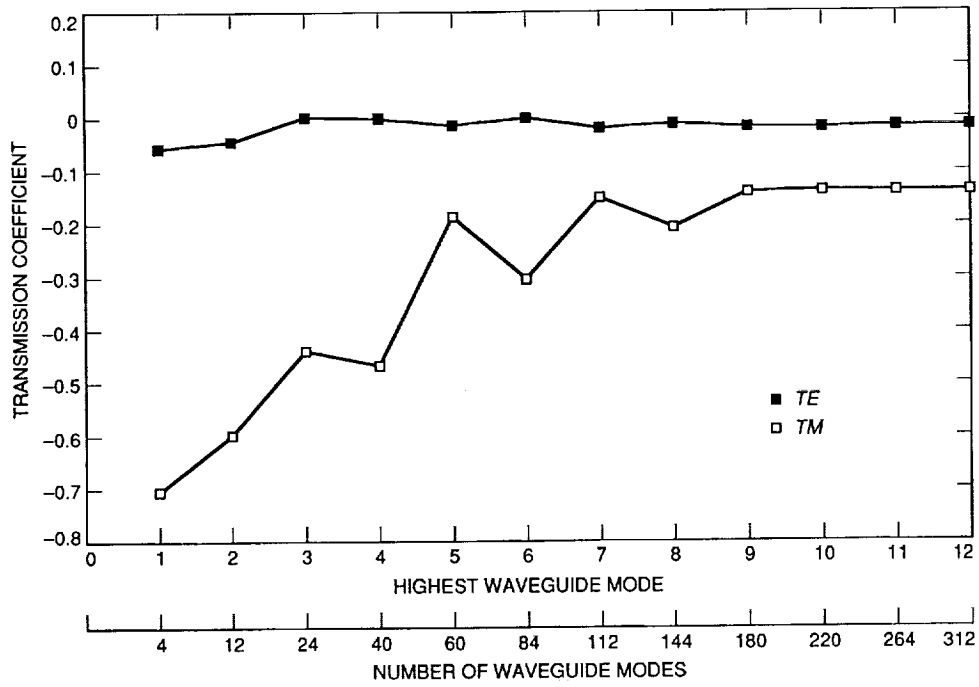


Fig. 1. Convergence of the calculated transmission coefficient versus the number of waveguide modes used at 34.7 GHz.

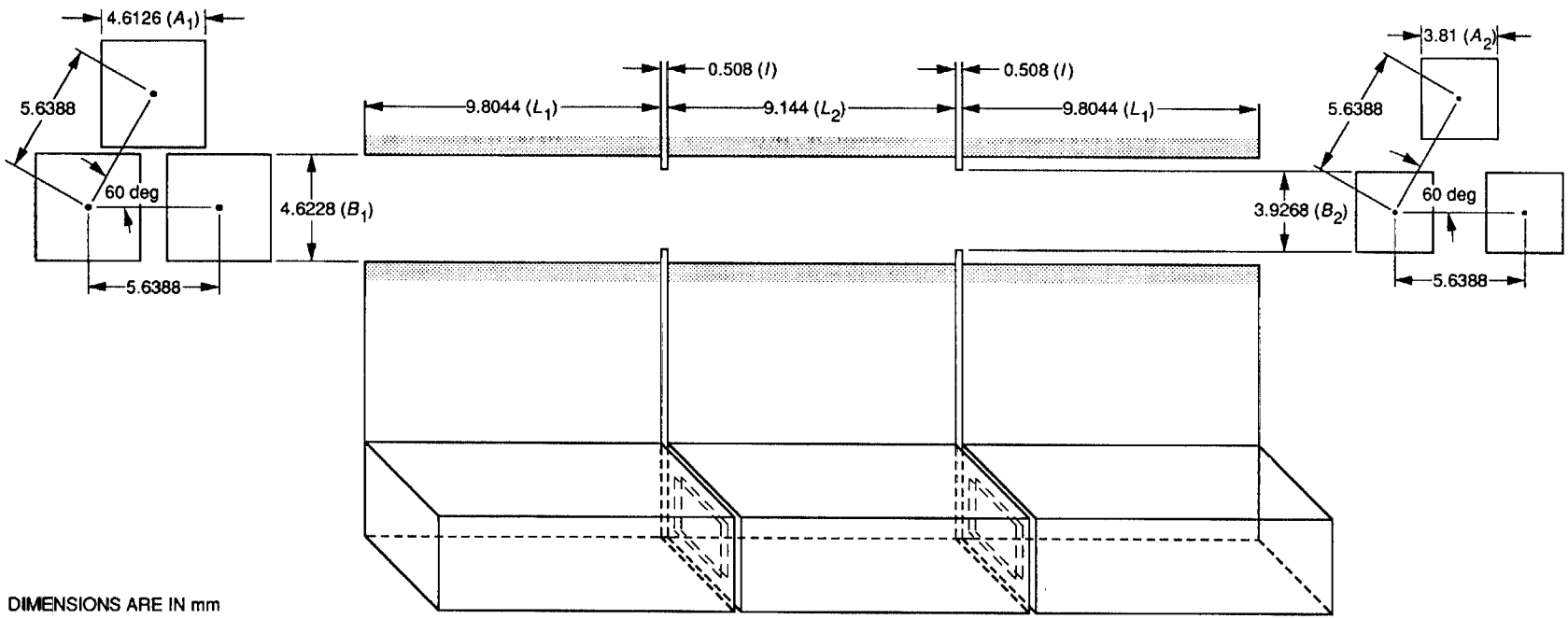


Fig. 2. Geometry of Ka-/Ka-band dichroic plate.

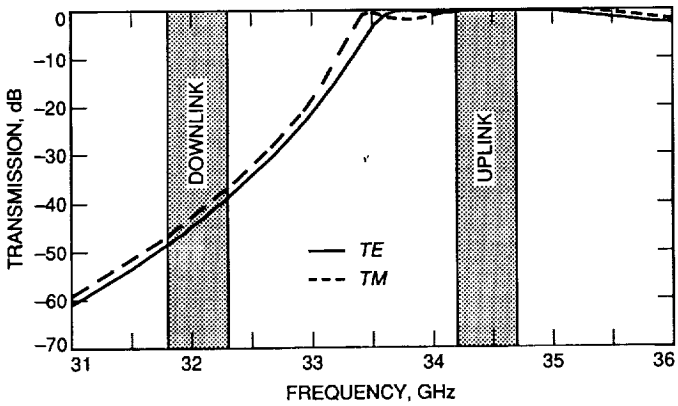


Fig. 3. Transmission coefficient versus frequency for Ka-/Ka-band dichroic plate with stepped rectangular apertures for *TE* and *TM* polarizations.

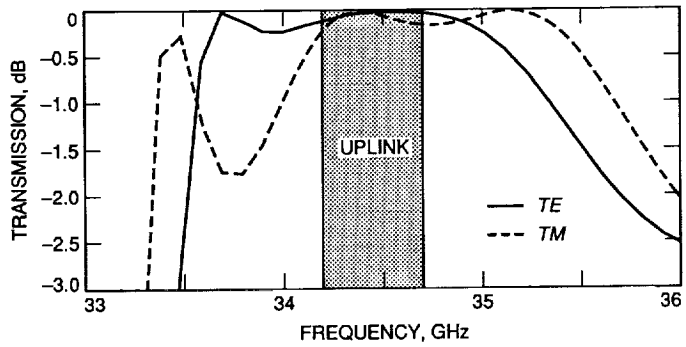


Fig. 4. Transmission coefficient versus frequency for Ka-/Ka-band dichroic plate with stepped rectangular apertures for *TE* and *TM* polarizations at Ka-band uplink.

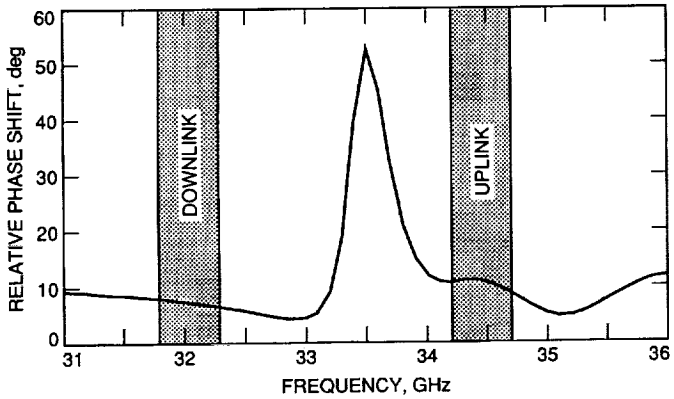


Fig. 5. Relative phase shift versus frequency for Ka-/Ka-band dichroic plate with stepped rectangular apertures.

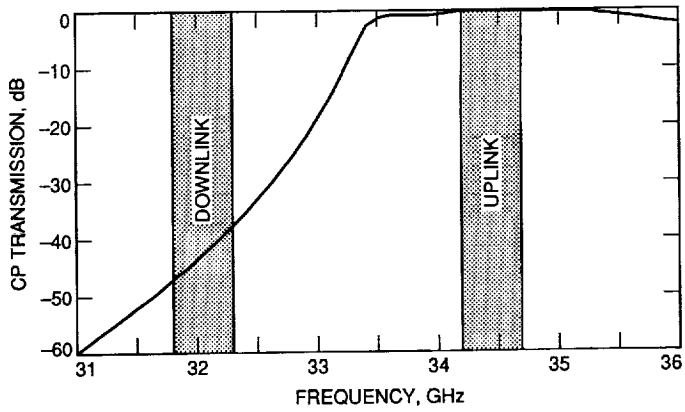


Fig. 6. Transmission power versus frequency for Ka-/Ka-band dichroic plate with stepped rectangular apertures for circularly polarized wave incidence.

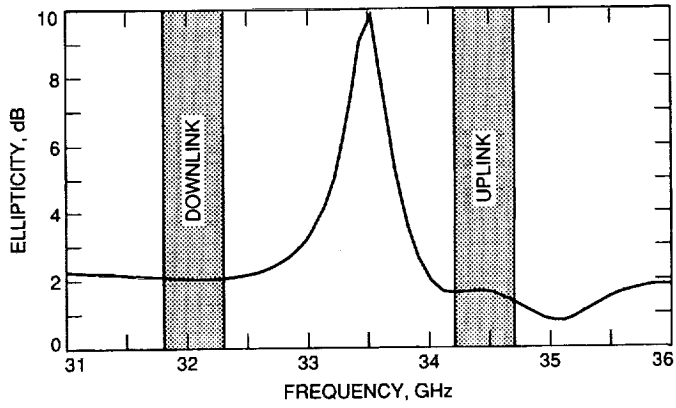


Fig. 7. Ellipticity versus frequency for Ka-/Ka-band dichroic plate with stepped rectangular apertures for circularly polarized wave.

A Technique for Computation of Noise Temperature Due to a Beam Waveguide Shroud

W. Veruttipong and M. M. Franco
Ground Antennas and Facilities Engineering Section

Direct analytical computation of the noise temperature of real beam waveguide (BWG) systems, including all mirrors and the surrounding shroud, is an extremely complex problem and virtually impossible to achieve. Yet the DSN antennas are required to be ultra low-noise in order to be effective, and a reasonably accurate prediction is essential. This article presents a relatively simple technique to compute a real BWG system noise temperature by combining analytical techniques with data from experimental tests. Specific expressions and parameters for X-band (8.45-GHz) BWG noise computation are obtained for DSS 13 and DSS 24, now under construction. These expressions are also valid for various conditions of the BWG feed systems, including horn sizes and positions, and mirror sizes, curvatures, and positions. Parameters for S- and Ka-bands (2.3 and 32.0 GHz) have not been determined; however, those can be obtained following the same procedure as for X-band.

I. Introduction

Noise temperature due to a BWG assembly is one of the major contributers to total antenna receive system noise temperature, especially for an ultra-low-noise system or a system with high spillover power in the BWG shroud. A reasonably accurate prediction of the BWG assembly noise temperature is essential. Direct analytical computation of the noise temperature of real BWG systems, including all mirrors, is an extremely complex problem and virtually impossible to achieve.

This article presents a new technique that combines an analytical approach with data from measurement tests to construct a specific expression with some parameters to compute noise temperature in a real BWG system, including all mirrors. Appendix A gives the needed relationships among reflector material conductivity, resistivity, loss, and resultant noise. Experimental tests that have helped to establish this technique are presented in Appendix B. Appendix C develops the effective noise temperatures, as seen by spillovers in the upper and lower portions of the BWG assembly. Appendix D contains physical explanations of the behavior of the field and noise temperature in the BWG shroud. The results from this technique have been used to predict BWG noise temperatures for DSS-13 Phase II and DSS 24.

II. Characteristics of the Electromagnetic Fields Inside a BWG Shroud

The total RF power originating from a horn aperture (viewed in transmission, for convenience) and propagating

through a BWG shroud (P_{BWG}) can be separated into two parts (see Fig. 1)

$$P_{BWG} = P_m + P_{spill} \quad (1)$$

where P_m is the majority of the total power (usually >95 percent of P_{BWG}) that is always confined inside *all* BWG mirrors; it does not contact the BWG wall and there are no multiple reflection, diffraction, or creeping wave components. The power P_m reflects from each and every mirror once before leaving the BWG shroud through the opening near F_1. P_m can be computed easily because BWG wall and mirror interactions are small and ignored, which drastically simplifies the problem. The power that is confined inside the lower mirrors M6 (or M5, M4, etc.) is not considered as P_m because leaks occur along the way up from M6 to M1. For most practical cases where the total spillover loss is less than 5 percent, the total power confined inside the last mirror M1 is approximately used as P_m.

P_{spill} is the sum of spillover powers of each mirror. The P_{spill} fraction creeps and bounces around the BWG walls, mirrors, brackets (behind the mirrors), and the edges, etc., and suffers dissipation loss and consequent noise. On an average, the P_{spill} power largely dissipates before a small remainder exits the BWG opening near F_1 (some power may get back into the horn, but most is dissipated, with consequent transformation into noise). Even though P_{spill} can be computed accurately ($P_{spill} = P_{BWG} - P_m$), its field distribution and its chaotic behavior inside the lossy BWG shroud is virtually impossible to compute analytically.

III. Noise Temperature Contributions Inside the BWG Shroud

From Eq. (1), the corresponding noise temperatures are

$$T_{BWG} = T_m + T_{spill} \quad (2)$$

where T_{BWG} is the total noise temperature (in kelvins) due to the BWG (including the shroud, mirrors, brackets, etc.). The values T_m and T_{spill} are the noise temperature contributions from P_m and P_{spill}, respectively. Because of the simplicity of P_m, its corresponding noise temperature T_m can be computed with acceptable accuracy. The details are given in Appendix A. From Appendix A [Eq. (A-6)], the noise temperature T_m for DSS 13 or DSS 24 in kelvins at X-band is

$$T_m = 0.734 P_m / P_{BWG} = 0.734 \alpha_m \quad (3)$$

where α_m is the P_m fraction of P_{BWG}, dimensionless.

Similarly, because of the complexity of the field that contributes to P_{spill}, it is not possible to analytically compute T_{spill}. Unfortunately, T_{spill} is usually a major contribution to T_{BWG} (for example, DSS-13 Phase II, X-band $T_{spill} \sim 3.0$ K and $T_m \sim 0.7$ K) and a reasonably accurate prediction is essential. The following section shows a technique used to compute T_{spill}.

IV. A Technique for T_{spill} Computation

Noise temperature due to spillover power P_{spill} is given in a very simple form as

$$T_{spill} = (P_1/P_{BWG})T_1 + (P_2/P_{BWG})T_2 = \alpha_1 T_1 + \alpha_2 T_2 \quad (4)$$

where P_1 is the total spillover power of the two mirrors (M5 and M6) in the basement and the value P_2 is the total spillover power of the four mirrors (M1, M2, M3, and M4) above the basement ceiling. The values α_1 and α_2 are the normalized powers (with respect to P_{BWG}) of P_1 and P_2, respectively. The values T_1 and T_2 are source (or sink) effective noise temperatures seen by P_1 and P_2, in kelvins, respectively. In other words, T_1 and T_2 are effective noise temperatures that α_1 and α_2 see in the BWG assembly (including the shroud, mirrors, brackets, etc.). For example, if the BWG shroud were very short and highly conducting, T_1 and T_2 would be equal and approximately T_{sky}. If the shroud were very long and lossy, T_1 and T_2 would again be equal and approximately $273.16 + T_{celsius}$ (~290 K).

If one substitutes Eqs. (3) and (4) into Eq. (2), the BWG shroud noise temperature at X-band becomes

$$T_{BWG} = 0.734 \alpha_m + \alpha_1 T_1 + \alpha_2 T_2 \quad (5)$$

It is noted that $\alpha_m + \alpha_1 + \alpha_2 = 1.0$.

The value T_{BWG} was measured at DSS 13. The power fractions α_m, α_1, and α_2 in Eq. (5) are known and the unknown quantities are T_1 and T_2. By performing various measurements and perturbations at X-band, the coefficients T_1 and T_2 at X-band have been obtained (see details in Appendices B and C).

DSS 13, no basement shroud:

$$T_1 = 300 \pm 10 \text{ K}$$
$$T_2 = 240 \pm 45 \text{ K} \quad \text{(6a)}$$

DSS 24, full shroud:

$$T_1 = 280 \pm 20 \text{ K}$$
$$T_2 = 230 \pm 45 \text{ K} \quad \text{(6b)}$$

It is noted that Eqs. (6a) and (6b) both have steel shrouds (the same conductivity as DSS 13) above the basement ceiling, while DSS 24 has an additional large aluminum shroud in the basement.

In general, the coefficients (T_1, T_2) apply to any BWG antenna with a similar shroud structure and (nearly) the same conductivity. The same coefficients are valid for various conditions in a BWG system, including the horn sizes and positions, and mirror sizes, curvatures, and positions. See Appendix D for more physical explanations about the behavior of field and noise temperature due to the BWG assembly.

The effective temperatures T_1 and T_2 for S- and Ka-bands are expected to be slightly smaller and larger than the X-band coefficients, respectively. The accurate values of the S- and Ka-bands coefficients have not been determined.

V. Conclusion

A technique for computing a BWG assembly noise temperature has been established and the technique can now be applied to the low-noise DSN BWG antennas. Specific expressions and parameters for X-band BWG noise temperature computation of DSS 13 and DSS 24 are obtained from Eqs. (5) and (6). These expressions are valid for various conditions of the BWG feed system, including the horn sizes and positions, and the mirror sizes, curvatures, and positions. The coefficients in Eq. (6) are not sensitive to small variations of BWG shroud structure (diameter, length, etc.), shroud conductivity, or operating frequency. Parameters for S- and Ka-bands have not been determined; however, they will not be dramatically different and can be obtained by following the same procedure as for X-band. The BWG noise temperature computed by this technique should be, at best, as accurate as a measurement result (plus some small error for the P_{spill} computation) and should be valid for all engineering applications.

Acknowledgments

The authors wish to thank T. Y. Otoshi for initiating an experimental test of DSS 13. The test enabled a better understanding of the problems of noise temperature due to a BWG assembly and provided valuable data used in this article. Thanks also to D. A. Bathker for his help with Appendix A and to the personnel of DSS 13 for their support.

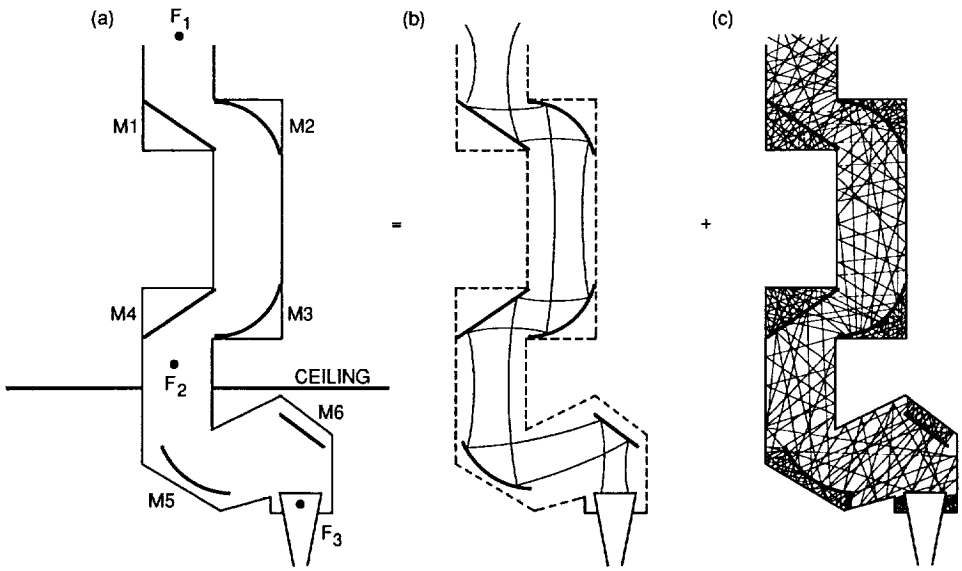


Fig. 1. Characteristics of the fields inside a BWG shroud and their corresponding noise temperatures.

Appendix A

Noise Temperature Due to BWG Mirror Surface Resistivity

Noise temperature due to one reflection of a wave normally incident on a mirror surface is[1]

$$T \sim \frac{4R_s}{Z_o} T_p \alpha, \text{ kelvins} \tag{A-1}$$

$$R_s = 0.02\pi\sqrt{\frac{f}{10\sigma_N}}, \text{ ohms/square} \tag{A-2}$$

where

R_s = surface resistivity of the mirror, ohms/square

f = frequency, GHz

σ_N = normalized conductivity = $\sigma \times 10^{-7}$, mho/meter

Z_o = free-space wave impedance, ohms

T_p = physical temperature, kelvins

α = power fraction illuminating the mirror

For a circularly polarized wave with an average incident angle to the mirror equal to θ, an approximate expression for T, valid where θ is away from grazing incidence, is

$$T \sim \frac{4R_s}{Z_o} T_p \alpha \frac{1}{2}\left(\cos\theta + \frac{1}{\cos\theta}\right) \tag{A-3}$$

The total noise temperature of n mirrors is

$$T = \frac{2T_p}{Z_o} \sum_{i=1}^{n} R_{si}\alpha_i \left(\cos\theta_i + \frac{1}{\cos\theta_i}\right) \tag{A-4}$$

where R_{si}, α_i, and θ_i are R_s, α, and θ of the ith mirror, respectively.

[1] D. A. Bathker, "Planewave Reflection Noise Temperature Due to Surface Resistivity," JPL Interoffice Memorandum 3328-92-0144 (internal document), Jet Propulsion Laboratory, Pasadena, California, August 20, 1992.

In order to compute T_m [as shown in Eq. (2) for DSS 13 or DSS 24], the noise temperature in Eq. (A-4) can be used with the following conditions:

(1) The surface conductivity is the same for all six mirrors with $\sigma = 2.3 \times 10^7$ mho/meter (for 6061T6 aluminum).

(2) The power fraction is also the same for all mirrors and equal to α_m, where α_m is the main fraction of $P_{BWG}(\alpha_m = P_m/P_{BWG})$.

(3) The incident angle is $\theta = 45$ deg for the four mirrors (M1, M2, M3, and M4) above the ceiling and $\theta = 30$ deg for the two basement mirrors (M5 and M6).

(4) The physical temperature $T_p = 290$ K.

(5) The frequency $f = 8.45$ GHz will be used because all measured data in Appendix C are at 8.45 GHz.

From Eq. (A-4) and the above five conditions, the total noise temperature due to surface resistivity of all six mirrors of DSS 13 or DSS 24 becomes

$$T_m = \frac{2T_p R_s \alpha_m}{Z_o}\left\{2\left(\cos 30 + \frac{1}{\cos 30}\right) + 4\left(\cos 45 + \frac{1}{\cos 45}\right)\right\} \tag{A-5}$$

Substitute $T_p = 290, Z_o = 120\pi$, $f = 8.45$, and $\sigma_N = 2.3$ into Eqs. (A-2) and (A-5), and one obtains

$$T_m = 0.734\alpha_m \tag{A-6}$$

Note that in a reasonable design with $\alpha_m = 0.95$ or more, the dissipation of six aluminum BWG mirrors alone causes a noise component of about 0.7 kelvin at X-band.

Appendix B

BWG Noise Temperature Experimental Tests

T. Y. Otoshi and M. M. Franco performed several measurements at DSS 13 in order to gain a better understanding of the BWG assembly noise temperature. The results show a higher than (initially) expected noise temperature (which was about 4 K). By adding skirts around the rim of the BWG mirrors in these tests, the noise temperature was reduced only a fraction of a kelvin. The two mirrors in the basement normally do not have shrouds. After constructing a temporary shroud (made with lightweight insulating material, aluminum foil–coated on two sides, and a foam center) that partially enclosed the two mirrors, the noise temperature dropped only 0.25 K.

The above results led to further investigation by W. Veruttipong and M. M. Franco. The experimental tests described below (performed by Veruttipong and Franco between May 31 and June 3, 1991) help to explain why the BWG noise temperature is higher than initially expected. More important, their results led to a new technique used to compute the BWG noise temperature.

Experiment 1. The purpose of this experiment was to observe the effect of the basement shroud on system noise temperature.

A tightly fitted and fully enclosed basement shroud was made from a partially enclosed shroud left over from Otoshi and Franco's experiment (see Fig. B-1). The same foil material was used to patch all the opening spaces. Aluminum tape was used to seal all small gaps. J. Withington built an RF probe to check for any RF signal leaks around the new shroud. The JPL standard 22-dB X-band corrugated horn was used, and its aperture was positioned at 3.34 in. above F_3 throughout the experiments. The following system noise temperatures were recorded:

Fully enclosed basement shroud

$$T_{op} = 33.62 \text{ K}$$

Partially enclosed basement shroud $\Big\rangle \Delta = 0.23$ K

$$T_{op} = 33.85 \text{ K}$$

No basement shroud $\Big\rangle \Delta = 0.25$ K

$$T_{op} = 34.10 \text{ K}$$

Conclusion 1. The results clearly show that the basement shroud does not significantly reduce the BWG noise temperature. In other words, the spillover field in the basement still sees a high effective noise temperature (close to 300 K) with or without the basement shroud.

Experiment 2. The purpose of this experiment was to reconfirm the results from Experiment 1 that spillover power sees a high effective noise temperature in the BWG shroud.

The main field inside the BWG shroud (P_m) was excited so that it had a characteristic similar to P_{spill} in Fig. 1. By causing mirrors M5 and M6 to be rippled (by putting curly aluminum tape all over the mirror surfaces), the field inside the shroud was obviously chaotic and similar to Fig. E-1(c). The tightly fitted basement shroud was still in place and the noise temperature reading was 235.5 K.

Conclusion 2. The simulated scattered field is not quickly guided by the shroud to the sky. In fact, it bounces around and effectively creates a high dissipation loss. The 235.5 K does not represent any specific parameter. It is just an indicator showing that the simulated scattered field (or spillover field in the real BWG) sees an effective noise temperature on the order of 200 K. The accurate value is computed in Appendix C. Earlier in the experiment, it was not fully understood that the effective noise temperature for spilled or scattered power was approximately T_{sky}, and that is why the measured noise temperature was higher than expected.

Experiment 3. The purpose of this test was to see whether there would be any unusual noise temperature measurements when the shroud was short-circuited at various locations. The intent was to see whether the noise temperature readings were high (300 K $\pm\Delta$), medium (150 K $\pm\Delta$), or low (50 K $\pm\Delta$). Therefore, a very high-precision experimental setup was not necessary.

The short circuit was done by closing the BWG tube with a foam-backed foil disk. Five of the six mirror surfaces were intentionally rippled (see Experiment 2) and the noise temperature readings were

Short circuit near F_1: $T_{op,1} = 296$ K

Short circuit at the ceiling: $T_{op,2} = 304$ K

Short circuit just above the cement: $T_{op,3}$ = 304 K

Short circuit near F_1 (all mirror surfaces are good): $T_{op,4}$ = 131 K

Conclusion 3. All readings (except $T_{op,4}$) are 300 K ±5 K, as expected, because a perfectly closed cavity has an ambient noise temperature. The slightly lower reading of $T_{op,1}$ results from an imperfect short circuit near F_1 due to a relatively strong wind. It is noted that if all the mirror surfaces are good during the measurement, the noise temperatures will differ greatly, depending upon the location of the short circuit. For example, if the short circuit is exactly at the waist of the beam (near F_1 or F_2), the noise temperature will be low (less than 100 K) because most of the energy will get back to the horn. The noise temperature $T_{op,4}$ is not less than 100 K because the short-circuit location is not at the waist of the field and the short-circuit disk is neither perfectly level nor flat.

Experiment 4. The purpose of this test was to measure the effect of a section of concrete in the shroud (see the location of the concrete in Fig. B-1).

To investigate the impact of a concrete surface segment (unlined shroud portion) on the BWG noise temperature, the BWG system was returned to its normal operational condition, i.e., no rippled surfaces, no short circuit, and no basement shroud. One-inch foam-backed foil was used to cover the concrete surface. System noise temperatures were recorded with and without the covering. The result shows practically no change in the noise temperature.

Conclusion 4. The concrete in the BWG shroud is not a major source of BWG shroud noise and the concrete does not create extra noise temperature in the BWG system. This was expected, as the concrete section is near a beam waist.

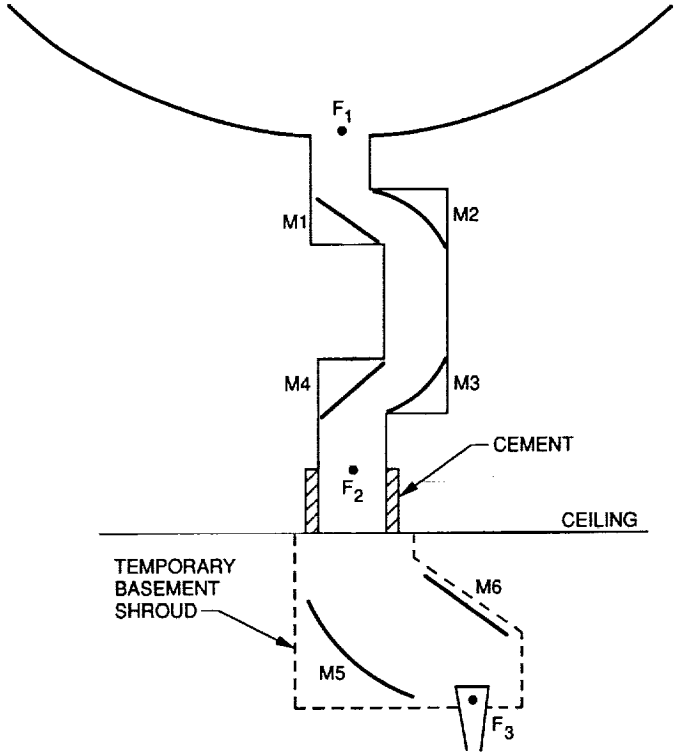


Fig. B-1. DSS-13 BWG antenna with a temporary basement shroud.

Appendix C

Effective Noise Temperatures (T_1 and T_2)

The noise temperature coefficients or effective noise temperatures, as shown in Eq. (4), will be evaluated at X-band for DSS 13 and DSS 24. Recall Eq. (5),

$$T_{\mathrm{BWG}} = 0.734\alpha_{\mathrm{m}} + \alpha_1 T_1 + \alpha_2 T_2$$

The coefficients for DSS 13 will be considered first. The measurement at DSS 13 reveals $T_{\mathrm{BWG}} = 8.9 \pm 0.4$ K [1]. The spillover power of every mirror for DSS 13 is computed, without the BWG wall, by using Physical Optics (PO) software, and the power is normalized to the total horn output power. The results are

$$\begin{aligned} \alpha_{\mathrm{m}} &= 0.9694 \\ \alpha_1 &= 0.0138 \\ \alpha_2 &= 0.0168 \end{aligned} \qquad \text{(C-1)}$$

Since there is no shroud in the basement of DSS 13, α_1 is mostly dissipated in the basement, therefore,

$$T_1 = 300 \text{ K} \qquad \text{(C-2)}$$

Substitute $T_{\mathrm{BWG}} = 8.9$ K, Eqs. (C-1) and (C-2) into Eq. (5), and one obtains

$$T_2 \sim 240 \text{ K} \qquad \text{(C-3)}$$

For the deviation of T_{BWG} of ± 0.4 K and the spillover power within ± 0.0005 for α_1 and ± 0.0010 for α_2, the deviation of T_2 becomes ± 45 K. It is noted that the value of $T_1 = 300$ K is quite well known and is an accurate value (a deviation of ± 10 K may be applicable). Therefore, the accuracy of T_2 is nearly as good as the measurement can get (plus a small error in PO computation).

Noise temperature coefficients for DSS 24 will be considered next. From Experiment 1 in Appendix B, the difference in noise temperatures of DSS 13 with and without a basement shroud is about 0.5 K. DSS 24 has a larger basement shroud than the one in the test (Fig. B-1), which results in a slightly higher system noise temperature (about 0.1 to 0.3 K higher). However, the smaller gap between mirrors and the BWG wall of DSS 24 helps to reduce the noise temperature by about 0.2 K. On balance, it is reasonable to say that the result from DSS 13 with the shroud should approximately represent DSS 24. Recall that with the temporary basement shroud, the DSS-13 noise temperature decreased by 0.5 K. One can assume that 0.3 K is related to α_1 and 0.2 K is related to α_2 (gap effect included). Even though this assumption of the ratio contribution may be questionable, the total noise temperature difference of 0.5 K is always preserved.

From Eqs. (5) and (C-1), one can write

$$\begin{aligned} \Delta T_1 &= \frac{0.3}{\alpha_1} = \frac{0.3}{0.0138} \sim 20 \text{ K} \\ \Delta T_2 &= \frac{0.2}{\alpha_2} = \frac{0.2}{0.0168} \sim 10 \text{ K} \end{aligned} \qquad \text{(C-4)}$$

From Eqs. (C-2), (C-3), and (C-4), the coefficients for DSS 24 are

$$\begin{aligned} T_1 &= 300 - 20 = 280 \text{ K} \\ T_2 &= 240 - 10 = 230 \text{ K} \end{aligned} \qquad \text{(C-5)}$$

The deviation of T_1 for DSS 24 should be increased to ± 20 K due to a small uncertainty ($\sim \pm 0.1$ K) of 0.5-K power distribution. The deviation of T_2 for DSS 24 should be approximately the same as DSS 13, which is about ± 45 K.

Reference

[1] T. Y. Otoshi, S. R. Stewart, and M. M. Franco, "Portable Microwave Test Packages for BWG Antenna Performance Evaluations," *IEEE Trans. Microwave Theory Tech.*, vol. 40, no. 6, June 1992.

Appendix D

Field and Noise Temperature Due to a BWG Shroud

The spillover power past a mirror is defined here as a power originating from a single source that is not incident on the mirror in the absence of a BWG wall and other mirrors. In reality, with BWG walls and all mirrors included, the spillover power of each mirror will spread (bounce around) nonuniformly throughout the shroud. The dissipation losses are associated with the conductivities of the walls, mirrors, brackets, etc., in a very complex and undetermined combination. Total spillover noise temperature is the sum of the contributions of all the mirrors. Besides the complexities of the scattered fields related to the mirror brackets, edges, bolts and nuts, welding scars, gaps between mirrors and BWG walls, etc., one must also deal with a partially open cavity problem of the lossy BWG shroud. Based on over five years' experience in fundamental research with reflection, diffraction, and creeping waves of various surfaces and objects, it is concluded that the characteristic of P_{spill} (and T_{spill}) is so chaotic in a very lossy and complex system that it is not possible to compute analytically.

One can draw a similarity between the expression $T_{\text{spill}} = \alpha_1 T_1 + \alpha_2 T_2$ in Eq. (4) and Ohm's law $V = IR$. The resistance R (in ohms) is a result of thermal agitation of electrons, which prevents current (free electrons) from moving freely through the resistor R. The relation $V = IR$ applies for a macroscopic region in an electric circuit, and R is obtained from a relation $R = V/I$. It is extremely difficult to compute R from a microscopic point of view by computing the total energy loss from continual collisions of electrons or by determining an average rate of flow of free electrons through the thermally excited lattice structure of the resistor. The resistance R is linear and isotropic within a wide range of I (similary, T_1 and T_2 are linear and perhaps isotropic). The linearity of T_1 (similarly for T_2) in Eq. (4) physically means that the behavior (or the degree of chaos) of α_1 is approximately the same, regardless of the magnitude of α_1 (as long as there is no dielectric breakdown).

The reason why T_{spill} in Eq. (4) is separated into two parts (above and below the subterranean room ceiling) is that the shroud in the basement is very different from the shroud above. In fact, DSS 13 does not have a basement shroud at all. Fortunately, this helps to improve the accuracy of Eq. (6a) because $T_1 = 300$ K is an accurate effective noise temperature for a very large basement room, provided that the exit "hole" is a small area, as compared with the total basement surface area. This leads to an accurate determination of T_2 because it is the only remaining unknown. Adding small perturbations with the guidance from measurement results leads to T_1 and T_2 of DSS 24, as shown in Eq. (6b). Suppose that a horn were located at F_2 and there were no basement. The coefficient T_1 does not exist, T_2 will be smaller than the one given in Eq. (6b) because there is no (close to) room temperature load below F_2.

For low spillover loss in a BWG shroud, the total antenna system noise temperature is not sensitive to a small variation (say, ±20 K) of T_1 or T_2. For example, for DSS-13 Phase II X-band with a total system noise temperature of 35 K and normalized spillover power $\alpha_{\text{spill}} = 0.012$, ±20 K variations in T resulted in only $0.012(\pm 20) = \pm 0.24$ K. Also, for an extremely low spillover loss (one mirror spill <0.05 percent), the noise temperature computation may not give a sensible result because of the possibility of inaccurate spillover computation.

Soft Decoding a Self-Dual (48,24;12) Code

G. Solomon[1]
Communications Systems Research Section

A self-dual (48,24;12) code comes from restricting a binary cyclic (63,18;36) code to a 6 × 7 matrix, adding an eighth all-zero column, and then adjoining six dimensions to this extended 6 × 8 matrix. These six dimensions are generated by linear combinations of row permutations of a 6 × 8 matrix of weight 12, whose sums of rows and columns add to one. A soft decoding using these properties and approximating maximum likelihood is presented here. This is preliminary to a possible soft decoding of the box (72,36;15) code that promises a 7.7-dB theoretical coding under maximum likelihood.

I. A Self-Dual (48,24;12) Code

Consider the BCH (63,18;24) code of length 63 generated by the recursion polynomial $f_1(x)f_3(x)f_{-1}(x)$, where $f_1(x) = x^6 + x + 1$ with a root β that is a primitive generator of the 63rd roots of unity in $GF(64)$. Here $f_3(x)f_1(x)$ contains β^3 and β^{-1} as roots, respectively. Restrict the values of the code to the coordinates $9i+7j$ for $0 \le i \le 6$, $j = 1, 2, 4, 5, 7, 8$. Thus, a (42,18;12) code has been constructed. To prove this, one examines the matrix in a Mattson-Solomon (MS) polynomial formulation over the rows.

For $z = xy$, where $x^7 = 1$, $y^9 = 1$, $x = \beta^{9i}$, and $y = \beta^{7j}$, indexing the rows by y, the MS polynomial for each row is $P_y(x) = \mathrm{Tr}\,(C_1 y + (C_1 y)^8)x + (C_3 y^3 + C_3^8 y^6)x^3 + (C_1 y^1 + C_1^8 y^1)x^5$. This polynomial becomes, in the Solomon-McEliece Γ_2 Formulation, $P_y(x) = \mathrm{Tr}\,(C_1 y + (C_1 y)^8)x + (C_3^2 y^6 + C_3^{16} y^3 + C_1^4 y^1 + C_1^8 y^8)x^6$.

Thus, the coefficient in x is seen to be a (6,2;5) code over $GF(8)$, while the coefficient of x^6 is a (6,4;3) code over $GF(8)$. The minimum binary weight of the six rows is ≥ 10. Now, summing Γ_2 over the rows, one can see that this adds to 0, yielding weights that are multiples of 4, and thus proving that the minimum distance of the code is ≥ 12. Note that the sum of the rows is the (7,3;4) codeword given by $\mathrm{Tr}\,(C_3 + C_3^8)x^6$.

Adjoin an eighth column to this 7×6 matrix. Now add six more dimensions by forming linear combinations of all cyclic row permutations of the single matrix

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

The newly constructed code of length 48 and dimension 24 has a minimum distance of 12. The dimension 23 code coming from pairs of rows with weight 24 is easily seen to have distance 12 and row sums equal to zero. For the 24th dimension, whose row sums are odd, one need only check

[1] Consultant.

that certain weight patterns in the dimension 18 code did not exist.

To verify the results, note that $\sum_j \Gamma_2 = 0$, $j = 1, 2, 4, 5, 7, 8$, and investigate the weight forms in any row permutation (6,6,6,6,6,6), (6,6,6,6,2,2), (6,6,6,6,4,0), (2,2,2,2,2,2), etc., to verify that this addition does not alter the basic minimum distance and self-orthogonality.

II. Soft Decoding This Code

To decode this code using soft-decision information, first assume that the rows are of even parity. There are six (8,7;2) binary codes with the coefficients of x forming a (6,2;5) code over $GF(8)$. There are 64 such possibilities, and these are stored as six (8,3;4) codes that are cyclic extensions of the maximal length shift register codes Tr cx, $x = \beta^i$, where $0 \leq i \leq 6$. Adding these six codewords to a received word, six extended BCH-Hamming (8,4;4) words are left to maximum likely decode. These words must have coefficients of x^6 that form a (6,4;3) pseudocode. Conducting fifteen trials where four words are assumed correct and generating the rest of the words will give a set of soft-decoded values. Thus, in 64×15 trials a candidate emerges for maximum likelihood decoding. This technique will correct all hard-decision five-error patterns, and almost all six-error patterns. Assuming an odd parity in the eighth column, one uses $2 \times 64 \times 15$ total examinations (2020 trials) in total. How close this is to maximum likelihood is a yet unanswered question.

The 64 words are generated by taking the recursion $x^6 + x + 1$ to generate six linearly independent words, placing the word in the 9×7 matrix as prescribed, and then limiting each codeword to the rows 1, 2, 4, 5, 7, and 8. This will give six generators of the (6,2;5) code and so will yield 64 words. For the (6,4;3) code, generate the cyclic code formed by $(x^6 + x^4 + x^2 + x + 1)(x^6 + x^5 + 1)$. Generate 12 linearly independent words as above and limit each codeword to the rows 1, 2, 4, 5, 7, and 8 to give 12 generators of the (6,4;3) code over $GF(8)$. This is the form desired.

III. Analysis of Performance

How close is this to maximum likelihood performance? Performance here consists of maximum likelihood decoding of the six BCH-Hamming (8,4;4) codes and assumes at least four rows are correct. This clearly will not work if three or more of the decoded rows are incorrect. This is the key factor to decoding correctly. If p is the decoding error under the maximum likelihood of the (8,4;4) code, then the decoding error is $\sum_{i=3}^{6} p^i (1-p)^{6-i}$. This is roughly $20p^3$ for the entire code.

IV. Soft Decoding the (72,36;15) Code

To decode the code in [1] using soft-decision information, first assume that the rows are of even parity. There are nine binary (8,7;2) codes with coefficients of x^6 forming a maximal-distance-separable (MDS) (9,3;7) code over $GF(8)$. There are 128 such possibilities, and these are stored as nine (8,3;4) codes that are cyclic extensions of the maximal length shift register codes Tr cx^6, $x = \beta^i$, where $0 \leq i \leq 6$. Adding these six codewords to a received word, nine extended BCH-Hamming (8,4;4) words are left to maximum likely decode. These decoded words are now symbols that are coefficients of x that form a (9,6;4) code over $GF(8)$. Eighty-four trials, where six symbols are assumed correct to generate the rest of the symbols, will give a set of values for soft decoding. Thus, in $128 \times 84 = 10,752$ trials, there emerges a candidate for maximum likelihood decoding. This technique will correct all hard-decision, seven-error patterns and almost all eight-error patterns. Assuming an odd parity in the eighth column, there have been $2 \times 10,752 = 21,504$ total examinations.

Reference

[1] G. Solomon, "Self-Dual (48,24;12) Codes," *The Telecommunications and Data Acquisition Progress Report 42-111, vol. July–September*, pp. 75–79, November 15, 1992.

A (72,36;15) Box Code

G. Solomon[1]
Communications Systems Research Section

A (72,36;15) box code is constructed as a 9 × 8 matrix whose columns add to form an extended BCH-Hamming (8,4;4) code and whose rows sum to odd or even parity. The newly constructed code, due to its matrix form, is easily decodable for all seven-error and many eight-error patterns. The code comes from a slight modification in the parity (eighth) dimension of the Reed-Solomon (8,4;5) code over GF(512). Error correction uses the row sum parity information to detect errors, which then become erasures in a Reed-Solomon correction algorithm.

I. The Code Construction

The first 27 dimensions of the codes constructed basically constitute a (63,27;16) code represented as a 9×7 matrix. This arises from a Reed-Solomon (RS) cyclic (7,3;5) code over $GF(512)$. The last nine dimensions are constructed by modifying the construction of the extended RS (8,4;5) code over $GF(512)$. For eight of the nine dimensions, this is exactly the extended Reed-Solomon code. For the ninth dimension, the encoding algorithm is modified. Encoding is direct and systematic. Decoding the code uses error/erasure techniques as discussed in [1].

II. A Modified Reed-Solomon (8,4;5) Code Over $GF(512)$

Recall first that the extended RS (8,4;5) code over $GF(8)$ represented in binary form in a normal basis is isomorphic to the extended Golay (24,12;8) code. However, the modified extended RS (8,4;5) code over $GF(64)$ represented in binary form is a self-dual (48,24;12) code. These were shown in [1]. In both these codes, the decoding must sometimes go through eight such trials corresponding to an ambiguity of elements in $GF(8)$. The decoding procedure here will exhibit the same ambiguity for one special case of seven errors that appear as erasures.

Using techniques similar to those in [1], if one starts with an RS (8,4;5) code over $GF(512)$, and represents the code in binary using a particular normal basis with the special property defined below, one can generate a code of length 72 and dimension 36 with even weights that are multiples of 4 and odd weights of form $4m - 1$.

The binary representation of the usual RS (8,4;5) code over $GF(512)$, yields nine (8,7;2) codewords whose decomposition into two cyclic code components and a constant component looks, respectively, like (9,7;3) and (9,2;8) RS or maximal-distance-separable (MDS) codes over $GF(8)$ and a binary (9,9;1) code. However, the code here is constructed by modifying the extended coding rule for the parity symbol.

In particular, let γ be a root of the polynomial $f(x) = x^9 + x^8 + x^6 + x^5 + x^4 + x + 1$, where γ is a primitive generator of the 511 roots of unity. Represent the elements of $GF(512)$ in the normal representation using

[1] Consultant.

the roots of $f(x)$. The roots are γ^j, where $j \in J$ for $J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$.

Note that for this particular choice of $f(x)$, one has

$$\text{Tr } \gamma^j = 1, \quad j \in J, \quad J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$$

$$\text{Tr } \gamma^i \gamma^k = 0, \quad i \neq k, \quad i, k \in J$$

Let β be a root of the polynomial $g(x) = x^3 + x^2 + 1$. Here β is an element of $GF(8)$, a subfield of $GF(512)$, and $\beta = \gamma^{73}$.

A. Encoding

Now use the recursion or check polynomial $h(x) = \prod_{i=0}^{3}(x+\beta^i)$ to generate an RS (7,4;4) code over $GF(512)$. This means that the initial shift register contains four elements in $GF(512)$ expressed as coefficients in the normal representation above. The cyclic portion of the code is of length seven, and the overall parity symbol, the eighth dimension, is defined differently. Representing the binary code as components Tr $P(x)\gamma^i$, where $i = 1, 2, 4, 8, 16, 32, 64, 128, 256$, extends the codes to the eighth coordinates by the rules; the binary value at the row indexed by the ith coordinate is given by Tr $C_0\gamma^i$ + Tr $\sum_{j \in J} C_0\gamma^i$.

Thus, for the constant term C_0 with Tr $C_0 = 0$, this symbol behaves like the normal parity symbol, which is a sum over the values of the cyclic code coordinates.

The general Mattson-Solomon (MS) polynomial of a codeword $\mathbf{a}$ is $P_{\mathbf{a}}(x) = C_0 + C_1 x + C_2 x^2 + C_3 x^3$, where $C_i \in GF(512)$ for $0 \leq i \leq 3$ and $x \in GF(8)$. Encode the codeword in its cyclic portion. The extended codeword $\mathbf{a}$ expressed in terms of the MS polynomial is

$$\mathbf{a} = \left(P_{\mathbf{a}}(\beta^i), \quad 0 \leq i \leq 6, \quad P_{\mathbf{a}}(0)\right)$$

Writing the codewords in binary and using the normal basis γ^j for $j \in J$ above, there are nine binary codewords of length eight

$$\text{Tr } P(x)\gamma^j, \quad j = 1, 2, 4, 8, 16, 32, 64, 128, 256$$

where Tr a denotes the value in $GF(2)$ given by the trace of an element $a \in GF(512)$

$$\text{Tr } a = a + a^2 + a^4 + a^8 + a^{16} + a^{32} + a^{64} + a^{128} + a^{256}$$

Consider one of the nine binary words in its Mattson-Solomon setting,

$$\begin{aligned}\text{Tr } P_{\mathbf{a}}(x)\gamma^j &= \text{Tr } (C_0 + C_1 x + C_2 x^2 + C_3 x^3)\gamma^j \\ &= \text{Tr } C_0\gamma^j + \text{Tr }'[(C_1 x + C_2 x^2 + C_3 x^3)\gamma^j] \\ &\quad + \text{Tr }'[((C_1 x + C_2 x^2 + C_3 x^3)\gamma^j)^8 \\ &\quad + ((C_1 x + C_2 x^2 + C_3 x^3)\gamma^j)^{64}]\end{aligned}$$

$$\text{Tr }' a = a + a^2 + a^4, \quad a \in GF(8)$$

Set $C_0 = 0$ temporarily, as this does not affect the arguments to follow. Then,

$$\begin{aligned}\text{Tr } P(x)\gamma^j &= \text{Tr }'(C_1\gamma^j + (C_1\gamma^j)^8 + (C_1\gamma^j)^{64} \\ &\quad + (C_2\gamma^j)^{256} + (C_2\gamma^{256j})^8 + (C_2\gamma^{256j})^{64})x \\ &\quad + \text{Tr }'((C_3\gamma^j)^2 + (C_3\gamma^j)^{16} + (C_3\gamma^j)^{128})x^6\end{aligned}$$

Lemma. The coefficient of x is a (9,6:4) code over $GF(8)$. The coefficient of x^3, and consequently of x^6, is a (9,3;7) code over $GF(8)$. The code is indexed by the values of γ^j, where $j \in J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$.

Proof. The set γ^j, where $j \in J = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ can only take zero values one less than the number of terms in the coefficients of x and x^6. An argument that clarifies this follows. Consider the coefficient of x^3. This is a polynomial of degree 64 for which, if γ is a solution for a value C_3, then $A\gamma$ is also a solution for all $A \in GF(8)$. Thus, there can only be at most two values of γ that make the coefficient of x^3 equal to zero and consequently the coefficient of x^6 equal to zero. A similar examination of the degrees in the coefficient of x will yield the above result. The term Tr $C_0\gamma^j$ in the code's expression when Tr $C_0 = 0$, i.e., the constant terms, forms a binary (9,8;2) code.

Theorem. The RS code determined by codewords with MS polynomials $P_{\mathbf{a}}(x)$ and Tr $C_0 = 0$ forms a binary (72,35;16) code with weights that are multiples of 4.

Proof. The multiple-of-4 property of the weights using the Solomon-McEliece Γ_2 Formula follows:

$$\text{Tr } P(x)\gamma = \text{Tr } ((C_1\gamma + (C_2\gamma)^4 + (C_3\gamma^2)x^6)$$

where Tr is defined in $GF(64)$.

Now

$$\begin{aligned}\Gamma_2(\text{Tr } P(x)\gamma) = \text{Tr } [&C_1C_3^2\gamma^3 + C_1^8C_3^2\gamma^{10} + C_1^{64}C_3^2\gamma^{66} \\ &+ C_1C_3^{16}\gamma^{17} + C_1^8C_3^{16}\gamma^{24} + C_1^{64}C_3^{16}\gamma^{80} \\ &+ C_1C_3^{128}\gamma^{129} + C_1^8C_3^{128}\gamma^{136} + C_1^{64}C_3^{128}\gamma^{192} \\ &+ C_2^{256}C_3^2\gamma^{258} + C_2^{256}C_3^{16}\gamma^{272} + C_2^{256}C_3^{128}\gamma^{384} \\ &+ C_2^4C_3^2\gamma^6 + C_2^4C_3^{16}\gamma^{20} + C_2^4C_3^{128}\gamma^{132}C_2^{32}C_3^2\gamma^{34} \\ &+ C_2^{32}C_3^{16}\gamma^{48} + C_2^{32}C_3^{128}\gamma^{160}]\end{aligned}$$

Similarly, one can compute $\Gamma_2(\text{Tr } P(x)\gamma^{2^j})$, where $j \leq 8$ and take the sum over all $0 \leq j \leq 8$ to obtain $\sum_{j\in J}\Gamma_2(\text{Tr } P(x)\gamma^j) = 0$. This results from the choice of the normal basis so that

$$\text{Tr } \gamma^j = 1,\ j \in J,\ \text{Tr } \gamma^{j+k} = 0,\ j,k \in J$$

It has been demonstrated that the binary weight of any codeword in the RS code above is a multiple of 4.

III. Structure of the Code

An examination of the binary version of the RS code reveals nine words whose cyclic components form a (9,6;4) and a (9,3;7) code over $GF(8)$. Thus, the symbol weights are 4, 5, 6, 7, 8, and 9. For weights 4, 5, and 6, one has binary weights of the code 16, 20, and 24. When the (9,3;7) code is nonzero, one has a minimum code weight of 14. But, since the codewords have weights that are multiples of 4, one has a minimum weight of 16.

Now consider C_0 by itself when Tr $C_0 = 0$. This is a binary (72,35;16) code. The encoding here has ceased to be systematic since the condition Tr $C_0 = 0$ is nonsystematic.

The extension parity rule for C_0 has been changed to give the following: For $i = 1$, Tr $P(x)\gamma^i = 1$, and Tr $P(x)\gamma^i = 0$, where $i \neq 1$, an eighth row of weight 8 is adjoined, e.g., 011111111. The argument invoked above [1] can then be used to prove that the minimum code distance is 15 and all odd weight words have weights of form $4m - 1$.

IV. Decoding Binary

Assume at first that Tr $C_0 = 0$. Then the parity sums over all nine binary codes give odd error-pattern information. Thus, with seven errors or less spread out over the nine words, at least two values of $C_3\gamma^j$ must be correct. A trial of eight other values for C_3 will eliminate the cyclic component attached to x^6. Since each of the nine binary codes is now an odd-error-detecting/single-error-correcting code, the parity information is usable to correct single errors when they occur. In the case of seven single binary error patterns in seven different rows, a complete correct decoding emerges. This is the most complex decoding to perform, as it requires eight decoding trials. If the seven or less error patterns are in a smaller set of the nine binary codes, then once C_3 is determined and single errors are corrected, there will be at least six correct coefficients of x.

Undecodable eight-error patterns occur when there are less than three of the nine binary codes that are error-free. Once the x^6 coefficient is obtainable without too many trials, then there must emerge, after single-error correction of the erasure codes, at least six correct coefficients of x. One can of course try 512 values of C_3 and then correct for single errors in what remains and finally decode the remaining (9,6;4) code over $GF(8)$.

Reference

[1] G. Solomon, "Self-Dual (48,24;12) Codes," *The Telecommunications and Data Acquisition Progress Report 42-111, vol. July–September*, pp. 75–79, November 15, 1992.

The Analysis of Convolutional Codes via the Extended Smith Algorithm

R. J. McEliece
California Institute of Technology, Department of Electrical Engineering

I. Onyszchuk
Communications Systems Research Section

Convolutional codes have been the central part of most error-control systems in deep-space communication for many years. Almost all such applications, however, have used the restricted class of (n,1), also known as "rate 1/n," convolutional codes. The more general class of (n,k) convolutional codes contains many potentially useful codes, but their algebraic theory is difficult and has proved to be a stumbling block in the evolution of convolutional coding systems. In this article, the situation is improved by describing a set of practical algorithms for computing certain basic things about a convolutional code (among them the degree, the Forney indices, a minimal generator matrix, and a parity-check matrix), which are usually needed before a system using the code can be built. The approach is based on the classic Forney theory for convolutional codes, together with the extended Smith algorithm for polynomial matrices, which is introduced in this article.

I. Introduction

In his celebrated paper on the algebraic structure of convolutional codes, Forney [2] showed that by using the algebra of $k \times n$ polynomial matrices, in particular the invariant-factor theorem (aka the Smith Form), one can transform an arbitrary generator matrix for an (n, k) convolutional code C into a noncatastrophic, basic, and ultimately minimal, generator matrix for C. He also showed how to find a polynomial inverse for a basic generator matrix for C, and a basic generator matrix for the dual code $C^{\perp}$. In this article, efficient ways are discussed to do all these things. The main tool is an algorithm, called the extended Smith algorithm, which is used to find the invariant factors of an arbitrary $k \times n$ matrix over an Euclidean domain, which bears the same relationship to the usual invariant factor algorithm as the extended Euclid's algorithm bears to the usual Euclid's algorithm.

A brief review of Euclid's algorithm (see e.g. [5, Section 1.1] or [9, Chapter 2]) is helpful. The algorithm's goal is to take a pair (a, b) of elements from a Euclidean domain R, and by repeated application of the division algorithm for R to find the greatest common divisor d of a and b. The goal of the *extended Euclidean algorithm* (see e.g. [6, Section 4.5.2] or [8, Section 8.4]) is to take the same pair, and not only find d, but also find elements s and t of R, such that $sa+tb = d$. The extended Euclidean algorithm has several important applications in coding theory. For example, it can be used to compute inverses in finite fields [9, Exam-

ple 4.2], decode BCH codes [8, Section 8.5], and to find finite impulse response (FIR) inverses for noncatastrophic $(n, 1)$ convolutional generator matrices [1, Section 12.2]. Similarly, the goal of the *Smith algorithm* (see e.g. [3, Section 6.2.4], [4, Section 6.3.3], or Smith's nineteenth-century original article [10, Section 12.2]) is to take an arbitrary $k \times n$ matrix G (with $k \leq n$) over an Euclidean domain R, and by a sequence of elementary row and column operations, to reduce G to a $k \times n$ diagonal matrix $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_k)$, whose diagonal entries are the invariant factors of G, i.e., $\gamma_i = \Delta_i / \Delta_{i-1}$, where Δ_i is the greatest common denominator of the $i \times i$ minors of G. (Here, $\Delta_0 = 1$ is taken by convention.) Smith's algorithm is reviewed in Section II.

The goal of the *extended Smith algorithm*, which is introduced in this article, is to take the same input, and not only find Γ, but also to find a $k \times k$ unimodular matrix X, and an $n \times n$ unimodular matrix Y, such that $XGY = \Gamma$. It is worthwhile to note that in the special case $k = 1$ and $n = 2$, the (extended) Smith algorithm reduces to the (extended) Euclidean algorithm. The extended Smith algorithm is described in detail in Section III.

Section IV describes how, given an arbitrary generator matrix G for an (n, k) convolutional code, the extended Smith algorithm can be used to efficiently compute the things mentioned above (the degree, the Forney indices, a minimal generator matrix, a polynomial inverse, a parity-check matrix, etc., for the given code). Throughout this article, all results are illustrated with an example of a 2×4 matrix of polynomials over the binary field $GF(2)$, which is a generator matrix for a $(4, 2)$ convolutional code over $GF(2)$.

II. Smith's Algorithm

In this section, a careful description is given of Smith's algorithm (which is often called the invariant-factor algorithm), but a formal proof of its correctness is not given. For that, refer to [3, Section 6.2.4], [4, Section 6.3.3], or [11, Section 12.2].

Begin by recalling the definition of a general Euclidean domain [8, Chapter 2]. It is an integral domain, i.e., a ring that satisfies the cancellation property, together with a "size" function $|a|$ defined for every nonzero element $a \in R$. The size function must satisfy $|a| \leq |ab|$ if $b \neq 0$. Furthermore, if a and b are arbitrary, and $b \neq 0$, a can be "divided" by b in the sense that there exist elements q (the "quotient") and r (the "remainder") such that $a = qb + r$, where either $r = 0$ or else $|r| < |b|$. For application to the study of convolutional codes, always take R to the ring of polynomials over a field F, where the "size" of a polynomial is its degree. However, there are many other Euclidean domains [8, Chapter 2], and Smith's algorithm applies equally to all of them.

The central part of Smith's algorithm is the following subalgorithm E, which takes as input an arbitrary matrix A with entries from an Euclidean domain R, at least one of which is nonzero, and, via elementary row and column operations, transforms A into a matrix with the $(1, 1)$ entry nonzero, every other entry in row 1 and column 1 zero, and every other entry in the matrix divisible by the $(1, 1)$ entry.

E1. Move the entry in A of least size to position $(1, 1)$.

E2. If there is a nonzero entry in either row 1 or column 1 that is not divisible by a_{11}, use it to reduce the size of a_{11}, as follows:

E2a. If a_{1j}, the entry in row 1 and column j, is not divisible by a_{11}, then by the division algorithm there exist nonzero elements q and r such that $a_{1j} = qa_{11} + r$, with $|r| < |a_{1,1}|$. Thus, if q times column 1 is subtracted from column j, the $(1, j)$ entry will be changed to r, which has a smaller size than $a_{1,1}$. If this entry is moved to position $(1, 1)$, the size of a_{11} will have been reduced.

E2b. If a_{i1}, the entry in row i and column 1, is not divisible by a_{11}, repeat the procedure outlined in step E2a, with rows and columns interchanged.

E3. Reduce all the entries in row 1 and column 1 (except $a_{1,1}$ itself) to zero, as follows:

E3a. Since (by step E2) a_{1j} for $j \geq 2$ is divisible by a_{11}, $a_{1j} = q_j a_{11}$. Thus, by subtracting q_j times column 1 from column j, the $(1, j)$ entry will be transformed to zero. Repeating this step for $j = 2, \ldots, r$, the first row will "zero out."

E3b. Repeat step E3a with rows and columns interchanged.

E4. If there is a nonzero entry in A, say a_{ij}, which is not divisible by a_{11}, add column j to column 1. (This produces a nonzero entry in column 1 that is not divisible by a_{11}.) Return to step E2.

E5. Stop. The matrix A now has the desired property.

In the Smith algorithm itself, one applies subalgorithm E successively to the original matrix G, then to the matrix obtained from G by deleting row 1 and column 1, then by deleting row 2 and column 2, etc., until either an all-zero matrix is encountered, or until all rows and columns have been processed. The result is a $k \times n$ diagonal matrix $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_k)$, whose diagonal entries are the invariant factors of G. The following example illustrates the Smith algorithm, when the underlying Euclidean domain is the ring of polynomials over the field $GF(2)$, in which the "size" of an element is its degree.

Example 1. Consider the following 2×4 matrix with entries that are polynomials over $GF(2)$:

$$G = G_0 = \begin{pmatrix} 1 & 1+D+D^2 & 1+D^2 & 1+D \\ D & 1+D+D^2 & D^2 & 1 \end{pmatrix}$$

Since the lowest degree term already appears in the $(1,1)$ position, skip step E1. Since the $(1,1)$ entry is 1, every nonzero entry in row 1 and column 1 is divisible by the $(1,1)$ entry, so skip step E2 as well. Executing step E3: "zero out" the first row by adding $1+D+D^2$ times column 1 to column 2, $1+D^2$ times column 1 to column 3, and $1+D$ times column 1 to column 4, thereby obtaining successively

$$G_1 = \begin{pmatrix} 1 & 0 & 1+D^2 & 1+D \\ D & 1+D^3 & D^2 & 1 \end{pmatrix}$$

$$G_2 = \begin{pmatrix} 1 & 0 & 0 & 1+D \\ D & 1+D^3 & D+D^2+D^3 & 1 \end{pmatrix}$$

$$G_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ D & 1+D^3 & D+D^2+D^3 & 1+D+D^2 \end{pmatrix}$$

To zero the entry in position $(2,1)$, add D times row 1 to row 2, thereby obtaining

$$G_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1+D^3 & D+D^2+D^3 & 1+D+D^2 \end{pmatrix}$$

This completes the operation of subalgorithm E on the original matrix G. Now apply subalgorithm E to the 1×3 matrix obtained by deleting row 1 and column 1 from G_4. The lowest degree term in G_4 appears in position $(2,4)$, so by interchanging columns 2 and 4, it is moved to position $(2,2)$:

$$G_5 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1+D+D^2 & D+D^2+D^3 & 1+D^3 \end{pmatrix}$$

At this point, all of the remaining entries in row 2 are divisible by the $(2,2)$ entry: $D+D^2+D^3 = D\cdot(1+D+D^2)$, and $1+D^3 = (1+D)\cdot(1+D+D^2)$. Thus by adding D times column 2 to column 3, and then $1+D$ times column 2 to column 4, compute successively

$$G_6 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1+D+D^2 & 0 & 1+D^3 \end{pmatrix}$$

$$G_7 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1+D+D^2 & 0 & 0 \end{pmatrix}$$

This completes the operation of the Smith algorithm on G. The matrix G_7 is the invariant-factor form for G; in particular, the invariant factors of G are $\gamma_1 = 1$ and $\gamma_2 = 1+D+D^2$. ■

III. The Extended Smith Algorithm

The previous section showed, in computational detail, how the Smith algorithm works. Beginning with the matrix $G_0 = G$, it produces a sequence of $k \times n$ matrices G_i, where G_{i+1} is derived from G_i by either an elementary row operation or an elementary column operation. This is represented algebraically as

$$G_{i+1} = E_{i+1} G_i F_{i+1} \tag{1}$$

where E_{i+1} and F_{i+1} are $k \times k$ and $n \times n$ elementary matrices, respectively. If G_{i+1} is obtained from G_i via a *row* operation, then E_{i+1} is a nontrivial $k \times k$ elementary matrix, but $F_{i+1} = I_n$. If G_{i+1} is obtained from G_i via a *column* operation, then F_{i+1} is a nontrivial $n \times n$ elementary matrix, but $E_{i+1} = I_k$. After a finite number N of steps, $G_N = \Gamma$.

The extended Smith algorithm builds on the Smith algorithm. Whereas the Smith algorithm works only with the sequence $G_0, G_1, \ldots, G_N$, the extended Smith algorithm also works with a sequence of unimodular $k \times k$ matrices $X_0, \ldots, X_N$, and a sequence of unimodular $n \times n$ matrices $Y_0, \ldots, Y_N$. The sequences (X_i) and (Y_i) are initialized as $X_0 = I_k$ and $Y_0 = I_n$, and updated via the rule [cf. Eq. (1)]

$$X_{i+1} = E_{i+1} X_i \tag{2}$$

$$Y_{i+1} = Y_i F_{i+1} \tag{3}$$

The following simple lemma is the key to the extended Smith algorithm.

Lemma.

$$X_i G Y_i = G_i \qquad \text{for } i = 0, 1, \ldots, N \tag{4}$$

Proof. Since $X_0 = I_k$ and $Y_0 = I_n$, Eq. (4) holds for $i = 0$. Assuming now that Eq. (4) holds for a given value of i, multiply both sides of Eq. (4) on the left by E_{i+1} and on the right by F_{i+1}

$$E_{i+1} X_i G Y_i F_{i+1} = E_{i+1} G_i F_{i+1} \tag{5}$$

By Eq. (1), the right side of Eq. (5) is equal to G_{i+1}. Thus

$$(E_{i+1} X_i) G (Y_i F_{i+1}) = G_{i+1} \tag{6}$$

But $E_{i+1} X_i = X_{i+1}$ by Eq. (2) and $Y_i F_{i+1} = Y_{i+1}$ by Eq. (3), so Eq. (6) becomes simply $X_{i+1} G Y_{i+1} = G_{i+1}$, which proves Eq. (4) by induction. ∎

If Eq. (4) is specialized with $i = N$,

$$X_N G Y_N = \Gamma \tag{7}$$

which is the desired extended Smith diagonalization of G. A convenient way to implement the extended Smith algorithm is to extend the original matrix G to dimensions $(n+k) \times (n+k)$ as follows:

$$G' = \begin{pmatrix} G & I_k \\ I_n & 0_{n \times k} \end{pmatrix} \tag{8}$$

where in Eq. (8) $0_{n \times k}$ is an $n \times k$ matrix of zeros. Then if the sequence of elementary row and column operations generated by the Smith algorithm applied to G is performed on the extended matrix G', after i iterations, the resulting matrix G'_i has the form

$$G'_i = \begin{pmatrix} G_i & X_i \\ Y_i & 0_{n \times k} \end{pmatrix} \tag{9}$$

Thus, after N steps, the matrices X_N and Y_N in Eq. (7) appear as the upper-right and lower-left blocks of G'_N, respectively. All this is illustrated by extending the example from Section II.

Example 2. Begin as in Example 1, with the following 2×4 matrix over $GF(2)[D]$:

$$G = G_0 = \begin{pmatrix} 1 & 1+D+D^2 & 1+D^2 & 1+D \\ D & 1+D+D^2 & D^2 & 1 \end{pmatrix}$$

Then the corresponding matrix G' [cf. Eq. (8)] is

$$G' = G'_0 = \begin{pmatrix} 1 & 1+D+D^2 & 1+D^2 & 1+D & 1 & 0 \\ D & 1+D+D^2 & D^2 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

In Example 1, G_1, G_2, and G_3 are obtained from G_0 by successively adding $1 + D + D^2$ times column 1 to column 2, $1 + D^2$ times column 1 to column 3, and $1 + D$ times column 1 to column 4. If the same sequence of operations is performed to G'_0, one obtains successively

$$G_1' = \begin{pmatrix} 1 & 0 & 1+D^2 & 1+D & 1 & 0 \\ D & 1+D^3 & D^2 & 1 & 0 & 1 \\ 1 & 1+D+D^2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$G_2' = \begin{pmatrix} 1 & 0 & 0 & 1+D & 1 & 0 \\ D & 1+D^3 & D+D^2+D^3 & 1 & 0 & 1 \\ 1 & 1+D+D^2 & 1+D^2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$G_3' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ D & 1+D^3 & D+D^2+D^3 & 1+D+D^2 & 0 & 1 \\ 1 & 1+D+D^2 & 1+D^2 & 1+D & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Next, adding D times row 1 to row 2, as was done to obtain G_4 from G_3, obtain

$$G_4' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1+D^3 & D+D^2+D^3 & 1+D+D^2 & D & 1 \\ 1 & 1+D+D^2 & 1+D^2 & 1+D & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Interchanging columns 2 and 4, obtain

$$G_5' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1+D+D^2 & D+D^2+D^3 & 1+D^3 & D & 1 \\ 1 & 1+D & 1+D^2 & 1+D+D^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Finally, adding D times column 2 to column 3, and $1+D$ times column 2 to column 4, one computes successively

$$G_6' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1+D+D^2 & 0 & 1+D^3 & D & 1 \\ 1 & 1+D & 1+D & 1+D+D^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & D & 0 & 0 & 0 \end{pmatrix}$$

and

$$G_7' = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1+D+D^2 & 0 & 0 & D & 1 \\ 1 & 1+D & 1+D & D & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & D & 1+D & 0 & 0 \end{pmatrix}$$

Thus the extended Smith decomposition of the original matrix G is given by [cf. Eq. (7)]

$$\begin{pmatrix} 1 & 0 \\ D & 1 \end{pmatrix} \cdot G \cdot \begin{pmatrix} 1 & 1+D & 1+D & D \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & D & 1+D \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1+D+D^2 & 0 & 0 \end{pmatrix}$$

In Section V, the decomposition of Example 2 will be used in the analysis of the $(4, 2)$ convolutional code generated by G, as an illustration of the general results to be expounded upon in Section IV.

IV. Application to the Analysis of Convolutional Codes

In this section, given an arbitrary generator matrix G for an (n, k) convolutional code C, one can efficiently find, among other things: a basic generator matrix, say G_{basic} for C; a polynomial inverse for G_{basic}; a minimal generator matrix for C; and a basic generator matrix for the dual code $C^{\perp}$. The central tool needed to do all this is the extended Smith algorithm introduced in Section III.

Assume then that G is a $k \times n$ generator matrix for an (n, k) convolutional code over the field F, i.e., a $k \times n$ matrix of rank k over the field $F(D)$ of rational functions over F. By multiplying the ith row of G by the least common multiple of the denominators in that row, one easily obtains an equivalent generator matrix, all of whose entries are polynomials over F, i.e., elements of $F[D]$. Since $F[D]$ is an Euclidean domain, one may apply the extended Smith algorithm described in Section III, thereby obtaining a decomposition of Eq. (7). In what follows, the matrices X_N and Y_N produced by the extended Smith algorithm will be denoted simply by X and Y.

The matrices X, Y, and Γ contain much valuable information about the code C and the generator matrix G. To extract this information, however, first define several useful pieces of these matrices, called Γ_k, $\tilde{\Gamma}_k$, K, and H (in what follows, $r = n - k$)

$$\Gamma_k = \text{leftmost } k \text{ columns of } \Gamma = \text{diag}(\gamma_1, \ldots, \gamma_k)$$

$$\text{(dimensions: } k \times k) \tag{10}$$

$$\tilde{\Gamma}_k = \gamma_k \cdot \Gamma_k^{-1} = \text{diag}(\gamma_k/\gamma_1, \ldots, \gamma_k/\gamma_k)$$

$$\text{(dimensions: } k \times k) \tag{11}$$

$$K = \text{leftmost } k \text{ columns of } Y$$

$$\text{(dimensions: } n \times k) \tag{12}$$

$$H = \text{rightmost } r \text{ columns of } Y$$

$$\text{(dimensions: } n \times r) \tag{13}$$

The following theorem describes the useful outputs of the extended Smith algorithm when applied to G. (For terminology not fully explained here, refer to Forney [2].)

Theorem 1. With the matrices Γ_k, $\tilde{\Gamma}_k$, K, and H defined as in Eqs. (10)–(13), one has the following:

(a) *A basic generator matrix for C:* $G_{\text{basic}} = \Gamma_k^{-1} X G$. (That is, G_{basic} is obtained by dividing the ith row of XG by the invariant factor γ_i, for $i = 1, \ldots, k$.)

(b) *A polynomial inverse for G_{basic}:* K.

(c) *A polynomial pseudo-inverse for G, with factor γ_k:* $K\tilde{\Gamma}_k X$. (In particular, if G is already basic, i.e., if $\Gamma_k = I_k$, then KX is a polynomial inverse for G.)

(d) *A basic generator matrix for $C^{\perp}$, i.e., a parity-check matrix for C:* H^T.

Proof. From Eq. (7) with $X = X_N$ and $Y = Y_N$, one has

$$XG = \Gamma B \tag{14}$$

where $B = Y^{-1}$. Now B is an $n \times n$ unimodular polynomial matrix. Let U be the $k \times n$ matrix consisting of the first k rows of B, and L be the $r \times n$ matrix consisting of the last r rows of B. Then, since $\Gamma = (\Gamma_k \quad 0_{k\times r})$ where $0_{k\times r}$ is a $k \times r$ all-zeros matrix, it follows that

$$\Gamma B = (\Gamma_k \quad 0_{k\times r}) \begin{pmatrix} U \\ L \end{pmatrix} = \Gamma_k U \tag{15}$$

so that, combining Eqs. (14) and (15)

$$U = \Gamma_k^{-1} XG \tag{16}$$

From Eq. (16), U is row equivalent to G. Furthermore, since $U = (I_k \quad 0_{k\times r})B$, and B is unimodular, it follows that the invariant factors of U are all equal to 1, and so U is a basic generator matrix for C. This proves Theorem 1(a).

To prove Theorem 1(b), use the fact that Y and B are inverse matrices, so that

$$BY = \begin{pmatrix} U \\ L \end{pmatrix} (K \quad H) = \begin{pmatrix} UK & UH \\ LK & LH \end{pmatrix}$$

$$= I_n = \begin{pmatrix} I_k & 0_{k\times r} \\ 0_{r\times k} & I_r \end{pmatrix} \tag{17}$$

It follows from Eq. (17) that $UK = I_k$, so that K is a polynomial inverse for the basic generator matrix U, which proves Theorem 1(b).

To prove Theorem 1(c), suppose that J is an $n \times k$ polynomial pseudo-inverse for G with (polynomial) factor $\psi(D)$, i.e., that $GJ = \psi I_k$. Then

$$XGYY^{-1}JX^{-1} = XGJX^{-1} = X(\psi I_k)X^{-1} = \psi I_k \tag{18}$$

But from Eq. (7), $XGY = \Gamma = (\Gamma_k \quad 0_{k\times r})$. Thus, if the matrix $Y^{-1}JX^{-1}$ is denoted by J', J' is a polynomial matrix (since X and Y are unimodular) and Eq. (18) becomes

$$(\Gamma_k \quad 0)J' = \psi I_k \tag{19}$$

If now $J' = \begin{pmatrix} J'_0 \\ J'_1 \end{pmatrix}$ where J'_0 is $k \times k$ and J'_1 is $r \times k$, it follows from Eq. (19) that $J'_0 = \psi\Gamma_k^{-1}$. But since J' is a polynomial matrix, and $\Gamma_k^{-1} = \mathrm{diag}(\gamma_1^{-1}, \ldots, \gamma_k^{-1})$, it follows that ψ must be a multiple of γ_k. Conversely, if $\psi = \gamma_k$, so that $J'_0 = \gamma_k\Gamma_k^{-1} = \Gamma_k$ and $J'_1 = 0_{r\times k}$, then $J' = \begin{pmatrix} \tilde{\Gamma}_k \\ 0_{r\times k} \end{pmatrix}$. Thus, $J = YJ'X = K\tilde{\Gamma}_k X$ is a polynomial pseudo-inverse with factor γ_k, which proves Theorem 1(c).

To prove Theorem 1(d), one has to show that the set of codewords in the code generated by G, i.e., the set of n-dimensional $F(D)$-vectors y of the form $y = xG$, is identical to the set of vectors y such that $yH = 0$. Thus, let $Y_1 = \{y : y = xG\}$ and $Y_2 = \{y : yH = 0\}$. It will be shown that $Y_1 = Y_2$.

Since, as in Theorem 1(a), the matrix U, defined to be the first k rows of $B = Y^{-1}$, is (a basic generator matrix) equivalent to G, it follows that $Y_1 = \{y : y = xU\}$. Thus, if $y \in Y_1$, $y = xU$ for some x and so $yH = (xU)H = x(UH) = 0$, since by Eq. (17), $UH = 0_{k\times r}$. Thus, $Y_1 \subseteq Y_2$. On the other hand, if y is arbitrary, then since $B^{-1} = Y$,

$$y = yI_n = y(YB) = y\left((K \quad H)\begin{pmatrix} U \\ L \end{pmatrix}\right)$$

$$= y(KU + HL) = (yK)U + (yH)L \tag{20}$$

Now suppose $y \in Y_2$, i.e., $yH = 0$. Then from Eq. (20) $y = (yK)U = xU$, where $x = yK$, and so $y \in Y_1$. Thus, $Y_1 \subseteq Y_2$, which completes the proof of Theorem 1(d). ∎

In the next section these results will be briefly illustrated with a simple example.

V. An Example Convolutional Code

To illustrate the results of this article, consider the following generator matrix G for a binary $(4, 2)$ convolutional code:

$$G = \begin{pmatrix} 1 & 1+D+D^2 & 1+D^2 & 1+D \\ D & 1+D+D^2 & D^2 & 1 \end{pmatrix}$$

In Section III the extended invariant-factor decomposition of G was found to be $XGY = \Gamma$, where

$$\Gamma = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1+D+D^2 & 0 & 0 \end{pmatrix}$$

$$X = \begin{pmatrix} 1 & 0 \\ D & 1 \end{pmatrix}$$

$$Y = \begin{pmatrix} 1 & 1+D & 1+D & D \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & D & 1+D \end{pmatrix}$$

Thus, the matrices defined in Eqs. (10)–(13) are as follows:

$$\Gamma_k = \begin{pmatrix} 1 & 0 \\ 0 & 1+D+D^2 \end{pmatrix}$$

$$\tilde{\Gamma}_k = \begin{pmatrix} 1+D+D^2 & 0 \\ 0 & 1 \end{pmatrix}$$

$$K = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1+D & 0 & 0 & 1 \end{pmatrix}^T$$

$$H = \begin{pmatrix} 1+D & 0 & 1 & D \\ D & 1 & 0 & 1+D \end{pmatrix}^T$$

Using the prescriptions in Theorem 1, one quickly obtains the following:

(a) *A basic generator matrix for C:*

$$G_{\text{basic}} = \Gamma_k^{-1} X G$$

$$= \begin{pmatrix} 1 & 1+D+D^2 & 1+D^2 & 1+D \\ 0 & 1+D & D & 1 \end{pmatrix}$$

(b) *A polynomial inverse for G_{basic}:*

$$K = \begin{pmatrix} 1 & 1+D \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

(c) *A polynomial pseudo-inverse for G, with factor $\gamma_2 = 1+D+D^2$:*

$$K\tilde{\Gamma}_k X = \begin{pmatrix} 1 & 1+D \\ 0 & 0 \\ 0 & 0 \\ D & 1 \end{pmatrix}$$

(d) *A (basic) generator matrix for $C^\perp$:*

$$H^T = \begin{pmatrix} 1+D & 0 & 1 & D \\ D & 1 & 0 & 1+D \end{pmatrix}$$

Finally, note that in this case neither of the basic generator matrices G_{basic} or H^T is minimal. To minimize them, follow the simple algorithm originally described in [2], or perhaps more lucidly in Kailath [4, Section 6.3.2] (where minimal matrices are, however, called row-reduced). The idea is to use elementary row operations on G to reduce the degree of the highest degree terms in some row, as long as this is possible. For example, to minimize G_{basic}, multiply the second row by D and add it to the first row, obtaining

$$G' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1+D & D & 1 \end{pmatrix}$$

Since elementary row operations cannot reduce the row degrees further, the matrix G' is a minimal generator matrix for the code C. (Indeed, at this point, one can recognize C as the $d_{\text{free}} = 4$ partial-unit-memory $(4,2)$ code first found by Lauer [7, Table 1].) To find a polynomial inverse for G', note that $G' = TG$, where T is a unimodular $k \times k$ polynomial matrix, and so $K' = KT^{-1}$ is a polynomial inverse for G'. In this example, simply multiply the *first column* of K by D and *subtract* it from the *second column*, thereby obtaining the following polynomial inverse for the minimal generator matrix G':

$$K' = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}$$

To minimize H^T, simply add row 2 to row 1, thereby obtaining the following minimal generator matrix for the dual code:

$$H' = \begin{pmatrix} 1 & 1 & 1 & 1 \\ D & 1 & 0 & 1+D \end{pmatrix}$$

In this case, the dual code is isomorphic to the original code.

The *Forney indices* of an (n, k) convolutional code are the degrees of the rows of a minimal generator matrix, and the *degree* of the code is the sum of the Forney indices. In this example, the *degree* of both C and $C^{\perp}$ is 1, and the Forney index of both codes is $(0, 1)$. It is a general theorem that $\deg C = \deg C^{\perp}$ (see [2, Theorem 7]), but the equality of the Forney indices in this case is more or less accidental.

References

[1] R. E. Blahut, *Theory and Practice of Error-Correcting Codes*, Reading, Massachusetts: Addison-Wesley, 1983.

[2] G. D. Forney, "Convolutional Codes I: Algebraic Structure," *IEEE Transactions on Inform. Theory*, vol. IT-16, pp. 720–738, November 1970.

[3] F. R. Gantmacher, *The Theory of Matrices, Volume I*, New York: Chelsea Publishing Company, 1977.

[4] T. Kailath, *Linear Systems*, Englewood Cliffs, N. J.: Prentice-Hall, 1980.

[5] D. E. Knuth, *The Art of Computer Algorithms, Volume 1: Fundamental Algorithms*, second edition, Reading, Massachusetts: Addison-Wesley, 1973.

[6] D. E. Knuth, *The Art of Computer Algorithms, Volume 2: Seminumerical Algorithms*, second edition, Reading, Massachusetts: Addison-Wesley, 1981.

[7] G. S. Lauer, "Some Optimal Partial-Unit-Memory Codes," *IEEE Transactions on Inform. Theory*, vol. IT-25, pp. 240–243, March 1979.

[8] R. J. McEliece, *The Theory of Information and Coding*, Reading, Massachusetts: Addison-Wesley, 1977.

[9] R. J. McEliece, *Finite Fields for Computer Scientists and Engineers*, Boston: Kluwer, 1987.

[10] H. J. S. Smith, "On Systems of Linear Indeterminate Equations and Congruences," *Philos. Transactions of the Royal Society of London*, vol. 151, pp. 293–326, 1861.

[11] B. L. van der Waerden, *Algebra, Volume 2*, New York: Frederick Ungar, 1970.

Image Coding Via Wavelets

M. Shahshahani
Communications Systems Research Section

The application of two wavelet transforms to image compression is discussed. It is noted that the Haar transform, with proper bit allocation, has performance that is visually superior to an algorithm based on a Daubechies filter and to the discrete-cosine-transform-based Joint Photographic Experts Group (JPEG) algorithm at compression ratios exceeding 20:1. In terms of the root-mean-square error, the performance of the Haar transform method is basically comparable to that of the JPEG algorithm. The implementation of the Haar transform can be achieved in integer arithmetic, making it very suitable for applications requiring real-time performance.

I. Introduction

In an earlier article [1], the author reported on his work on the application of a wavelet transform to image coding. It was noted that the images processed by this method did not suffer from the blockiness which is typical of algorithms based on the discrete cosine transform (DCT) at compression ratios exceeding 20 to 1; however, the edges of the objects in an image were blurred and created an unsatisfactory visual impression. Two approaches were taken to overcome this problem. The wavelet transform used in [1] was essentially one dimensional in nature, and it was applied along rows and columns of the image. In the first approach, it was surmised that the reason for the blurriness of the edges was that the application of the transform along rows and columns did not properly take advantage of the proximity of the pixels. Several modified versions of this transform reflecting the contiguity of the pixels were tested. Improvements in the edges and the general quality of the processed images were observed for certain modified transforms. The second approach was based on a two-dimensional Haar transform. The edges and the general quality of the processed images were in general superior to those of the first method and to those of the Joint Photographic Experts Group (JPEG) DCT-based algorithm at compression ratios exceeding 20 to 1. In terms of the root-mean-square error (RMSE), the performance of this method is essentially comparable to that of the JPEG DCT-based algorithm. This is remarkable, since the quantization and bit-allocation algorithms used in the tests reported here were of a much cruder nature than those utilized in the DCT-based approach. While there is some blockiness in images processed with the Haar transform method, it is not nearly as severe as in images processed with the DCT-based algorithm, and the edges appear as sharp as those obtained by the latter method.

In the application of wavelet transforms, one assumes that the wavelet transform coefficients are encoded by an entropy encoder. This method is analogous to algorithms based on other transforms such as DCT. An advantage of the Haar transform approach is that both the algorithm and its inversion can be implemented in integer arithmetic and are very suitable for applications where real-time performance is required.

II. A Daubechies Filter

The work in [1] was based on the filter $\mathcal{F}$ defined by the matrix

$$\begin{pmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 & 0 & 0 & \dots & \dots & 0 \\ \alpha_3 & -\alpha_2 & \alpha_1 & -\alpha_0 & 0 & 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 & \dots & \dots & 0 \\ 0 & 0 & \alpha_3 & -\alpha_2 & \alpha_1 & -\alpha_0 & 0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & & & \vdots \\ \alpha_2 & \alpha_3 & 0 & 0 & \dots & \dots & \dots & 0 & \alpha_0 & \alpha_1 \\ \alpha_1 & -\alpha_0 & 0 & 0 & \dots & \dots & \dots & 0 & \alpha_3 & -\alpha_2 \end{pmatrix}$$

where

$$\alpha_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad \alpha_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}$$

$$\alpha_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \alpha_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}$$

This filter, the discrete version of one of Daubechies' compactly supported wavelets [2], is one dimensional and was applied in the work reported in [1] along rows and columns of an image. Instead, it is possible to reorder the pixels differently to take advantage of their proximity and then apply the filter. There is no unique way of reordering or reconfiguring the pixels. A number of experiments were carried out with widely different results. Certain configurations led to improvement of the edges, while others resulted in significant deterioration of the processed image. The author does not know of any theoretical procedure for obtaining the optimal configuration. The experiments did, however, yield some insight into what a desirable configuration may look like. A method which produced improvements in the processed image is a two-step application of the filter $\mathcal{F}$. The first step is described in Figure 1. Starting with an even-numbered row, the weighted average of the pixels with even coordinates $(2m, k)$, $(2m+1, k)$, $(2m, k+1)$, and $(2m+1, k+1)$ is computed by using the weights α_0, α_1, α_2, and α_3. This weighted average is the transformed value at the pixel with coordinates $(2m, k)$. To compute the transformed values at $(2m+1, k)$, the weighted average of the pixels $(2m, k)$, $(2m+1, k)$, $(2m, k+1)$, and $(2m+1, k+1)$ with weights α_3, $-\alpha_2$, α_1, and $-\alpha_0$ is calculated. The second step is the application of the filter $\mathcal{F}$ along the even-numbered rows after the image is transformed according to the procedure of step 1. The hierarchy of the coefficients is different from that described in [1]. After step 1 of the first iteration of the two-step process, the coefficients in odd-numbered rows are placed at the lowest level of the hierarchy. The coefficients in the even-numbered rows are then transformed according to step 2, and those coefficients corresponding to odd-numbered columns are placed immediately above the lowest level. The next iteration is the application of the same procedure to the subgrid of points with even coordinates. Evenness and oddness are then replaced by divisibility by 4 and congruence to 2 mod 4. The procedure can be repeated in the obvious manner. In the terminology of [1], the coefficients at pixels with coordinates divisible by 2^r form the projection of the data from $\mathcal{L}_{r-1}$ to $\mathcal{L}_r$. Figure 2 is an image of peppers that was processed by this algorithm. The compression ratio is 27:1.

III. The Haar Transform

The Haar transform is defined by the matrix

$$\mathcal{H} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 \end{pmatrix}$$

which is a Hadamard matrix (see [3]). Just as is the case with the Daubechies filter $\mathcal{F}$, this is a one-dimensional filter which may be applied in different ways to the two-dimensional image. The configuration which appears to best take advantage of the proximity of the pixels and produce the most appealing results visually is shown in Fig. 3. The image is subdivided into 2×2 blocks. Within each block the pixels are reordered as $(2k, 2m)$, $(2k, 2m+1)$, $(2k+1, 2m)$, and $(2k+1, 2m+1)$, and the matrix $\mathcal{H}$ is applied to each block separately to yield the transformed image which is of the same size. The coefficients in the transformed image are separated into two groups which were designated in [1] as "smooth" and "detail." In the terminology of [1], those coefficients corresponding to the pixels with even coordinates form the smooth group (i.e., the image of the projection from $\mathcal{L}_0$ to $\mathcal{L}_1$), and the remaining are the details (i.e., the image of the projection from $\mathcal{L}_0$ to $\mathcal{E}_0$). The filter $\mathcal{H}$ is then applied to the smooth group in the obvious manner, leading to a hierarchy of transformed data. Figures 4(a) and 4(b) are two images

processed by the Haar transform. The compression ratios are 12:1 and 27:1, respectively. Notice that the edges are clearly defined, even with a compression factor of 27.

Notice that the Haar transform is integer valued, and therefore its application can be implemented in integer arithmetic. Since averaging over four pixel values generates large numbers rapidly, it was convenient to divide the values by four, which from the implementation point of view is easily achievable. Naturally, one loses some resolution in this manner; nevertheless, this approach seems to be convenient.

IV. Bit Allocation and Quantization

As noted in [1], the best method of quantization is by truncation to the nearest integer, irrespective of the distribution of the coefficients. Bit allocation is done according to the desired compression ratio and the level of the hierarchy in the wavelet pyramid. Figure 5 exhibits RMSE versus compression ratio for what, after much experimentation, appears to be a good choice of bit assignments to the wavelet coefficients obtained via the application of the filter $\mathcal{F}$ as described in Section II. The actual bits assigned to each level are given in Table 1.

The actual bit allocation affects the compression ratio versus root-mean-square error significantly. In the application of the Haar transform, the actual effect of different bit-allocation schemes on RMSE is shown in Fig. 6. Each circled point in the graph corresponds to a particular assignment of bits to different levels of the hierarchy of the wavelet coefficients. It is clear from the graph that certain assignments do not produce desirable effects. The desirable bit-allocation schemes correspond to the lower envelope of the curve. This lower envelope is reproduced in Fig. 7. Figure 8 gives a comparison between the Haar transform algorithm and the standard JPEG DCT-based algorithm on the basis of the root-mean-square error. Note that the performance of the two methods is comparable. This is remarkable, since the latter approach makes use of refined quantization and bit-allocation algorithms, while in the tests reported here, rather crude methods were used. It is therefore reasonable to expect that with further refinement of the Haar method, it will have superior performance even in terms of RMSE. The actual bits assigned to each level of the hierarchy are given in Table 2. For practical applications, the tables and the graphs may be used as guidelines for bit allocation.

V. Conclusion

The Haar transform is a viable alternative to the discrete cosine transform for image compression. With proper bit-allocation, this approach has performance visually superior to that of the commonly used DCT-based JPEG algorithm at compression ratios exceeding 20:1. In terms of RMSE, the performances of the two methods are comparable. It is therefore reasonable to expect that with further refinement of the quantization and bit-allocation algorithms, the performance of the Haar transform will become superior to that of the DCT-based algorithm even in terms of RMSE.

Acknowledgment

The author thanks F. Pollara for providing the data for the application of the JPEG DCT-based algorithm.

References

[1] M. Shahshahani, "Data Compression by Wavelet Transforms," *The Telecommunications and Data Acquisition Progress Report 42-110*, vol. April–June 1992, Jet Propulsion Laboratory, Pasadena, California, pp. 179–187, August 15, 1992.

[2] I. Daubechies, "Orthonormal Bases of Compactly Supported Wavelets," *Communications in Pure and Applied Mathematics*, vol. 41, pp. 909–996, November 1988.

[3] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image Compression Through Wavelet Transform Coding," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 719–746, March 1992.

Table 1. The assignment of bits resulting in the graph shown in Fig. 5.

Compression ratio	Bit allocation
20.7	2,2,4,4,6,6
25	2,2,3,3,6,6
29.7	2,2,3,3,4,6
36.2	0,2,3,3,4,5
42.8	0,2,3,3,4,5,6,6
46.8	0,0,3,3,4,4,6,6

Table 2. The assignment of bits resulting in the plots in Fig. 8.

Compression ratio	Bit allocation
9.1	4,6,8
24.2	2,4,6,8
27.1	0,4,6,8
29.2	0,4,6,6
35.5	2,3,5,7
37.3	2,3,5,6
39.7	2,3,5,6,6
42.0	0,3,5,7
43.3	2,3,4,6
43.6	0,2,6,6
48.1	0,3,5,6,6
49.0	2,2,4,7
56.7	2,2,3,7
57.5	0,2,4,8
62.4	0,2,4,7
67.9	0,2,4,6

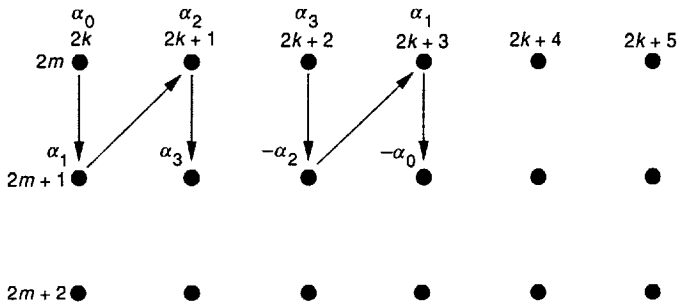


Fig. 1. Step 1 of reconfiguration.

Fig. 2. An image of peppers that was processed by a transform based on the Daubechies filter.

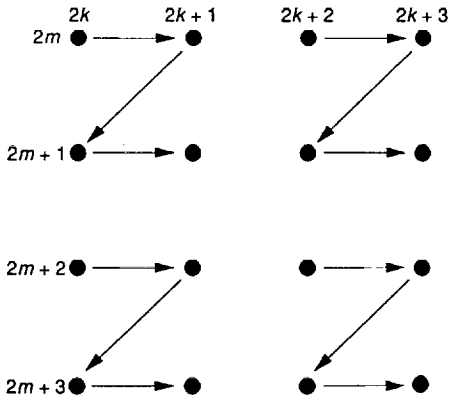


Fig. 3. Configuration of pixels for the Haar transform.

Fig. 4. Images processed by the Haar transform when (a) the compression ratio is 12:1 and (b) the compression ratio is 27:1.

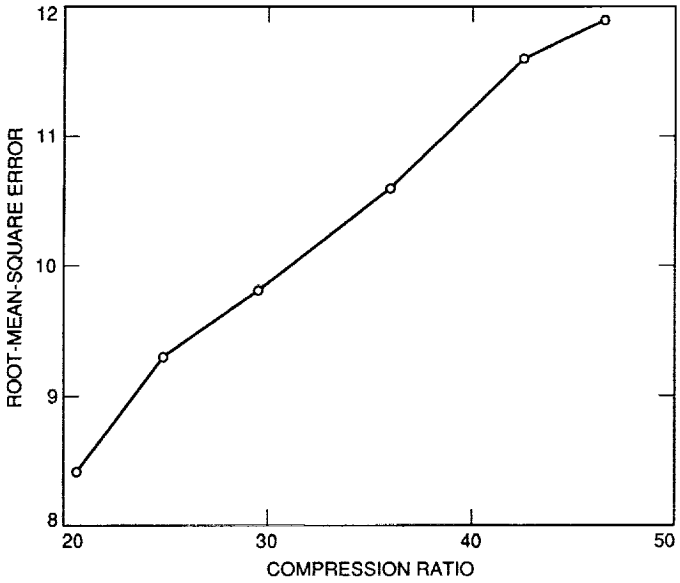


Fig. 5. Root-mean-square error versus compression ratio for good bit assignments. (See Table 1 for actual bits assigned.)

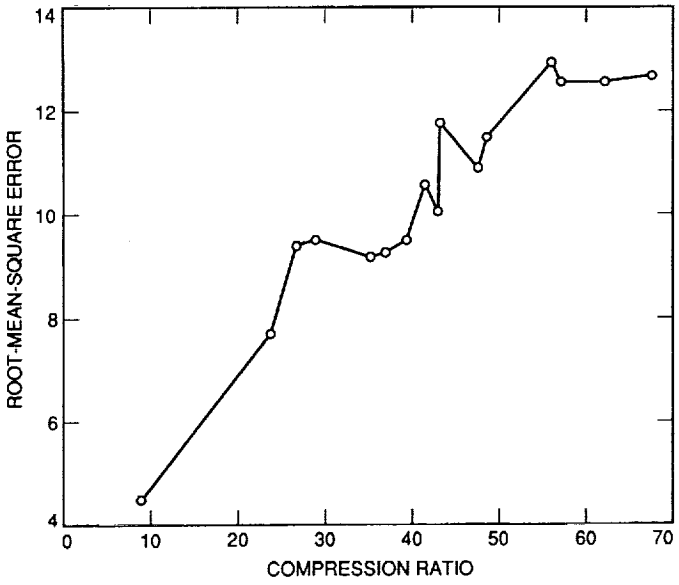


Fig. 6. The effect on root-mean-square error of different bit-allocation schemes resulting from application of the Haar transform.

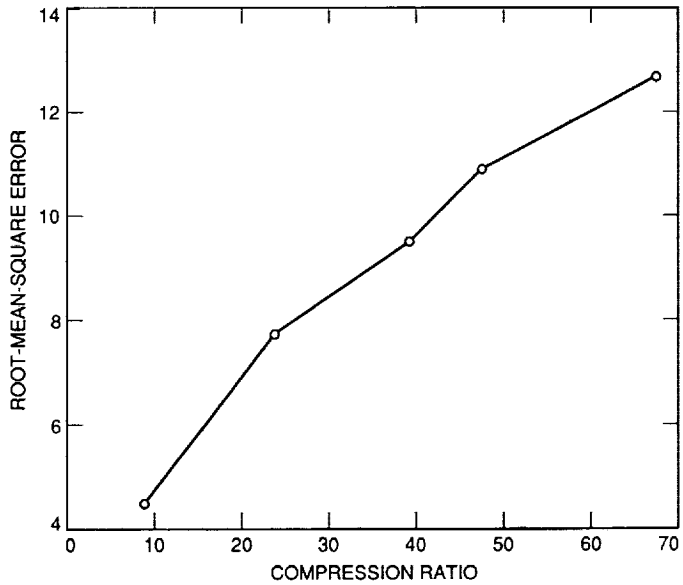


Fig. 7. Lower envelope of the curve in Fig. 6.

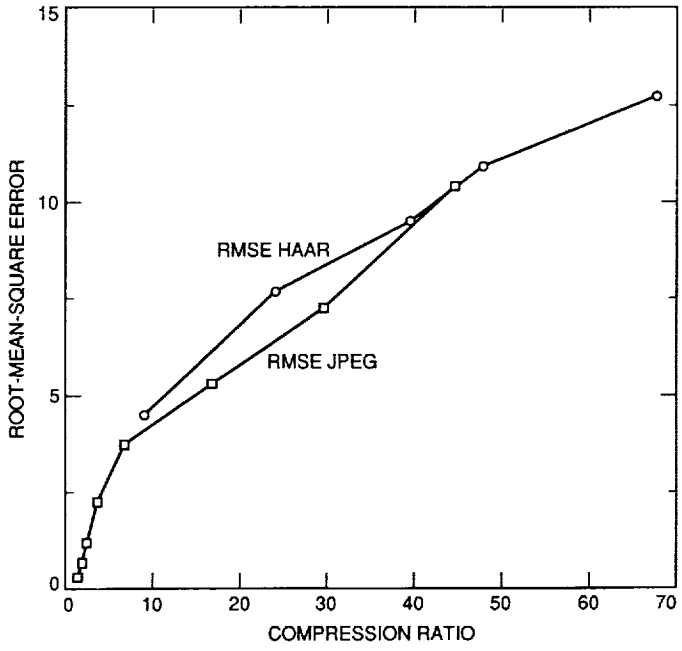


Fig. 8. Root-mean-square error comparison of the Haar transform and JPEG DCT-based algorithms.

Failure Monitoring in Dynamic Systems: Model Construction Without Fault Training Data

P. Smyth and J. Mellstrom
Communications Systems Research Section

Advances in the use of autoregressive models, pattern recognition methods, and hidden Markov models for on-line health monitoring of dynamic systems (such as DSN antennas) have recently been reported. However, the algorithms described in previous work have the significant drawback that data acquired under fault conditions are assumed to be available in order to train the model used for monitoring the system under observation. This article reports that this assumption can be relaxed and that hidden Markov monitoring models can be constructed using only data acquired under normal conditions and prior knowledge of the system characteristics being measured. The method is described and evaluated on data from the DSS 13 34-m beam waveguide antenna. The primary conclusion from the experimental results is that the method is indeed practical and holds considerable promise for application at the 70-m antenna sites where acquisition of fault data under controlled conditions is not realistic.

I. Introduction and Background

In previous articles, the problem of on-line health monitoring of a dynamic system (in particular, a DSN 34-m beam waveguide [BWG] antenna) has been investigated [1–3]. The problem can be stated in the following simple manner: let the observed data be denoted by $\underline{X}(t) = \{\underline{x}(t), \underline{x}(t-\tau), \ldots, \underline{x}(0)\}$ where each of the $\underline{x}(t)$ is a k-dimensional vector measurement of sensor data sampled at discrete time intervals τ. Given $\underline{X}(t)$, the problem is to determine the most likely current state of the system at time t, where the system is assumed to be in one of m states $\{\omega_1, \ldots, \omega_m\}$. The states are unobservable directly, but can be inferred from the observable data $\underline{X}(t)$. In probabilistic terms, the *modelling* goal is to accurately model $p[\omega_i(t)|\underline{X}(t)]$ (either from prior knowledge, training data, or a combination of both), while for *prediction*, $p[\omega_i(t)|\underline{X}(t)]$ is used to predict the current state given a specific set of data $\underline{X}(t)$ for which the system state is unknown. Typically ω_1 corresponds to the normal operating state of the system, while the other states represent various system faults that may occur. The quality of a particular model for $p[\omega_i(t)|\underline{X}(t)]$ can be obtained by measuring an empirical estimate of the *prediction accuracy*, which is simply the percentage of time that the state predicted by the model agrees with the true state—the test is performed over a period of time where the system cycles through various states (not known to the model) using data that are independent of those on which the model was trained.

In [1] and [2], an autoregressive-exogenous (ARX) time series model coupled to a pattern recognition component was used as the basis for estimating $p[\omega_i|\underline{x}(t)]$. This is a relatively simple model providing state estimates based only on instantaneous measurements $\underline{x}(t)$ but ignoring

past data. This model resulted in prediction accuracies of about 90 percent on independent test data sets obtained at DSS 13. A significant improvement on this method was reported in [3] whereby the past data were used in the state estimates by embedding the problem in a hidden Markov model (HMM) framework. The key point of the HMM method is that prior knowledge regarding the temporal behavior of the states can be used to effectively model temporal correlations in the system at the state level. On-line tests of this method at DSS 13 in November 1991 resulted in no prediction errors during a 1-hr test with state estimates being provided by the model every 6 sec [3].

It should be pointed out that the autoregressive and hidden Markov modelling methods are not the only approach for the fault detection problem. In [4] a number of statistical change detection methods were investigated. It was found that change detection methods require significant prior knowledge of the behavior of parameter characteristics when the system enters a fault state. In practice this type of detailed prior knowledge is unlikely to be available, limiting the applicability of these methods in practice.

II. Limitations of Previously Reported Methods

While the models described in [1–3] display useful capabilities in terms of on-line fault detection, they suffer from two major limitations:

(1) The models assume that the known states are exhaustive, i.e., the set of states $\{\omega_1, \ldots, \omega_m\}$ covers all possible states in which the system may be.

(2) The models also require that labelled training data are available for each state, i.e., for each state ω_i there is a set of data $\{\underline{x}(t), \underline{x}(t-\tau), \ldots, \underline{x}(0)\}$ which was measured when the system was known to be in state ω_i.

Clearly both of these requirements cannot be satisfied in most real-world fault detection applications. For fault detection, the assumption that all system states due to faults can be specified in advance is clearly inappropriate except for the simplest of systems—real-world systems (such as DSN antenna pointing systems) often contain large numbers of interacting components with feedback and nonlinearities, making prior prediction of all possible system behaviors under fault conditions unrealistic. However, it should be pointed out that it is usually possible to model system behavior under a small set of likely system faults—this point will be expanded upon later in this article.

The second requirement, that training data are available for each possible system state, is coupled to the first assumption: if all possible states cannot be described in advance, then the notion of having training data for such states is moot. However, even if the first assumption were satisfied and all fault states could be described in advance, the requirement that data can be recorded when the system is in each of these states is often unrealistic. A good example is a DSN 70-m antenna where hardware simulation of fault conditions is not a practical option due to operational considerations (as compared to the DSN 34-m antenna at DSS 13).

Hence, there is considerable practical motivation to develop methods that relax the assumptions on which the earlier-reported models are based, while still retaining the accurate prediction capabilities of these models. This article describes a relatively simple yet effective method that can detect the presence of states for which no training data sets were available, i.e., states about which the model has no knowledge. It is assumed that training data (or else a strong prior model) for at least one state is available—this is not restrictive since data under normal conditions are almost always available. The proposed method is based on the use of prior knowledge to constrain the possible distribution of system parameters, which when coupled with the model derived from the training data, allows detection of both known states and a generic, unknown state category.

This article outlines the general model, illustrates its use and effectiveness on data collected from the elevation axis of the DSS 13 BWG antenna pointing servomechanism, and describes the limitations of the current approach.

III. Notation and Assumptions

For the purposes of this article, the distinction is made between the observable data at time t which is $\underline{x}(t)$ and the estimated parameters at time t, denoted by the vector $\underline{\theta}(t)$. Typically $\underline{x}(t)$ is the original sensor data or time series (such as the motor current in an antenna pointing system), whereas the values of $\underline{\theta}(t)$ are typically statistical estimates of some characteristics of the time series such as the mean, variance, or autoregressive (AR) coefficients. In this article, attention will be limited to *block* estimation methods whereby $\underline{\theta}(t) = f[\underline{x}(t), \underline{x}(t-\tau), \ldots, \underline{x}(t-N\tau)]$, etc. Hence, each of the parameter estimates is derived from disjoint windows or blocks of the original data, where N is chosen to be large enough to enable reasonably reliable statistical estimates.

Let $\Phi_t = \{\underline{\theta}(t), \underline{\theta}(t-N\tau), \ldots, \underline{\theta}(0)\}$. In effect, Φ_t is then viewed as the observable data sequence and the prob-

lem can be treated as that of recovering the likely system states given the estimates Φ_t, i.e., find $p[\omega_i(t)|\Phi_t]$. Issues such as choosing appropriate estimators, block sizes, etc., will not be dealt with in this article. For the experimental results reported later in this article, values of $\tau = 20$ msec and $N = 200$ are used. However, for the purposes of simplification of notation, it will be assumed without loss of generality that $N\tau = 1$ during the development of the probabilistic models that follow.

It is also assumed that there are $m-1$ states for which prior information is available either in the form of: (1) specific parametric models for the dependence of the states on the observable data, or (2) training data. An additional mth state is used in the model as a single state which accounts for all other possible behaviors of the system that are qualitatively different from the known states. This state will be referred to as the unknown fault state. Hence, in the simplest case, for example, if prior information is only available for the normal state, then the model has two states: normal and the unknown fault state.

IV. The General Model

The goal of the modelling process is to provide a means of estimating the posterior state probabilities

$$p[\omega_i(t)|\Phi_t] = p[\omega_i(t)|\underline{\theta}(t), \underline{\theta}(t-1), \ldots, \underline{\theta}(0)] \qquad 1 \leq i \leq m \tag{1}$$

which are required for prediction. In the Appendix it is shown how the hidden Markov framework can be used such that the full number of conditioning terms in Eq. (1) is not necessary if the appropriate assumptions are met. The hidden Markov model leads to recursive estimates of the form

$$p[\omega_i(t)|\underline{\theta}(t), \underline{\theta}(t-1), \ldots, \underline{\theta}(0)] \approx p[\underline{\theta}(t)|\omega_i(t)] \times \sum_{j=1}^{m} f\Big(p[\omega_j(t-1)|\underline{\theta}(t-1), \ldots, \underline{\theta}(0)]\Big) \tag{2}$$

so that knowledge of the likelihood $p[\underline{\theta}(t)|\omega_i(t)]$ at each time t (in addition to the Markov transition matrix $\mathbf{A}$) is *sufficient* to calculate the posterior estimates.

Note that it will be assumed that the statistics of interest are time-invariant, hence reference to a specific time t can be dropped at this point.

In previous work, direct forward models of $p(\omega_i|\underline{\theta})$ were estimated and then $p(\underline{\theta}|\omega_i)$ was estimated by the use of Bayes' rule in Eq. (2) [3]. In this article, it is proposed to use models of the form $p(\underline{\theta}|\omega_i)$ as the direct basis for the model. The rationale behind this approach is simple: based on prior knowledge alone it is impossible except in simple cases to specify the form of $p(\omega_i|\underline{\theta})$. However, it is much more likely that one can model the dependence of the data on the state, i.e., a prior density can be assigned to the likelihood $p(\underline{\theta}|\omega_i)$ based on prior knowledge. In particular, for state ω_m, which is the state that covers all possible states not included in the set $\{\omega_1, \ldots, \omega_{m-1}\}$, one can typically specify a noninformative uniform prior density over the set of possible parameter values for $\underline{\theta}$. In addition, one must also supply models for $p(\underline{\theta}|\omega_i)$, $1 \leq i \leq m-1$, which are typically estimated from training data.

The key difference between this method and those methods proposed in previous literature is that the model works with likelihoods (the probability of the observable data given the states), rather than directly with the posterior state probabilities (the probabilities of the states given the data). This approach rules out the use of many discriminant-based methods that only provide estimates of the posterior probabilities, but do not provide estimates of the likelihoods (for example, logistic regression, feedforward neural networks, decision trees, etc.). Methods that provide the required estimates include (naturally) both parametric methods (such as maximum likelihood classifiers based on a specific parametric form for $p(\underline{\theta}|\omega_i)$ and nonparametric methods such as kernel density estimators. For a more extensive general discussion of the differences between such models, see [5–7].

The proposed method can be summarized as follows:

(1) Specify or estimate prior density models, $p(\underline{\theta}|\omega_i)$, for the *known* classes, $\omega_1, \ldots, \omega_{m-1}$. As mentioned above, this requires the use of either a parametric model (such as a multivariate Gaussian) or a nonparametric density estimation method.

(2) Specify a prior density for $p(\underline{\theta}|\omega_m)$ where ω_m is the special unknown state. This is typically done by establishing bounds or constraints on each of the parameters in $\underline{\theta}$ and then (in the absence of any other information) specifying a uniform density over the bounded parameter space.

(3) The remainder of the method is the same as before: simply estimate the hidden Markov model parameters from reliability data (as described in the Appendix) and run the model for prediction.

Note in step (2) it is important that the derived parameters can be bounded in some manner. The stronger the

constraints that can be placed on the parameters, the better will be the detection performance of the model. These constraints could be due to the basic physics of the system, such as energy limits, or a function of the particular representation being used, such as spectral or autoregressive estimates (a specific example is provided in Section V). If there are no constraints at all, then it is still possible to specify a prior model, such as a Gaussian model, although the choice of model may now be somewhat more problematic since it will inevitably reflect a prior bias which may not be appropriate. A better alternative (in the case of no constraints) would be not to use a prior model at all for ω_m and just detect data which appear to be outliers from the other $m-1$ models. However, outlier detection can be problematic—it is a central theme of this article that prior constraints can usually be placed on the parameter space of interest and that this provides the natural avenue for detecting data from ω_m. In essence, it is argued that if such prior constraints exist, this information should be used in the model, and should in principle provide better detection capabilities than any outlier detection method.

V. Applying the Likelihood Method in Practice

One significant difference between modelling the likelihoods and posterior probabilities is the issue of dimensionality, namely, that a high-dimensional parameter space will be potentially more problematic for the likelihood modelling method than for the posterior (or discriminant) modelling method. In a discriminant model (which calculates posterior probabilities), input dimensions can be ignored in the model if they are irrelevant to the state, allowing more efficient estimation at small training sample sizes. However, in the likelihood model, all input dimensions must be included in the model. If there are a significant number of irrelevant or redundant inputs, this can lead to a poor model, particularly as the ratio of sample size to input dimensions gets small. Hence, parsimony in parameter choice is recommended.

From previous work with the DSS-13 BWG-antenna pointing system, it has been found that autoregressive coefficients and standard deviation estimates are both particularly useful characteristics of the motor current for the purpose of detecting abnormal events [2,3]. In this article, three such characteristics as estimated from the motor current signal will be chosen: the two coefficients of a second-order autoregressive model [AR(2)], ϕ_1 and ϕ_2, and the standard deviation, σ. Hence, $\underline{\theta} = (\phi_1, \phi_2, \sigma)$. In [2] and [3], an eighth-order ARX model was used to model the motor current signal, using the rate command as the forcing term. However, in the interests of keeping the input dimensionality relatively low, a simpler AR(2) model was used for the purposes of this experiment. While the simpler model is not appropriate for complete system identification, it is sufficient for the purposes at hand to extract useful signal characteristics that can be used to discriminate between normal and abnormal operating conditions.

The next step is to specify a prior density over the AR parameters ϕ_1 and ϕ_2. In accordance with standard time series theory, if the estimated process (as represented by the two coefficients) is to be stationary, then the coefficients must obey the following restrictions [8]:

$$\phi_1 + \phi_2 < 1$$

$$\phi_2 - \phi_1 < 1$$

$$-1 < \phi_1 < 1$$

It will be assumed that the estimated coefficients are in fact stationary, thus providing bounds on the possible parameter values (see Fig. 1). A uniform density is specified over all such allowable values of ϕ_1 and ϕ_2. Of course, this does not allow for the fact that in practice (and in particular for fault conditions) there is no guarantee that the estimated coefficients will obey these bounds. The following approach is adopted: if the estimated coefficients lie outside the bounds of the stationary region, then the probability of the normal state $p(\omega_1|\underline{\theta})$ is set to zero.

The third parameter, the standard deviation of the voltage from the Hall effect sensor, which measures motor current, is about 20 mV under normal conditions. Based on experience from observing the motor current signal under a variety of conditions, it is estimated that under any fault condition the standard deviation should not exceed 1 V. Hence, in the absence of any other prior information, a uniform density is placed on the standard deviation over this range 0 to 1 V for σ. This density is assumed independent of the AR(2) coefficient density. This completes the specification of the likelihood model for ω_m.

For the other $m-1$ states, normal and any known fault conditions, likelihood models can be found via the use of Gaussian assumptions with maximum likelihood parameter estimation or nonparametric density estimation.

VI. Experimental Results

In [2] the acquisition of data at DSS 13 was described. Specifically, sensor data were measured under controlled conditions from the elevation axis servomechanism of the 34-m BWG antenna. Data are available for two different

days of antenna operation, referred to as day 42 and day 53. Data were recorded for about 30 min with four different fault conditions present. The faults are: tachometer noise, tachometer failure, compensation loss in the amplifier, and encoder failure. The fourth fault, encoder failure, was a real fault that was subsequently repaired. It shows up in the data as being intermittent in nature. The other three faults were purposely introduced into the hardware in a controlled manner.

The same model for the prior likelihood $p(\underline{\theta}|\omega_m)$ as described in Section V was used in all experiments. The Markov transition matrix was set to have probability of 0.99998 of remaining in the normal state, which corresponds to a mean time between failure of about 2 days. The probability of transiting to any particular fault state was set uniformly, and the probability of remaining in a fault state was set to 0.95 (corresponding to a mean failure duration of 1 min before shutdown occurred).

A model was trained on normal data and on one of the known faults (the compensation loss fault), giving a three-state model (normal, known fault, and unknown fault). The normal and known fault likelihood models were constructed using a multivariate Gaussian density where the mean and covariance parameters were estimated from the data using maximum likelihood estimators.

Two models were constructed in this manner (one on each of the day 42 and day 53 data sets) and then tested on the independent data from the other day. The goal of the experiment was to see if the model could correctly identify data as being either normal, a known fault, or an unknown fault. The ability to classify data into the third unknown category was of particular relevance, since, as described earlier in this article, previously developed models did not have this capability, i.e., all data were classified into one of the known states.

The state sequence in each test data set was as follows: normal, unknown fault (tachometer failure); known fault (compensation loss); normal conditions, unknown fault (tachometer noise); and finally an unknown intermittent fault (encoder failure). Each state lasted roughly 5 min in duration. Figure 2 shows how one of the AR(2) coefficients changed as a function of the underlying state (for day 53). Note how noisy the estimates are, due in part to the fact that an AR(2) model is too simple to capture the full dynamics of the data.

Figures 3(a) and 3(b) show the state estimation results in terms of estimated state probabilities [as in Eq. (2)] for each of the three states in the model. The results clearly indicate that the likelihood model has the ability to infer the correct state of the system from the observable data. As in [3], the Markov model adds stability to the estimates, reducing false alarms while still allowing a rapid transition when the underlying state changes.

The important aspect of this new model is its ability to identify data as being of the unknown category, namely, between minutes 5 and 10, minutes 20 and 25, and the intermittent fault that occurred between minutes 25 and 30. The response of the model is not entirely perfect. For example, during the test of day 42 data [Fig. 3(a)], in the first 5 min of normal operations, there appear to be at least two short false alarms, i.e., where the probability of normal conditions drops significantly below 1 even though the system is supposed to be in the normal state. This can be attributed to one of two possible causes: either the model is not quite accurate, or, more interestingly, although the system is assumed to be normal it is in fact in some other transient state. Closer examination of the original sensor data revealed that the second explanation was more likely to be true: the model detected the possibility of an unknown transient state that had not been noticed when these data were originally recorded. While this is a relatively simple example, it nonetheless demonstrates the basic concept of a model which can detect subtle changes and abnormalities in the behavior of a dynamic system—changes that are not noticeable to the human observer.

It is also worth noting that the present model assigns a relatively low probability to short a priori states. An obvious extension to the model proposed here would be to further refine the unknown state into substates based on their temporal characteristics, i.e., intermittent or transient, or permanent.

VII. Discussion

This article has described the basic principles behind the construction of dynamic system monitoring models that can classify system states into an "unknown" category. Although the basic idea is quite simple, it has some very useful properties. In addition, it is worth noting that all previous fault monitoring methods described in the literature (of which the authors of this article are aware) implicitly assume that all system states of interest are known in advance. For large-scale complex systems, this is clearly an undesirable and unrealistic assumption.

A possible criticism of the proposed method is the possibly arbitrary nature by which the prior density for the unknown state is assigned. Certainly it must be admitted that this can never be a purely objective choice and requires the careful judgment of the modeller. However, any

model is by nature the result of various implicit biases and subjective judgments, and, hence the standard argument of the Bayesian school of statistical modelling can be appealed to: if any reasonable prior information exists, then it is judicious to include it in the model. The astute reader will have noted that by simply changing the boundaries or constraints on the parameters of interest, the modeller can in effect control the detection to false alarm trade-off characteristic of the model [also known as the receiver operating characteristic (ROC) in signal detection theory]. The use of decision theoretic methods to minimize the relevant loss function (in the context of choosing the prior density) would seem the appropriate avenue by which to control this aspect of the model.

The ability to detect new system states does not come without a cost. As alluded to earlier, the mapping describing how the observed data depend on the system states (the likelihood) is generally more difficult to estimate than the mapping describing how the states depend on the observed data. Hence, for example, in the case where one has three known faults, a model of the type which is proposed here may not be as accurate in terms of discriminating among these faults as the types of discrimination models that focus exclusively on these faults but which ignore the possibility of an unknown fault. One way to avoid this problem is to improve the quality of the likelihood modelling process. For example, a Gaussian assumption is often not appropriate: nonparametric density estimation methods, if used correctly, may provide more accurate models for the known states.

Another possibility would be to use both likelihood models and discriminative models as part of one overall model. Letting $p(\omega_m)$ be the posterior probability that the data are from an unknown state (as calculated by a likelihood model of the type described in this article), and letting the symbol $\omega_{\{1,\ldots,m-1\}}$ denote the event that the true system state is one of the known states, one can estimate the true posterior probability of individual known states as

$$p_d(\omega_i|\Phi,\omega_{\{1,\ldots,m-1\}}) \times [1 - p(\omega_m)], \quad 1 \leq i \leq m-1$$

where $p_d(\omega_i|\Phi,\omega_{\{1,\ldots,m-1\}})$ is the posterior probability estimate of the known states as provided by a discriminative model such as described in [2] and [3]. Note that this method does not in any way help to improve the ability of the overall model to detect unknown states since that estimate remains unchanged; however, in principle, it should improve the ability of the model to distinguish between specific known states. The possibility of improving the model described in this article by using this particular technique has not been tested in an experimental manner at this point.

A final comment is that the ability of the likelihood model to detect unknown states is necessarily limited by the information in the observable data. For example, although the simple AR models reported here have given very useful information in terms of discriminating between normal and various fault states, it is quite possible that a fault state may not be well modelled by a simple linear AR model, i.e., that the AR coefficients will not yield any useful information. Hence, in general, the use of more robust signal characteristics should improve the model performance.

VIII. Summary

A new method was proposed that allows the construction of HMM monitoring algorithms without the requirement that training data for each of a prescribed set of faults be made available. Naturally, if such data (or equivalent prior knowledge) are available, then these data can also be incorporated into the new model. The proposed method was validated on data from the DSS-13 BWG-antenna pointing system. In particular, the model was able to detect system states that could not have been detected using previously reported methods. While there is still room for improvement in terms of the performance of this class of models, the results are nonetheless quite accurate and of significant practical importance in the context of monitoring 70-m antenna data where fault training data are unlikely to be available.

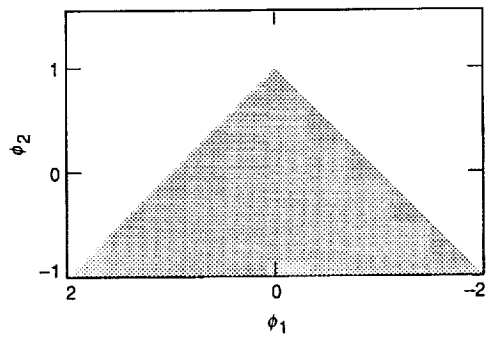


Fig. 1. Admissible region for AR(2) parameters.

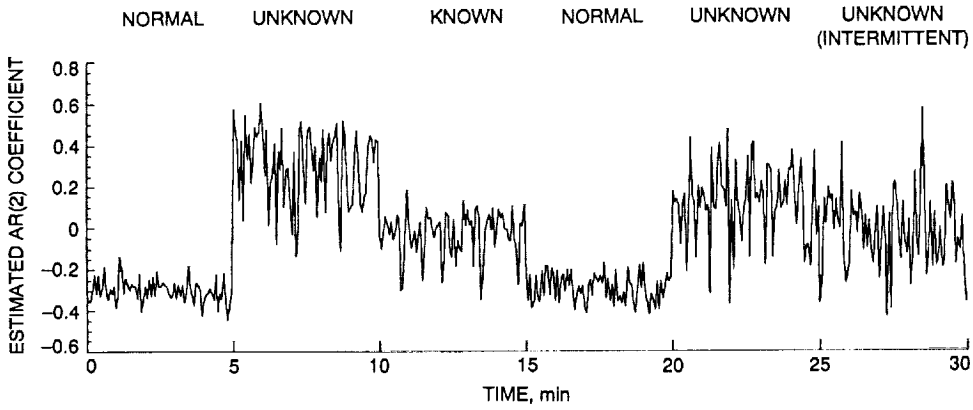


Fig. 2. Estimates of coefficient ϕ_2 as a function of the system state (day 53 data).

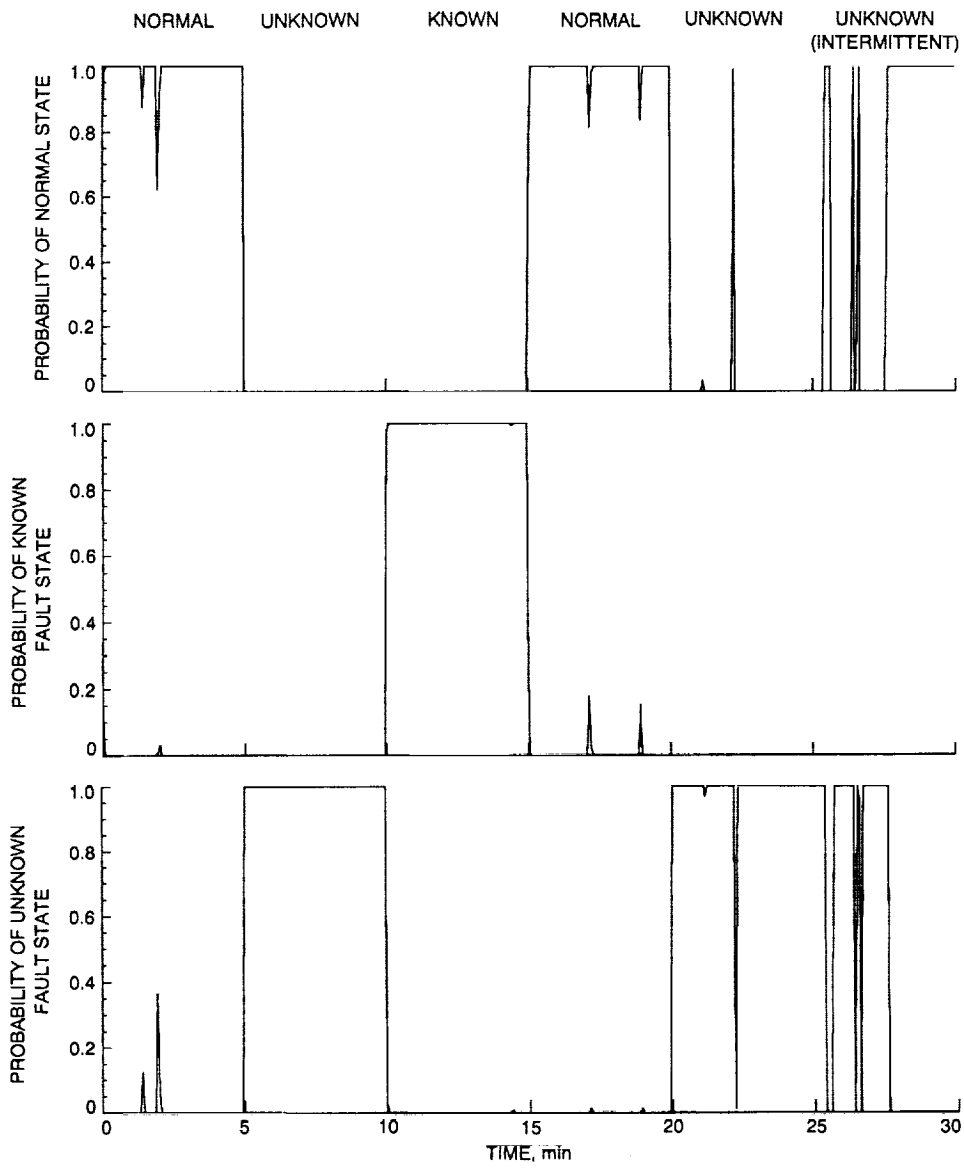


Fig. 3. Estimates of posterior state probabilities as provided by the likelihood and hidden Markov models: (a) training on day 53 and testing on day 42 data and (b) vice versa.

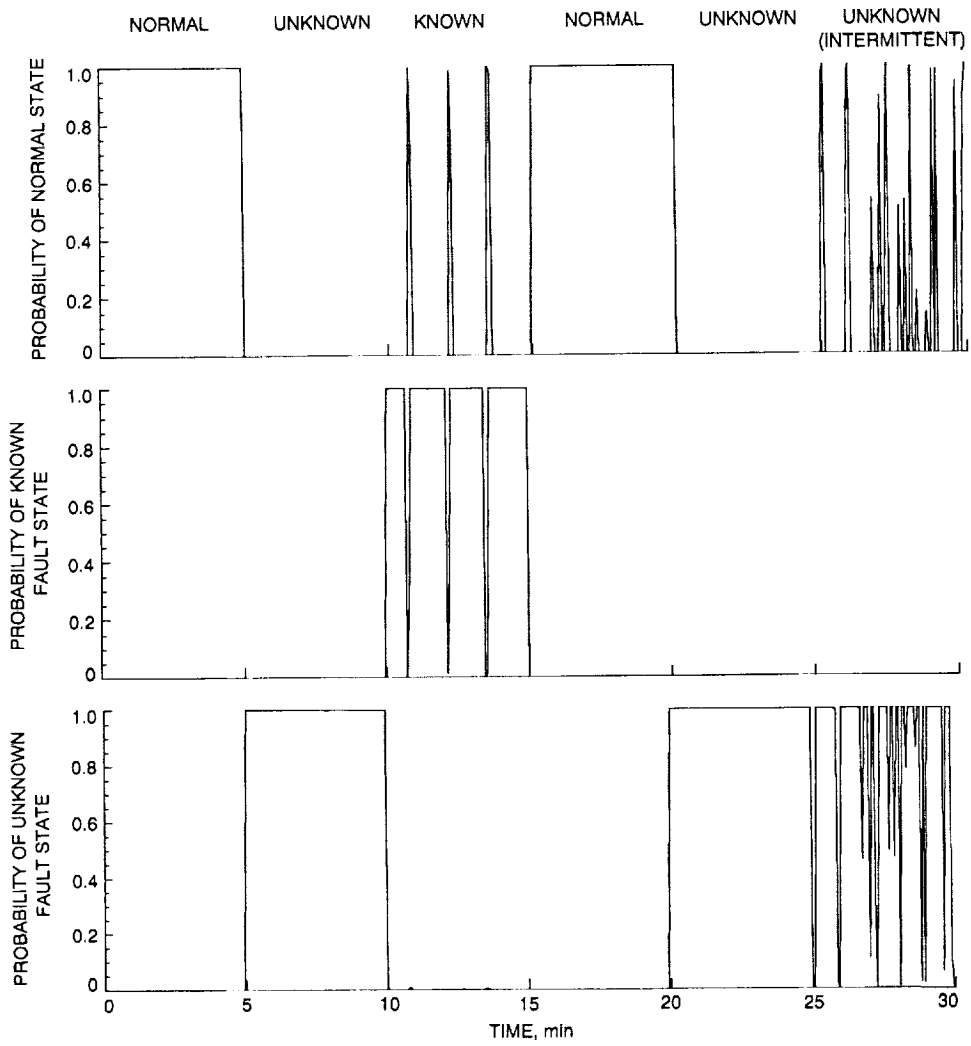

NORMAL
UNKNOWN
KNOWN
NORMAL
UNKNOWN
UNKNOWN (INTERMITTENT)
PROBABILITY OF NORMAL STATE
PROBABILITY OF KNOWN FAULT STATE
PROBABILITY OF UNKNOWN FAULT STATE
1.0
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
TIME, min

Fig. 3. (contd)

Appendix

Hidden Markov Model Description

Let Ω denote the discrete-valued state variable taking values in the set $\{\omega_1, \ldots, \omega_m\}$. A first-order discrete-time Markov model is characterized by the assumption that

$$p[\Omega(t)|\Omega(t-1), \ldots, \Omega(0)] = p[\Omega(t)|\Omega(t-1)] \qquad \text{(A-1)}$$

That is, that the conditional probability of any current state given knowledge of all previous states is the same as the conditional probability of the current state given knowledge of the system state at time $t-1$. This is equivalent to the well-known assumption of "memorylessness" in that the evolution of the system depends only on the present state and not on the past state. A direct consequence is the fact that the number of consecutive time steps that the system spends in any given state will be a discrete random variable with a geometric distribution.

In a standard, *nonhidden* Markov model, to calculate the probability that the system is in a given state at time t, one needs only to know the initial state probabilities $\pi = p[\omega_1(0), \ldots, \omega_m(0)]$ and the values $a_{ij} = p[\omega_i(t)|\omega_j(t-1)]$, $1 \leq i, j \leq m$. The $m \times m$ matrix $\mathbf{A}$ is known as the transition matrix and characterizes the Markov model. The first-order Markov assumption governing state evolution in time may appear restrictive at first glance, but has been found in practice to be an extremely robust model for many real-world applications. In principle, the theory for higher-order Markov models can be developed, but at the cost of increased complexity in terms of specifying the model and of increased computational complexity in terms of on-line calculation of the posterior probabilities.

Under the first-order Markov assumption, it can easily be shown that the probability in which the system remains in the normal state from one instant to the next can be expressed as

$$a_{11} = 1 - \frac{\tau}{\text{MTBF}} \qquad \text{(A-2)}$$

where the MTBF is the mean time between failures of the system and τ is the time between states (both expressed in the same units). Similarly, the other elements in the transition matrix $\mathbf{A}$ can be estimated from information concerning the general nature of system faults, which may be available from an existing database or can be estimated based on known physical properties of the system. Augmented models may have a wide variety of additional states. For example, it may be useful to include a state to account for the transient behavior of the system. Similarly, states which account for known operational modes of the system, such as powered off and brakes on, may be necessary in practice. The specification of the Markov transition matrix corresponds to the explicit modelling of high-level prior knowledge concerning system behavior at the *state* level. In particular, it does not involve the specification of prior models for *observable data* over time since typically this is much more difficult to model. This is precisely the advantage of the HMM decomposition: the temporal behavior of the system needs only to be specified at a relatively high level.

Denote the observed data up to time t to be $\Phi_t = \{\underline{\theta}(t), \ldots, \underline{\theta}(0)\}$. The hidden aspect of the Markov model is derived from the fact that the observed data Φ_t is a stochastic function of the underlying Markov states. These states are hidden in the sense that they cannot be measured directly. It is the state identities which one wishes to estimate, hence, the purpose of the modelling is to represent the relationship between the states and the observable data such that the most likely state sequence can be inferred. Figure A-1 shows an illustration of the concept for a three-state HMM. For on-line monitoring of a dynamic system, the observed data simply consist of observed sensor data (or derived parameters) while the states reflect the underlying system states, in particular normal and fault operational states.

An estimate of the instantaneous likelihood, the probability of the observed data at time t conditioned on the state variable, $p[\underline{\theta}(t)|\Omega(t)]$, is assumed to be known. The goal is to take advantage of *all* the symptom information and to estimate $p[\Omega(t)|\Phi_t]$. It is convenient to work in terms of an intermediate variable α, where

$$\alpha_i(t) = p[\omega_i(t), \Phi_t] \qquad \text{(A-3)}$$

To find the posterior probabilities of interest, it is sufficient to be able to calculate the α's at any time t, since by Bayes' rule

$$p(\omega_i(t)|\Phi_t) = \frac{1}{p(\Phi_t)}\alpha_i(t) = \frac{\alpha_i(t)}{\sum_{j=1}^{m}\alpha_j(t)} \tag{A-4}$$

The derivation of a recursive estimate follows:

$$\alpha_i(t) = \sum_{j=1}^{m} p\Big(\omega_i(t), \Phi_t, \omega_j(t-1)\Big) = \sum_{j=1}^{m} p\Big(\omega_i(t), \underline{\theta}(t), \Phi_{t-1}, \omega_j(t-1)\Big)$$

$$= \sum_{j=1}^{m} p\Big(\omega_i(t), \underline{\theta}(t)|\Phi_{t-1}, \omega_j(t-1)\Big)\alpha_j(t-1)$$

by the definition of α_j

$$= \sum_{j=1}^{m} p\Big(\underline{\theta}(t)|\omega_i(t), \Phi_{t-1}, \omega_j(t-1)\Big)p\Big(\omega_i(t)|\Phi_{t-1}, \omega_j(t-1)\Big)\alpha_j(t-1)$$

$$= \sum_{j=1}^{m} p\Big(\underline{\theta}(t)|\omega_i(t)\Big)p\Big(\omega_i(t)|\Phi_{t-1}, \omega_j(t-1)\Big)\alpha_j(t-1)$$

assuming that $\underline{\theta}(t)$ is independent of past observations and past states, *given* the present state

$$= \sum_{j=1}^{m} p\Big(\underline{\theta}(t)|\omega_i(t)\Big)p\Big(\omega_i(t)|\omega_j(t-1)\Big)\alpha_j(t-1)$$

assuming that $\omega_i(t)$ is independent of past observations *given* the past states

$$= p\Big(\underline{\theta}(t)|\omega_i(t)\Big)\sum_{j=1}^{m} a_{ij}\alpha_j(t-1) \tag{A-5}$$

The first term is the likelihood (assumed to be known). The terms in the sum are just a linear combination of the α's from the previous time step. Hence, Eq. (A-5) provides the basic recursive relationship for estimating state probabilities at any time t.

The additional assumptions made in the derivation of Eq. (A-5) (besides the first-order Markov assumption on state dependence) require some comment. The first assumption is that $\underline{\theta}(t)$ is independent of both the most recent state and the observed past data, *given* that the present state is known. This implies that the observed symptoms are assumed to be statistically independent from one time window to the next, given the state information. This will generally be true when the values of $\underline{\theta}(t)$ consist of derived parameters and τ is much greater than any significant time constants of the dynamic system. Even if it is known that the $\underline{\theta}$'s exhibit temporal correlations, this can also in principle be modelled directly in Eq. (A-5), although the model will now be much more complex. The second assumption, that the present state only depends on the previous state but not the past observations, simply reflects the causal relationship between symptoms and states.

Note that the method described above only calculates the state probabilities based on *past* information. Alternative estimation strategies are possible. For example, using the well-known forward-backward recurrence relations [9], one can update the state probability estimates using symptom information which occurred *later* in time.

References

[1] P. Smyth and J. Mellstrom, "Initial Results on Fault Diagnosis of DSN Antenna Control Assemblies Using Pattern Recognition Techniques," *The Telecommunications and Data Acquisition Progress Report 42-101*, vol. January–March 1990, Jet Propulsion Laboratory, Pasadena, California, pp. 136–151, May 15, 1990.

[2] J. Mellstrom and P. Smyth,"Pattern Recognition Techniques Applied to Performance Monitoring of the DSS 13 34-m Antenna Control Assembly," *The Telecommunications and Data Acquisition Progress Report 42-106*, vol. April–June 1991, Jet Propulsion Laboratory, Pasadena, California, pp. 30–51, August 15, 1991.

[3] J. Mellstrom, C. Pierson, and P. Smyth, "Real-time Antenna Fault Diagnosis Experiments at DSS 13," *The Telecommunications and Data Acquisition Progress Report 42-108*, vol. October–December 1991, Jet Propulsion Laboratory, Pasadena, California, pp. 96–108, February 15, 1992.

[4] P. Scholtz and P. Smyth, "Fault Detection Using a Two-Model Test for Changes in the Parameters in a Time Series," *The Telecommunications and Data Acquisition Progress Report 42-110*, Jet Propulsion Laboratory, Pasadena, California, August 15, 1992.

[5] P. Smyth and J. Mellstrom, "Fault Diagnosis of Antenna Pointing Systems Using Hybrid Neural Networks and Signal Processing Techniques," *Advances in Neural Information Processing Systems 4*, J. E. Moody, S. J. Hanson, and R. P. Lippmann (eds.), Los Altos, California: Morgan Kaufmann Publishers, pp. 667–674, 1992.

[6] P. Smyth and J. Mellstrom, "Detecting Novel Classes With Applications to Fault Diagnosis," *Proceedings of the Ninth International Conference on Machine Learning*, Los Altos, California: Morgan Kaufmann Publishers, pp. 416–425, 1992.

[7] P. Smyth, "Probability Density Estimation and Local Basis Function Neural Networks," *Computational Learning Theory and Natural Learning Systems 2*, T. Petcshe, M. Kearns, S. Hanson, and R. Rivest (eds.), Cambridge, Massachusetts: MIT Press, in press, 1993.

[8] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco, California: Holden-Day, 1970.

[9] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

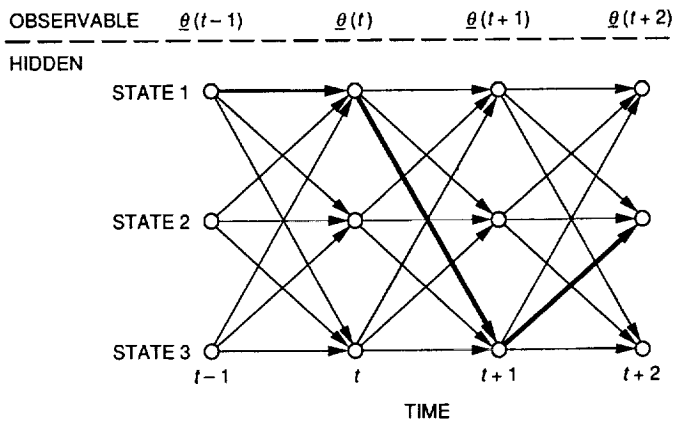


Fig. A-1. An illustrative example of a three-state hidden Markov model.

A Functional Description of the Buffered Telemetry Demodulator (BTD)

H. Tsou, B. Shah, R. Lee, and S. Hinedi
Communications Systems Research Section

This article gives a functional description of the buffered telemetry demodulator (BTD), which operates on recorded digital samples to extract the symbols from the received signal. The key advantages of the BTD are (1) its ability to reprocess the signal to reduce acquisition time, (2) its ability to use future information about the signal and to perform smoothing on past samples, and (3) its minimum transmission bandwidth requirement as each subcarrier harmonic is processed individually. The first application of the BTD would be the Galileo S-band contingency mission, where the signal is so weak that reprocessing to reduce the acquisition time is crucial. Moreover, in the event of employing antenna arraying with full spectrum combining, only the subcarrier harmonics need to be transmitted between sites, resulting in significant reduction in data rate transmission requirements. Software implementation of the BTD is described for various general-purpose computers.

I. Introduction

Current DSN receivers operate on the received signal in real time and do not buffer or store the received signal for reprocessing. As a result, data lost during acquisition or cycle slips is not recoverable. In some DSN missions, the link margin is so low that signal acquisition (carrier, subcarrier, and symbol) takes too long, resulting in significant data loss. Such a scenario occurs during operation at low data rates (on the order of a few hundred symbols per second) with rate 1/4, 1/5, and 1/6 convolutional codes, resulting in symbol signal-to-noise ratios (SNR's) between −4 dB and −6 dB. This combination of low symbol SNR and low data rate produces unusually long acquisition times dominated by the behavior of the subcarrier- and symbol-synchronization loops. The acquisition time can, in some instances, be as long as 20 minutes and depends on the scenario of interest. The problem is augmented when suppressed carriers are employed, since the Costas loop SNR is a nonlinear function of the symbol SNR; significant squaring loss results. When the data rate on the spacecraft is programmed to take advantage of the antenna aperture on the ground, several data rate changes can occur in a pass or a single day, each resulting in data loss due to resynchronization in the ground receiver.

The buffered telemetry demodulator (BTD) overcomes these problems by recording the signal and then processing and reprocessing the samples. Once acquisition occurs, the BTD can operate on the past data, using estimates of the signal state (such as phase, frequency, and Doppler rate) to basically go back in time and recover those symbols lost during acquisition. This is referred to as smoothing,

where "future" data are used to provide an estimate in the "past." The optimum strategy on the data playback is still under investigation and will be the topic of a future report. The BTD is designed mainly for low-data-rate applications and can be implemented on a general-purpose workstation. In order to reduce the computational requirement, the signal is recorded on a subcarrier harmonic basis, thus rendering the BTD independent of the subcarrier frequency. This results in a computational throughput requirement based on the data rate rather than on the bandwidth of the transmitted signal. In addition, when antenna arraying is employed and full spectrum combining is utilized, only a fraction of the total signal bandwidth needs to be transmitted between sites, resulting in significant reduction in transmission requirements. In this article, the BTD is functionally described and its implementation on various workstations is evaluated. First, the carrier and subcarrier synchronization loops are reviewed. Then, the recording of the signal is described; each subcarrier harmonic is individually demodulated to baseband and recorded. Finally, the coherent baseband demodulation is discussed along with requirements for software implementation of the BTD. Also, a glossary is included.

II. A Review of the Advanced Receiver Carrier and Subcarrier Loops

The advanced receiver II (ARX II) [1] is a receiving system that has been developed to digitally demodulate and process signals from deep-space spacecraft. It is a breadboard system for the Block V receiver, which will eventually replace various receivers currently used in the DSN [2]. This section briefly reviews the ARX II carrier and subcarrier loops so that they can be compared to the carrier and subcarrier loops of the BTD. The main difference between the ARX II and BTD is that the ARX II carrier and subcarrier loops process the received signal after it has been open-loop downconverted to an intermediate frequency (IF), whereas the BTD loops process the received signal after it has been open-loop downconverted to baseband on a subcarrier harmonic basis. All the other receiving functions, e.g., symbol synchronization, symbol detection, lock detection, etc., are implemented the same way in both the ARX II and BTD. Consequently, they are not discussed in this article. Instead, the interested reader is referred to [1] for a complete review of these functions.

A block diagram of the ARX II carrier and subcarrier loops is shown in Fig. 1. The received signal, which is assumed to be a typical deep space signal, consists of a residual sinusoidal carrier and a square-wave subcarrier. After downconversion to an appropriate IF, it can be represented as

$$r(t) = \sqrt{2P}\cos(\Delta)\sin[\Psi_c(t)] + \sqrt{2P}\sin(\Delta)d(t)\ \mathrm{sgn}\{\sin[\Psi_{su}(t)]\} \times \cos[\Psi_c(t)] + n(t) \quad (1)$$

where $P_c = P\cos^2(\Delta)$ and $P_d = P\sin^2(\Delta)$ in watts are the received carrier power and data power, respectively; $\Psi_c(t)$ is the total carrier phase; and $\Psi_{su}(t)$ is the total subcarrier phase, both in radians. The data, $d(t)$, are given by

$$d(t) = \sum_{l=-\infty}^{\infty} a_l p(t - lT) \quad (2)$$

where $p(t)$ is the unit-power square pulse limited to T seconds and $\{a_l\}$ represents the independent and equally likely binary (± 1) symbols. The noise process, $n(t)$, is modeled as bandpass with a flat, one-sided power spectral density (PSD) level equal to N_0 W/Hz. By expressing the square-wave subcarrier as an infinite sum, i.e.,

$$\mathrm{sgn}\{\sin[\Psi_{su}(t)]\} = \frac{4}{\pi}\sum_{\substack{j=1 \\ j:odd}}^{\infty}\frac{1}{j}\sin[j\Psi_{su}(t)] \quad (3)$$

the digitized signal, which is sampled at rate $R_s = 1/T_s$, can be written as

$$r(kT_s) = \sqrt{2P_c}\sin[\Psi_c(kT_s)] + \sqrt{2P_d}d(kT_s)\left(\frac{4}{\pi}\sum_{\substack{j=1 \\ j:odd}}^{L}\frac{\sin[j\Psi_{su}(kT_s)]}{j}\right) \times \cos[\Psi_c(kT_s)] + n(kT_s) \quad (4)$$

where only up to the Lth subcarrier harmonics (L is an odd integer) are assumed to be present at the analog-to-digital (A/D) converter output in Fig. 1. Samples of the bandpass noise can be expressed as

$$n(kT_s) = \sqrt{2}n_1(kT_s)\cos[\Psi_c(kT_s)] - \sqrt{2}n_2(kT_s)\sin[\Psi_c(kT_s)] \quad (5)$$

where the random variables $n_1(kT_s)$ and $n_2(kT_s)$ are samples of the baseband noise processes $n_1(t)$ and $n_2(t)$, which are assumed to be statistically independent, white Gaussian noise processes with a flat, one-sided PSD level equal to N_0 W/Hz within a one-sided bandwidth equal to $1/(2T_s)$ Hz. As a result, $n_1(kT_s)$ and $n_2(kT_s)$ are zero-mean with variance $\sigma^2_{n_1} = \sigma^2_{n_2} = N_0/(2T_s)$. The time index kT_s will now be omitted to allow a simpler notation.

The residual carrier tracking loop control signal, U_q in Fig. 1, is derived by mixing the digitized IF signal with its carrier quadrature component, lowpass filtering the mixer output, and then accumulating the filtered samples over the loop update interval. Mathematically,

$$U_q = \sqrt{P_c}\sin(\phi_c) + n_{U_q} \tag{6}$$

where ϕ_c is the carrier phase error, n_{U_q} is a zero-mean Gaussian random variable with variance $N_0/(2T_u)$, and $T_u = M_1T_s$ is the loop update interval in seconds. Note that the data have been neglected in writing Eq. (6), since it is assumed that the accumulator in the carrier quadrature arm averages over several subcarrier cycles (i.e., $f_{su}T_u \gg 1$) and that the data sum is thus approximately zero. It will be shown in Section IV that the BTD carrier loop control signal has the same signal power and noise statistics as Eq. (6) and, therefore, both loops will have the same tracking performance.

Next, consider the subcarrier tracking loop. As shown in Fig. 1, the input to the subcarrier loop is the quadrature component of the residual carrier. The subcarrier control signal is obtained by forming the product $Z_iZ_{q,su}$, where Z_i is the output of the subcarrier inphase arm and $Z_{q,su}$ is the output of the subcarrier quadrature arm. Note that the subscript "su" in $Z_{q,su}$ is used to distinguish it from $Z_{q,sca}$, the quadrature arm of the suppressed carrier loop. The signals Z_i and $Z_{q,su}$ are obtained after mixing the residual carrier quadrature component with, respectively, the subcarrier inphase and quadrature references and then accumulating these samples over a symbol duration (perfect symbol synchronization assumed). These signals are given as [1]

$$Z_i = \sqrt{P_d}a_lF_I(\phi_{su})\cos(\phi_c) + n_{Z_i} \tag{7}$$

and

$$Z_{q,su} = \sqrt{P_d}a_lF_Q(\phi_{su})\cos(\phi_c) + n_{Z_{q,su}} \tag{8}$$

where ϕ_{su} is the subcarrier phase error and the functions $F_I(\phi_{su})$ and $F_Q(\phi_{su})$ are periodic functions of ϕ_{su} (period 2π) given in the principal phase interval $[-\pi, \pi]$ as follows [1]:

$$F_I(\phi_{su}) = 1 - \frac{2}{\pi}|\phi_{su}|, \quad |\phi_{su}| \le \pi \tag{9}$$

and

$$F_Q(\phi_{su}) = \begin{cases} (2/\pi)\phi_{su} & |\phi_{su}| \le (\pi/2)W_{su} \\ \mathrm{sgn}(\phi_{su})W_{su} & (\pi/2)W_{su} \le |\phi_{su}| < \pi(1 - W_{su}/2) \\ 2\mathrm{sgn}(\phi_{su}) - (2/\pi)\phi_{su} & \pi(1 - W_{su}/2) \le |\phi_{su}| \le \pi \end{cases} \tag{10}$$

The parameter W_{su}, in fractions of cycles, is the width of the subcarrier window ($W_{su} \le 1$, and $W_{su} = 1$ corresponds to the "no window" case). It is used in the quadrature arm to improve the square-wave tracking performance [3]. The noises n_{Z_i} and $n_{Z_{q,su}}$ in Eqs. (7) and (8) are independent with respective variances $N_0/(2T)$ and $N_0W_{su}/(2T)$, where T denotes symbol duration. It will be shown in Section IV that the BTD inphase and quadrature accumulator outputs have the same signal and noise statistics as Eqs. (7) and (8). Consequently, both subcarrier loops will have the same control signal and, therefore, the same tracking performance.

Finally, consider the case of the suppressed carrier with subcarrier ($\Delta = 90$ deg in Eq. [1]). In this scenario, the suppressed carrier loop control signal is obtained by form-

ing the product $Z_i Z_{q,sca}$, as shown in Fig. 1. The signal Z_i, which was shown to be the subcarrier inphase component, now becomes the suppressed carrier inphase component because it is already proportional to $\cos(\phi_c)$ and $F_I(\phi_{su})$, as indicated by Eq. (7). The suppressed carrier quadrature component $Z_{q,sca}$ is generated by mixing the digitized IF signal with the quadrature carrier reference and the inphase subcarrier reference, and then accumulating the resulting samples over a symbol duration. Hence, the suppressed carrier inphase and quadrature signals are given as [1]

$$Z_i = \sqrt{P_d} a_l F_I(\phi_{su}) \cos(\phi_c) + n_{Z_i} \tag{11}$$

and

$$Z_{q,sca} = \sqrt{P_d} a_l F_I(\phi_{su}) \sin(\phi_c) + n_{Z_{q,sca}} \tag{12}$$

where n_{Z_i} and $n_{Z_{q,sca}}$ are independent and have variance $N_0/(2T)$. It will be shown in Section IV that the inphase and quadrature signals of the BTD suppressed carrier loop have the same statistics as are found in Eqs. (11) and (12). Hence, both suppressed carrier loops will have the same error signal and tracking performance.

III. Baseband Recording

As indicated earlier, the BTD processes the received signal after the signal has been open-loop downconverted to baseband on a subcarrier harmonic basis and recorded on tape. This section describes two methods for recording the received signal at baseband. The first recording scheme, depicted in Fig. 2, distributes the sampled IF signal into two carrier channels and $4n$ subcarrier channels, where n denotes the number of subcarrier harmonics present at the A/D output and is related to L (the highest recorded subcarrier harmonic) through $L = 2n - 1$. Recording of the carrier requires two channels because baseband demodulation requires both inphase and quadrature samples. Similarly, recording of each subcarrier harmonic requires four channels—two channels (one inphase subcarrier channel and one quadrature subcarrier channel) per carrier channel (inphase and quadrature). Table 1 depicts the loss in data power as a function of the highest recorded subcarrier harmonic L.

Each band of the sampled IF signal is recorded at baseband after mixing it with tones tuned to its predicted center frequency. The predicts are assumed to be close, but not exactly equal, to the actual band center frequency. For example, consider the two channels that record the residual carrier. In the residual carrier sine channel, the sampled IF signal is mixed with $\sqrt{2}\sin\hat{\Psi}_c(kT_s)$, where $\hat{\Psi}_c(kT_s)$ is an estimate of the actual carrier phase $\Psi_c(kT_s)$, and is lowpass filtered to obtain the recorded waveform $C_s(kT_s)$. Similarly, in the residual carrier cosine channel, the sampled IF signal is mixed with $\sqrt{2}\cos\hat{\Psi}_c(kT_s)$ and lowpass filtered to obtain $C_c(kT_s)$. Given the information in Appendix A, let $\Theta_c(kT_s) \triangleq \Psi_c(kT_s) - \hat{\Psi}_c(kT_s)$, the recorded carrier is given by

$$C_s = \sqrt{P_c}\cos(\Theta_c) + n_{c_s} \tag{13}$$

$$C_c = \sqrt{P_c}\sin(\Theta_c) + n_{c_c} \tag{14}$$

where, using the Heaviside operator notation,[1]

$$n_{c_s} = H_c(z)\{-n_1\sin(\Theta_c) - n_2\cos(\Theta_c)\} \tag{15}$$

$$n_{c_c} = H_c(z)\{n_1\cos(\Theta_c) - n_2\sin(\Theta_c)\} \tag{16}$$

The lowpass filter $H_c(z)$ is assumed to be bandlimited to B_c Hz. Furthermore, since the carrier frequency error (derivative of Θ_c) is assumed to be much smaller than B_c, both n_{c_c} and n_{c_s} are baseband processes limited to B_c and have a flat, one-sided PSD level equal to N_0 W/Hz.

In the first approach, the recording of the subcarrier harmonics at baseband requires a two-stage downconversion of the IF signal. The first stage centers the IF spectrum near DC by mixing it with sine and cosine references tuned to the predicted carrier frequency. The second stage mixes each of the subcarrier harmonics to baseband individually by mixing the output of the first stage with sine and cosine references tuned to multiples of the predicted subcarrier frequency. Hence, to record the mth subcarrier harmonic, the IF signal is first mixed with $\sin(\hat{\Psi}_c)$ and $\cos(\hat{\Psi}_c)$, and then with $\sin(m\hat{\Psi}_{su})$ and $\cos(m\hat{\Psi}_{su})$. It is important to note that a phase relationship has to be maintained between the various tones that demodulate the subcarrier harmonics to baseband. Given the information in Appendix A, the four signals corresponding to the recording of the mth subcarrier harmonic are represented as (where m is an odd integer)

$$SU_{ss}^{(m)} = -\sqrt{P_d}\frac{2d}{\pi m}\cos[m\Theta_{su}]\sin(\Theta_c) + n_{SU_{ss}}^{(m)} \tag{17}$$

[1] The Heaviside operation $H(z)\{x(m)\} \triangleq \sum_{n=-\infty}^{m} x(n)h(m-n)$ denotes the convolution of $x(m)$ with $h(m)$.

$$SU_{sc}^{(m)} = -\sqrt{P_d}\frac{2d}{\pi m}\sin[m\Theta_{su}]\sin(\Theta_c) + n_{SU_{sc}}^{(m)} \quad (18)$$

$$SU_{cs}^{(m)} = +\sqrt{P_d}\frac{2d}{\pi m}\cos[m\Theta_{su}]\cos(\Theta_c) + n_{SU_{cs}}^{(m)} \quad (19)$$

$$SU_{cc}^{(m)} = +\sqrt{P_d}\frac{2d}{\pi m}\sin[m\Theta_{su}]\cos(\Theta_c) + n_{SU_{cc}}^{(m)} \quad (20)$$

where the noise terms are given by

$$n_{SU_{ss}}^{(m)} = -H_{su}(z)\left\{[n_1\sin(\Theta_c) + n_2\cos(\Theta_c)]\sin[m\hat{\Psi}_{su}]\right\} \quad (21)$$

$$n_{SU_{sc}}^{(m)} = -H_{su}(z)\left\{[n_1\sin(\Theta_c) + n_2\cos(\Theta_c)]\cos[m\hat{\Psi}_{su}]\right\} \quad (22)$$

$$n_{SU_{cs}}^{(m)} = H_{su}(z)\left\{[n_1\cos(\Theta_c) - n_2\sin(\Theta_c)]\sin[m\hat{\Psi}_{su}]\right\} \quad (23)$$

$$n_{SU_{cc}}^{(m)} = H_{su}(z)\left\{[n_1\cos(\Theta_c) - n_2\sin(\Theta_c)]\cos[m\hat{\Psi}_{su}]\right\} \quad (24)$$

It can be verified that $n_{SU_{sc}}^{(m)}$, $n_{SU_{ss}}^{(m)}$, $n_{SU_{cc}}^{(m)}$, and $n_{SU_{cs}}^{(m)}$ are baseband processes limited to B_{su}, the bandwidth of the lowpass filter $H_{su}(z)$, and have a flat, one-sided PSD level equal to $N_0/2$ W/Hz. Moreover, it can be shown that these recorded noises are not only mutually independent of each other for every subcarrier harmonic (say, m), but are also independent of the recorded noises at other harmonics (any $n \neq m$).

Another approach to baseband recording of the received signal is shown in Fig. 3. Unlike the previous recording scheme, which uses the two-stage downconversion technique to record the subcarrier, this scheme records each subcarrier harmonic by directly downconverting it to baseband. As shown in Fig. 3, the sampled IF signal is still distributed into $(4n+2)$ channels. The two channels used for recording the residual carrier remain the same for this new scheme. Consequently, the recorded carrier is as given by Eqs. (13) and (14). However, the subcarrier harmonics are recorded at baseband by directly mixing the upper and lower sidebands of each harmonic to baseband. That is, the mth subcarrier harmonic is recorded after mixing the A/D output with sine and cosine references tuned to $\hat{f}_c + m\hat{f}_{su}$ (upper sideband) and $\hat{f}_c - m\hat{f}_{su}$ (lower sideband) where $\hat{f}_c$ and $\hat{f}_{su}$ denote, respectively, the predicted carrier and subcarrier frequencies. Following the same procedure used in the derivation of Eqs. (17)-(20) (see Appendix A), the recorded signals for this scheme can be shown to be (where m is an odd integer)

$$SU_{s+}^{(m)} = \sqrt{P_d}\frac{2d}{\pi m}\cos[\Theta_c + m\Theta_{su}] + n_{SU_{s+}}^{(m)} \quad (25)$$

$$SU_{s-}^{(m)} = -\sqrt{P_d}\frac{2d}{\pi m}\cos[\Theta_c - m\Theta_{su}] + n_{SU_{s-}}^{(m)} \quad (26)$$

$$SU_{c+}^{(m)} = \sqrt{P_d}\frac{2d}{\pi m}\sin[\Theta_c + m\Theta_{su}] + n_{SU_{c+}}^{(m)} \quad (27)$$

$$SU_{c-}^{(m)} = -\sqrt{P_d}\frac{2d}{\pi m}\sin[\Theta_c - m\Theta_{su}] + n_{SU_{c-}}^{(m)} \quad (28)$$

where the subscripts "+" and "−", respectively, denote the upper and lower sidebands. The noises $n_{SU_{s+}}^{(m)}$, $n_{SU_{s-}}^{(m)}$, $n_{SU_{c+}}^{(m)}$, and $n_{SU_{s+}}^{(m)}$ can be shown to have a flat, one-sided PSD level equal to N_0 within B_{su} Hz, the one-sided bandwidth of the lowpass filter $H_{su}(z)$ in Fig. 3. These upper and lower sideband noises are correlated with respect to each other because the recorded signals of the direct recording scheme are correlated. Specifically, they are related to each other by Eqs. (17)-(20) as follows; namely,

$$SU_{s+}^{(m)} = SU_{sc}^{(m)} + SU_{cs}^{(m)} \quad (29)$$

$$SU_{s-}^{(m)} = SU_{sc}^{(m)} - SU_{cs}^{(m)} \quad (30)$$

$$SU_{c+}^{(m)} = SU_{cc}^{(m)} - SU_{ss}^{(m)} \quad (31)$$

$$SU_{c-}^{(m)} = SU_{cc}^{(m)} + SU_{ss}^{(m)} \quad (32)$$

and consequently,

$$n_{SU_{s+}}^{(m)} = n_{SU_{sc}}^{(m)} + n_{SU_{cs}}^{(m)} \quad (33)$$

$$n_{SU_{s-}}^{(m)} = n_{SU_{sc}}^{(m)} - n_{SU_{cs}}^{(m)} \quad (34)$$

$$n_{SU_{c+}}^{(m)} = n_{SU_{cc}}^{(m)} - n_{SU_{ss}}^{(m)} \quad (35)$$

$$n_{SU_{c-}}^{(m)} = n_{SU_{cc}}^{(m)} + n_{SU_{ss}}^{(m)} \quad (36)$$

Clearly, $n_{SU_{s+}}^{(m)}$ is correlated with $n_{SU_{s-}}^{(m)}$ and $n_{SU_{c+}}^{(m)}$ is correlated with $n_{SU_{c-}}^{(m)}$.

IV. Coherent Baseband Demodulation

This section considers the coherent baseband demodulation of the carrier and subcarrier when the modulation index (Δ in Eq. [1]) is such that the received signal consists of a sinusoidal carrier and square-wave subcarrier; it also considers the case when the received signal consists of a suppressed carrier with square-wave subcarrier (i.e., $\Delta = 90$ deg). The demodulation functions begin by reading the recorded signals of Section III from a tape drive and then processing them to obtain control signals for the carrier, subcarrier, and suppressed carrier tracking loops. Recall that two methods for baseband recording of the IF signal were considered in Section III. It will be shown that the baseband processing corresponding to these two methods is very similar except for a little extra processing that is required for the direct downconversion scheme, the second recording scheme in Section III. In this section, the carrier tracking loop, which is the same for both recording schemes, is described first. Then, the subcarrier and suppressed carrier loops for the two-stage downconversion recording scheme are described, and followed by the additional processing needed for using the signals of the direct downconversion scheme.

The carrier tracking loop, depicted near the bottom of Fig. 4, mixes the signals C_c and C_s with $\cos(\hat{\Theta}_c)$ and $\sin(\hat{\Theta}_c)$, respectively, before subtracting the mixer outputs and accumulating the resulting samples over the loop update interval to form the carrier loop control signal, V_q. Hence,

$$V_q = \frac{1}{M_1} \sum_{M_1} [C_c \cos(\hat{\Theta}_c) - C_s \sin(\hat{\Theta}_c)] \quad (37)$$

where $\hat{\Theta}_c$ is an estimate of the recorded carrier phase Θ_c. Substituting Eqs. (13) and (14) for C_s and C_c, then expanding and canceling common terms, and assuming that the carrier phase error, $\phi_c \triangleq \Theta_c - \hat{\Theta}_c$, varies slowly over the loop update interval, $T_u = M_1 T_s$, yields

$$V_q = \sqrt{P_c} \sin(\phi_c) + n_{V_q} \quad (38)$$

where

$$n_{V_q} = \frac{1}{M_1} \sum_{M_1} [n_{c_c} \cos(\hat{\Theta}_c) - n_{c_s} \sin(\hat{\Theta}_c)] \quad (39)$$

and n_{c_s} and n_{c_c} are given by Eqs. (15) and (16). Since both n_{c_s} and n_{c_c} are zero mean with a flat, one-sided power spectral density level equal to N_0 W/Hz within a much wider recording bandwidth B_c than the loop update rate $1/T_u$ has, it can be shown that n_{V_q} is Gaussian with zero mean and variance $N_0/2T_u$. Comparing Eq. (38) with Eq. (6) indicates that both the baseband carrier loop and the IF carrier loop have the same control signal in terms of signal power and noise variance. As a result, the carrier tracking variance of the baseband loop will be the same as that of a phase-locked loop tuned to the intermediate frequency.

Recall that the square-wave subcarrier was recorded at baseband by mixing each of its harmonics down to baseband individually and then recording each harmonic separately. For each subcarrier harmonic retained, four baseband signals given by Eqs. (17) through (20) were actually recorded. As shown in Fig. 4, each harmonic of the subcarrier is first read from the tape recorder and demodulated individually, and then properly weighted and summed to reconstruct the proper inphase and quadrature samples for the various Costas loops. The demodulation procedures require that the four recorded signals of each subcarrier harmonic be multiplied by the estimated carrier references, $\sin(\hat{\Theta}_c)$ and $\cos(\hat{\Theta}_c)$, and the estimated subcarrier harmonic references, $\sin(m\hat{\Theta}_c)$ and $\cos(m\hat{\Theta}_c)$, where m is an odd integer that depends on the subcarrier harmonic under consideration. As depicted in Fig. 4, the resulting signals are either added or subtracted to retain terms at $\Theta_c - \hat{\Theta}_c$ and $m(\Theta_{su} - \hat{\Theta}_{su})$, and eliminate terms at $\Theta_c + \hat{\Theta}_c$ and $m(\Theta_{su} + \hat{\Theta}_{su})$. The details of these demodulation procedures are provided in Appendix A. Each of the demodulated signals of a particular subcarrier harmonic is then properly weighted and summed to reconstruct the desired control signals for various tracking loops. For example, in the subcarrier loop depicted in the top half of Fig. 4, the mth harmonic is scaled by $4/(m\pi)$ on the inphase arm, and $[4/(m\pi)]\sin[(m/2)\pi W_{su}]$ on the quadrature arm (W_{su} is the subcarrier window size). Subsequently, the scaled inphase signals corresponding to each harmonic are added to form the input to the inphase symbol accumulator, whereas the scaled quadrature signals are added to form the input to the quadrature symbol accumulator. The product of the inphase accumulator output, Y_i in Fig. 4, and the quadrature accumulator output, $Y_{q,su}$ in Fig. 4, is the subcarrier loop control signal. Given the information in Appendix A, the symbol accumulator outputs corresponding to the lth symbol are given as

$$Y_i = \sqrt{P_d} a_l \cos(\phi_c) \frac{8}{\pi^2} \sum_{\substack{j=1 \\ j:odd}}^{L} \left(\frac{1}{j^2} \cos(j\phi_{su}) \right) + n_{Y_i} \quad (40)$$

and

$$Y_{q,su} = \sqrt{P_d} a_l \cos(\phi_c) \frac{8}{\pi^2} \sum_{\substack{j=1 \\ j:odd}}^{L} \left(\frac{\sin[(j/2)\pi W_{su}]}{j^2} \sin(j\phi_{su}) \right)$$

$$+ n_{Y_{q,su}} \quad (41)$$

where the independent noise terms, n_{Y_i} and $n_{Y_{q,su}}$, are shown in Appendix A to have variance $N_0/(2T)$ and $N_0 W_{su}/(2T)$, respectively, as L approaches infinity in the above two equations. That is, for sufficiently large L, the noise variances of n_{Y_i} and $n_{Y_{q,su}}$ are approximately the same as the variances of n_{Z_i} and $n_{Z_{q,su}}$ in Eqs. (7) and (8), respectively. Furthermore, the signal parts of Eqs. (40) and (7), and of Eqs. (41) and (8), are also equal as L approaches infinity. This can be easily shown by using the Fourier series expansion of $F_I(\phi_{su})$ and $F_Q(\phi_{su})$ in Eqs. (7) and (8), given as follows:

$$F_I(\phi_{su}) = \frac{8}{\pi^2} \sum_{\substack{j=1 \\ j:odd}}^{\infty} \frac{1}{j^2} \cos(j\phi_{su}) \quad (42)$$

and

$$F_Q(\phi_{su}) = \frac{8}{\pi^2} \sum_{\substack{j=1 \\ j:odd}}^{\infty} \frac{\sin[(j/2)\pi W_{su}]}{j^2} \sin(j\phi_{su}) \quad (43)$$

As a result of the subcarrier loop accumulator outputs of the ARX II and BTD being approximately equal for sufficiently large L, the control signals, $Y_i Y_{q,su}$ and $Z_i Z_{q,su}$, and tracking variance of both loops can also be shown to be equal.

For suppressed carrier tracking, the suppressed carrier loop, also shown in Fig. 4, is a Costas loop sharing the same inphase arm with the subcarrier Costas loop. Hence, the suppressed carrier inphase accumulator output, Y_i, is as given in Eq. (40). The processing in the suppressed carrier loop quadrature arm is very similar to that of the subcarrier loop. Hence, following the same derivation as in the subcarrier loop quadrature arm, the suppressed carrier quadrature arm accumulation over the lth symbol can be shown to be

$$Y_{q,sca} = \sqrt{P_d} a_l \sin(\phi_c) \frac{8}{\pi^2} \sum_{\substack{j=1 \\ j:odd}}^{L} \frac{1}{j^2} \cos(j\phi_{su}) + n_{Y_{q,sca}} \quad (44)$$

As before, the equivalence of the suppressed carrier loops of the BTD and ARX II is established by comparing the inphase and quadrature accumulator outputs of both loops. The equivalence of the inphase arms has already been shown earlier. The signal parts of the quadrature arm accumulations given by Eqs. (12) and (44) are seen to be equivalent after letting L go to infinity in Eq. (44) and then using Eq. (42) for the infinite sum. The noise parts of Eqs. (12) and (44), as L approaches infinity in Eq. (44), can also be shown to have the same variance ($N_0/2T$). Furthermore, it can be shown that n_{Y_i} and $n_{Y_{q,sca}}$ are uncorrelated.

Finally, consider the coherent baseband demodulation of the signals that were recorded as a result of the direct downconversion scheme, the second recording scheme described in Section III. As indicated earlier, the demodulator for the direct scheme is very similar to the two-stage downconversion scheme demodulator except that the former requires a little extra processing. Specifically, the recorded signals for the direct scheme, given by Eqs. (25)–(28), are combined to yield the signals $SU_{ss}^{(m)}$, $SU_{sc}^{(m)}$, $SU_{cs}^{(m)}$, and $SU_{cc}^{(m)}$, which were the input signals to the two-stage demodulator described in this section. From Eqs. (29)–(32), it is clear that the input signals to the two-stage demodulator can be generated by using two adders and two subtractors (per harmonic) on the signals $SU_{s+}^{(m)}$, $SU_{s-}^{(m)}$, $SU_{c+}^{(m)}$, and $SU_{c-}^{(m)}$. After these additions and subtractions, shown in Fig. 5, the rest of the signal processing remains the same as in the previous scheme.

V. Requirements for Software Implementation of the BTD

As indicated earlier, the BTD depicted in Fig. 4 is targeted for implementation on a general-purpose engineering workstation. The primary concern with a software implementation is whether a general-purpose workstation can provide the computational speed required for real-time processing of the recorded signals. For example, if the received symbol rate is 1K symbols/sec and the A/D sampling rate is 10K samples/sec, then a real-time demodulator would process 10K samples/sec and detect 1K symbols/sec. This section first determines the computational speed required (in operations/sec) for implementing the BTD in real time when the symbol rate is 1K symbols/sec and the sample rate is 10K samples/sec. Next, it deter-

mines the feasibility of implementing the BTD on a Sun Sparc 2 (SS-2) and Sun Sparc 10 workstation.

Recall that the two modes for the BTD are the residual carrier with subcarrier mode ($\Delta \neq 90$ deg) and the suppressed carrier with subcarrier mode ($\Delta = 90$ deg). The number of (addition and multiplication) operations/sec, for a given mode, can be computed by separately counting the operations/sample and operations/symbol in Fig. 4 and then multiplying by the operations at the sampling rate and the operations at the symbol rate. Mathematically,

$$\text{operations} = \left(\frac{(\text{number of "+" and "}\times\text{"}) \text{ operations}}{\text{sample}}\right) \times \left(\frac{10\text{K samples}}{\text{sec}}\right) + \left(\frac{(\text{number of "+" and "}\times\text{"}) \text{ operations}}{\text{symbol}}\right) \times \left(\frac{1\text{K symbols}}{\text{sec}}\right) \quad (45)$$

where the sample and symbol (floating point) operations for the coherent demodulator in Fig. 4 are tabulated in Table 2(a). The total number of operations in Table 2(a) is accounted for in more detail in Table 2(b). The coherent demodulator for the direct downconversion scheme needs four more sample operations per harmonic than that of the two-stage downconversion scheme, as shown in Fig. 5, and is not considered separately. Note that both addition and multiplication are counted as one operation each. Furthermore, the lookup table operations are done at the 10K samples/sec rate, since the lookup table is considered to be part of the carrier and subcarrier numerically controlled oscillators (NCO's). With the number of harmonics, n, equal to three (i.e., the first, third, and fifth harmonics are recorded) in Table 2(a), there are 71 operations/sample and 38 operations/symbol for the residual carrier mode, and there are 98 operations/sample and 39 operations/symbol in the suppressed carrier mode. Consequently, at 1K samples/sec and 10 KHz sampling rate, a real-time processor would be required to provide 748K and 1019K operations for the residual carrier and suppressed carrier modes, respectively. The number of operations per second and the SNR degradation as a function of subcarrier harmonics are tabulated in Table 2(b). Table 3 lists the throughput utilized by a software implementation of the BTD for various computers. The throughput is defined as the number of instructions a computer can execute per second. In Table 3, the percentage of throughput utilized by a Sun SS-2 was obtained by simulating a portion of the BTD on the Sun SS-2 and then linearly extrapolating the results to determine the throughput utilization for a software implementation of the entire BTD. The throughput of the remaining computers was obtained by linearly extrapolating the Sun SS-2 results using the SPECfp92 benchmark. The specmark rating of a computer, obtained by running a collection of commonly used programs on a target system, is a measure of the computer's performance relative to a VAX-11/780. For example, the Sun SS-2 is expected to perform 22.8 times better than a VAX-11/780, since it specmarks at 22.8. Alternatively stated, the throughput utilization of the Sun SS-2 is expected to be (1/22.8) that of a VAX-11/780. It is clear from Table 3 that several computers will be able to process the recorded data in real time.

References

[1] S. Hinedi, "A Functional Description of the Advanced Receiver," *The Telecommunications and Data Acquisition Progress Report 42-100*, vol. October–December 1989, Jet Propulsion Laboratory, Pasadena, California, pp. 131–149, February 15, 1990.

[2] J. H. Yuen, *Deep Space Telecommunications Systems Engineering*, New York: Plenum Press, 1983.

[3] W. J. Hurd and S. Aguirre, "A Method to Dramatically Improve Subcarrier Tracking," *The Telecommunications and Data Acquisition Progress Report 42-86*, vol. April–June 1986, Jet Propulsion Laboratory, Pasadena, California, pp. 103–110, August 15, 1986.

Glossary

Ψ_c = Incoming carrier phase

$\hat{\Psi}_c$ = Predicted carrier phase

Θ_c = Carrier phase offset, $\Psi_c - \hat{\Psi}_c$

$\hat{\Theta}_c$ = Carrier phase offset estimate

ϕ_c = Carrier phase offset error, $\Theta_c - \hat{\Theta}_c$

Ψ_{su} = Incoming subcarrier phase

$\hat{\Psi}_{su}$ = Predicted subcarrier phase

Θ_{su} = Subcarrier phase offset, $\Psi_{su} - \hat{\Psi}_{su}$

$\hat{\Theta}_{su}$ = Subcarrier phase offset estimate

ϕ_{su} = Subcarrier phase offset error, $\Theta_{su} - \hat{\Theta}_{su}$

$XSU_{ss}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\sin(\hat{\Psi}_c)\sin(m\hat{\Psi}_{su})$

$SU_{ss}^{(m)}$ = The lowpass filtered version of $XSU_{ss}^{(m)}$

$XSU_{sc}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\sin(\hat{\Psi}_c)\cos(m\hat{\Psi}_{su})$

$SU_{sc}^{(m)}$ = The lowpass filtered version of $XSU_{sc}^{(m)}$

$XSU_{cs}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\cos(\hat{\Psi}_c)\sin(m\hat{\Psi}_{su})$

$SU_{cs}^{(m)}$ = The lowpass filtered version of $XSU_{cs}^{(m)}$

$XSU_{cc}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\cos(\hat{\Psi}_c)\cos(m\hat{\Psi}_{su})$

$SU_{cc}^{(m)}$ = The lowpass filtered version of $XSU_{cc}^{(m)}$

$XSU_{s+}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\sin(\hat{\Psi}_c + m\hat{\Psi}_{su})$

$SU_{s+}^{(m)}$ = The lowpass filtered version of $XSU_{s+}^{(m)}$

$XSU_{s-}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\sin(\hat{\Psi}_c - m\hat{\Psi}_{su})$

$SU_{s-}^{(m)}$ = The lowpass filtered version of $XSU_{s-}^{(m)}$

$XSU_{c+}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\cos(\hat{\Psi}_c + m\hat{\Psi}_{su})$

$SU_{c+}^{(m)}$ = The lowpass filtered version of $XSU_{c+}^{(m)}$

$XSU_{c-}^{(m)}$ = mth subcarrier harmonic, derived from mixing the received signal with $\sqrt{2}\cos(\hat{\Psi}_c - m\hat{\Psi}_{su})$

$SU_{c-}^{(m)}$ = The lowpass filtered version of $XSU_{c-}^{(m)}$

Appendix A

I. Derivation of Eqs. (13) and (14)

As shown in Fig. 2, the lowpass filter inputs XC_s and XC_c are given as

$$XC_s = r\sqrt{2}\sin(\hat{\Psi}_c) \tag{A-1}$$

$$XC_c = r\sqrt{2}\cos(\hat{\Psi}_c) \tag{A-2}$$

where the digitized IF, r, is given by Eq. (4) and $\hat{\Psi}_c$ is the predicted carrier phase. (The index kT_s is omitted for simplicity of notation.) Substituting Eq. (4) into Eqs. (A-1) and (A-2), then expanding and letting $\Theta_c = \Psi_c - \hat{\Psi}_c$, yields

$$XC_s = \sqrt{P_c}\cos(\Theta_c) - n_1\sin(\Theta_c) - n_2\cos(\Theta_c) - \sqrt{P_d}d\left(\frac{4}{\pi}\sum_{\substack{j=1\\ j:odd}}^{L}\frac{1}{j}\sin[j\Psi_{su}]\right)\sin(\Theta_c)$$

$$+ \left(\text{terms at angular frequency } (\Psi_c + \hat{\Psi}_c)'\right) \tag{A-3}$$

and

$$XC_c = \sqrt{P_c}\sin(\Theta_c) + n_1\cos(\Theta_c) - n_2\sin(\Theta_c) + \sqrt{P_d}d\left(\frac{4}{\pi}\sum_{\substack{j=1\\ j:odd}}^{L}\frac{1}{j}\sin[j\Psi_{su}]\right)\cos(\Theta_c)$$

$$+ \left(\text{terms at angular frequency } (\Psi_c + \hat{\Psi}_c)'\right) \tag{A-4}$$

where the instantaneous angular frequency (in radians/sec) of the high-frequency components is denoted as $(\Psi_c + \hat{\Psi}_c)'$, the time derivative of the total phase $\Psi_c + \hat{\Psi}_c$.

From Eqs. (A-3) and (A-4), it is clear that XC_s and XC_c have the desired baseband carrier at angular frequency, Θ_c'—as well as the undesired data spectra centered at $\Theta_c' \pm j\Psi_{su}'$, where Ψ_{su}' is the angular subcarrier frequency and j is an odd integer, $1 \le j \le L$—and other undesired terms at frequency $(\Psi_c + \hat{\Psi}_c)'$. The recorded baseband carrier given by Eqs. (13) and (14) can be obtained by lowpass filtering XC_s and XC_c with the filter $H_c(z)$ whose bandwidth B_c is small enough to accommodate only the desired baseband carrier, i.e.,

$$B_c \le (-\Theta_c' + \Psi_{su}' - \omega_d)/(2\pi) \quad \text{(in Hz)}$$

where ω_d is the bandwidth (in radians/sec) wherein most of the data power resides.

II. Derivation of Eqs. (17) Through (20)

As shown in Fig. 2, recording of the mth harmonic of the data-modulated subcarrier requires downconverting this particular harmonic to baseband by mixing XC_s and XC_c with both the inphase and quadrature subcarrier references tuned to the predicted frequency of the mth harmonic. Here, m is an odd integer with $1 \leq m \leq L$, where L is the highest subcarrier harmonic to be recorded. The four signals that are generated as a result are as follows:

$$XSU_{ss}^{(m)} = XC_s \sin(m\hat{\Psi}_{su}) \tag{A-5}$$

$$XSU_{sc}^{(m)} = XC_s \cos(m\hat{\Psi}_{su}) \tag{A-6}$$

$$XSU_{cs}^{(m)} = XC_c \sin(m\hat{\Psi}_{su}) \tag{A-7}$$

$$XSU_{cc}^{(m)} = XC_c \cos(m\hat{\Psi}_{su}) \tag{A-8}$$

It can be found that (with $\Theta_{su} = \Psi_{su} - \hat{\Psi}_{su}$)

$$\begin{aligned} XSU_{ss}^{(m)} &= \sqrt{P_c}\cos(\Theta_c)\sin(m\hat{\Psi}_{su}) - \sqrt{P_d}\frac{2d}{m\pi}\sin(\Theta_c)\cos(m\Theta_{su}) \\ &\quad - \sqrt{P_d}\frac{2d}{\pi}\sin(\Theta_c)\left(\sum_{\substack{j=1 \\ j:odd,\ j\neq m}}^{L}\frac{1}{j}\cos\left(j\Psi_{su} - m\hat{\Psi}_{su}\right)\right) - [n_1\sin(\Theta_c) + n_2\cos(\Theta_c)]\sin(m\hat{\Psi}_{su}) \\ &\quad + \left(\text{terms at angular frequency } (\Psi_c + \hat{\Psi}_c)'\right) \end{aligned} \tag{A-9}$$

$$\begin{aligned} XSU_{sc}^{(m)} &= \sqrt{P_c}\cos(\Theta_c)\cos(m\hat{\Psi}_{su}) - \sqrt{P_d}\frac{2d}{m\pi}\sin(\Theta_c)\sin(m\Theta_{su}) \\ &\quad - \sqrt{P_d}\frac{2d}{\pi}\sin(\Theta_c)\left(\sum_{\substack{j=1 \\ j:odd,\ j\neq m}}^{L}\frac{1}{j}\sin\left(j\Psi_{su} - m\hat{\Psi}_{su}\right)\right) - [n_1\sin(\Theta_c) + n_2\cos(\Theta_c)]\cos(m\hat{\Psi}_{su}) \\ &\quad + \left(\text{terms at angular frequency } (\Psi_c + \hat{\Psi}_c)'\right) \end{aligned} \tag{A-10}$$

$$\begin{aligned} XSU_{cs}^{(m)} &= \sqrt{P_c}\sin(\Theta_c)\sin(m\hat{\Psi}_{su}) + \sqrt{P_d}\frac{2d}{m\pi}\cos(\Theta_c)\cos(m\Theta_{su}) \\ &\quad + \sqrt{P_d}\frac{2d}{\pi}\cos(\Theta_c)\left(\sum_{\substack{j=1 \\ j:odd,\ j\neq m}}^{L}\frac{1}{j}\cos\left(j\Psi_{su} - m\hat{\Psi}_{su}\right)\right) + [n_1\cos(\Theta_c) - n_2\sin(\Theta_c)]\sin(m\hat{\Psi}_{su}) \\ &\quad + \left(\text{terms at angular frequency } (\Psi_c + \hat{\Psi}_c)'\right) \end{aligned} \tag{A-11}$$

$$XSU_{cc}^{(m)} = \sqrt{P_c}\sin(\Theta_c)\cos(m\hat{\Psi}_{su}) + \sqrt{P_d}\frac{2d}{m\pi}\cos(\Theta_c)\sin(m\Theta_{su})$$

$$+ \sqrt{P_d}\frac{2d}{\pi}\cos(\Theta_c)\left(\sum_{\substack{j=1 \\ j:odd,\ j\neq m}}^{L}\frac{1}{j}\sin\left(j\Psi_{su} - m\hat{\Psi}_{su}\right)\right) + [n_1\cos(\Theta_c) - n_2\sin(\Theta_c)]\cos(m\hat{\Psi}_{su})$$

$$+ \left(\text{terms at angular frequency } (\Psi_c + \hat{\Psi}_c)'\right) \qquad \text{(A-12)}$$

It is clear that the baseband signal component in each of $XSU_{ss}^{(m)}$, $XSU_{sc}^{(m)}$, $XSU_{cs}^{(m)}$, and $XSU_{cc}^{(m)}$ is represented by the second term in each of the above four equations, since the subcarrier frequency error, denoted by Θ'_{su}, is assumed to be very close to zero. The recorded mth subcarrier harmonic signals given in Eqs. (17) to (20) for any odd integer $1 \leq m \leq L$ are the corresponding baseband components in Eqs. (A-9) to (A-12). These signals can be extracted by lowpass filtering $XSU_{ss}^{(m)}$, $XSU_{sc}^{(m)}$, $XSU_{cs}^{(m)}$, and $XSU_{cc}^{(m)}$ with the filter $H_{su}(z)$, of which the bandwidth B_{su} is wide enough to accommodate both the carrier frequency offset and the maximum subcarrier frequency offsets (up to the highest subcarrier harmonic recorded) as well as the data modulation, namely,

$$B_{su} \geq (\Theta'_c + L\Theta'_{su} + \omega_d)/(2\pi) \quad \text{(in Hz)}$$

where ω_d is the bandwidth (in radians/sec) wherein most of the data power resides.

III. Derivation of Eqs. (40) and (41)

In referring to the subcarrier loop in Fig. 4, it is seen that the mth subcarrier harmonic waveforms $SU_{sc}^{(m)}$, $SU_{ss}^{(m)}$, $SU_{cc}^{(m)}$, and $SU_{cs}^{(m)}$ are demodulated as follows:

$$A^{(m)} = SU_{cs}^{(m)}\cos(\hat{\Theta}_c) - SU_{ss}^{(m)}\sin(\hat{\Theta}_c) \qquad \text{(A-13)}$$

$$B^{(m)} = SU_{cc}^{(m)}\cos(\hat{\Theta}_c) - SU_{sc}^{(m)}\sin(\hat{\Theta}_c) \qquad \text{(A-14)}$$

where the references $\sin(\hat{\Theta}_c)$ and $\cos(\hat{\Theta}_c)$ have the estimated carrier error phase $\hat{\Theta}_c$ obtained from the carrier tracking. It can be found that

$$A^{(m)} = \sqrt{P_d}\frac{2d}{m\pi}\cos(m\Theta_{su})\cos(\phi_c) + n_A^{(m)} \qquad \text{(A-15)}$$

$$B^{(m)} = \sqrt{P_d}\frac{2d}{m\pi}\sin(m\Theta_{su})\cos(\phi_c) + n_B^{(m)} \qquad \text{(A-16)}$$

where

$$n_A^{(m)} = n_{SU_{cs}}^{(m)}\cos(\hat{\Theta}_c) - n_{SU_{ss}}^{(m)}\sin(\hat{\Theta}_c) \qquad \text{(A-17)}$$

$$n_B^{(m)} = n_{SU_{cc}}^{(m)}\cos(\hat{\Theta}_c) - n_{SU_{sc}}^{(m)}\sin(\hat{\Theta}_c) \qquad \text{(A-18)}$$

It is clear that both $n_A^{(m)}$ and $n_B^{(m)}$ have a flat, one-sided PSD level equal to $N_0/2$ W/Hz within the frequency interval $[0, B_{su}]$ Hz, since $n_{SU_{ss}}^{(m)}$, $n_{SU_{sc}}^{(m)}$, $n_{SU_{cs}}^{(m)}$, and $n_{SU_{cc}}^{(m)}$ have the same PSD level ($N_0/2$, one-sided) within the same bandwidth (B_{su}). Note that B_{su} is the bandwidth of the filter $H_{su}(z)$ in Fig. 2. Furthermore, it can be shown that both $n_A^{(m)}$ and $n_B^{(m)}$ are not only independent of each other, but also independent of their counterparts at other subcarrier harmonics.

The signals $A^{(m)}$ and $B^{(m)}$ are multiplied by the subcarrier references and then added or subtracted, as shown in Fig. 4, to remove the subcarrier frequency error. The resulting signals, $X_i^{(m)}$ and $X_{q,su}^{(m)}$, are given as follows:

$$X_i^{(m)} = A^{(m)}\cos(m\hat{\Theta}_{su}) + B^{(m)}\sin(m\hat{\Theta}_{su}) \qquad \text{(A-19)}$$

$$X_{q,su}^{(m)} = B^{(m)}\cos(m\hat{\Theta}_{su}) - A^{(m)}\sin(m\hat{\Theta}_{su}) \qquad \text{(A-20)}$$

Substituting the expressions for $A^{(m)}$ and $B^{(m)}$, then letting $\phi_c = \Theta_c - \hat{\Theta}_c$ and $\phi_{su} = \Theta_{su} - \hat{\Theta}_{su}$, yields

$$X_i^{(m)} = \sqrt{P_d}\frac{2d}{m\pi}\cos(m\phi_{su})\cos(\phi_c) + n_{X_i}^{(m)} \qquad \text{(A-21)}$$

$$X_{q,su}^{(m)} = \sqrt{P_d}\frac{2d}{m\pi}\sin(m\phi_{su})\cos(\phi_c) + n_{X_{q,su}}^{(m)} \qquad \text{(A-22)}$$

where

$$n_{X_i}^{(m)} = n_A^{(m)} \cos(m\hat{\Theta}_{su}) + n_B^{(m)} \sin(m\hat{\Theta}_{su}) \quad \text{(A-23)}$$

$$n_{X_{q,su}}^{(m)} = n_B^{(m)} \cos(m\hat{\Theta}_{su}) - n_A^{(m)} \sin(m\hat{\Theta}_{su}) \quad \text{(A-24)}$$

In referring to the weighted summer in Fig. 4, it is seen that the mth subcarrier harmonic $X_i^{(m)}$ is weighted by $4/(m\pi)$ and then added to the other inphase harmonics to obtain the input to the inphase symbol accumulator, denoted as X_i. Hence,

$$X_i = \sum_{\substack{m=1 \\ m:odd}}^{L} X_i^{(m)} \frac{4}{m\pi}$$

$$= \sqrt{P_d} d \cos(\phi_c) \frac{8}{\pi^2} \sum_{\substack{m=1 \\ m:odd}}^{L} \frac{\cos(m\phi_{su})}{m^2} + n_{X_i} \quad \text{(A-25)}$$

where

$$n_{X_i} = \frac{4}{\pi} \sum_{\substack{m=1 \\ m:odd}}^{L} \frac{n_A^{(m)} \cos(m\hat{\Theta}_{su}) + n_B^{(m)} \sin(m\hat{\Theta}_{su})}{m} \quad \text{(A-26)}$$

Similarly, the input to the quadrature symbol accumulator, $X_{q,su}$, is obtained by weighing the mth subcarrier harmonic $X_{q,su}^{(m)}$ by $[4/(m\pi)]\sin[(m/2)\pi W_{su}]$ and then adding it to the other quadrature harmonics. Namely,

$$X_{q,su} = \sum_{\substack{m=1 \\ m:odd}}^{L} X_{q,su}^{(m)} \left[\frac{4}{m\pi} \sin\left(\frac{m}{2}\pi W_{su}\right) \right]$$

$$= \sqrt{P_d} d \cos(\phi_c) \frac{8}{\pi^2} \sum_{\substack{m=1 \\ m:odd}}^{L} \sin\left(\frac{m}{2}\pi W_{su}\right) \frac{\sin(m\phi_{su})}{m^2}$$

$$+ n_{X_{q,su}} \quad \text{(A-27)}$$

where

$$n_{X_{q,su}} = \frac{4}{\pi} \sum_{\substack{m=1 \\ m:odd}}^{L} \sin\left(\frac{m}{2}\pi W_{su}\right) \times \frac{n_B^{(m)} \cos(m\hat{\Theta}_{su}) - n_A^{(m)} \sin(m\hat{\Theta}_{su})}{m} \quad \text{(A-28)}$$

The variance of n_{X_i} in Eq. (A-26) can be found as

$$\sigma_{n_{X_i}}^2 = \mathbf{E}\left\{ \left(\frac{4}{\pi} \sum_{\substack{m=1 \\ m:odd}}^{L} \frac{n_A^{(m)} \cos(m\hat{\Theta}_{su}) + n_B^{(m)} \sin(m\hat{\Theta}_{su})}{m} \right)^2 \right\}$$

$$= \left(\frac{4}{\pi}\right)^2 \sum_{\substack{m=1 \\ m:odd}}^{L} \frac{\mathbf{E}\left\{(n_A^{(m)})^2\right\} \cos^2(m\hat{\Theta}_{su}) + \mathbf{E}\left\{(n_B^{(m)})^2\right\} \sin^2(m\hat{\Theta}_{su})}{m^2} \quad \text{(A-29)}$$

where

$$\mathbf{E}\left\{(n_A^{(m)})^2\right\} = \mathbf{E}\left\{(n_B^{(m)})^2\right\} = \frac{N_0}{2} B_{su} \quad \forall m$$

Here, B_{su} is the bandwidth of the filter $H_{su}(z)$ in Fig. 2. The last equality in Eq. (A-29) holds because $n_A^{(m)}$ and $n_B^{(m)}$ are mutually independent with respect to each other as well as their counterparts at other subcarrier harmonics. After letting the highest recorded subcarrier harmonic order L go to infinity, and then using the identity

$$\sum_{\substack{m=1 \\ m:odd}}^{\infty} \frac{1}{m^2} = \frac{\pi^2}{8}$$

it is easily shown that the variance of n_{X_i} approaches twice the variance of either $n_A^{(m)}$ or $n_B^{(m)}$. Therefore, n_{X_i} has a flat, one-sided PSD level equal to N_0 W/Hz, which is twice the PSD level of $n_A^{(m)}$ or $n_B^{(m)}$, within B_{su} Hz. Similarly, it can also be shown that $n_{X_{q,su}}$ in Eq. (A-28) has a flat, one-sided power spectral density level approximately equal to $(W_{su}N_0)$ W/Hz within the lowpass arm filter bandwidth B_{su} Hz. This result is as expected since windowing the quadrature subcarrier reference reduces the noise power by the factor W_{su}.

In referring to Fig. 4, it is seen that the subcarrier loop inphase and quadrature accumulator outputs, Y_i and $Y_{q,su}$, are obtained by averaging samples of X_i and $X_{q,su}$ over a symbol duration. Assuming that ϕ_c and $m\phi_{su}$ for all m are approximately constant over a symbol duration, Y_i and $Y_{q,su}$ are as given by Eqs. (40) and (41). It is also true that, since the one-sided PSD levels of n_{X_i} (N_0 W/Hz) and $n_{X_{q,su}}$ ([$W_{su}N_0$] W/Hz) are flat over B_{su} Hz and since B_{su} is much greater than the symbol rate ($1/T$), the variance of n_{Y_i} and the variance of $n_{Y_{q,su}}$ are equal to $N_0/(2T)$ and $W_{su}N_0/(2T)$, respectively. Furthermore, since

$$\left\{n_A^{(m)}, n_B^{(m)} \mid \text{ odd integer m }, \ 1 \le m \le L\right\}$$

is a set of mutually independent random samples, it is straightforward to show that n_{X_i} and $n_{X_{q,su}}$ are independent, which in turn assures that n_{Y_i} and $n_{Y_{q,su}}$ are also independent.

IV. Derivation of Eq. (44)

In referring to the suppressed carrier loop in Fig. 4, it is seen that the recorded mth subcarrier harmonic waveforms $SU_{sc}^{(m)}$, $SU_{ss}^{(m)}$, $SU_{cc}^{(m)}$, and $SU_{cs}^{(m)}$ are demodulated as follows:

$$C^{(m)} = SU_{ss}^{(m)} \cos(\hat{\Theta}_c) + SU_{cs}^{(m)} \sin(\hat{\Theta}_c) \qquad \text{(A-30)}$$

$$D^{(m)} = SU_{sc}^{(m)} \cos(\hat{\Theta}_c) + SU_{cc}^{(m)} \sin(\hat{\Theta}_c) \qquad \text{(A-31)}$$

where $\hat{\Theta}_c$ represents the estimated carrier error phase. After substituting Eqs. (17) to (20) into the above equations, it can be found that

$$C^{(m)} = -\sqrt{P_d}\frac{2d}{m\pi} \cos(m\Theta_{su}) \cos(\phi_c) + n_C^{(m)} \qquad \text{(A-32)}$$

$$D^{(m)} = -\sqrt{P_d}\frac{2d}{m\pi} \sin(m\Theta_{su}) \sin(\phi_c) + n_D^{(m)} \qquad \text{(A-33)}$$

where

$$n_C^{(m)} = n_{SU_{ss}}^{(m)} \cos(\hat{\Theta}_c) + n_{SU_{cs}}^{(m)} \sin(\hat{\Theta}_c) \qquad \text{(A-34)}$$

$$n_D^{(m)} = n_{SU_{sc}}^{(m)} \cos(\hat{\Theta}_c) + n_{SU_{cc}}^{(m)} \sin(\hat{\Theta}_c) \qquad \text{(A-35)}$$

and $n_{SU_{ss}}^{(m)}$, $n_{SU_{sc}}^{(m)}$, $n_{SU_{cs}}^{(m)}$, and $n_{SU_{cc}}^{(m)}$ are as given by Eqs. (21) to (24).

It is clear that both $n_C^{(m)}$ and $n_D^{(m)}$ have a flat, one-sided PSD level equal to $N_0/2$ W/Hz within the frequency interval $[0, B_{su}]$ Hz, since $n_{SU_{ss}}^{(m)}$, $n_{SU_{sc}}^{(m)}$, $n_{SU_{cs}}^{(m)}$, and $n_{SU_{cc}}^{(m)}$ have the PSD level equal to $N_0/2$ W/Hz within the same band. Note that B_{su} is the bandwidth of the filter $H_{su}(z)$ in Fig. 2. It is easy to show that both $n_C^{(m)}$ and $n_D^{(m)}$ are not only independent of each other, but also independent of their counterparts at other subcarrier harmonics. Moreover, it can be shown that

$$\left\{n_A^{(m)}, n_B^{(m)}, n_C^{(m)}, n_D^{(m)} \mid \text{ odd integer m, } 1 \le m \le L\right\} \qquad \text{(A-36)}$$

is a set of mutually independent random samples.

The signals $C^{(m)}$ and $D^{(m)}$ are multiplied by the subcarrier references and then added to remove the subcarrier frequency error. The resulting signal $X_{q,sca}^{(m)}$ is as follows:

$$X_{q,sca}^{(m)} = C^{(m)} \cos(m\hat{\Theta}_{su}) + D^{(m)} \sin(m\hat{\Theta}_{su}) \qquad \text{(A-37)}$$

$$= \sqrt{P_d}\frac{2d}{m\pi} \cos(m\phi_{su}) \sin(\phi_c) + n_{X_{q,sca}}^{(m)} \qquad \text{(A-38)}$$

where

$$n_{X_{q,sca}}^{(m)} = n_C^{(m)} \cos(m\hat{\Theta}_{su}) + n_D^{(m)} \sin(m\hat{\Theta}_{su}) \qquad \text{(A-39)}$$

In a manner similar to forming the weighted sums in the subcarrier tracking, the mth subcarrier harmonic, $X_{q,sca}^{(m)}$, is weighted by $4/(m\pi)$ and then added to the other quadrature harmonics to obtain $X_{q,sca}$, the input to the quadrature arm accumulator. Thus,

$$X_{q,sca} = \sqrt{P_d} d \sin(\phi_c) \frac{8}{\pi^2} \sum_{\substack{m=1 \\ m:odd}}^{L} \frac{\cos(m\phi_{su})}{m^2} + n_{X_{q,sca}} \tag{A-40}$$

where

$$n_{X_{q,sca}} = \frac{4}{\pi} \sum_{\substack{m=1 \\ m:odd}}^{L} \frac{n_C^{(m)} \cos(m\hat{\Theta}_{su}) + n_D^{(m)} \sin(m\hat{\Theta}_{su})}{m} \tag{A-41}$$

Using the same procedure as in the derivation of the PSD level of n_{X_i}, it can be easily verified that $n_{X_{q,sca}}$ has a PSD level approximately equal to N_0 W/Hz within the frequency interval $[0, B_{su}]$ Hz. In referring to Fig. 4, it is seen that the suppressed carrier loop quadrature arm accumulator output, $Y_{q,sca}$, is obtained by averaging samples of $X_{q,sca}$ over a symbol duration. Assuming that ϕ_c and $m\phi_{su}$, for all m, are approximately constant over a symbol duration, the signal $Y_{q,sca}$ is as given in Eq. (44). Furthermore, assuming that B_{su} is much greater than the symbol rate $(1/T)$, the variance of $n_{Y_{q,sca}}$ can be shown to be equal to $N_0/(2T)$.

By using Eq. (A-36), it is straightforward to show that n_{X_i} and $n_{X_{q,sca}}$ are independent, which in turn assures that n_{Y_i} and $n_{Y_{q,sca}}$ are independent.

Table 1. Loss in data power as a function of subcarrier harmonics.

Highest recorded subcarrier harmonic, L	Number of subcarrier harmonics at A/D output, n	Loss, dB
1	1	0.91
3	2	0.45
5	3	0.30
7	4	0.22
9	5	0.18

Table 2(a). Required number of operations for the demodulation function (n is the number of harmonics recorded).

Mode	Multiplication operations per sample	Multiplication operations per symbol	Addition operations per sample	Addition operations per symbol	Lookup table
Residual carrier with subcarrier	$6 + 8n$	14	$13 + 6n$	24	$4 + 2n$
Suppressed carrier with subcarrier	$4 + 14n$	14	$12 + 10n$	25	$4 + 2n$
Suppressed carrier with no subcarrier	6	12	10	17	4

Table 2(b). Detailed list of operations.

Three modes with types of operations for each	Number of multiplication operations per sample	Number of multiplication operations per symbol	Number of addition operations per sample	Number of addition operations per symbol	Lookup table
Residual carrier with subcarrier					
Residual carrier phase detector	2	0	1	0	0
Residual carrier accumulator	0	0	1	0	0
Residual carrier loop filter	0	3	0	5	0
Residual carrier numerically controlled oscillator	0	1	1	1	2
Subcarrier phase detector	$8n$	0	$2+6n$	0	0
Subcarrier accumulator	0	0	0	1	0
Subcarrier loop filter	0	3	0	5	0
Subcarrier numerically controlled oscillator	0	1	1	1	$2n$
Symbol synchronization phase detector	0	1	1	1	0
Symbol synchronization accumulator	0	0	0	1	0
Symbol synchronization loop filter	0	3	0	5	0
Symbol synchronization numerically controlled oscillator	0	1	1	1	2
Residual carrier lock detector	2	0	2	0	0
Subcarrier lock detector	2	0	2	0	0
Split-symbol SSNR estimator	0	1	1	3	0
Total	$6+8n$	14	$13+6n$	24	$4+2n$
Suppressed carrier with subcarrier					
Carrier/subcarrier phase detector	$14n$	0	$3+10n$	0	0
Carrier/subcarrier accumulator	0	0	0	2	0
Carrier/subcarrier loop filter	0	6	0	10	0
Carrier/subcarrier numerically controlled oscillator	0	2	2	2	$2+2n$
Symbol synchronization phase detector	0	1	1	1	0
Symbol synchronization accumulator	0	0	0	1	0
Symbol synchronization loop filter	0	3	0	5	0
Symbol synchronization numerically controlled oscillator	0	1	1	1	2
Suppressed carrier lock detector	2	0	2	0	0
Subcarrier lock detector	2	0	2	0	0
Split-symbol SSNR estimator	0	1	1	3	0
Total	$4+14n$	14	$12+10n$	25	$4+2n$
Suppressed carrier with no subcarrier					
Suppressed carrier phase detector	4	1	4	0	0
Suppressed carrier accumulator	0	1	0	0	0
Suppressed carrier loop filter	0	3	0	5	0
Suppressed carrier numerically controlled oscillator	0	1	1	1	2
Symbol synchronization phase detector	0	1	1	1	0
Symbol synchronization accumulator	0	0	0	1	0
Symbol synchronization loop filter	0	3	0	5	0
Symbol synchronization numerically controlled oscillator	0	1	1	1	2
Suppressed carrier lock detector	2	0	2	0	0
Split-symbol SSNR estimator	0	1	1	3	0
Total	6	12	10	17	4

Table 3. Throughput comparison (n is the number of harmonics recorded).

System	SPECfp92	Percent throughput utilized ($n = 3$)	Percent throughput utilized ($n = 5$)
Sun SS-2	22.8	100	151
Sun SS-10/30	52.9	43	65
Sun SS-10/41	64.7	35	53
Sun SS-10/52	71.4/CPU	32 (on one CPU)	48 (on one CPU)
SGI Crimson	63.4	36	54
HP 710	47.6	48	72
HP 750	75.0	30	45
IBM 340	51.9	44	66
IBM 560	85.6	27	41
Intel Xpress (486)	14.0	160	242

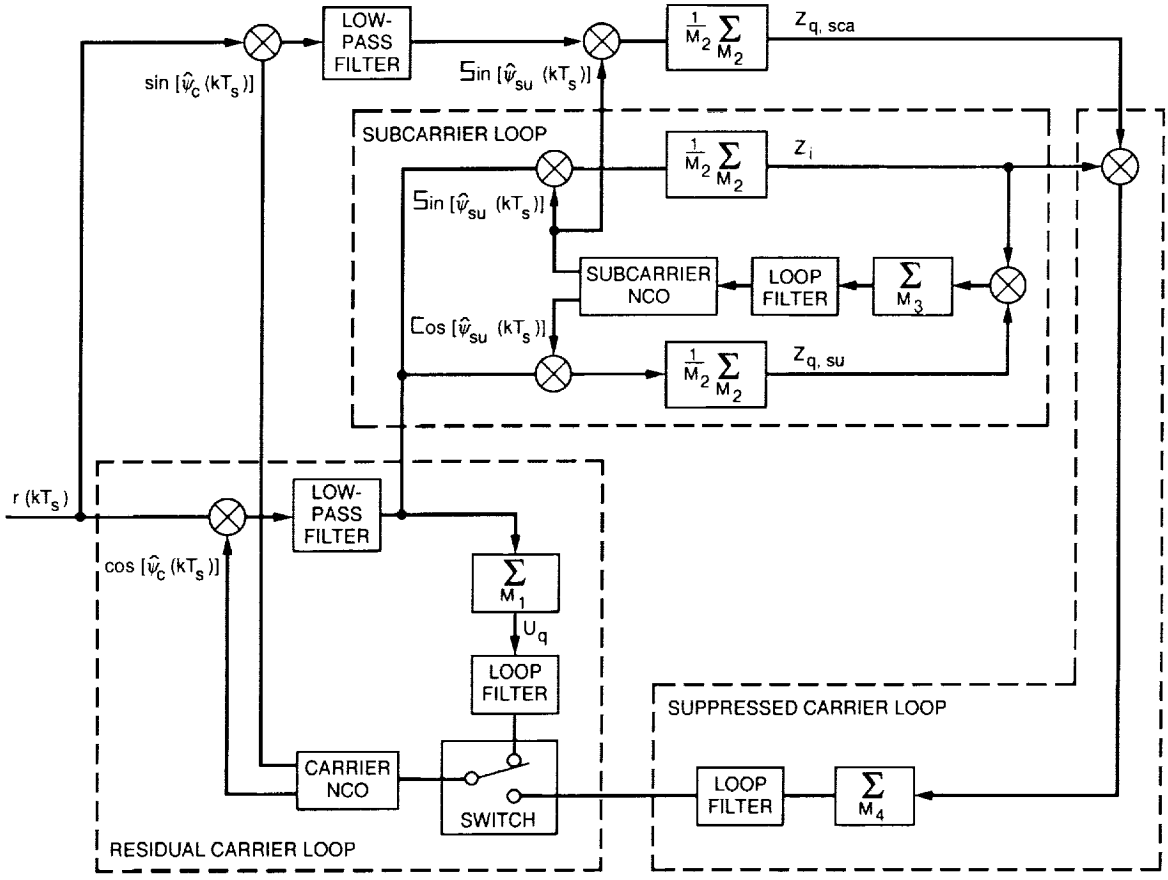


Fig. 1. A simplified block diagram of the ARX II.

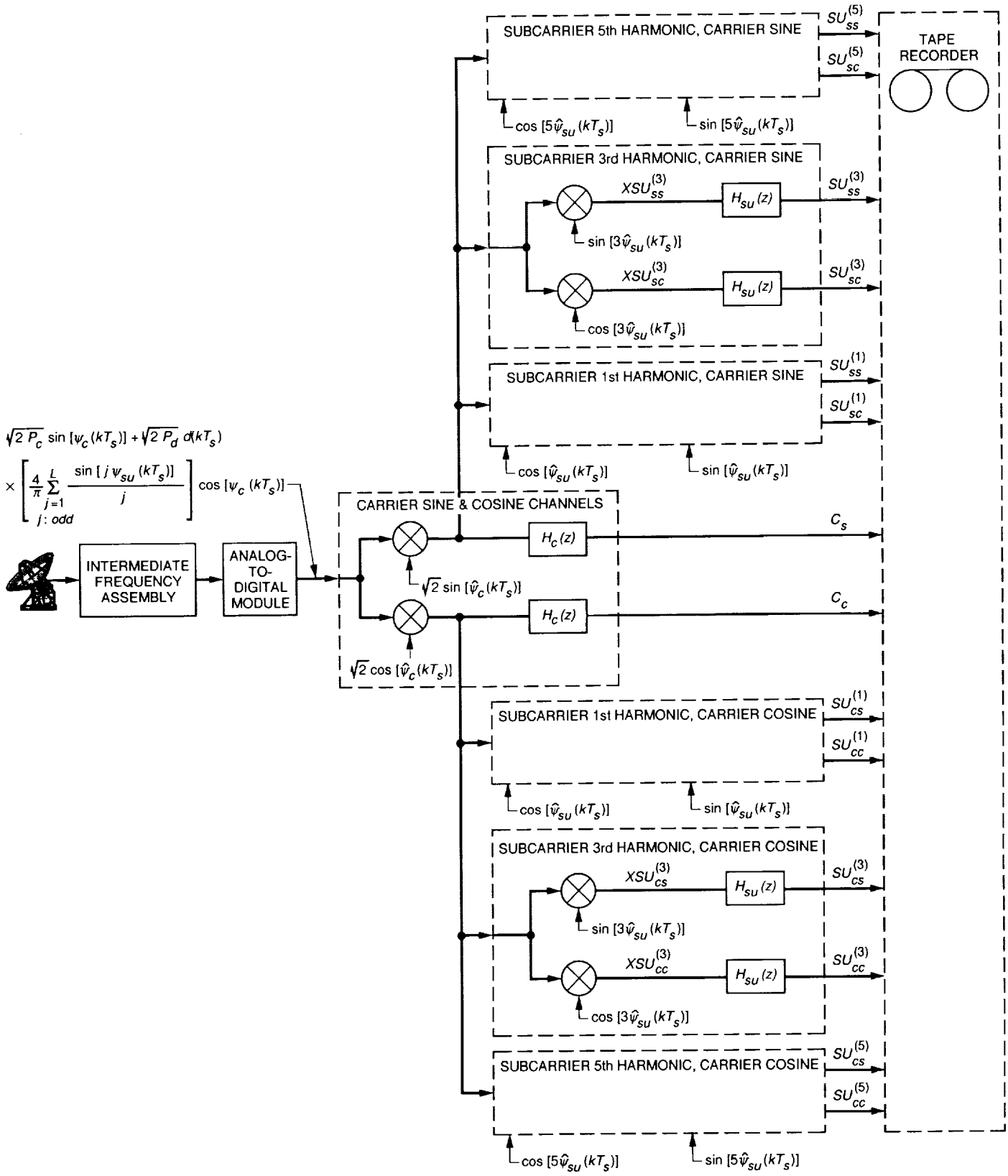


Fig. 2. The two-stage downconversion recording scheme.

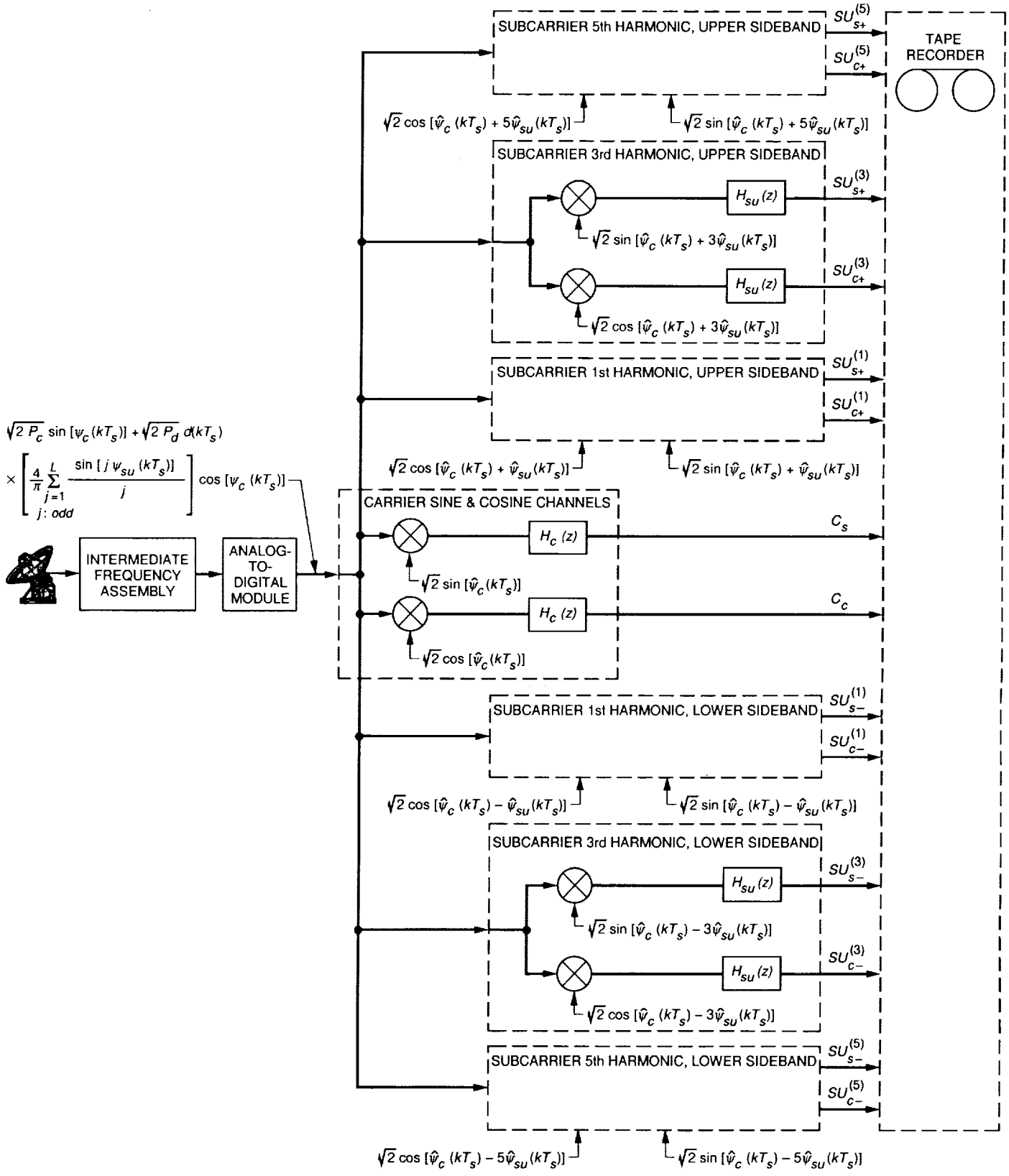


Fig. 3. The direct downconversion recording scheme.

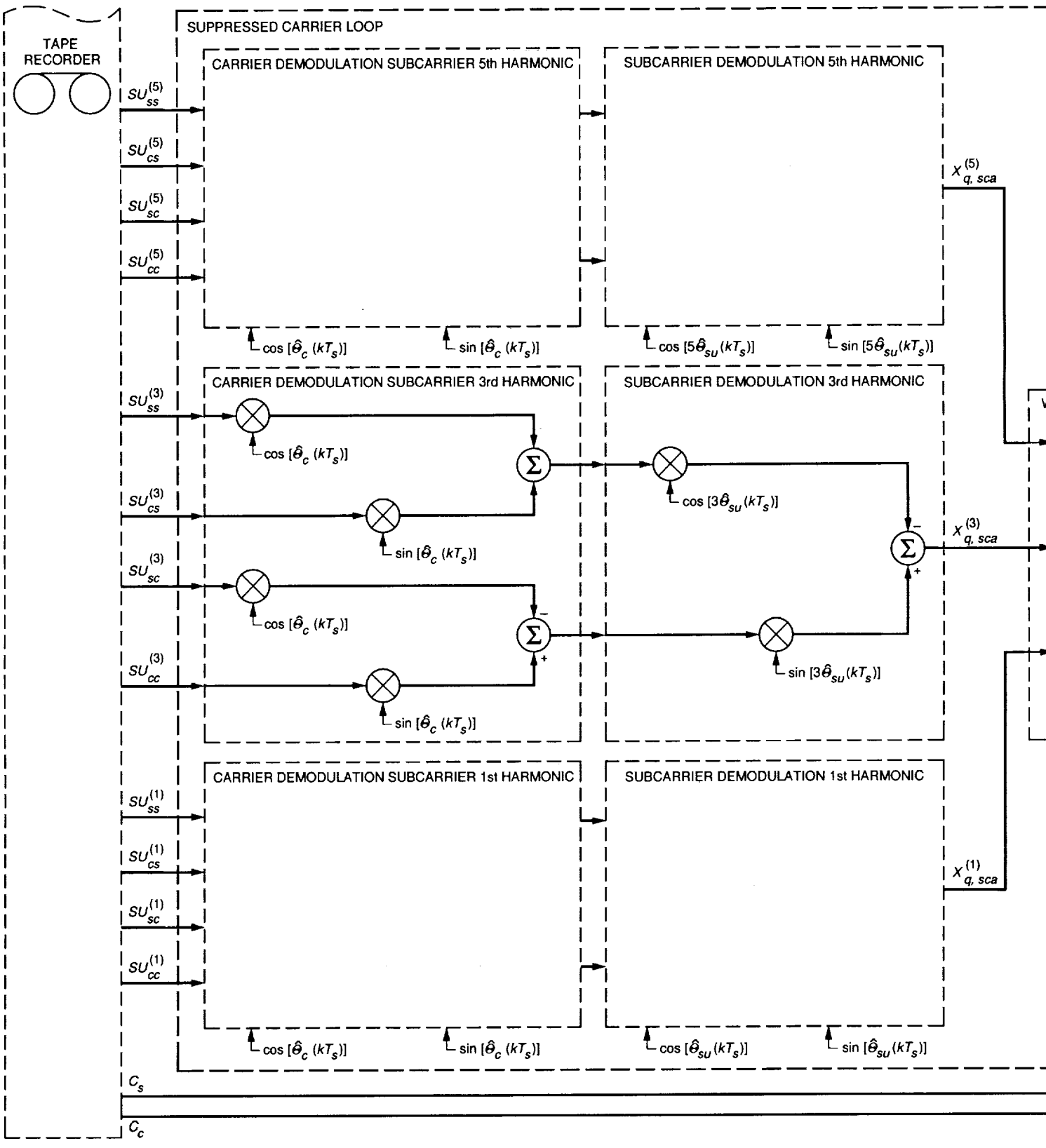


Fig. 4 (contd).

FOLDOUT FRAME

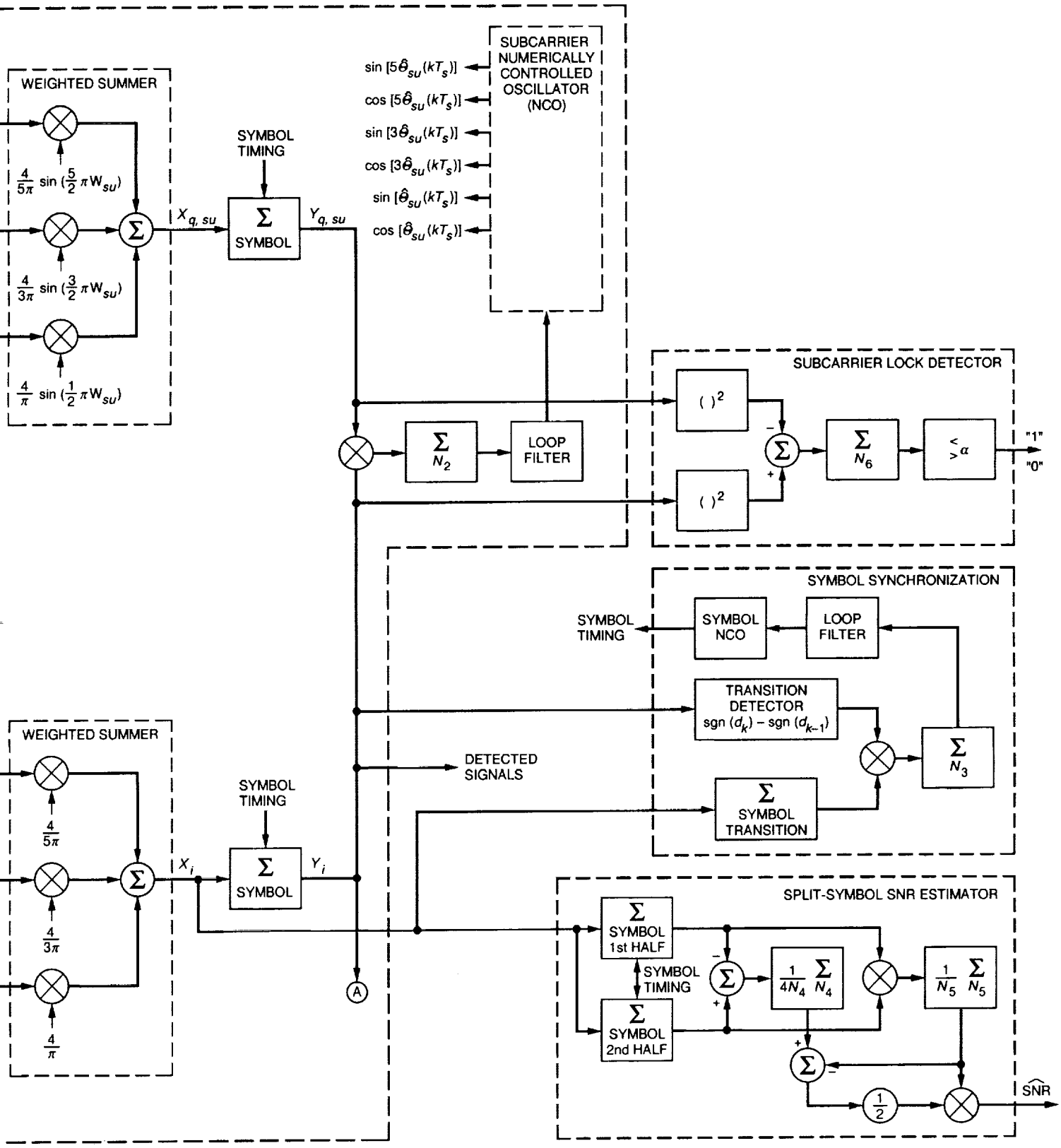
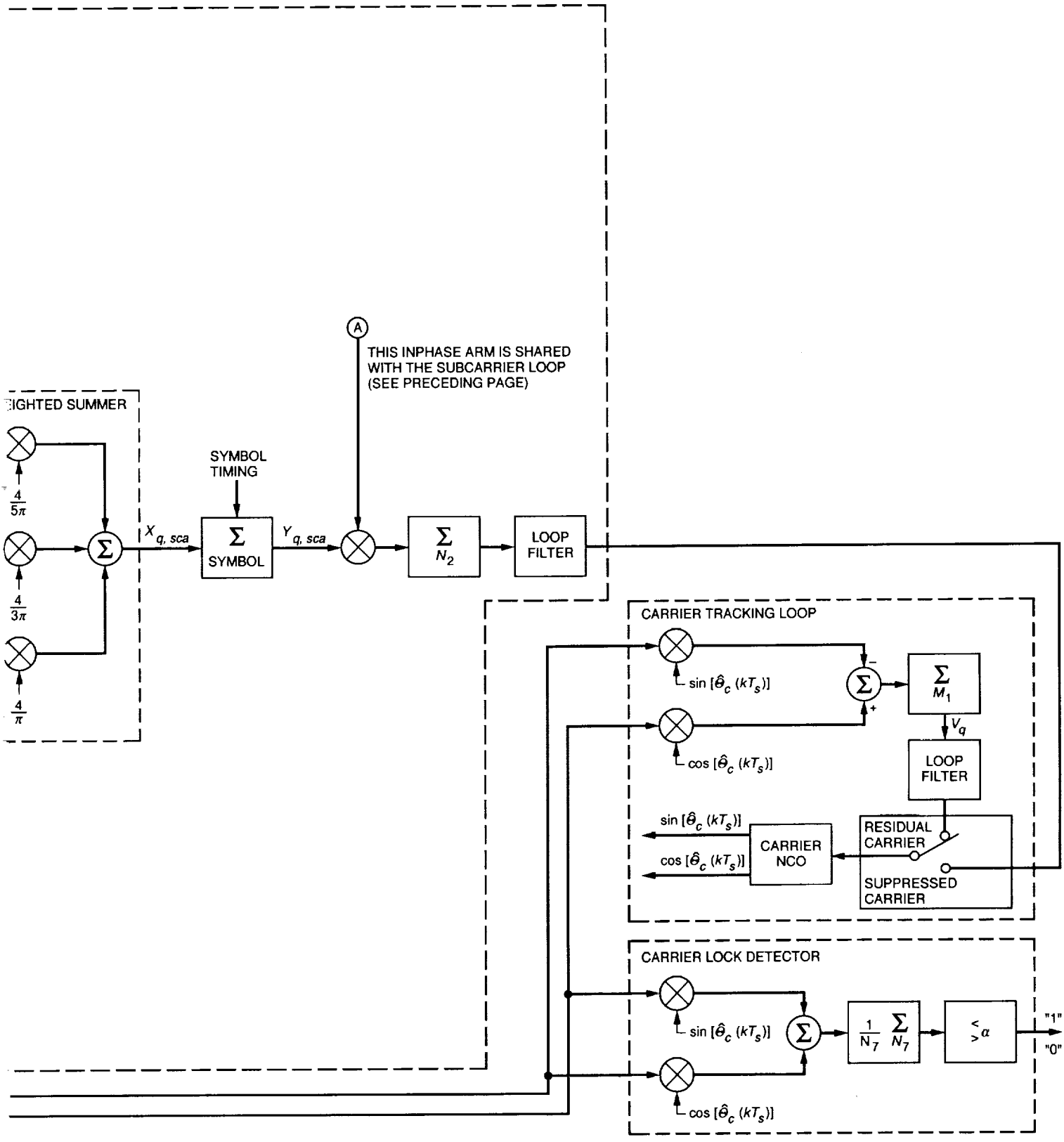
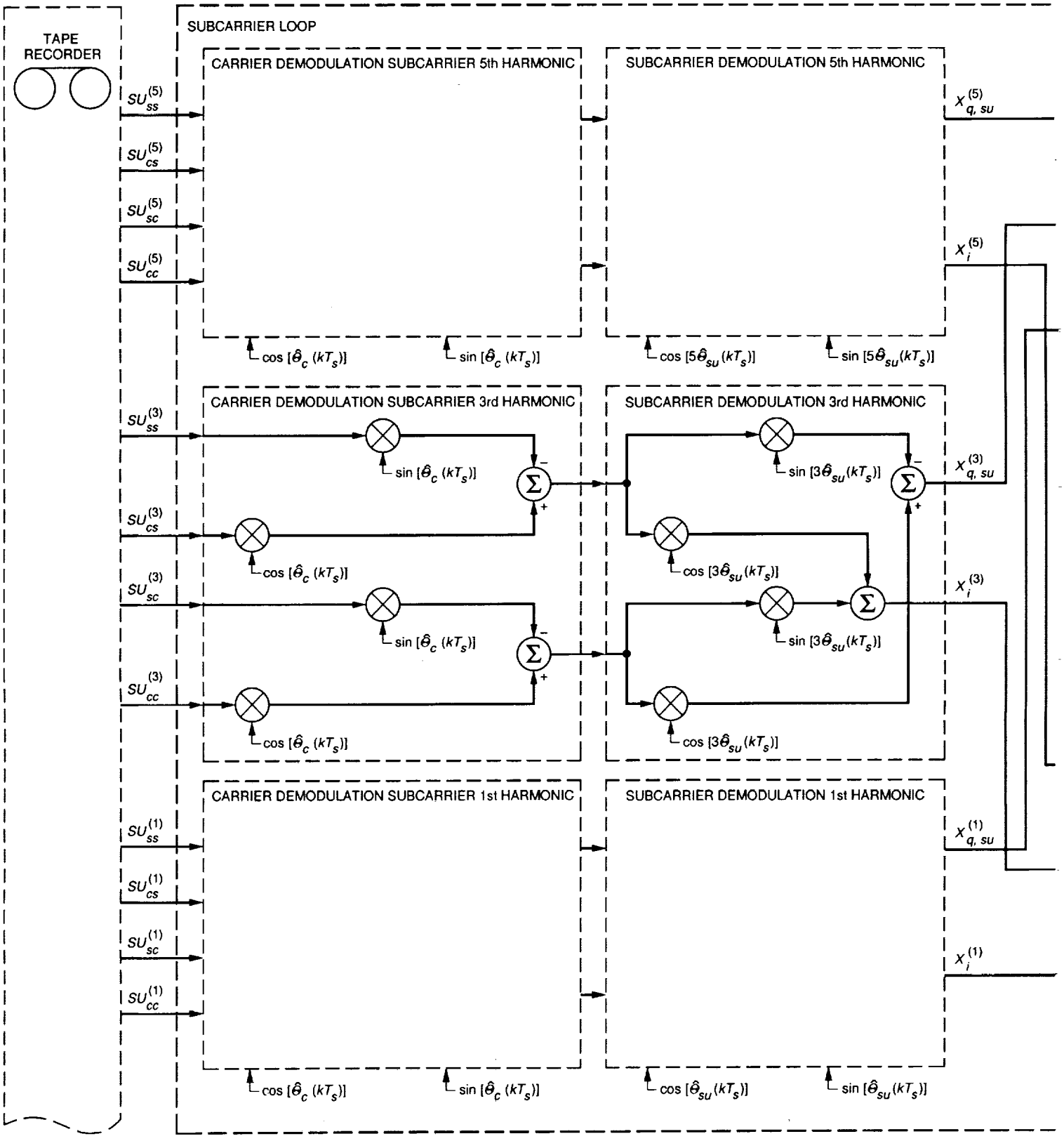


Fig. 4. The coherent baseband demodulator for the two-stage recording scheme.

FOLDOUT FRAME 2



FOLDOUT FRAME 2



FOLDOUT FRAME

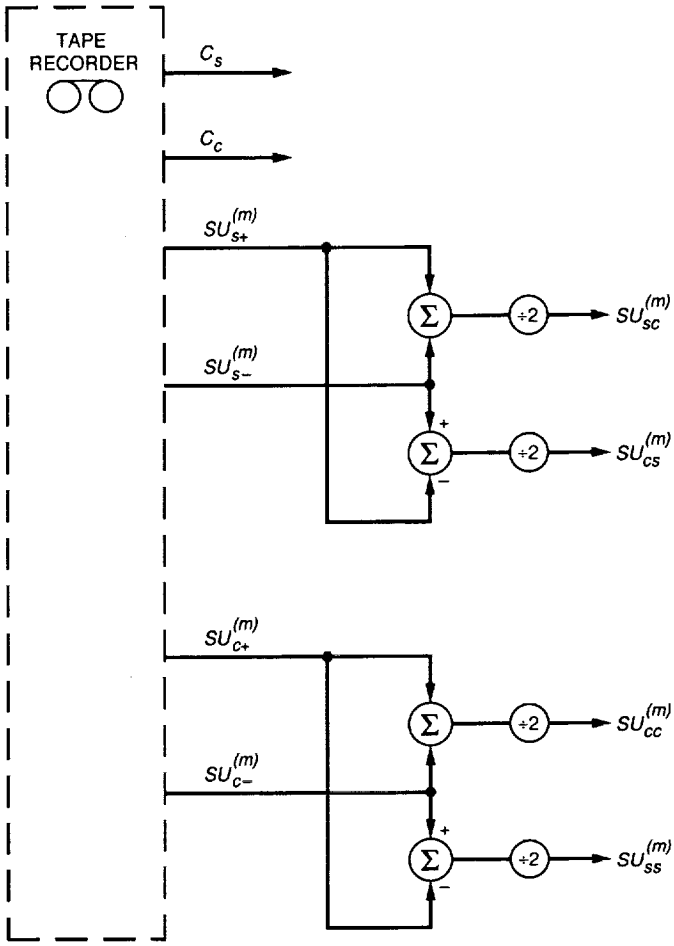


Fig. 5. The processing required to generate the signals $SU_{ss}^{(m)}$, $SU_{sc}^{(m)}$, $SU_{cs}^{(m)}$, $SU_{cc}^{(m)}$ from the signals $SU_{s+}^{(m)}$, $SU_{s-}^{(m)}$, $SU_{c+}^{(m)}$, and $SU_{c-}^{(m)}$.

Symbol Signal-to-Noise Ratio Loss in Square-Wave Subcarrier Downconversion

Y. Feria and J. Statman
Communications Systems Research Section

This article presents the simulated results of the signal-to-noise ratio (SNR) loss in the process of a square-wave subcarrier downconversion. In a previous article [2], the SNR degradation was evaluated at the output of the downconverter based on the signal and noise power change. Unlike in the previous article, the SNR loss is defined here as the difference between the actual and theoretical symbol SNR's for the same symbol-error rate at the output of the symbol matched filter. The results show that an average SNR loss of 0.3 dB can be achieved with tenth-order infinite impulse response (IIR) filters. This loss is a 0.2-dB increase over the SNR degradation in the previous analysis where neither the signal distortion nor the symbol detector was considered.

I. Introduction

A signal with a downconverted square-wave subcarrier suffers distortion caused by the nonideal filtering, the use of a finite number of harmonics, the data bandwidth cutoff, and so on. This distortion results in a loss of signal-to-noise ratio (SNR). In a previous article [2], the SNR degradation at the output of the downconverter was measured under the following definition:

$$SNR \text{ degradation, dB} = SNR_{\text{ideal}} - SNR_{\text{real}}$$

where SNR_{ideal} and SNR_{real} are the SNR's after the square-wave subcarrier downconversion using ideal and realizable filters, respectively. The analysis based on this definition, however, does not include the qualitative changes, such as distortion, that may have an impact on the symbol-error rate (see Fig. 1), nor does it include the effect of the symbol detector on the symbol SNR loss. For a complete study, the downconverter output is further decimated and fed into a symbol detector, and the symbol-error rate is then determined and compared with the theoretical symbol-error probability for binary phase-shift keying (BPSK) without downconversion [3]. Unlike the SNR degradation defined previously [2], the SNR loss now is defined as the difference between the required SNR and the theoretical SNR for the same symbol-error rate.

II. Simulation Procedure

The simulated procedure is illustrated in Fig. 2, where the subcarrier down-mixer has been described in [1,2]. The simulation assumes symbol synchronization, zero subcarrier phase, and known subcarrier frequency; hence, it focuses only on losses due to the nonideal filtering, the imperfect delay compensation, and the data bandwidth cutoff in the down-mixing and symbol-detecting processes. The simulation employs IIR filters with the bandwidths of 2, 4,

error rate is compared with both the theoretical symbol-error probability for BPSK and the simulated results from the direct symbol detection without the downconversion.

The simulation procedure is as follows: a square-wave subcarrier at a frequency of 22.5 kHz is modulated by a pseudo-random sequence with clock time of 1/1000 sec. A white Gaussian noise is added and its variance is adjusted so that the symbol SNR, E_s/N_o, takes values from -10 to 10 dB. The relationship between the random noise variance, σ^2, and the symbol SNR, E_s/N_o, is

$$\sigma^2 = \frac{P_s f_s}{2 f_d (E_s/N_o)}$$

where P_s is the subcarrier power, f_s is the sample frequency, and $1/f_d$ is the pseudo-noise (PN) sequence clock time. The noise-contaminated signal is then passed through a bank of bandpass filters (BPF's) with center frequencies at the first, third, and fifth harmonics of the square-wave subcarrier. Each of the BPF outputs is multiplied by a proper down-mixing signal, and then the products are added. The sum of the products is then passed through a lowpass filter (LPF). The subcarrier now has a lower frequency. The total signal bandwidth is reduced from 5 times the original subcarrier frequency plus a single-sided bandwidth of the data to 5 times the new subcarrier frequency plus a single-sided bandwidth of the data.

After the down-mixing, the original high sample rate is no longer necessary; hence, the signal is decimated by an integer number equal to $\lfloor f_s/(6f_B) \rfloor$, where f_s is the original sample rate and f_B is the BPF bandwidth. The decimated samples are then multiplied by the sum of the first three harmonics at the downconverted subcarrier frequency and fed into an integrate-and-dump filter for the symbol detection. The symbol-error rate is obtained by taking the ratio of the number of the incorrect symbol detections and the total number of symbols.

At the downconverter output, a delay due to the nonideal filtering is compensated for. The compensation is done in terms of sample periods before the decimation for the higher resolution. Note that this compensation, however, may be off by a fraction of a sample period since the delay period may not be an integer multiple of the sample period. The SNR degradation due to the symbol synchronization offset has been previously studied [4], and is thus not discussed in this article. The symbol SNR loss from the downconversion is then determined by taking the difference between the actual and the ideal E_s/N_o in decibels required to achieve the same symbol-error rate.

Three different BPF bandwidths are used in the simulation. The sample rates before and after the downconversion, the total number of samples, and the number of symbols used in each case are summarized in Table 1.

III. Results

For the BPF bandwidths of 2, 4, and 6 kHz, the symbol-error rates are obtained through the simulations with the symbol SNR varied from -10 to 10 dB. The comparisons between the simulated results and the theoretical symbol-error probabilities for BPSK are shown in Fig. 3. The theoretical symbol-error probability for BPSK is computed with the complementary error function [3], $\text{erfc}(\sqrt{E_s/N_o})$, with an accuracy of 10^{-5}. Therefore, the measured symbol-error rates for a symbol SNR greater than 6 dB may not be very accurate. Another problem is the limitation of the number of symbols used in the simulations. In this case, 250,000 symbols are used. For a symbol-error rate on the order of 10^{-5}, it may be necessary to simulate 10^6 symbols, which is beyond the available time and resource constraints.

To obtain the symbol SNR loss for a given symbol-error rate, first the square of the inverse of the complementary error function of two times the symbol-error rate is evaluated with an accuracy of 10^{-10}. The obtained actual SNR is then compared with the theoretical SNR. The difference of the two SNR's, ΔSNR, is the SNR loss. The SNR loss is computed for each simulation point, and the results are shown in Fig. 4. The points that are beyond 6 dB are not accurate, as mentioned earlier; therefore, they may not be shown in the figure.

For each BPF bandwidth, the SNR losses are averaged, and the results are given in Table 2, where DC stands for downconversion, and SD stands for symbol detection. The SNR degradations in the downconversion process without symbol detection in Table 2 were obtained in [2]. The SNR loss in the process of symbol detection without downconversion is also obtained and shown in Table 2. Furthermore, the relationship between the SNR loss and the BPF bandwidth is shown in Fig. 5.

For the BPF bandwidth varied from 2 to 6 kHz, the average symbol SNR loss decreased from 0.8745 to 0.2850 dB in the simulations. The loss is higher than the 0.5- to 0.1-dB SNR degradation computed in the previous analysis [2] without the symbol detection. However, the decrease in SNR loss with wider filter bandwidths agrees with the SNR degradation obtained in the previous analysis. The SNR loss includes the effect of the cutoff of the side lobes of the PN data signal spectrum, the effect of the nonideal filtering, and the imperfect delay compensation.

0.1-dB SNR degradation computed in the previous analysis [2] without the symbol detection. However, the decrease in SNR loss with wider filter bandwidths agrees with the SNR degradation obtained in the previous analysis. The SNR loss includes the effect of the cutoff of the side lobes of the PN data signal spectrum, the effect of the nonideal filtering, and the imperfect delay compensation.

For example, when the BPF bandwidth is 6 kHz, the average symbol SNR loss is 0.2850 dB. This loss is higher than the SNR degradation in the previous analysis [2], which is about 0.1 dB. The loss is due to several factors. First, the nonideal filtering causes signal distortion. Second, when the group delay of the output signal at the downconverter is not an integer multiple of the sample period, there is an SNR loss due to symbol timing misalignment. Third, when the decimated signal is multiplied by the sum of the harmonics, the high-frequency terms of the product can cause aliasing if the original signal is sampled at the Nyquist rate. Last, the integrate-and-dump filter acts as a lowpass filter and its performance is not as good as that of an ideal LPF. As the bandpass filter bandwidth gets narrower, the loss becomes larger. This is partly due to the cutoff of the PN data signal bandwidth [2], and also when the filter bandwidth needs to be narrower, the required filter order may be higher for a better performance.

Several attempts were unsuccessful in reducing the SNR loss. First, since the integrate-and-dump filter does not perform as an ideal LPF, an LPF is put in front of the integrate-and-dump filter. However, the symbol-error rate increases. The reason is that the transition region is enlarged by lowpass filtering.

It is also found that in the symbol detection process, if the incoming signal is multiplied by a square wave instead of by the first three harmonics of the square wave, the symbol SNR loss increases. The reason is that when a sine wave is multiplied by a square wave, assuming perfect timing and frequency alignment, the resulting rectified sine wave contains a constant term plus an infinite series of harmonics in its Fourier series expansion, which is not band limited. After sampling at a finite rate, aliasing occurs. If, on the other hand, only the first three harmonics are multiplied by the incoming signal, aliasing will not occur given a proper sample rate.

A higher sample rate of 324 kHz was also employed at the input of the downconverter for a 6-kHz BPF bandwidth, but the SNR loss did not reduce significantly.

IV. Conclusion

This article presents the symbol SNR loss due to the process of the square-wave subcarrier downconversion. The symbol SNR loss is measured as the difference between the actual symbol SNR and the theoretical SNR for the same symbol-error rate. An average loss of 0.3 dB was achieved with tenth-order IIR filters.

Acknowledgments

The authors thank Dr. Edgar Satorius for his many helpful suggestions and contributions to this work. Thanks are also due to David Rogstad for his helpful discussions. The authors are especially grateful to Todd Chauvin and David Watola for their considerable help in the simulation implementation.

References

[1] Y. Feria and J. Statman, "Bandwidth Compression of Noisy Signal With Square-Wave Subcarrier," *The Telecommunications and Data Acquisition Progress Report 42-109*, vol. January–March 1992, Jet Propulsion Laboratory, Pasadena, California, pp. 170–178, May 15, 1992.

[2] Y. Feria and J. Statman, "SNR Degradation in Square-Wave Subcarrier Down-conversion," *The Telecommunications and Data Acquisition Progress Report 42-111*, vol. July–September 1992, Jet Propulsion Laboratory, Pasadena, California, pp. 192–201, November 15, 1992.

[3] J. G. Proakis, *Digital Communications*, New York: McGraw-Hill, 1983.

[4] R. Sadr and W. Hurd, "Detection of Signals by the Digital Integrate-and-Dump Filter With Offset Sampling," *The Telecommunications and Data Acquisition Progress Report 42-91*, vol. July–September 1987, Jet Propulsion Laboratory, Pasadena, California, pp. 158–173, November 15, 1987.

Table 1. Simulation conditions.

BPF bandwidth, kHz	2	4	6
Original sample rate, kHz	264	264	288
Reduced sample rate, kHz	12	24	36
Samples per symbol	12	24	36
Number of samples	6.6×10^6	6.6×10^6	7.2×10^6
Number of symbols	25×10^3	25×10^3	25×10^3

Table 2. SNR loss and degradation, dB.

BPF bandwidth, kHz	2	4	6
Loss in DC with SD	0.8745	0.4218	0.2850
Degradation in DC without SD	0.3964	0.1655	0.0606
Loss in SD without DC	0.0288	0.0288	0.0288

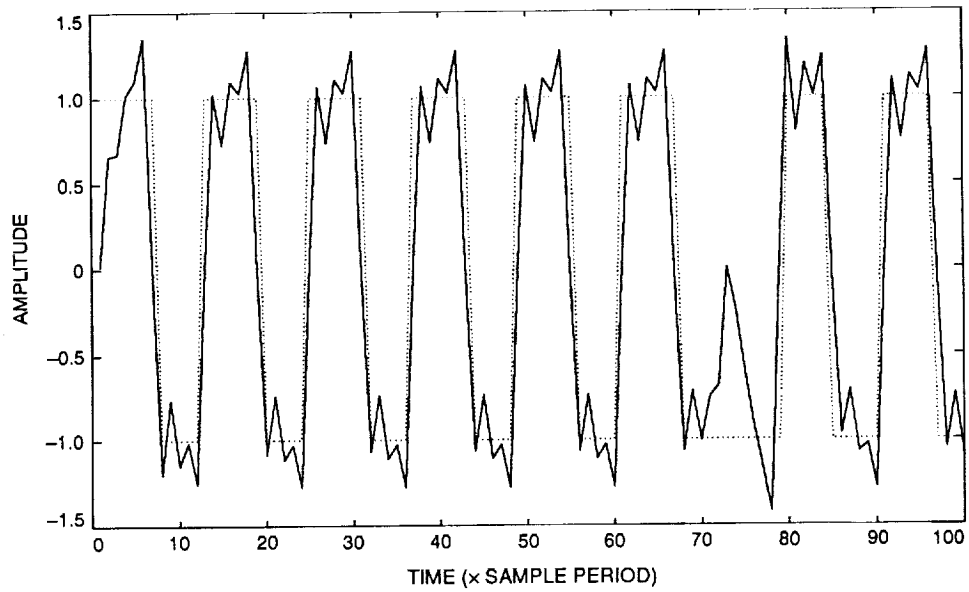


Fig. 1. Signal distortion due to the subcarrier downconversion.

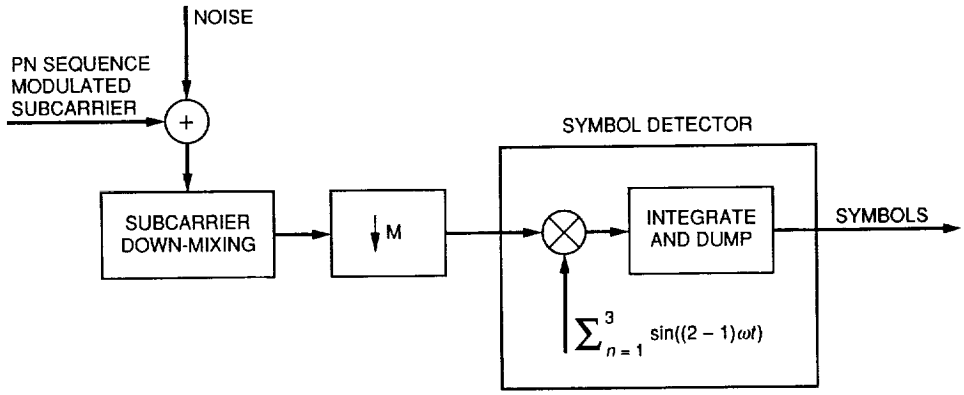
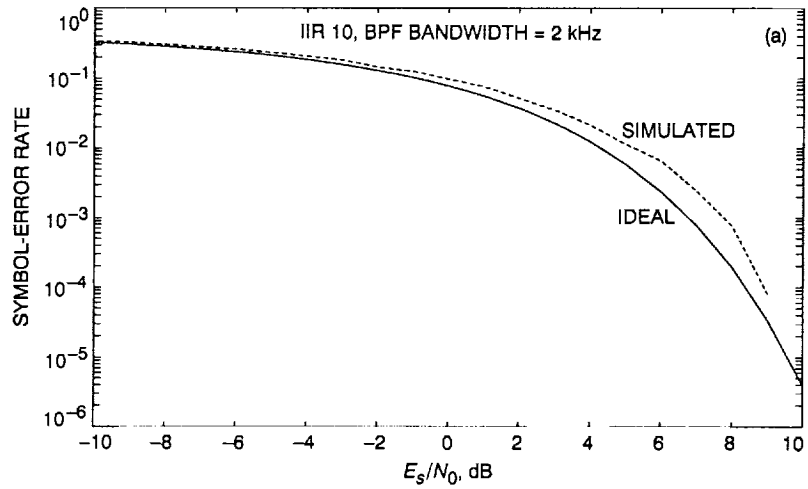
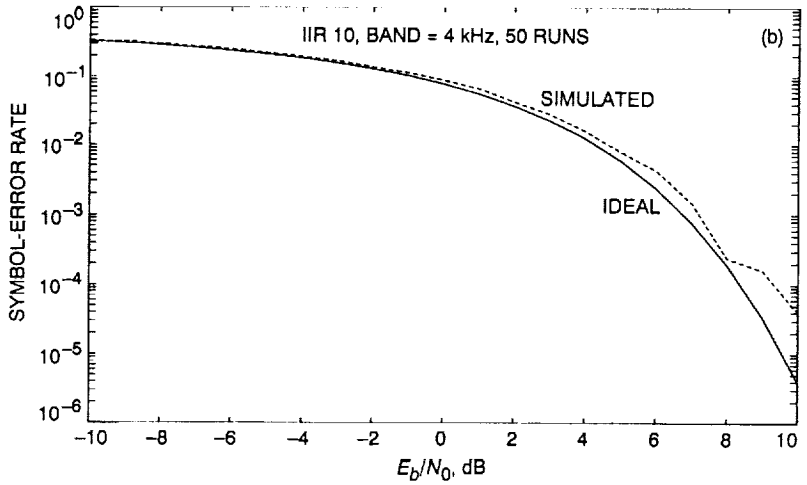


Fig. 2. Subcarrier downconversion and symbol detector.





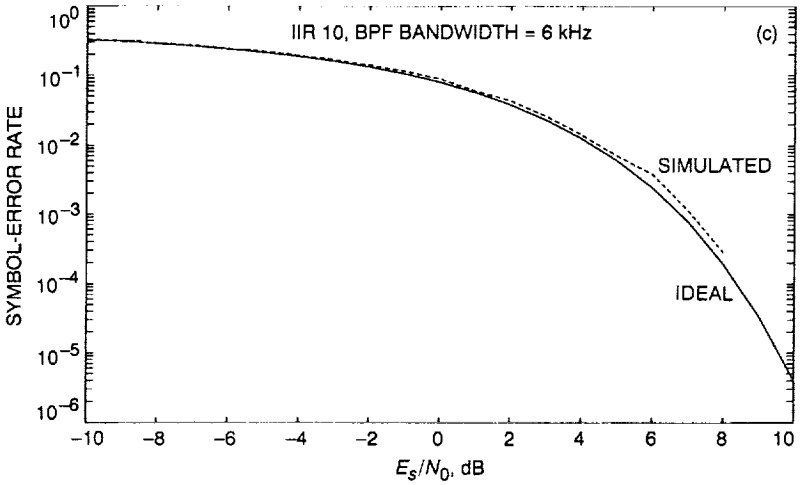


Fig. 3. Symbol-error rate versus symbol SNR for BPF bandwidth: (a) 2 kHz, (b) 4 kHz and (c) 6 kHz.

C.2

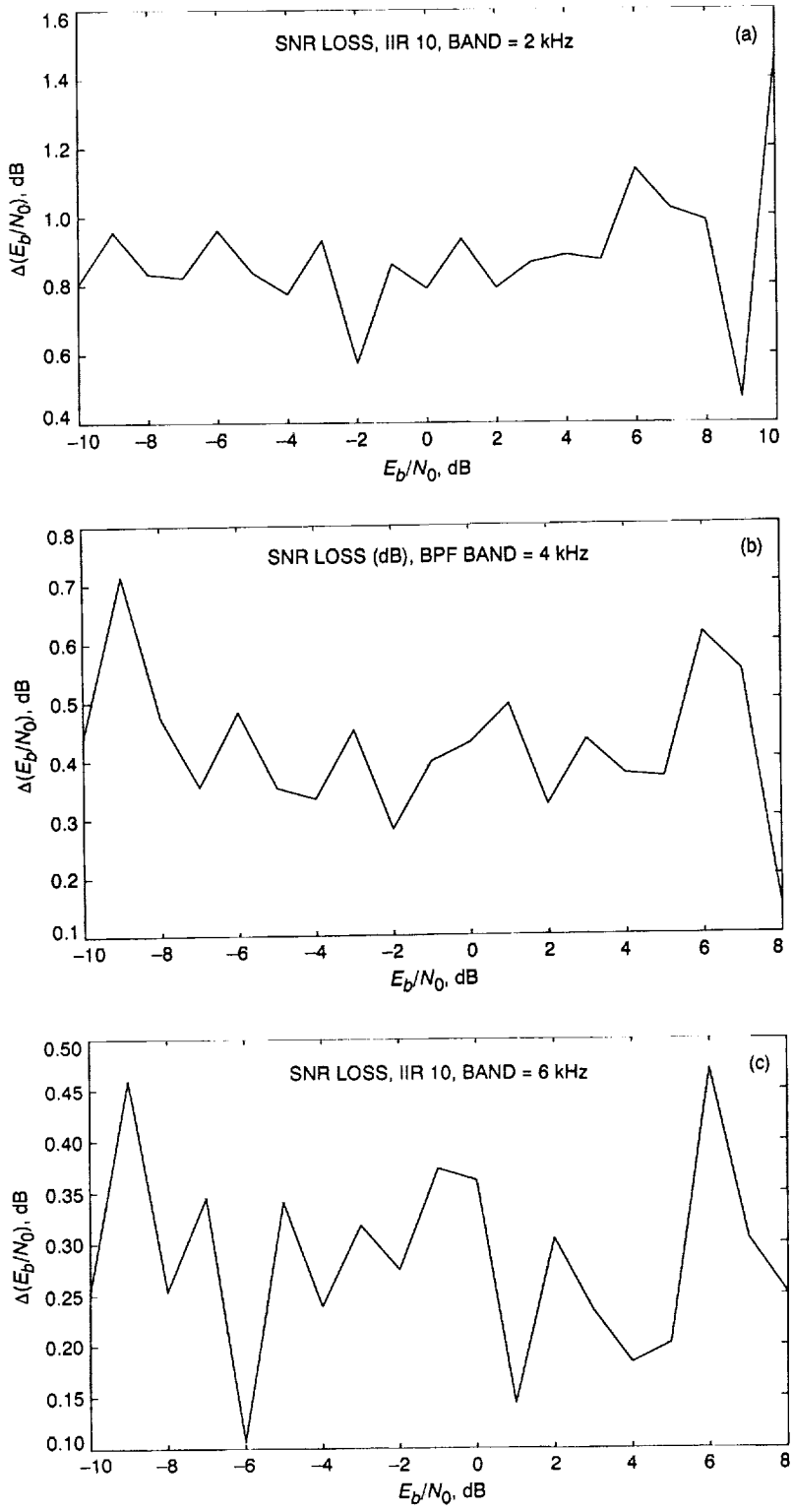


Fig. 4. Δ(SNR) versus symbol SNR for BPF bandwidth: (a) 2 kHz, (b) 4 kHz and (c) 6 kHz.

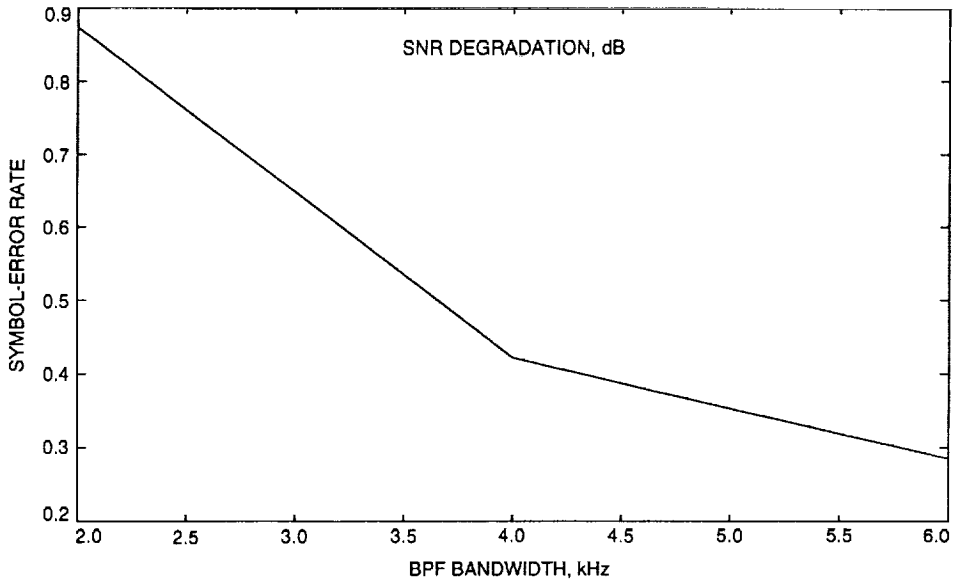


Fig. 5. Δ(SNR) versus BPF bandwidth.

Design of the Reduced LQG Compensator for the DSS-13 Antenna

W. Gawronski
Ground Antennas and Facilities Engineering Section

A linear-quadratic-Gaussian (LQG) compensator design procedure is proposed for the DSS-13 antenna. The procedure is based on two properties. It is shown that tracking and flexible motion of the antenna are almost independent (the separation property). As a consequence, compensators for the flexible and tracking parts can be designed separately. It is shown also that the balanced LQG compensator's effort is evenly divided between the controller and the estimator. This allows a minimization of the compensator order, which is important for implementation purposes. An efficient compensator reduction procedure that gives a stable low-order compensator of satisfactory performance is introduced. This approach is illustrated with a detailed compensator design for the DSS-13 antenna. The implementation of this compensator design requires an update of the antenna model.

I. Introduction

The linear quadratic controller for the DSS-14 antenna was designed by Alvarez and Nickerson [1], and a linear quadratic controller for the DSS-13 antenna was designed by Gawronski [2]. The design method presented in this article extends the results obtained in [2] for the case when full-state feedback is not available.

The development of new high-performance controllers for the DSN antennas is a current priority. The existing proportional and integral (PI) controllers satisfy the requirements for X-band (8.4-GHz) tracking; they remain simple, robust to parameter variations, and do not require detailed knowledge of the antenna dynamics. However, due to the recent pointing requirements for Ka-band (32 GHz), new performance requirements for the antenna controllers have emerged. The PI controllers have reached their performance limits, therefore a new generation of controllers has to be designed and developed. Also, in order to improve controller performance, more sophisticated and accurate antenna models have to be developed. As a rule, the better the knowledge of the plant dynamics, the better the performance that can be achieved by the controller. The recently developed antenna models [3] are accurate enough to give an opportunity to improve tracking performance. The models allow simulation of simultaneous tracking in azimuth and elevation, and include antenna flexible deformations up to 10 Hz.

Among the family of the model-based controllers, the linear-quadratic-Gaussian (LQG) compensator has been chosen for the DSN antennas because it is commonly known to be performing well in industrial applications. The LQG compensator consists of a controller and an estimator. The controller drives the antenna, and the driving control torques are determined from the knowledge of the full antenna dynamics. Since only a small part of the antenna dynamics is measured (by encoders), the estimator is implemented to reconstruct the "missing" dynamics.

The controller and the estimator designs consist of adjusting their gains through proper determination of the controller and estimator weights. For the antenna model of order n, $2n^2$ weights have to be determined (n^2 for the controller and n^2 for the estimator). This number is customarily dropped to $2n$ weights (n for the controller and n for the estimator). But in spite of this drastic reduction, the number of weights is still too large to make the search for the best weights reasonable (typically $n = 40$ for the antenna, i.e., 80 weights have to be found). The difficulty arises because no general procedure for weight determination is available, and the known procedures deal with simplified and/or specific cases.

The weight determination presented here becomes simple due to several properties of the antenna and the compensator investigated in this article. First, it is shown that the tracking and flexible motion of the antenna are almost separated. The tracking part consists of four states (elevation and azimuth encoder readings and their integrals); thus, instead of dealing with a model of order n, one obtains two separate models of order 4 and $n - 4$. Secondly, the LQG compensator is balanced such that the controller and compensator efforts are the same. For the balanced compensator, the weights of the controller and estimator are the same, thus instead of $2n$ weights, n weights need to be determined. Thirdly, it is shown that each component of the balanced compensator (consisting of two states) is almost independent of others. Thus, weights for each component are determined separately. In consequence, the search for $2n$ weights becomes a series of searches for 2 weights, which obviously is not a difficult task to perform.

As mentioned, the PI controller is easy to implement due to its simplicity. But the implementation of the 40-state model-based LQG compensator is not an easy task. It would result in a complex algorithm, and would be a huge computational burden. Therefore, a simplification of the LQG compensator is an important implementation requirement. The simplification is obtained through order reduction of the compensator. The size of the compensator is reduced, but one has to find a reduction procedure such that the reduced-order compensator is stable and its performance is still close to the full-order compensator. This task is solved by introducing the pole mobility index. The pole mobility index characterizes the importance of the components of the balanced compensator. The states with small pole mobility index are truncated, and the truncation marginally affects the closed-loop dynamics. The closed-loop system with the reduced compensator is stable, and its performance is close to the full-order compensator. It is shown that the reduced-order compensator of 12 states is stable and its performance is close to the performance of the full-order compensator of 40 states.

II. Problem Statement

The closed-loop system with an LQG compensator is shown in Fig. 1, with the plant state-space triple (A, B, C), the process noise v of intensity V, and the measurement noise w of intensity W, where both v and w are uncorrelated:

$$V = E(vv^T), \quad W = E(ww^T)$$
$$E(vw^T) = 0, \quad E(v) = 0, \quad E(w) = 0 \tag{1}$$

where $E(.)$ is an expectation operator. It is assumed that $W = I$ without loss of generality. The task is to determine the controller gain (K_p) and estimator gain (K_e) such that the performance index J

$$J = E\left(\int_0^\infty (x^T Q x + u^T R u) dt\right) \tag{2}$$

is minimal, where R is a positive definite input weight matrix, and Q is a positive semidefinite state weight matrix. It is assumed that $R = I$, also without loss of generality. The minimum of J is obtained for the feedback $u = -K_p x$, where the gain matrix

$$K_p = B^T S \tag{3}$$

is obtained from the solution S of the controller Riccati equation (CARE) [4,5]:

$$A^T S + SA - SBB^T S + Q = 0 \tag{4}$$

The optimal estimator gain is given by

$$K_e = PC^T \tag{5}$$

where P is the solution of the estimator Riccati equation (FARE):

$$AP + PA^T - PC^T CP + V = 0 \tag{6}$$

The LQG compensator performance can be significantly modified through variations of weight Q and variance V. Although V is formally predetermined by the process noise v, it can be modified in a search for a more suitable solution [5,6]. The determination of the weight and covariance is addressed in the following section.

Another nontrivial issue addressed here is the order of the compensator. Although the size of the plant determines the size of the compensator, in many cases a full-size compensator is not acceptable for implementation due to its complexity. Thus, its order must be minimized in such a way that the reduced compensator maintains the stability and performance of the full-order compensator. The solution to this problem is found through the approximate balancing of the CARE/FARE equations.

III. Quasi-Separation of Flexible and Tracking Subsystems

The open-loop (or rate-loop) state-space representation (A, B, C) of the DSS-13 antenna includes the input $u_p^T = [u_{pe}\ u_{pa}]$, which consists of the rate commands in elevation (u_{pe}) and azimuth (u_{pa}) and the outputs $y_p^T = [y_{pe}\ y_{pe}]$ and $y_i^T = [y_{ie}\ y_{ia}]$, which consist of the elevation and azimuth angles $(y_{pe},\ y_{pa})$ and their integrals $(y_{ie},\ y_{ia})$. Divide the state vector x of the open-loop antenna model into the tracking x_t and flexible x_f parts

$$x^T = [x_t^T \quad x_f^T] \tag{7}$$

where $x_t^T = [y_i^T \quad y_p^T]$ and x_f are the remaining states. It can be shown [2] that, in this case, the rate-loop representation (A, B, C) has the form

$$A = \begin{bmatrix} A_t & A_{tf} \\ 0 & A_f \end{bmatrix}, \quad B = \begin{bmatrix} B_t \\ B_f \end{bmatrix}, \quad C = [C_t \ \ 0] \tag{8}$$

and that

$$\| B_t \| \ll \| B_f \|, \ \| A_{tf} \| \ll \| A_t \|, \ \| A_{tf} \| \ll \| A_f \| \tag{9}$$

For the DSS-13 antenna, $\| B_t \| < 10^{-5}$, $\| B_f \| > 1$, $\| A_{tf} \| < 10^{-3}$, $\| A_f \| > 10$, and $\| A_t \| = 1$. Thus, the states of the tracking part are much weaker than the states of the flexible part. The strong and weak signal flow is shown in the block diagram of Fig. 2. The strong states of the flexible subsystem and the weak states of the tracking subsystem are shown in Fig. 3, which presents the transfer function plots of the rate-loop systems due to elevation rate command.

In the LQG design, the performance index is minimized and the minimum is obtained for $K = B^T S$, and S is a solution of the Riccati Eq. (4). Divide S and K into parts related to the triple (A, B, C) in Eq. (8)

$$S = \begin{bmatrix} S_t & S_{tf} \\ S_{tf}^T & S_f \end{bmatrix}, \quad K = [K_t \ K_f] \tag{10}$$

so that Eq. (4) can be written as follows

$$A_t^T S_t + S_t A_t - S_t B_t B_t^T S_t + Q_t - \Delta_{tf} = 0 \tag{11a}$$

$$A_t^T S_{tf} + S_{tf} A_f + S_t A_{tf} - K_t^T K_f = 0 \tag{11b}$$

$$A_f^T S_f + S_f A_f + S_f B_f B_f^T S_f + Q_f - \Delta_{ft} = 0 \tag{11c}$$

where

$$\left.\begin{aligned} \Delta_{ft} &= A_{tf}^T S_{tf} + S_{tf}^T A_{tf} - S_{tf}^T B_t K_f \\ &\quad + S_f B_f B_t^T S_{tf} \\ \Delta_{tf} &= S_t B_t B_f^T S_{tf}^T + S_{tf} B_f K_t \end{aligned}\right\} \tag{12a}$$

$$K_f = B_t^T S_{tf} + B_f^T S_f, \quad K_t = B_t^T S_t + B_f^T S_{tf}^T \tag{12b}$$

Taking a closer look at Eqs. (12), notice that there exist weights Q_t and Q_f such that the gain K_f depends on the flexible subsystem only. Namely, for large enough Q_f, such that $\| Q_f \| \gg \| \Delta_{ft} \|$, the solution S_f of Eq. (11c) is independent of the tracking system, and for small Q_t, one obtains $\| B_t^T S_{tf} \| \ll \| B_f^T S_f \|$. In terms of Eq. (12a), the latter inequality means that the gain K_f depends only on the flexible subsystem. However, due to the master-slave relationship between flexible and tracking subsystems, the situation is not quite symmetric: There are no such Q_t and Q_f that the gain K_t depends only on the tracking subsystem. To understand this, note that the term "small" has a different meaning for Q_f and Q_t. Magnitudes of small Q_f and small Q_t are of different order, namely small Q_f is such that $Q_f < 10^{-7}$, and small Q_t is such that $Q_t < 1$. Therefore, by increasing Q_t in order to obtain $\| Q_t \| \gg \| \Delta_{tf} \|$ one obtains $\| B_f^T S_{ft} \|$ and $\| B_t^T S_t \|$ of the same magnitude. According to Eq. (12b), the latter fact means that the gain K_t depends on the flexible subsystem, as well as on the tracking subsystem, and the solution S_t of Eq. (11a) is dependent on the flexible subsystem. This property can be validated by observation of the closed-loop transfer functions for different weights (Fig. 4). The plots show that the variations of Q_f changed the properties of the flexible subsystem only, while the variations of Q_t changed the properties of both subsystems.

The weight Q_t should be large enough to achieve the pointing performance requirements, and the increase of Q_t

causes increasing dependency of the gains on the tracking system. For this reason the above independence becomes a quasi-independence in the final stage of controller design. Nevertheless, the separation in the initial stages of controller design is very strong. The design consists, therefore, of the initial choice of relatively small weights for the tracking subsystem and determination of the controller gains of the flexible subsystem. It is followed by adjustment of weights of the tracking subsystem and a final tuning of the flexible weights.

The quasi-separation principle discussed above in the case of the controller design is also valid in the case of estimator design, since the compensator design consists of the independent designs of a controller and an estimator [4,5]. Additional properties of the LQG compensator that arise in controls of flexible structures are discussed in the following section.

IV. Balanced LQG Compensator for Flexible Structures

In this section the flexible subsystem is considered only (subscript "f" is dropped in this section for simplicity of notation). A flexible structure is defined as a controllable and observable linear system with distinct complex conjugate pairs of poles (N poles, N is even) and with small real parts of the poles. In other words, it is a linear system with vibrational properties. In the Moore balanced coordinates, it consists of $n = N/2$ components [7,8], and each component consists of two states.

An approximately balanced LQG compensator is considered. An approximate equality between two variables is used in the following sense: Two variables x and y are approximately equal ($x \cong y$) if $x = y + \epsilon$, and $\|\epsilon\|/\|y\| \ll 1$.

The block diagram of a closed-loop flexible system with the LQG compensator is shown in Fig. 5. Similar to the balancing of controllability and observability grammians is the balancing of CARE and FARE equations. Namely, there exists a diagonal positive definite $M = diag\,(\mu_i)$, $i = 1, ..., n, \mu_i > 0$, such that

$$S = P = M \tag{13}$$

A state-space representation with the condition Eq. (13) satisfied is called an LQG balanced representation, and μ_i, $i = 1, ..., n$ represents the characteristic values of (A, B, C). Jonckheere and Silverman [9] and Opdenacker and Jonckheere [10] have shown that a balanced solution for CARE and FARE equations exists in the case of $Q = C^T C$ and $V = BB^T$. Gawronski [11] has shown that the balanced LQG representation exists in the case of general Q and V, and has derived the transformation to the balanced LQG representation.

Let (A, B, C) be a state-space triple of an open-loop system. Its controllability and observability grammians W_c and W_o are positive-definite and satisfy the Lyapunov equations

$$\begin{aligned} AW_c + W_c A^T + BB^T &= 0 \\ A^T W_o + W_o A + CC^T &= 0 \end{aligned} \tag{14}$$

The system representation is balanced in the sense of Moore [12] if its controllability and observability grammians are diagonal and equal:

$$W_c = W_o = \Gamma^2, \quad \Gamma = diag(\gamma_1, ..., \gamma_N), \quad i = 1, ..., N \tag{15}$$

where $\gamma_i > 0$ is the ith Hankel singular value of the system. In [11] it is shown that, for flexible structures, the balanced representation (in the Moore sense) produces diagonally dominant solutions of CARE and FARE, and in the case of $Q = V$ produces approximate LQG balanced solutions S and P, such that $S \cong P \cong M$. Assume a diagonal weight matrix Q:

$$Q = diag(q_i I_2), \qquad i = 1, ..., n \tag{16}$$

Then there exists $q_i \leq q_{oi}$, where $q_{oi} > 0$, $i = 1, ..., n$, such that $S \cong diag\,(s_i I_2)$ is the solution of Eq. (4), where

$$s_i = (\beta_{pi} - 1)/2\gamma_i^2, \quad \beta_{pi}^2 = 1 + 2q_i\gamma_i^2/\zeta_i\omega_i \tag{17}$$

The proof is presented in the Appendix. The plots of s_i with respect to q_i and γ_i are shown in Fig. 6. They show an increase of s_i with the increase of weight q_i, and decrease of s_i with increase of γ_i^2 or $\zeta_i\omega_i$.

A similar result is obtained for the FARE equation, namely, for a diagonal V:

$$V = diag(v_i I_2), \quad i = 1, ..., n \tag{18}$$

there exists $v_i \leq v_{oi}$, where $v_{oi} > 0$, $i = 1, ..., n$, such that $P \cong diag(p_i I_2)$ is the solution of Eq. (6), where

$$p_i = (\beta_{ei} - 1)/2\gamma_i^2, \quad \beta_{ei}^2 = 1 + 2v_i\gamma_i^2/\zeta_i\omega \tag{19}$$

If the ith diagonal entry of P and the respective entry of S are equal, say to μ_i, i.e.,

$$p_i = s_i = \mu_i \tag{20}$$

the ith component is LQG balanced; if S and P are equal the system is LQG balanced. If S, P, and M are diagonally dominant, i.e. $v_i + \epsilon_{vi} \cong s_i + \epsilon_{si} \cong \mu_i$, with ϵ_{vi} and ϵ_{si} small ($|\epsilon_{vi}/v_i| \ll 1, |\epsilon_{si}/s_i| \ll 1$), the system is approximately LQG balanced.

It follows from Eqs. (17) and (19) that for $Q = diag(q_i) = V = diag(v_i)$, the system is approximately LQG balanced, such that

$$\begin{aligned} &S \cong P \cong M = diag(\mu_i) \\ &\mu_i = (\beta_i - 1)/2\gamma_i^2, \quad \beta_i^2 = 1 + 2q_i\gamma_i^2/\zeta_i\omega \end{aligned} \tag{21}$$

Next it is shown that the weight Q

$$Q = diag(0, 0, ..., q_i I_2, ...0, 0), \quad q_i \le q_{oi} \tag{22}$$

shifts the ith pair of complex poles of flexible structure, and leaves the remaining pairs of poles almost unchanged. Only the real part of the pair of poles is changed (just moving the pole apart from the imaginary axis and stabilizing the system), and the imaginary part of the poles remains unchanged. Namely, for the weight Q as in Eq. (22), the closed-loop pair of flexible poles (λ_{cri}, $\pm j\lambda_{cii}$) relates to the open-loop poles (λ_{ori}, $\pm j\lambda_{oii}$) as follows:

$$(\lambda_{cri}, \pm j\lambda_{cii}) \cong (\beta_{pi}\lambda_{ori}, \pm j\lambda_{oii}), \ i = 1, ..., n \tag{23}$$

where β_{pi} is defined in Eq. (17). The proof is presented in the Appendix.

The real part of the poles is shifted by β_{pi}, while the imaginary part remains unchanged. The above proposition has additional interpretations. Denote the real part of the open-loop pole by $\lambda_{ori} = -\zeta_i\omega_i$ and the real part of the closed-loop pole by $\lambda_{cri} = -\zeta_{ci}\omega_i$; note also that the height of the open-loop resonant peak is $\alpha_{oi} = \kappa/2\zeta_i\omega_i$, where κ is a constant, and the closed-loop resonant peak is $\alpha_{ci} = \kappa/2\zeta_{ci}\omega_i$. From Eq. (23), $\beta_{pi} = \lambda_{cri}/\lambda_{ori}$; thus, one obtains

$$\beta_{pi} = \zeta_{ci}/\zeta_i = \alpha_{oi}/\alpha_{ci} \tag{24}$$

i.e., β_{pi} is a ratio of closed- and open-loop damping factors, or it is a ratio of open- and closed-loop resonant peaks. Therefore, if a suppression of the ith resonant peak by the factor β_{pi} is required, the appropriate weight q_i is determined from Eq. (17):

$$q_i = 0.5(\beta_{pi}^2 - 1)\zeta_i\omega_i\gamma_i^{-2} \tag{25}$$

An alternative interpretation of β_i is as a ratio of the open-loop Hankel singular value to the closed-loop Hankel singular value:

$$\beta_i = \gamma_{oi}^2/\gamma_{ci}^2 \tag{26a}$$

or a ratio of variances of open-loop (σ_{oi}^2) and closed-loop (σ_{ci}^2) states excited by the white-noise input [13]:

$$\beta_i = \sigma_{oi}^2/\sigma_{ci}^2 \tag{26b}$$

The proof is presented in the Appendix.

The plots of β_{pi} with respect to q_i and γ_i are shown in Fig. 7. They show relatively large β_{pi} even for small q_i, i.e., a significant pole shift to the left. Also, β_{pi} increases with the increase of γ_i, and decreases with the increase of $\zeta_i\omega_i$, i.e., there is a significant pole shift for highly observable and controllable states with small damping. In terms of the transfer function profile, the weight q_i suppresses the resonant peak at frequency ω_i while leaving the natural frequency unchanged. Due to weak coupling between the states, the assignment of one pair of states does not significantly impact other states. Thus, the weight assignment can be done separately for each pair of states.

The estimator poles are shifted in a similar manner. Denote

$$V = diag(0, 0, ..., v_i I_2, ...0, 0), \quad v_i \le v_{oi} \tag{27}$$

then for the weight V as in Eq. (27) and $v_i \le v_{oi}$, the estimator pair of poles (λ_{eri}, $\pm j\lambda_{eii}$) relates to the open-loop poles (λ_{ori}, $\pm j\lambda_{oii}$) as follows:

$$(\lambda_{eri}, \pm j\lambda_{eii}) \cong (\beta_{ei}\lambda_{ori}, \pm j\lambda_{oii}), \quad i = 1, ..., n \tag{28}$$

where β_{ei} is defined in Eq. (19).

The limiting values q_{oi} and v_{oi} are determined. Their values are rather fuzzy numbers. Despite their fuzziness, they are not difficult to determine. There are several indicators that q_i is approaching q_{oi}, or that v_i is approaching v_{oi}. For the controller, q_{oi} is the weight for which the ith pair of complex poles of the plant departs from the horizontal trajectory in the root-locus plane, or it is the weight for which the ith resonant peak of the plant transfer function disappears (the peak is flattened). For the estimator, v_{oi} is the covariance for which the ith pair of complex poles of the estimator departs from the horizontal trajectory in the root-locus plane, or it is a covariance for which the ith resonant peak of the estimator transfer function disappears.

It is crucial from an implementation point of view to obtain a compensator of the smallest possible dimension that preserves the stability and performance of the full-order compensator. Although the size of a plant determines the size of a compensator, the plant model cannot be reduced excessively in advance in order to assure the quality of the closed-loop system. Therefore, compensator reduction is a part of the compensator design. The balanced LQG design procedure provides this opportunity.

To successfully perform the compensator reduction, an index of the importance of each compensator component is introduced. In the open-loop case, Hankel singular values serve as reduction indices. In the closed-loop case, the characteristic values of the system seem to be good candidates for the reduction indices, as suggested in [9]. This is not a good choice, however, since the characteristic values do not properly reflect the effectiveness of the compensator. The effectiveness of the closed-loop system can be evaluated by the relative suppression of the closed-loop output when compared to the open-loop output. Thus, the ratio of variances of the closed-loop output and the open-loop output excited by the white noise input is an appropriate measure of the suppression (alias the compensator performance). It will be shown later that the suppression depends on pole mobility in the complex plane. Therefore, if a particular pair of poles is easily moved (i.e., in the case when small weight is required), the respective states are easy to control and to estimate. On the contrary, if a particular pair of poles is difficult to move (i.e., a large weight is required to move the poles), the respective states are difficult to control and to estimate. In the latter case, the action of the compensator is irrelevant, and the states that are difficult to control and estimate can be reduced; thus, pole mobility is a good indicator of the importance of a particular compensator state.

Consider an LQG balanced system, and denote the pole mobility index π_i by

$$\pi_i = 0.5(\beta_i - 1) \tag{29}$$

Note that for $\beta_i = 1$, the ith pole is stationary and π_i is equal to zero; for a shifted pole, one obtains $\beta_i > 1$ and $\pi_i > 0$. If for small π_i a small pole shift (in plant, as well as in estimator) is observed, this component can be reduced.

Another useful interpretation follows from Eqs. (29) and (26):

$$\pi_i = 0.5(\gamma_{oi}^2 - \gamma_{ci}^2)/\gamma_{ci}^2 = 0.5(\sigma_{oi}^2 - \sigma_{ci}^2)/\sigma_{ci}^2 \tag{30}$$

i.e., the pole mobility index is proportional to the relative change in the white noise response of the open- and closed-loop systems. Furthermore, from Eqs. (21) and (29), it can be shown that the pole mobility index is a product of the square of a Hankel singular value and the characteristic value of the system:

$$\pi_i = \gamma_i^2 \mu_i \tag{31}$$

Thus, π_i combines the system observability and controllability properties of the open-loop system with the closed-loop performance. The more heavily weighted the component, the larger its pole mobility index, see Fig. 8(a). Also, the larger the Hankel singular value of the component, the larger the corresponding pole mobility index, c.f. Fig. 8(b).

The matrix Π of pole mobility indices is defined as

$$\Pi = diag(\pi_1, \pi_2, ..., \pi_{n-1}, \pi_n) \tag{32}$$

and it is obtained from Eqs. (15) and (29) as

$$\Pi = \Gamma^2 M \tag{33}$$

In the following, a reduction technique is discussed. Assume Π in Eq. (32) has a descending order, i.e., $\pi_i \geq 0$, $\pi_{i+1} \leq \pi_i$, $i = 1, ..., n$, and divide Π as follows:

$$\Pi = diag(\Pi_r, \Pi_t) \tag{34}$$

where Π_r consists of the first k entries of Π, and Π_t the remaining ones. If the entries of Π_t are small in comparison with the entries of Π_r, the compensator is reduced by truncating its last $n - k$ states.

It is shown in [11] that a system with the reduced-order compensator obtained by reducing states with small Π_i *is expected* to be stable. That is, although it is not guaranteed, there is a well-founded expectation to obtain a stable system with the reduced-order compensator. Also, the estimation errors of the full-order and reduced-order compensators are approximately the same.

V. Closed-Loop System

The LQG compensator configuration for the DSN antenna is shown in Fig. 1. The tracking command y_c is compared with the estimated antenna position $\hat{y}_p$, and the error $e = \hat{y}_p - y_c$ along with the integral e_i of the error e are the plant inputs. The equations for the integrator, plant, and the estimator are, respectively,

$$\dot{e}_i = e \tag{35a}$$

$$\dot{x} = Ax + Bu, \quad y_p = C_p x, \quad x_f = C_f x \tag{35b}$$

$$\dot{\hat{x}} = A\hat{x} + Bu + K_e(y_p - C_p\hat{x}) \tag{35c}$$

$$u = -K_f\hat{x}_f - K_p e - K_i e_i$$

Denoting $x_{cl}^T = [e_i^T \quad x^T \quad \hat{x}^T]$, one obtains

$$\dot{x}_{cl} = A_{cl}x_{cl} + B_{cl}y_c, \quad y = C_{cl}x_{cl} \tag{36a}$$

where

$$A_{cl} = \begin{bmatrix} 0 & 0 & C_p \\ -BK_i & A & -BK_fC_f - BK_pC_p \\ -BK_i & K_eC_p & A - BK_fC_f - BK_pC_p - K_eC_p \end{bmatrix}, \quad B_{cl} = \begin{bmatrix} -I \\ BK_p \\ BK_p \end{bmatrix}, \quad C_{cl} = [0 \quad C_p \quad 0] \tag{36b}$$

The triple $(A_{cl},\ B_{cl},\ C_{cl})$ for the LQG system with the reduced-order estimator $(A_r,\ B_r,\ C_r)$ is as follows:

$$A_{cl} = \begin{bmatrix} 0 & 0 & C_p \\ -BK_i & A & -BK_fC_{fr} - BK_pC_{pr} \\ -B_rK_i & K_{er}C_p & A_r - B_rK_fC_{fr} - B_rK_pC_{pr} - K_{er}C_{pr} \end{bmatrix}, \quad B_{cl} = \begin{bmatrix} -I \\ BK_p \\ B_rK_p \end{bmatrix}, \quad C_{cl} = [0 \quad C_p \quad 0] \tag{37}$$

where C_{pr} and C_{fr} are obtained from the partition of C_r, $C_r = [C_{pr} \quad C_{fr}]$.

VI. Compensator Design Algorithm

Weights for the balanced LQG controller and estimator are identical. Therefore, in the algorithm, the controller and estimator gains are adjusted simultaneously. The procedure for the antenna LQG compensator design is a sequential one. First, for the ad hoc (but relatively small) chosen tracking subsystem weights, the flexible subsystem weights are determined (recall that the controller and estimator weights are the same). Second, the adjustment of the tracking system weights is performed, followed by final adjustment of the flexible system weights. The flexible subsystem weights are determined sequentially, which gives more insight into the system performance and simplifies the procedure.

The estimator order is determined as a part of the weight tuning process. Only the modes with large index π_i are considered. If the number of flexible modes is n_f, the number of disregarded modes is n_o, and the size of the tracking system is n_t, then the controller order n_c is $n_c = n_t + 2(n_f - n_o)$. The following LQG compensator design algorithm is proposed:

(1) Determine the plant state-space representation in the form of Eq. (8), consisting of flexible and tracking parts.

(2) Choose ad hoc but reasonably small weights and variances for the tracking part $Q_t = V_t = Q_{\text{tah}}$.

(3) For each balanced coordinate of the flexible part, choose the weight q_i, $i = 1, ..., n_f$, and define the weight and covariance matrices $Q_{fi} = diag\ (0, 0, ..., q_i,\ q_i, 0, 0, ...0)$, $V_{fi} = Q_{fi}$ so that the closed-loop system performance for the weight $Q_i = diag(Q_{\text{tah}}, Q_{fi})$ and the covariance $V_i = Q_i$ is optimized. For example, determine the weights q_i to impose the required pole shift or to suppress the ith resonant peak to the required level without depreciating other properties of the closed-loop transfer function. Note that for small q_i, only the ith pair of poles is shifted (to the left), and the remaining poles are almost unaffected. Disregard the modes with small index π, for which the weighting does not improve the closed-loop system performance.

(4) For the already determined Q_f and V_f, tune weight Q_t and assume covariance $V_t = Q_t$ to obtain improvements in tracking properties of the antenna.

(5) Adjust flexible subsystem weights, if necessary.

VII. Applications

A balanced LQG compensator was designed for the DSS-13 antenna. The DSS-13 antenna model consists of two tracking states (azimuth and elevation angle) and 13 flexible modes (or 26 balanced states). The preliminary weights $q_{ie} = q_{pe} = q_{ia} = q_{pa} = 1$ for the tracking subsystem (for y_i and y_p) and zero weights for the flexible subsystem ($q_1 = q_2 = ... = q_{13} = 0$ for all 13 modes) were chosen. The closed-loop system step response is presented in Fig. 9 (azimuth encoder reading due to azimuth command) and the magnitudes of the closed-loop transfer function in Fig. 10. Both figures show that flexible motion of the antenna is excessive and should be damped out. This is achieved by adjusting weights for the flexible subsystem. For the tracking weights as before, the weight for the first mode (2.32 Hz) is chosen to be $q_1 = 10^{-7}$, and the remaining weights are zero, obtaining the closed-loop system responses as shown in Figs. 11 and 12. One can see that the 2.32-Hz resonance peak in the azimuth command response (Fig. 11) has disappeared, along with most of the flexible motion in the azimuth step response (Fig. 12). The elevation motion is unaffected, however, since the azimuth mode is almost nonexistent in the elevation motion.

The weight should be chosen carefully. Weight that is too small (e.g., 3×10^{-9} in the case considered) will not suppress the resonant peak, Fig. 13(a). Weight that is too large (e.g., 10^{-5}) will deteriorate the tracking performance: for the overweighted mode the transfer function is pressed down within a wide frequency range, Fig. 13(b). The proper weight suppresses the resonant peak, leaving the other peaks unchanged.

Similar procedures have been applied for the second (2.64-Hz), third (4.26-Hz), fourth (3.77-Hz), fifth (7.92-Hz), sixth (4.47-Hz), seventh (3.38- Hz), eighth (5.98-Hz), ninth (7.32-Hz), and tenth (9.48-Hz) modes, with weight 10^{-7} for each mode. As a result, the suppression of the remaining flexible motion is observed as shown in Fig. 14. Weights for the remaining modes (eleventh through thirteenth) have been set to zero.

The root locus of the closed-loop system due to weight variations of the 7.92-Hz (fifth) mode is shown in Fig. 15. The figure shows the horizontal departure of poles to the left (stabilizing property). It confirms the properties of the weighted LQG design described previously.

In the next step, the tracking properties of the system are improved by proper weight setting of the tracking subsystem. Namely, setting the integral weight to $q_{ie} = q_{ia} = 70$ and the proportional weight to $q_{pe} = q_{pa} = 100$ improves the system tracking properties, as shown in Fig. 16 (small overshoot and settling time) and in Fig. 17 (extended bandwidth up to 2 Hz). However, by improving the tracking properties, the transfer function has been raised dramatically in the frequency region of 1 to 3 Hz, which forces the first two modes located in this region to appear again in the step response. By sacrificing a bit of the tracking properties, the flexible motion in the step response is removed. This is done by slightly increasing the weights of the flexible subsystem, setting them as follows: $q_1 = q_2 = q_3 = q_4 = q_5 = q_6 = 10^{-6}$, $q_7 = q_8 = 10^{-7}$, and $q_9 = q_{10} = 10^{-5}$. The closed-loop system response with satisfactory tracking performance is shown in Figs. 18 and 19 (small overshoot, settling time, and 1-Hz bandwidth are observed).

The reduced-order compensator is obtained through evaluation of pole mobility indices π_i. The plot of π_i is shown in Fig. 20. Reducing the order of the estimator to 12 states [the first four are tracking states (states 1 to 4), the next six are flexible states (states 5 to 10), and the last two states are nonflexible components of the rate loop model (states 25 and 27)] gives the stable and accurate closed-loop system. The reduced-order compensator is compared with the full-order compensator in the step response plots in Fig. 21, and the transfer function plots in Fig. 22, showing satisfactory approximation.

VIII. Conclusions

A method for designing a reduced-order compensator for the DSS-13 antenna has been presented. A balanced LQG compensator has been introduced that uses the same amount of effort to control and to estimate the system.

The properties of the balanced LQG system are used to obtain a reduced-order compensator for the antenna. This compensator preserves the stability and performance of the full-order compensator. The performance of the reduced-order compensator has been verified by simulations.

References

[1] L. S. Alvarez and J. Nickerson, "Application of Optimal Control Theory to the Design of the NASA/JPL 70-Meter Antenna Axis Servos," *TDA Progress Report 42-97*, vol. January–March 1989, Jet Propulsion Laboratory, Pasadena, California, pp. 112–126, May 15, 1989.

[2] W. Gawronski, "Sequential Design of a Linear Quadratic Controller for the Deep Space Network Antennas," *Proceedings of the 1992 AIAA Guidance, Navigation, and Control Conference*, Hilton Head, South Carolina, pp. 1399–1408, 1992.

[3] W. Gawronski and J. A. Mellstrom, "Modeling and Simulations of the DSS 13 Antenna Control System," *TDA Progress Report 42-106*, vol. April–June 1991, Jet Propulsion Laboratory, Pasadena, California, pp. 205–248, August 5, 1991.

[4] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems*, New York: Wiley-Interscience, 1972.

[5] B. D. O. Anderson and J. B. Moore, *Optimal Control*, Englewood Cliffs, New Jersey: Prentice Hall, 1990.

[6] J. M. Maciejowski, *Multivariable Feedback Design*, Wokingham, England: Addison-Wesley, 1989.

[7] W. Gawronski and J.-N. Juang, "Model Reduction for Flexible Structures," *Control and Dynamics Systems*, edited by C. T. Leondes, vol. 36, New York: Academic Press, pp. 143–222, 1990.

[8] W. Gawronski and T. Williams, "Model Reduction for Flexible Space Structures," *J. of Guid. Control Dyn.*, vol. 14, no. 1, pp. 68–76, January 1991.

[9] E. A. Jonckheere and L. M. Silverman, "A New Set of Invariants for Linear Systems—Application to Reduced Order Compensator Design," *IEEE Trans. Autom. Control*, vol. AC-28, no. 10, pp. 953–964, 1983.

[10] P. Opdenacker and E. A. Jonckheere, "LQG Balancing and Reduced LQG Compensation of Symmetric Passive Systems," *Int. J. Control*, vol. 41, no. 1, pp. 73–109, 1985.

[11] W. Gawronski, "Balanced LQG Compensator for Flexible Structures," *Proceedings of the 1993 IEEE American Control Conference*, San Francisco, June 1993.

[12] B. C. Moore, "Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction," *IEEE Trans. Autom. Control*, vol. 26, pp. 17–32, 1981.

[13] W. Gawronski and H. G. Natke, "Balancing Linear Systems," *Int. J. Sys. Sci.*, vol. 17, pp. 237–249, 1987.

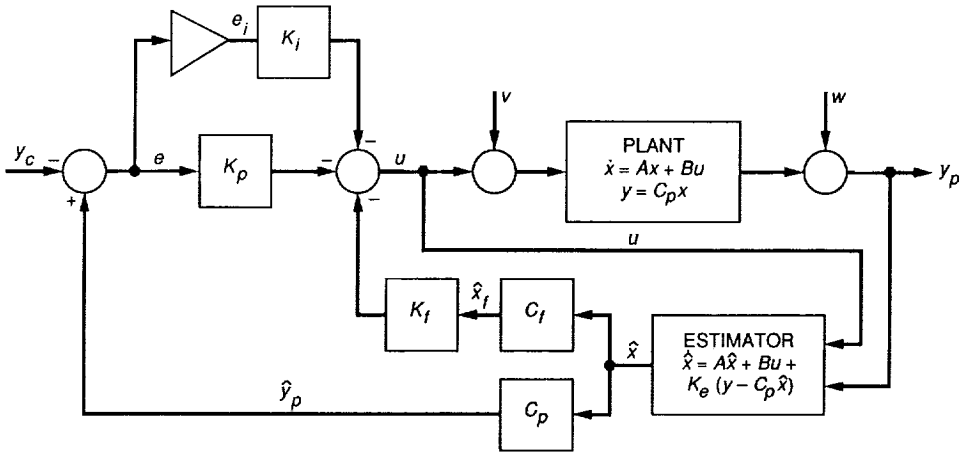


Fig. 1. LQG compensator configuration for DSS-13 antenna.

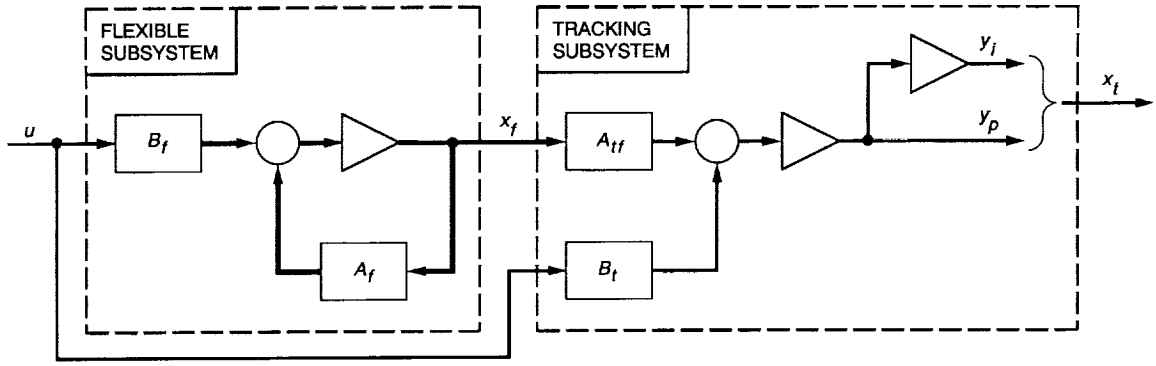


Fig. 2. System configuration.

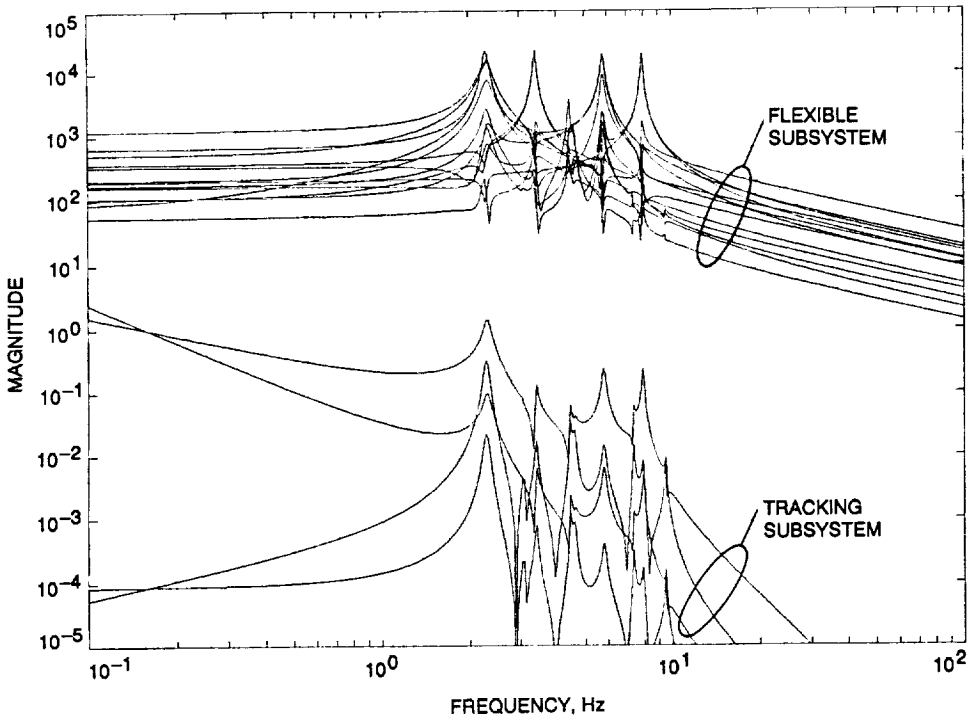


Fig. 3. Magnitudes of transfer function of tracking and flexible subsystems for the elevation rate input.

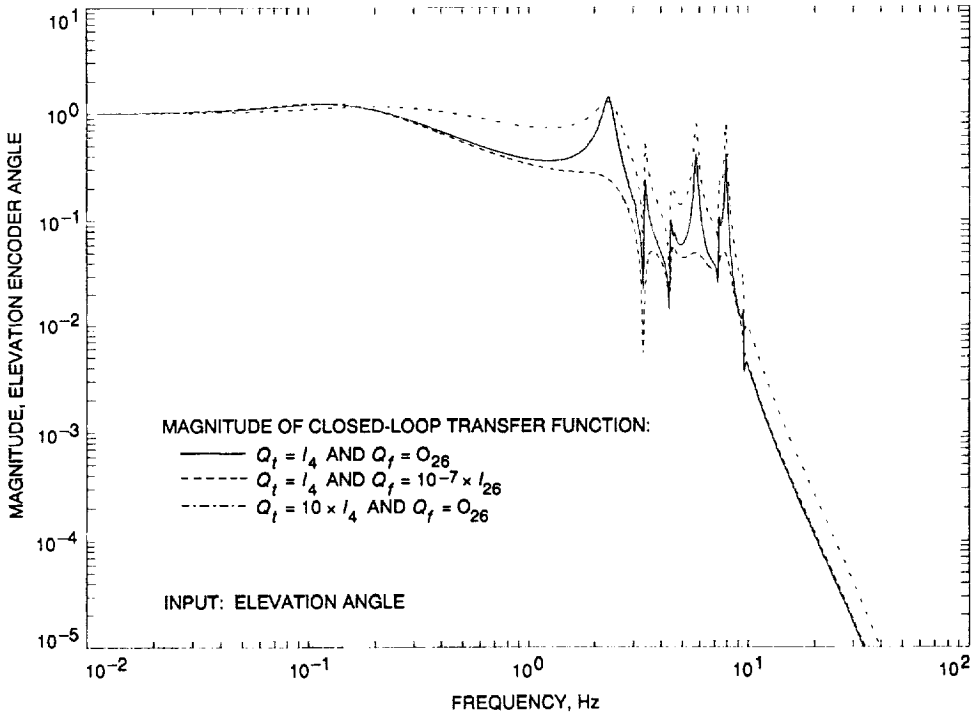


Fig. 4. Magnitudes of transfer function of the closed loop system for different weights of elevation angle to elevation command.

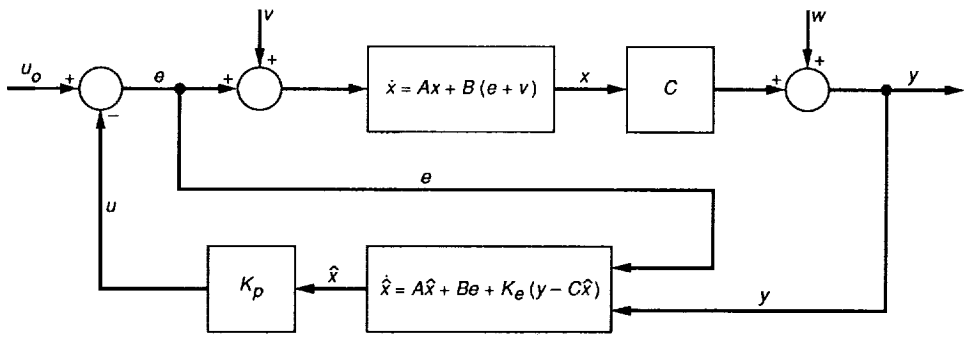


Fig. 5. A flexible structure with an LQG compensator.

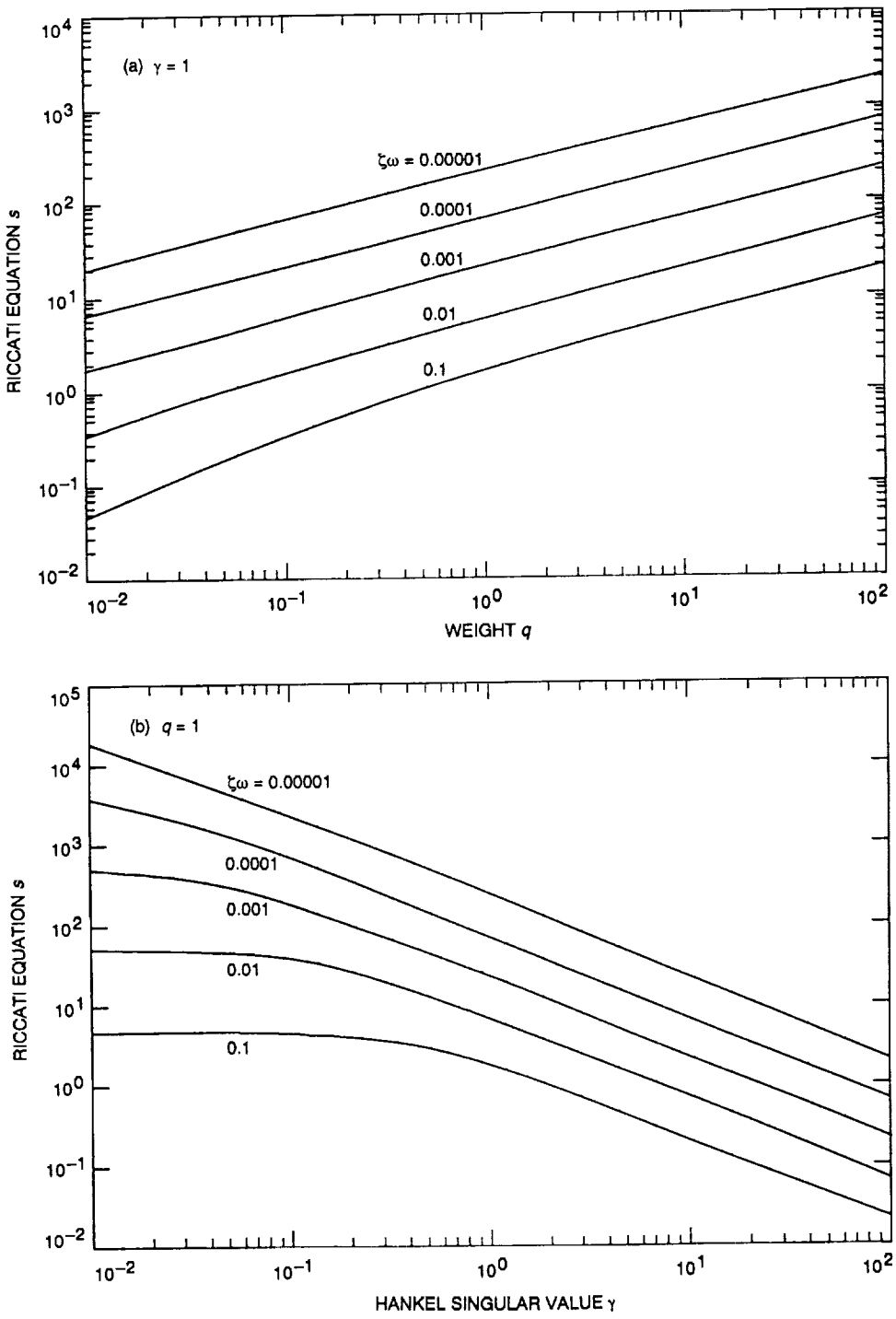


Fig. 6. Solution of the Riccati equation s versus: (a) weight q and (b) Hankel singular value γ.

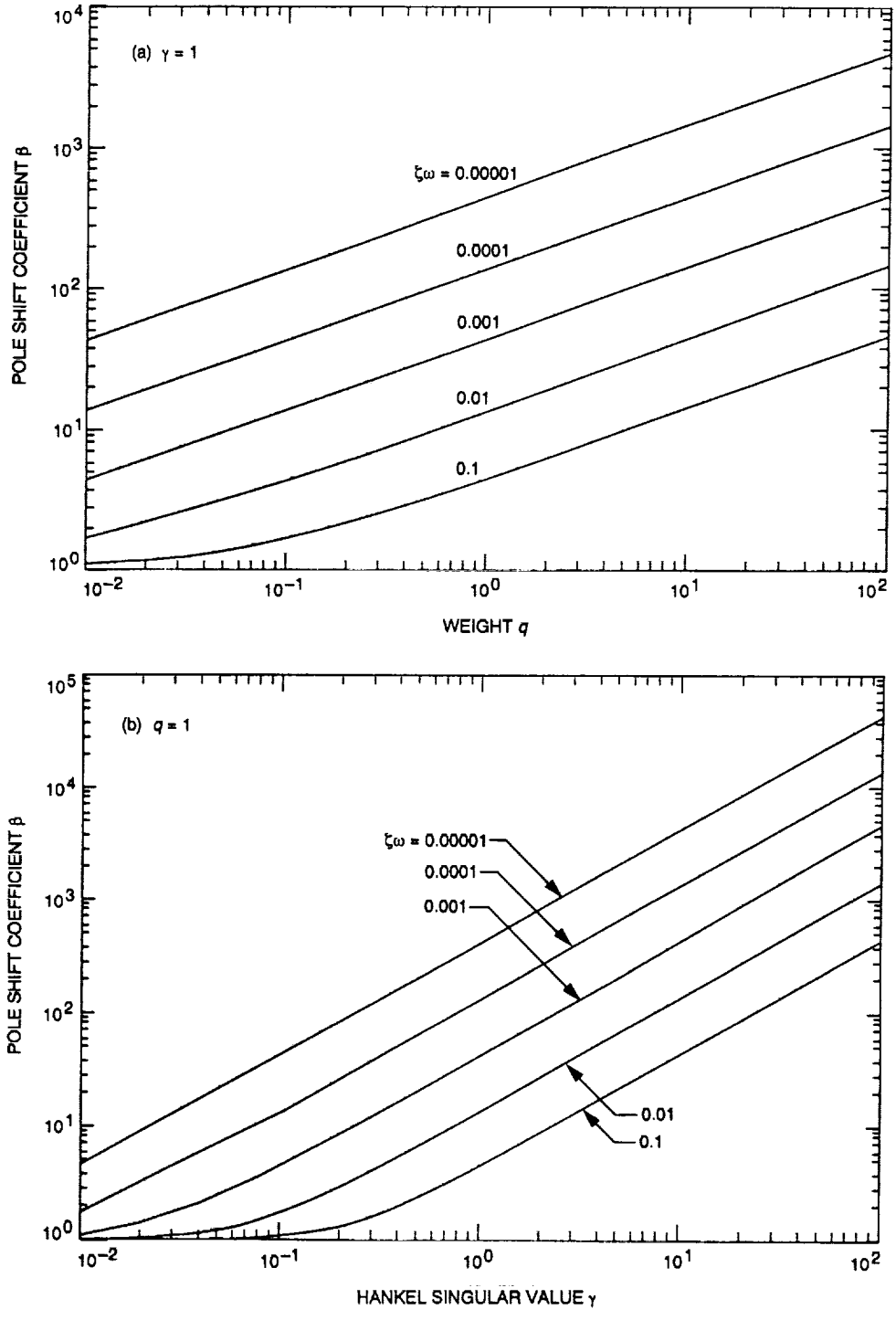


Fig. 7. Pole shift factor β versus: (a) weight q and (b) Hankel singular value γ.

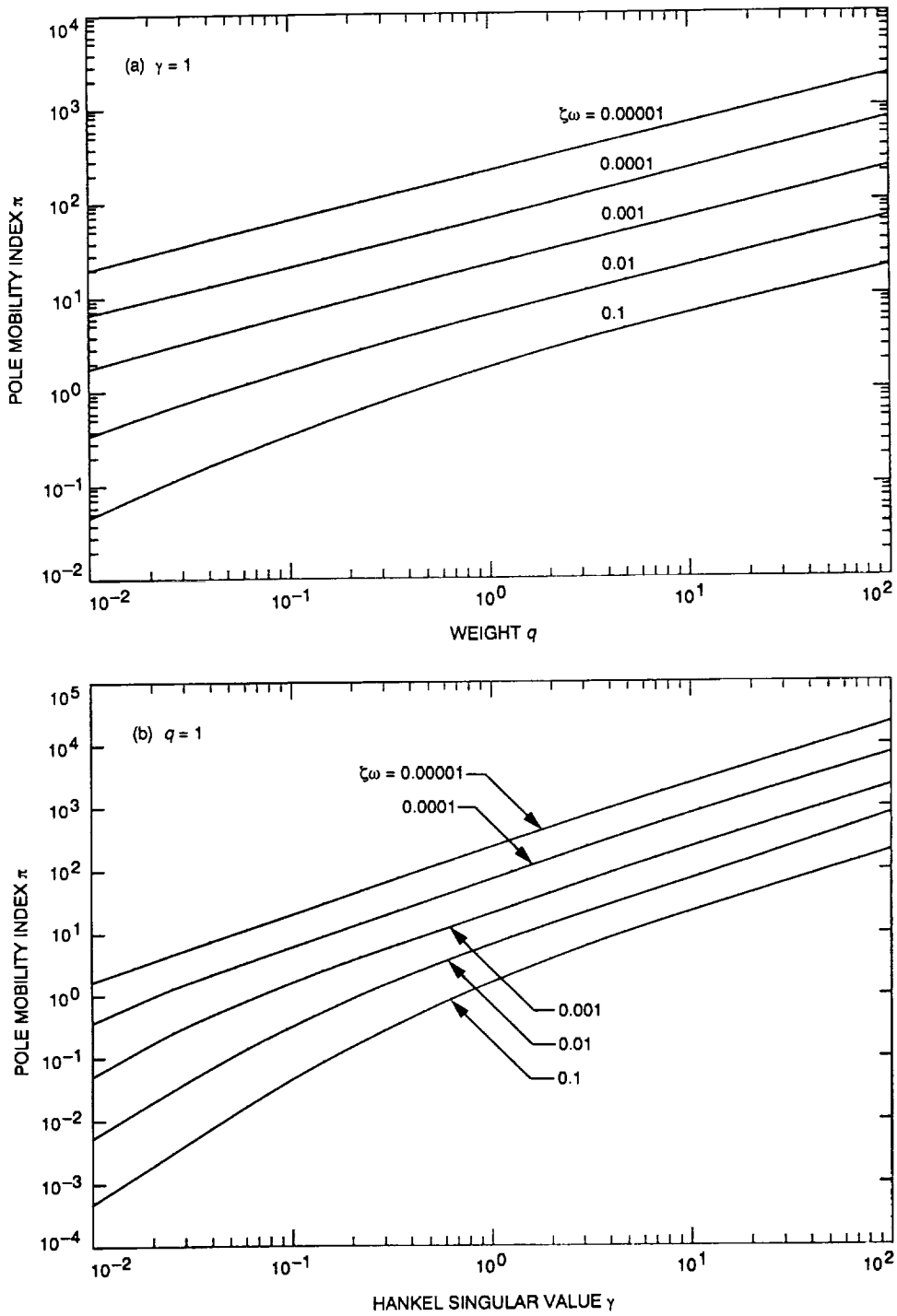


Fig. 8. Pole mobility index π versus: (a) weight q and (b) Hankel singular value γ.

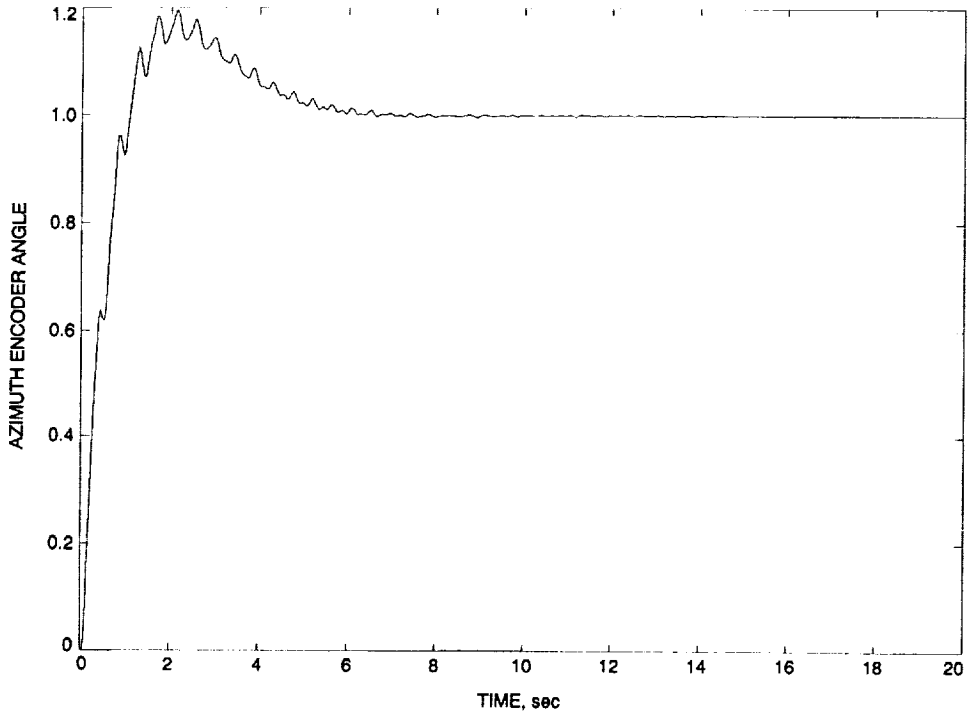


Fig. 9. Azimuth encoder response to azimuth step command for the unit proportional and integral weights in azimuth and elevation, and zero weight for flexible subsystem.

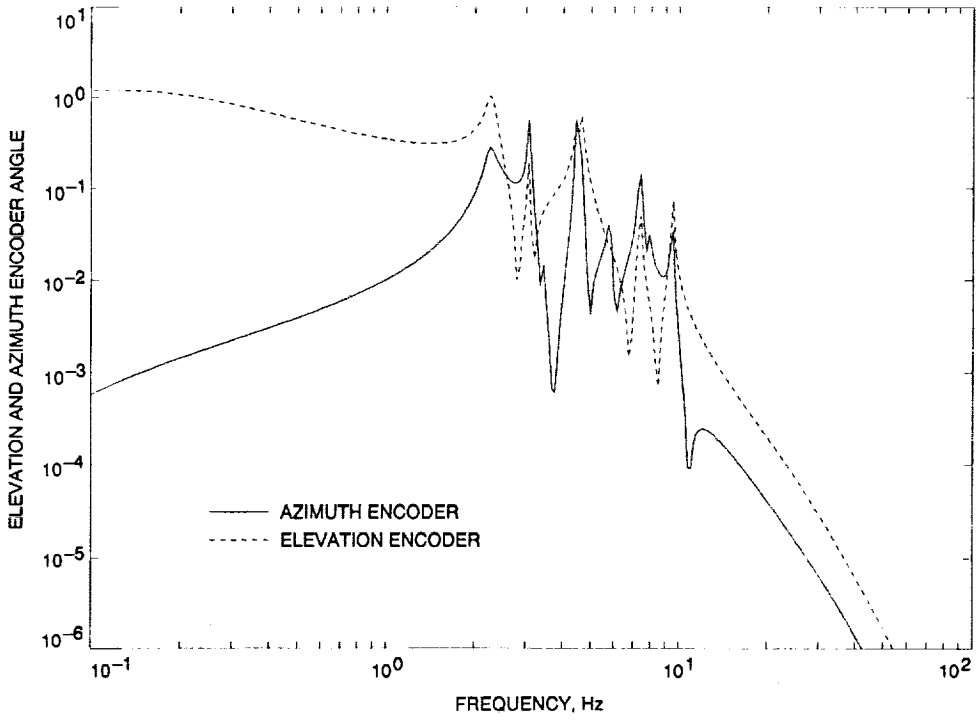


Fig. 10. Transfer function (elevation and azimuth encoder due to azimuth command) for weights the same as those in Fig. 9.

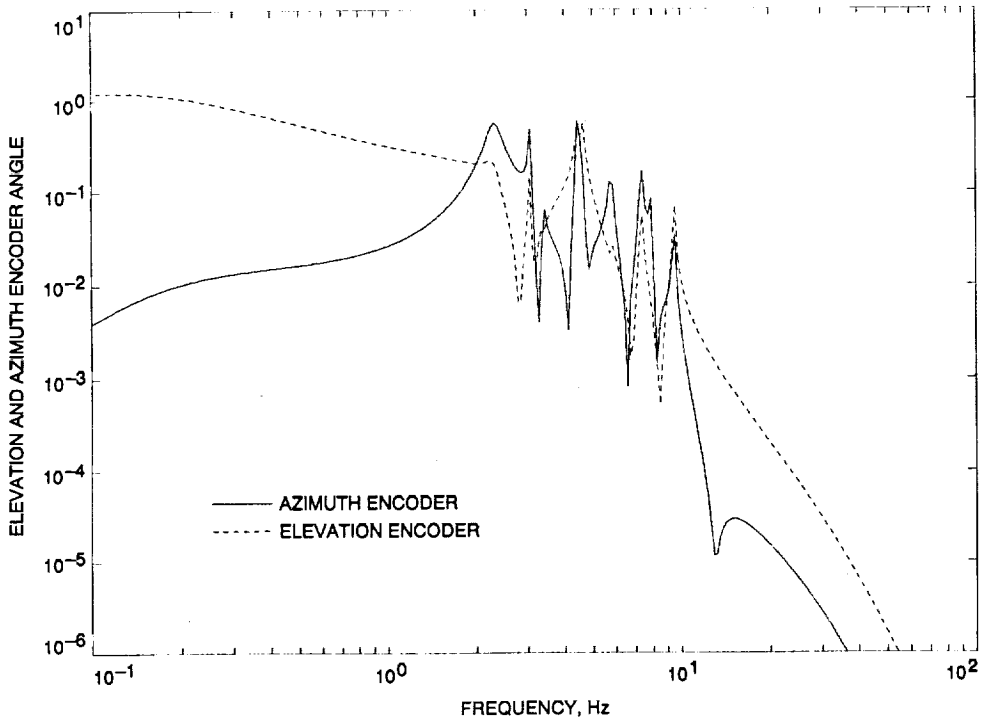


Fig. 11. Transfer function (elevation and azimuth encoder to azimuth command) for weights the same as those in Fig. 9, but for $q_{f1} = 10^{-7}$.

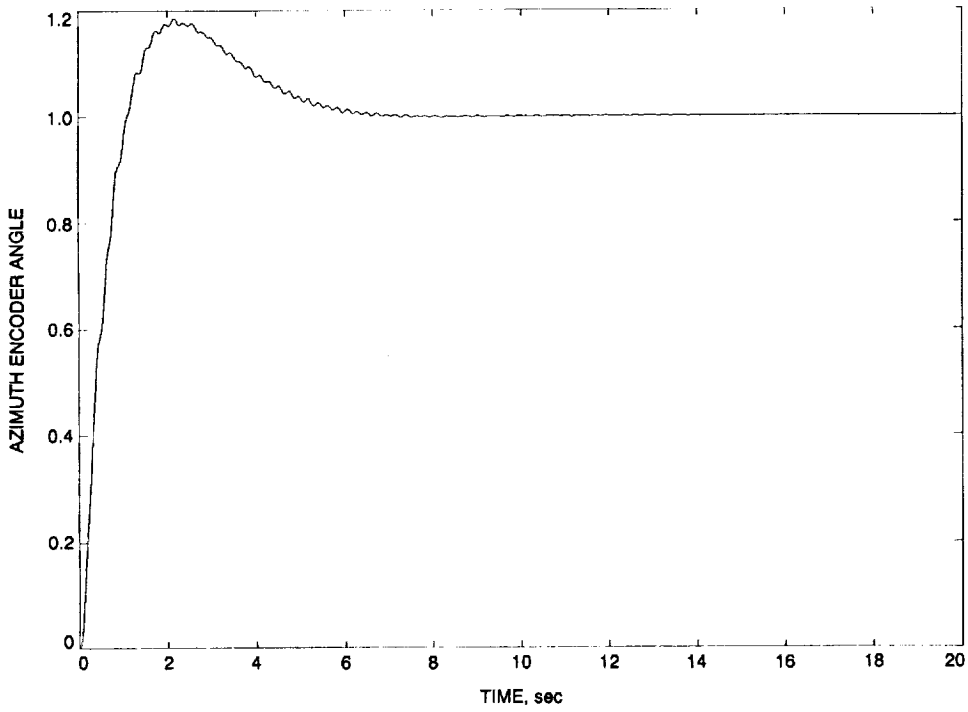


Fig. 12. Azimuth encoder response to azimuth step command for weights the same as those in Fig. 11.

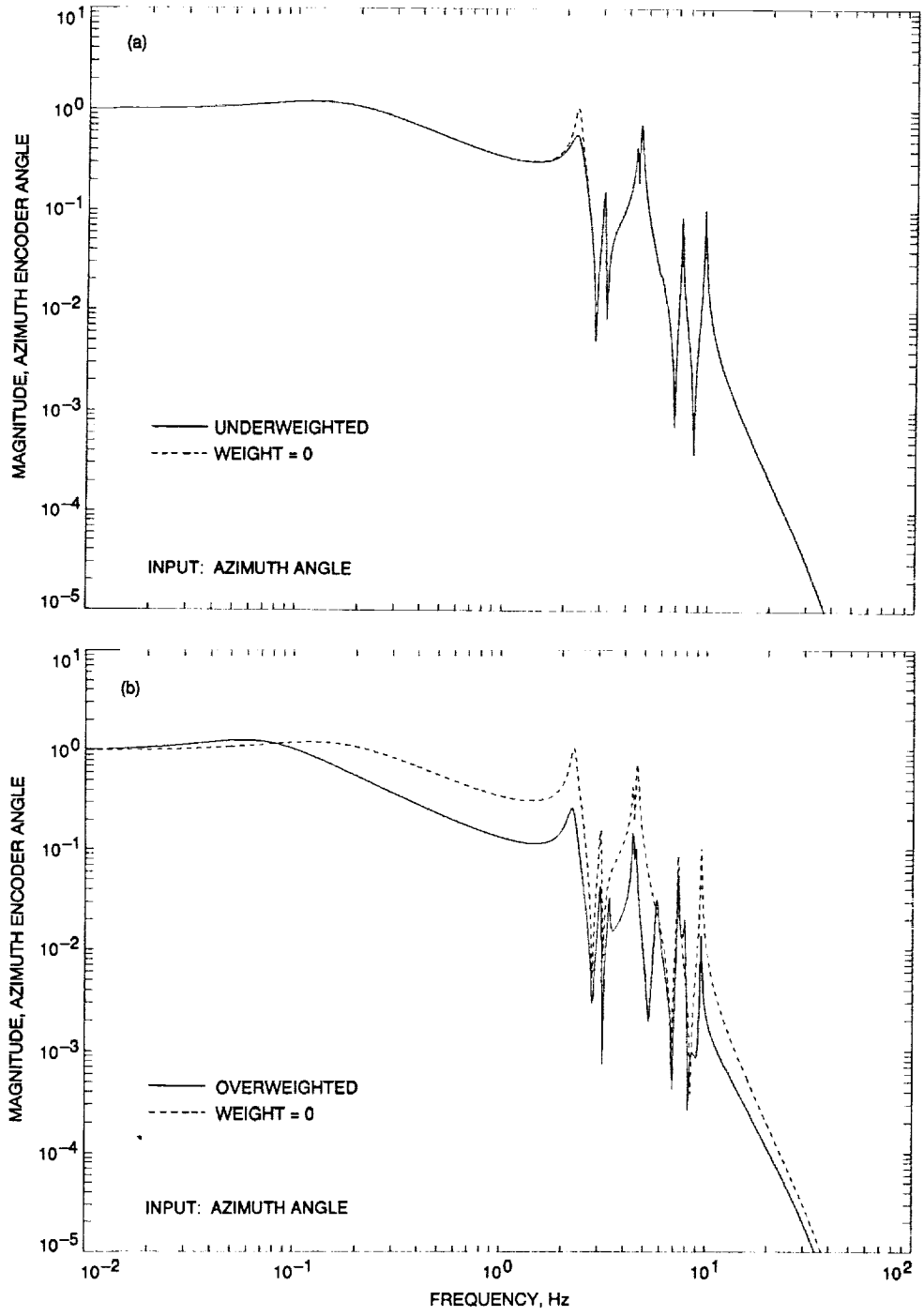


Fig. 13. The first component: (a) underweighted and (b) overweighted.

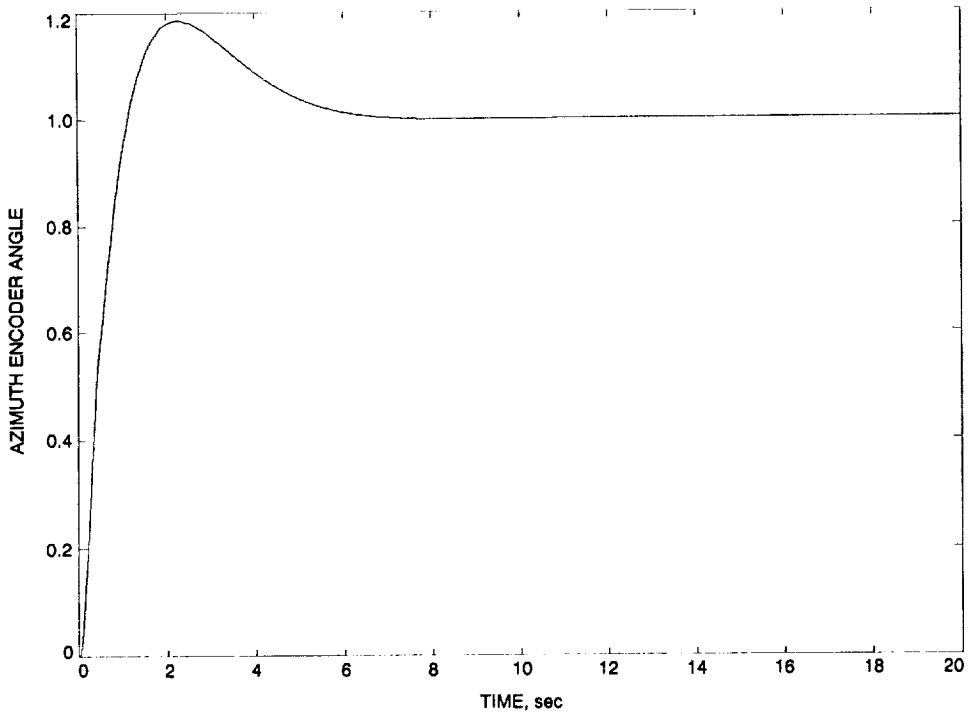


Fig. 14. Azimuth encoder response to azimuth step command for weights the same as those in Fig. 9, but for $q_{f1} = \ldots = q_{f10} = 10^{-7}$.

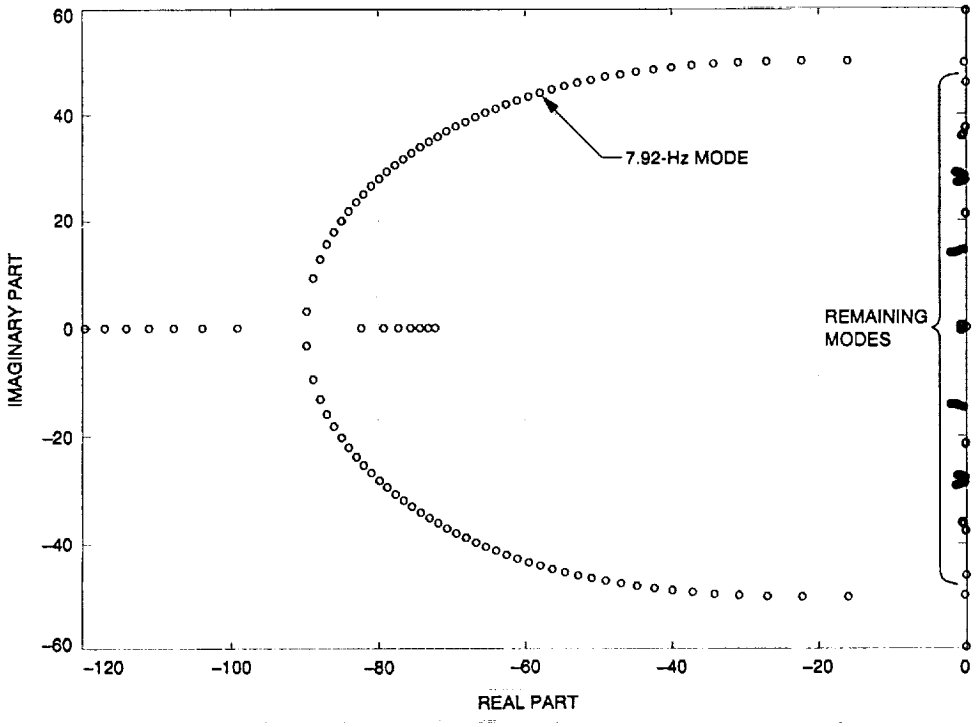


Fig. 15. Root locus for 7.92-Hz mode versus weight.

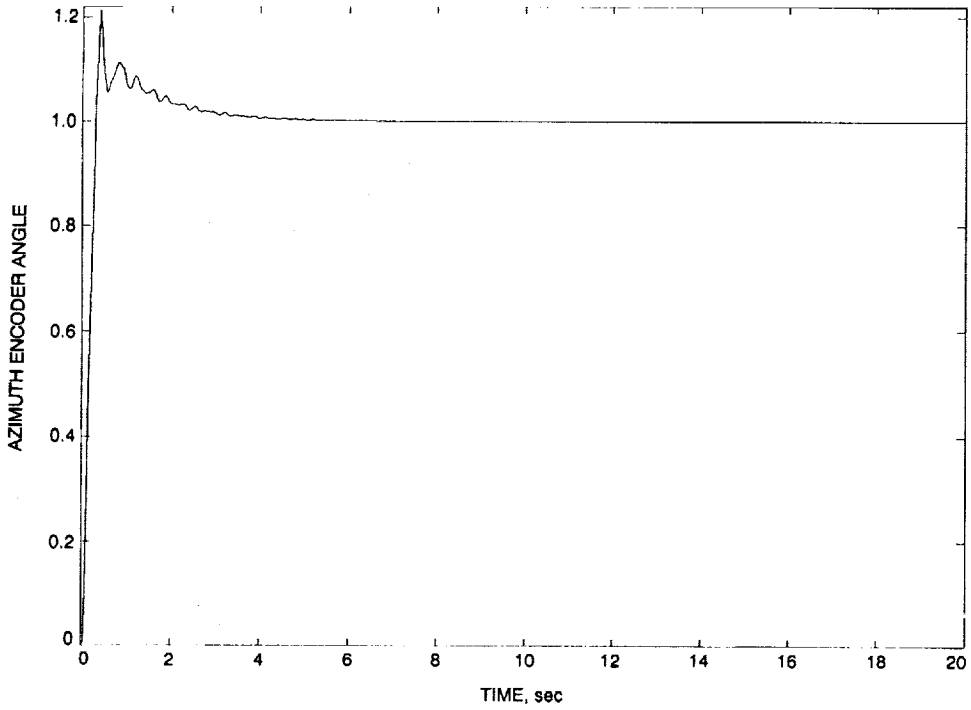


Fig. 16. Closed-loop response to step input for proportional weight 100 and integral weight 70 (both in azimuth and elevation), and weights for flexible subsystem equal to 10^{-7} (azimuth encoder to azimuth command).

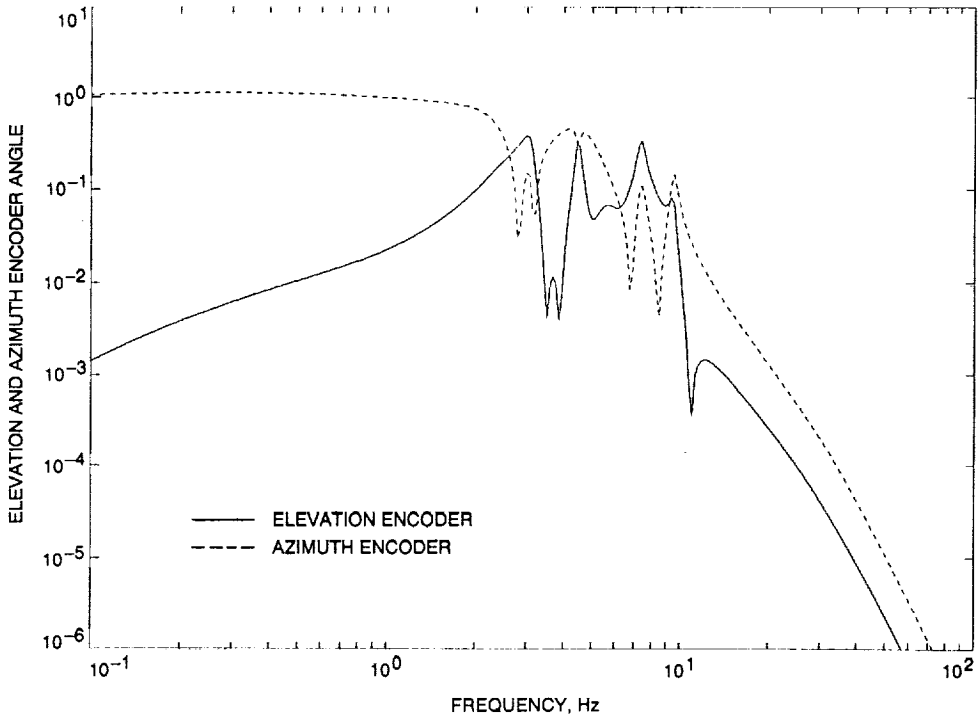


Fig. 17. Closed-loop transfer function for weights the same as those in Fig. 16 (elevation encoder and azimuth encoder to azimuth command).

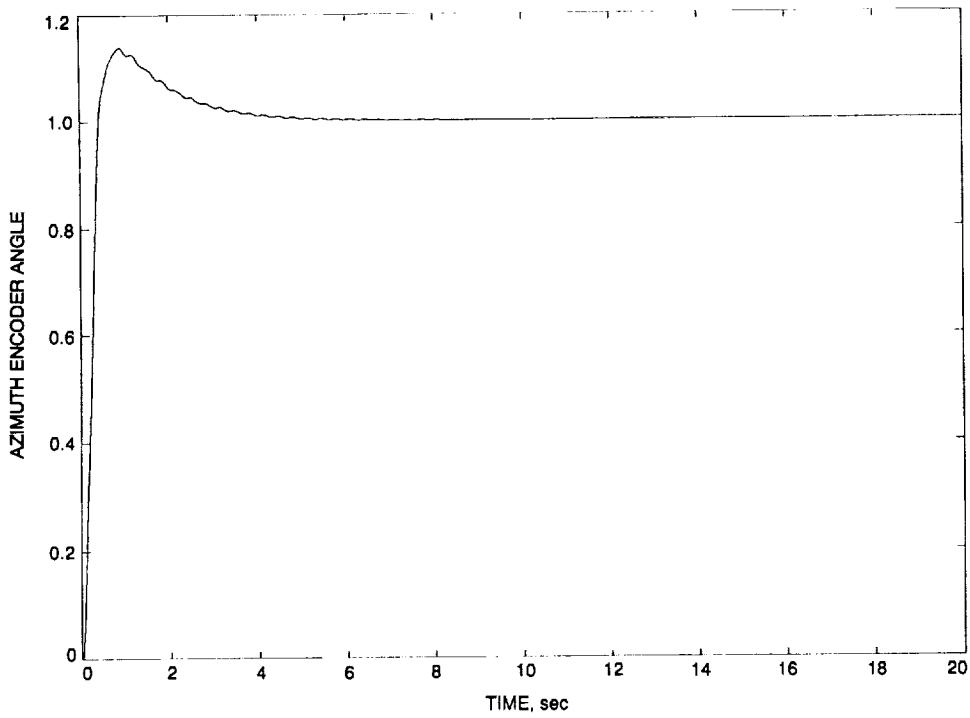


Fig. 18. Azimuth encoder response to azimuth step command for proportional weight 100, integral weight 70 (in azimuth and elevation), and for flexible weights $q_{f1} = \ldots = q_{f6} = 10^{-6}$, $q_{f7} = q_{f8} = 10^{-7}$, and $q_{f9} = q_{f10} = 10^{-5}$.

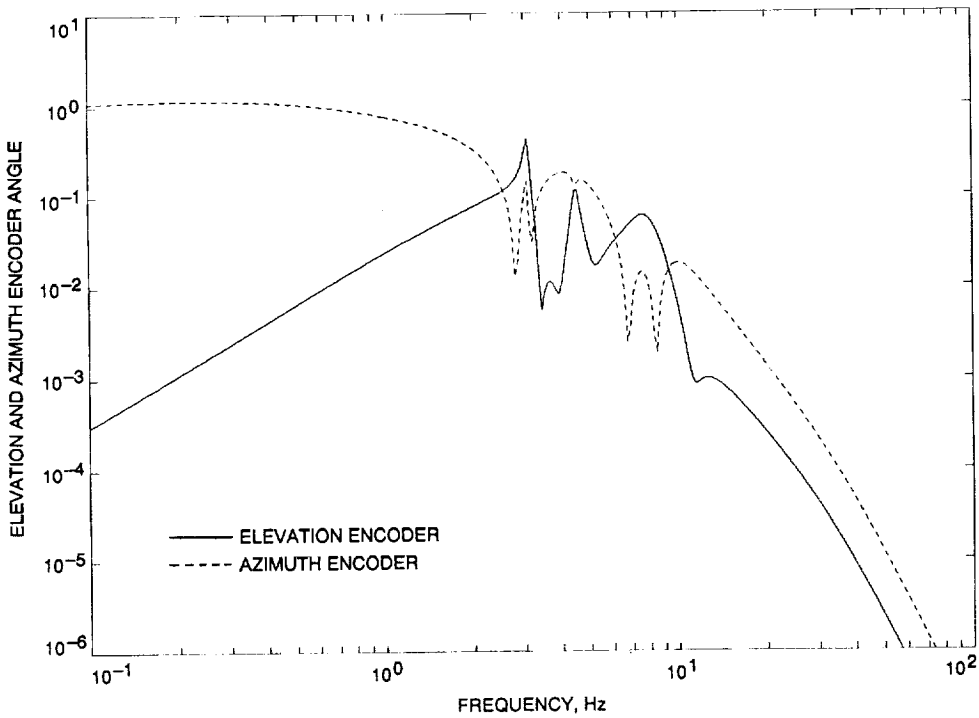


Fig. 19. Closed-loop transfer functions for weights the same as those in Fig. 18 (elevation encoder and azimuth encoder to azimuth command).

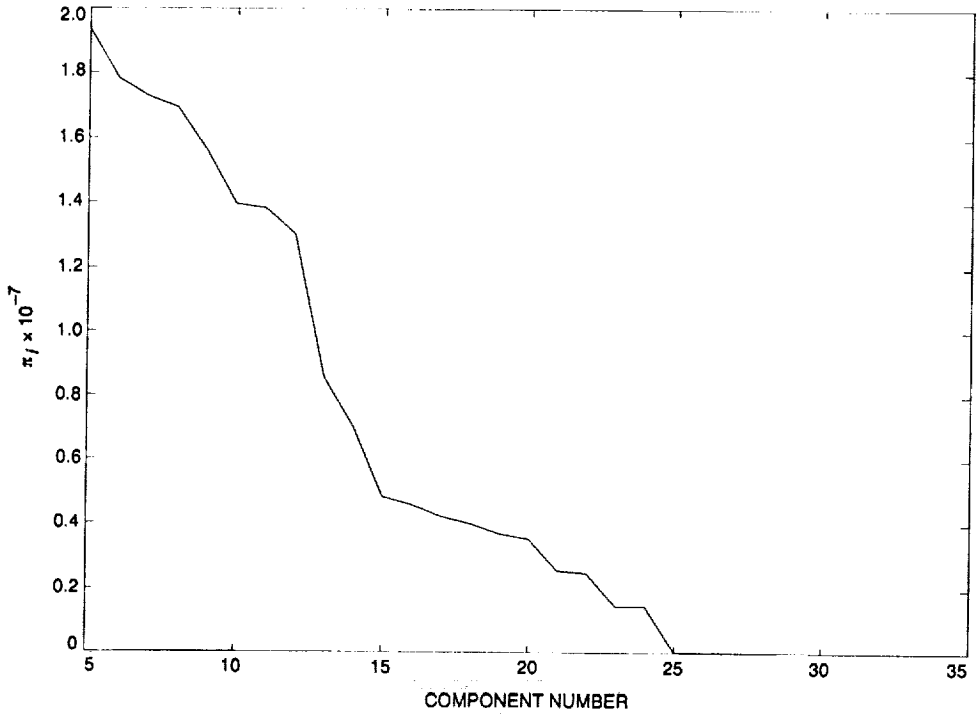


Fig. 20. Pole mobility indices for the DSS-13 antenna.

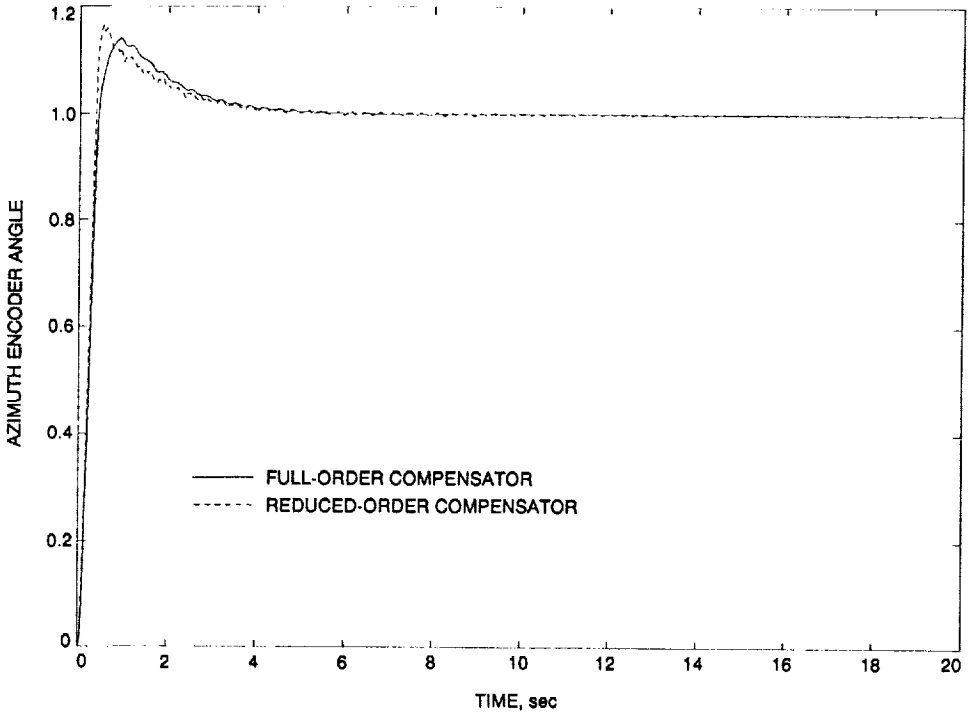


Fig. 21. Step response (azimuth encoder to azimuth command) of the DSS-13 antenna with full- and reduced-order compensators.

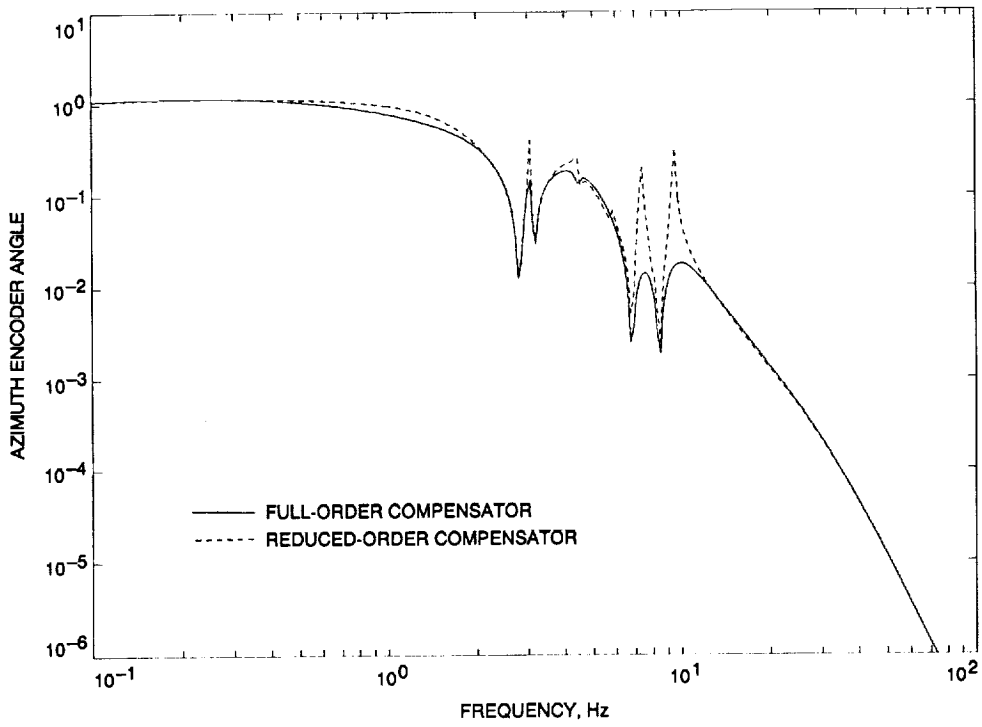


Fig. 22. Transfer function (azimuth encoder to azimuth command) of the DSS-13 antenna with full- and reduced-order compensators.

Appendix

I. Selected Properties of Flexible Systems

The balanced grammian for a flexible system with n components (or $N = 2n$ states) has the following form [7,8]:

$$\Gamma \cong diag(\gamma_1, \gamma_1, \gamma_2, \gamma_2, ..., \gamma_n, \gamma_n) \qquad \text{(A-1)}$$

The system matrix A is almost block diagonal, with dominant 2×2 blocks on the main diagonal:

$$A \cong diag(A_i), A_i = \begin{bmatrix} -\zeta_i\omega_i & -\omega_i \\ \omega_i & -\zeta_i\omega_i \end{bmatrix}, \quad i = 1, ..., n \qquad \text{(A-2)}$$

where ω_i is the ith natural frequency of the structure, and ζ_i is the ith modal damping. The matrices B and C are divided into two blocks, comparably to A, $B^T = [B_1^T, B_2^T, ..., B_n^T]$, and $C = [C_1, C_2, ..., C_n]$, with the following property:

$$B_iB_i^T \cong C_i^TC_i \cong -\gamma_i^2(A_i + A_i^T) \qquad \text{(A-3)}$$

II. Proof of Eq. (17)

Due to the diagonally dominant matrix A for a flexible structure in balanced representation, and for Q as in Eq. (16), there exists $q_i \leq q_{oi}$, $i = 1, ..., n$, such that the solution S of the Riccati Eq. (4) is also diagonally dominant with 2×2 blocks S_i on the main diagonal:

$$S_i \cong s_iI_2, \quad s_i > 0, \quad i = 1, ..., n \qquad \text{(A-4)}$$

Thus, Eq. (4) turns into a set of the following equations:

$$s_i(A_i + A_i^T) - s_i^2B_iB_i^T + q_iI_2 = 0, \quad i = 1, ..., n \qquad \text{(A-5)}$$

For a balanced system $B_iB_i^T \cong -\gamma_i^2 (A_i + A_i^T)$ and $A_i + A_i^T = -2\zeta_i\omega_iI_2$, see Eqs. (A-3) and (A-2), respectively. Therefore, Eq. (A-5) is now

$$s_i^2 + s_i/\gamma_i^2 - 0.5q_i/\zeta_i\omega_i\gamma_i^2 = 0, \quad i = 1, ..., n \qquad \text{(A-6)}$$

There are two solutions of Eq. (A-6), but for a stable system and for $q_i = 0$ it is required that $s_i = 0$, therefore Eq. (17) is the unique solution of Eq. (A-6).

III. Proof of Eq. (23)

For small q_i, the matrix A of the closed-loop system is diagonally dominant $A_o \cong diag\ (A_{oi})$, $i = 1, ..., n$, and $A_{oi} = A_i - B_iB_i^Ts_i$. Introducing Eq. (A-3), one obtains

$$A_{oi} \cong A_i + 2s_i\gamma_i^2(A_i + A_i^T) \qquad \text{(A-7)}$$

and introducing A_i as in Eq. (A-2) to Eq. (A-7) one obtains

$$A_{oi} = \begin{bmatrix} -\beta_{pi}\zeta_i\omega_i & -\omega_i \\ \omega_i & -\beta_{pi}\zeta_i\omega_i \end{bmatrix} \qquad \text{(A-8)}$$

with β_{pi} as in Eq. (17).

IV. Proof of Eqs. (26a) and (26b)

In order to prove Eqs. (26a) and (26b), the closed-loop Lyapunov equation is considered:

$$(A - BB^TS)\Gamma_c^2 + \Gamma_c^2(A - BB^TS)^T + BB^T \cong 0 \qquad \text{(A-9a)}$$

or, for the ith pair of variables,

$$(A_i - B_iB_i^Ts_i)\gamma_{ci}^2 + \gamma_{ci}^2(A_i - B_iB_i^Ts_i)^T + B_iB_i^T \cong 0 \qquad \text{(A-9b)}$$

Introducing Eq. (A-3) gives

$$\gamma_{ci}^2 + 2\gamma_{ci}^2\gamma_{oi}^2s_i - \gamma_{oi}^2 \cong 0 \qquad \text{(A-10)}$$

or

$$\gamma_{oi}^2/\gamma_{ci}^2 \cong 1 + 2s_i\gamma_{oi}^2 \qquad \text{(A-11)}$$

Comparing Eq. (A-11) and Eq. (17) gives Eq. (26a).

Performance Results of a 300-Degree Linear Phase Modulator for Spaceborne Communications Applications

N. R. Mysoor and R. O. Mueller
Spacecraft Telecommunications Equipment Section

A phase modulator capable of large linear phase deviation, low loss, and wideband operation with good thermal stability has been developed for deep space spacecraft transponder (DST) applications at X-band (8.415 GHz) and Ka-band (32 GHz) downlinks. The design uses a two-stage circulator-coupled reflection phase shifter with constant gamma hyperabrupt varactors and an efficient modulator driver circuit to obtain a phase deviation of ±2.5 rad with better than 8 percent linearity. The measured insertion loss is 6.6 dB ±0.35 dB at 8415 MHz. Measured carrier and relative sideband amplitudes resulting from phase modulation by sine wave and square modulating functions agree well with the predicted results.

I. Introduction

The telecommunication transponders [1] for Cassini and other future deep space missions require an X-band linear phase modulator. The design is to provide ±2.5 rad of peak phase deviation to accommodate downlink modulation of telemetry and ranging signals. The tolerance on the phase deviation linearity is 8 percent. The phase modulator design specifications [2] for the deep space transponder applications are listed in Table 1. The modulator design concepts, development, analysis, and measured results are presented in this article. The description of the breadboard phase modulator is presented in Section II, the test data and analysis are presented in Section III, and the conclusions are presented in Section IV.

II. Description of the Phase Modulator

The X-band phase modulator is designed as a cascaded two-stage circulator-coupled reflection phase shifter [2,3,4] with constant gamma hyperabrupt varactors. The gallium arsenide (GaAs) varactor diode is well suited for the phase modulator application as it can provide rapid phase change with the applied voltage. The diode capacity parameter, gamma (Γ), is the slope of the capacitance–voltage curve as plotted on log–log paper. The gamma of a typical hyperabrupt varactor varies widely with applied voltage, which makes these varactors unsuitable for linear phase-modulator applications. Recent device processing developments [4] have enabled the construction of varactors with gammas that remain constant over a limited voltage bias

range. The GaAs varactors[1] used in this investigation have a constant gamma equal to 1.5. The varactor diode is usually modeled as a junction capacitance in series with the resistance of the GaAs substrate and epitaxial layers. The junction capacitance of varactor diodes is modeled as

$$\frac{C_j(V)}{C_0} = \left[1 + \frac{|V|}{\Phi}\right]^{-\Gamma} \quad (1)$$

where $C_j(V)$ is the junction capacitance at reverse bias voltage, V. The value C_0 is the junction capacitance at $V = 0$, and Φ is the built-in potential, which is equal to 1.3 V for GaAs.

The series resistance (R_S) of varactor diodes can be determined from the measured diode's quality factor, Q. The Q factor is a measure of loss in the varactor diode. The expression for Q as a function of the bias voltage, used in the simulation, was obtained by curve fitting the manufacturer's (Microwave Associates') data at 50 MHz:

$$Q(V) = 1538 - 115.6|V| + 118.2|V|^2 + 21.96|V|^3 - 1.683|V|^4 \quad (2)$$

The diode series resistance is then given by

$$R_S(V) = [2\pi \times 50 \times 10^6 \times Q \times C_j(V)]^{-1} \quad (3)$$

A phase modulator consisting of a two-section reflection phase shifter [2] and a driver circuit was breadboarded and tested. The phase modulator photograph and schematic are illustrated in Figs. 1 and 2, respectively. The module is made out of aluminum alloy 6061, which is tin-lead plated to ensure ease of soldering and good electrical conductivity. The size of the module is 81 × 61 × 14.3 mm. The two-section phase shifter layout is etched on a 0.508-mm-thick RT/Duroid 6002 soft substrate of a dielectric constant equal to 2.94. The constant 1.5 gamma varactors[2] are mounted with a contact strip at the end of an optimized 30-ohm, 2.794-mm-wide transformer. The diode capacitance at −4 V is equal to 0.65 pF. The diode package parasitic capacitance and inductance are equal to 0.23 pF and 0.20 nH, respectively. TRAK Microwave 6.35-mm micropuck circulators [79*9001] and isolators [89*9001] are used in this breadboard. The driver circuit is etched on 0.787-mm-thick RT/Duroid 5880, which has a dielectric constant of 2.2. The phase modulator driver circuit schematic is shown in Fig. 3. The functions of the phase modulator drive circuit are to sum and amplify the modulation input signals and to provide composite drive voltage to the varactor diodes. The modulation input signals include the spacecraft telemetry, ranging, and differential one-way ranging (DOR) signals. The modulation frequency range is from 1 kHz to 20 MHz. The selected wideband op-amp for this application is Comlinear CLC 505.

III. Experimental Results

The measured values of phase deviation and insertion loss for the phase modulator are compared with the predicted results in Fig. 4. The predicted results were obtained by the computer microwave computer-assisted-design (CAD) simulation of an extensive circuit model of the circuit schematic shown in Fig. 2. The measurements were conducted using a test RF signal level of +9 dBm. The insertion loss for this unit is 6.35 ±0.25 dB, the phase deviation is ±150 deg, with a linearity better than ±8 percent of a best-fitted straight line (BSL). The varactor bias voltage range is 5 ±3.5 V. The model accurately predicted a linear phase slope of 42 deg/V. The predicted insertion loss, including circulator and isolator losses, is 7.0 ±0.4 dB, which is about 0.65 dB higher than the measured values. The discrepancy in the insertion loss is most likely due to the uncertainty in the measured insertion loss of the circulators and isolators. The insertion loss of these devices was measured in a separate test fixture. Circulators measured in this test fixture had a port-to-port one-way insertion loss of 1 ±0.1 dB. The test fixture is not enclosed and, therefore, has radiation losses which the module does not have. Also, circulators were not epoxied in the test fixture as they are in the module; this will result in higher losses since there is not a solid ground contact. The measured results of the phase modulator are compared with its design specifications in Table 1. The phase modulator meets or exceeds all specifications.

Figure 5 shows the predicted and measured gain versus frequency characteristics of the phase modulator driver circuit at −35 deg C, +25 deg C, and +85 deg C. A separate breadboard was used to test the driver circuit. The measured response is flat and drops only 0.4 dB, from 1 kHz to 20 MHz. The measured 3-dB bandwidths at −35 deg C, +25 deg C, and +85 deg C are 92 MHz, 84 MHz, and 75 MHz, respectively. The measured results are actually better than the Simulation Program with Integrated Circuit Emphasis (SPICE) predicted circuit response using a

[1] The varactors were manufactured by Microwave Associates, MA-46411-126.

[2] Ibid.

conservative circuit model for the op-amp integrated circuit. The model correctly predicted a larger bandwidth at lower temperatures. The dc power consumption with an 8-V peak-to-peak output swing is 112 mW; when no signal is applied, the power consumption is 32 mW.

Sinusoidal and square modulating waveforms were applied to the phase modulator, and their resulting spectra [5,6] were monitored on a calibrated spectrum analyzer. All measurements were performed at 25 deg C and with a carrier frequency of 8415 MHz. Figure 6 shows the resulting spectrum when a sinusoidal modulating wave was applied to the phase modulator. The modulating frequency used is 1 MHz, and the peak modulation index is 2.4 rad.

A comparison of measured and predicted carrier, first, second, and third sideband levels for the case of sinusoidal modulation is shown in Fig. 7. A modulation frequency of 10 MHz was used in the measurements. The peak phase-modulation index, β, ranges from 0.2 to 2.4 rad in 0.2-rad steps. The predicted relative carrier and sideband levels [6] were computed by evaluating the appropriate Bessel functions of the first kind of order n, $J_n(\beta)$. The subscript n is an integer. It represents the carrier for n equal to zero and sidebands for n not equal to zero. The relative carrier level for a given modulation index is determined by evaluating $20\log[|J_0(\beta)|]$. Similarly, the relative sideband levels may be determined by evaluating $20\log[|J_n(\beta)|]$ for $|n| > 0$. Figure 7 shows excellent agreement between theory and measurement for sinusoidal phase-modulating waves. Negligible amplitude modulation distortion was observed in this case.

Figure 8 shows the spectrum that results when a square modulating wave at a frequency of 100 kHz and a peak modulation index of 1.571 rad is applied to the phase modulator. A comparison of measured and predicted carrier, first, third, and fifth sideband levels for the case of square-wave modulation is shown in Fig. 9. A modulation frequency of 100 kHz was used in the measurements. The relative carrier level in decibels is determined [6] by evaluating $20\log[|\cos(\beta)|]$, where β is the peak modulation index. The relative sideband levels may be determined [6] by evaluating $20\log[|(2/n\pi)\sin(\beta)|]$, where n is the number of the sideband. Excellent agreement between predicted and measured square-wave modulation results is demonstrated in Fig. 9.

The data illustrated in Figs. 7 and 9 were obtained by the judicious use of a spectrum analyzer. For small modulation indices, setting the first sideband level is the most accurate way [5] to obtain a particular value of a modulation index. For example, if a modulation index of 0.2 rad (sinusoidal modulation) is needed, then the modulation power level should be adjusted until the first sideband level falls 20.0 dB below the unmodulated carrier level. The error in the measurement is determined by the accuracy of the spectrum analyzer. If the above reading is accurate to within 1 dB, then the modulation index would be accurate to within 0.03 rad. For larger modulation indices, the first sideband will give large errors. In this case, the carrier level should be used to set the modulation index. The decision of whether to use the carrier or the first sideband level can, in general, be made by choosing the one that has the steepest slope [5] near the desired modulation index.

IV. Conclusions

A two-section analog X-band reflection phase modulator with an efficient driver circuit was developed. The phase modulator performance is accurately predicted by the theoretical modeling and simulations. By using hyperabrupt junction GaAs varactor diodes of a constant gamma of 1.5, a full ± 2.5-rad deviation phase modulator was realized by cascading two phase shifter circuits. An efficient phase modulator driver circuit was also developed and tested. The measured gain-frequency characteristics of the modulator driver circuit over the design temperature range agreed well with the predicted results. The modulator driver 3-dB bandwidth and dc power consumption at 25 deg C are 84 MHz and 112 mW, respectively. At 8415 MHz, the measured phase shift and insertion loss of the phase modulator are equal to ± 2.5 rad, with better than ± 8 percent linearity, and 6.35 ± 0.25 dB, respectively. Measured carrier and sideband spectra showed excellent agreement with theory with negligible amplitude modulation distortion for both sinusoidal and square phase modulating waves.

Acknowledgments

The authors gratefully acknowledge the valuable support of A. W. Kermode for comments on design and analysis and J. H. Meysenburg for breadboard fabrication and testing.

References

[1] N. R. Mysoor, J. D. Perret, and A. W. Kermode, "An X-Band Spacecraft Transponder for Deep Space Applications—Design Concepts and Breadboard Performance," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-40, no. 6, pp. 1192–1198, June 1992.

[2] N. R. Mysoor and R. O. Mueller, "Design and Analysis of Low-Loss Linear Analog Phase Modulator for Deep Space Spacecraft X-band Transponder (DST) Application," *The Telecommunications and Data Acquisition Progress Report 42-105, vol. January–March 1991*, Jet Propulsion Laboratory, Pasadena, California, pp. 136–145, May 15, 1991.

[3] R. Garver, "360 deg Varactor Linear Phase Modulator," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-17, pp. 137–147, March 1969.

[4] E. C. Niehenke, V. V. DiMarco, and R. Friedberg, "Linear Analog Hyperabrupt Varactor Diode Phase Shifters," *IEEE 1985 International Microwave Theory and Techniques Symposium*, Digest of papers, pp. 657–660, June 1985.

[5] E. A. Whitman, "Phase Modulation Measurement Techniques for Improved Accuracy," *Microwave Journal*, pp. 113–116, June 1978.

[6] F. Stocklin, *Relative Sideband Amplitudes vs. Modulation Index for Common Functions Using Frequency and Phase Modulation*, Goddard Space Flight Center, distributed by National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia, November 1973.

Table 1. Measured results and design specifications for the phase modulator.

Parameter	Specification	Measurement (25 deg C)
Insertion loss	10 dB, max	6.6 dB, max
Amplitude modulation	±0.5 dB, max	+0.15/−0.35 dB
Input return loss	14 dB, min	25 dB (8415 ±50 MHz)
Output return loss	14 dB, min	33 dB (8415 ±50 MHz)
RF input level (for linear operation)	+10 dBm, max	+10 dBm, max
Linearity	±5 percent (best straight line) to ±2 rad	+4.8/−3.1 percent
Modulation sensitivity	2 ±0.2 rad/V	2.1 rad/V
Modulation input signal level at ±2.5 rad	±2.5 V, max	±1.28 V, max
Op-amp gain	—	2.8 V/V
RF modulation bandwidth		
−0.25 dB bandwidth	20 MHz, min	28 MHz
−3 dB bandwidth	82 MHz, min	160 MHz
DC power (at 20 MHz and ±2.5 rad)		
+5 V	25 mA, max	7.0 mA
−5 V	25 mA, max	6.9 mA
Modulation voltage range (to varactors)	—	−1.5 to −8.5 V

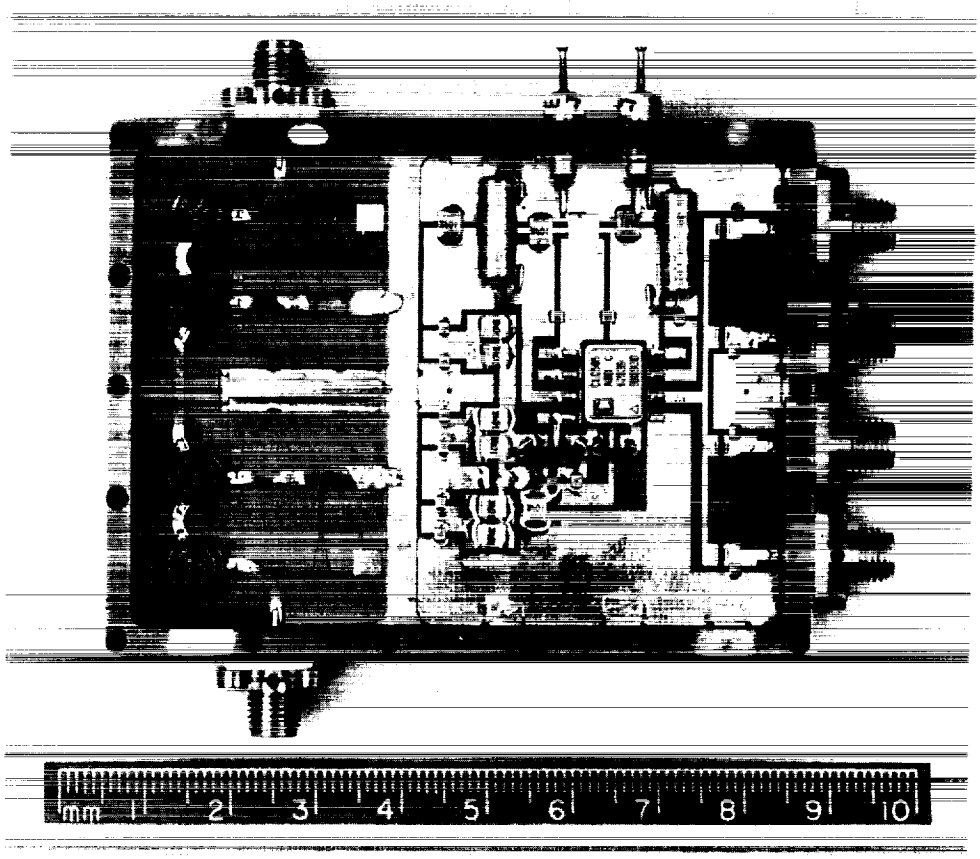


Fig. 1. DST X-band phase modulator and driver circuit assembly.

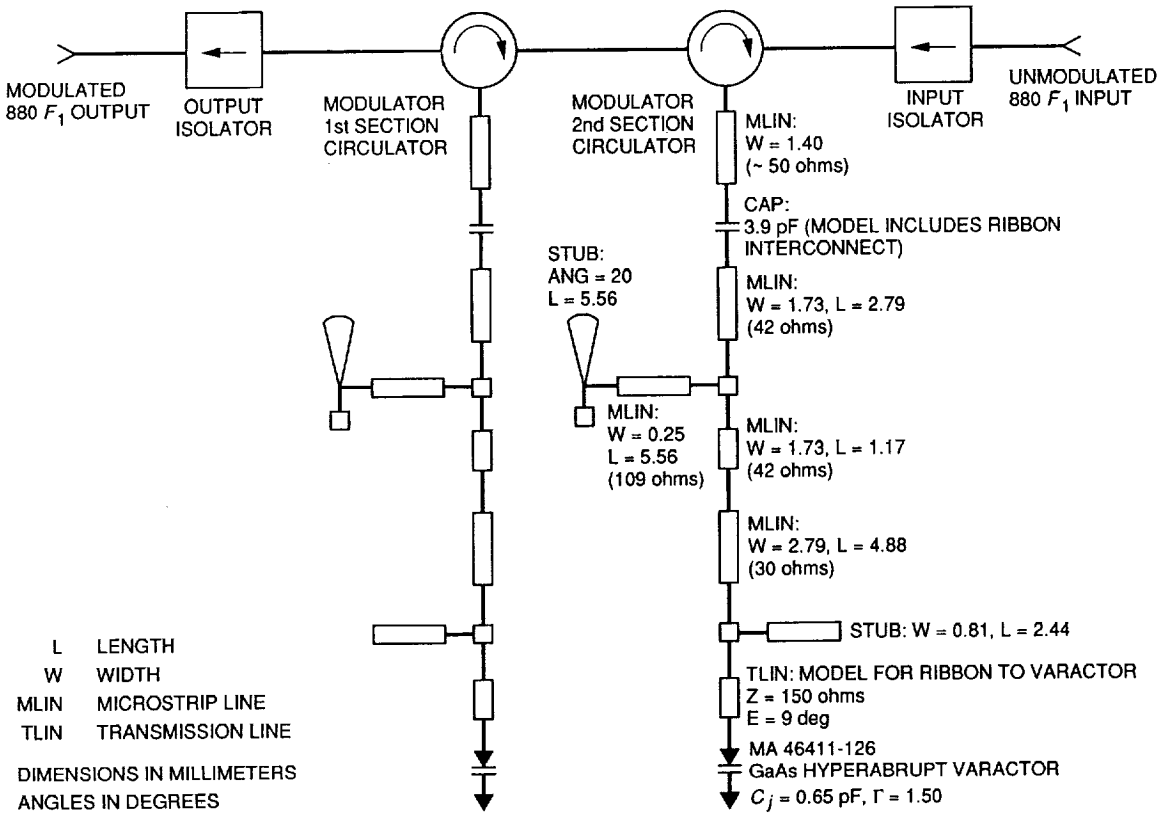


Fig. 2. DST X-band phase modulator schematic.

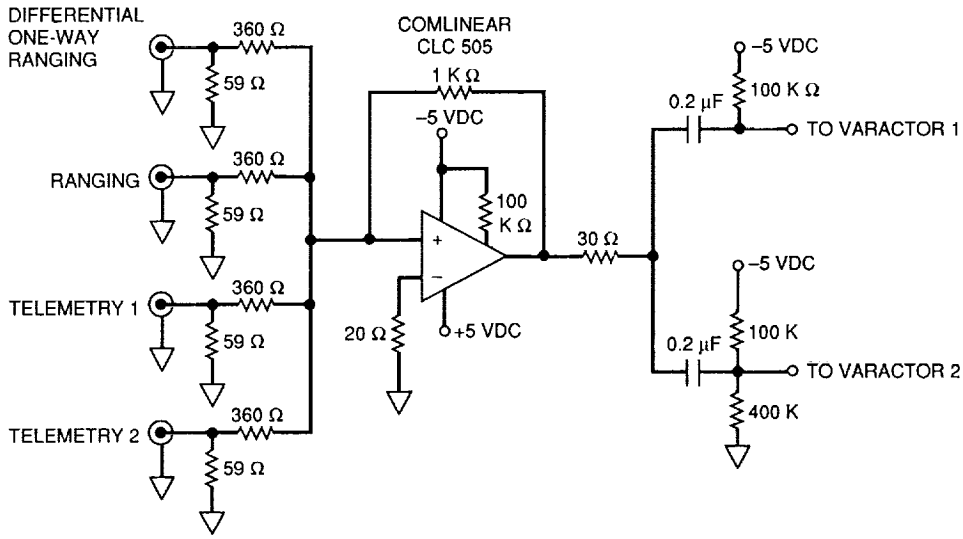


Fig. 3. DST phase modulator driver circuit schematic.

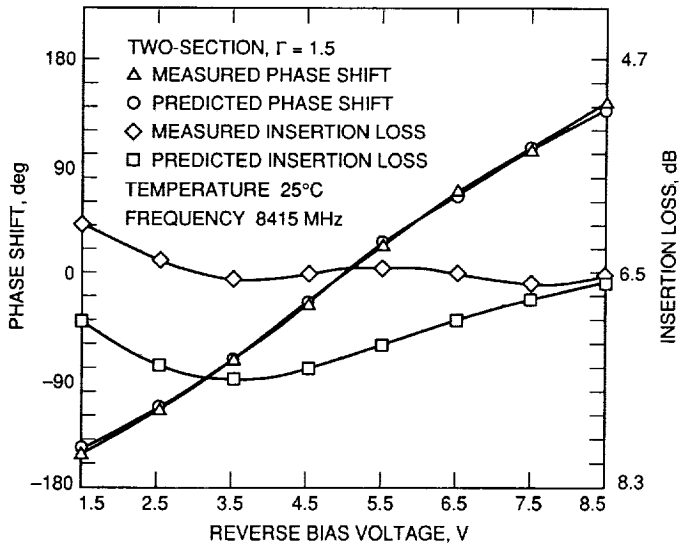


Fig. 4. Measured and predicted results for the phase modulator.

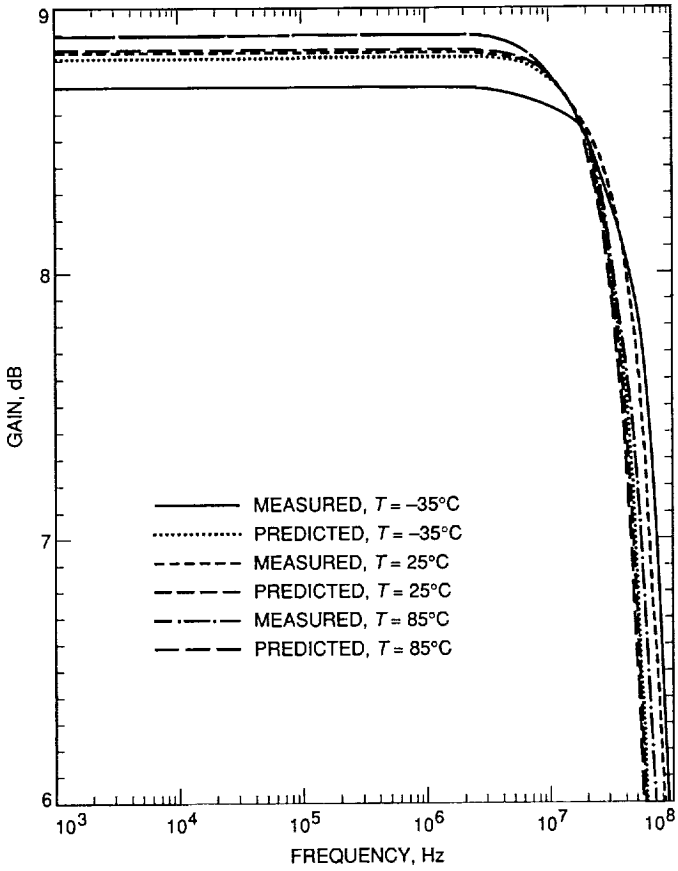


Fig. 5. Measured and predicted gain versus frequency for the phase modulator driver circuit at temperatures of –35, 25, and 85 deg C.

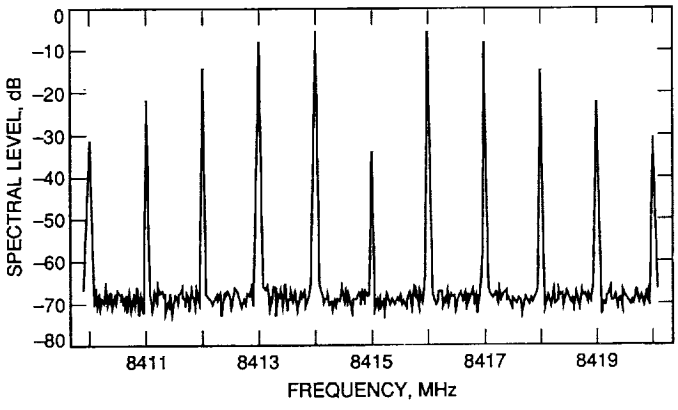


Fig. 6. Measured phase-modulated spectrum for the case of a sinusoidal modulating wave at a frequency of 1 MHz and a peak modulation index of 2.4 rad.

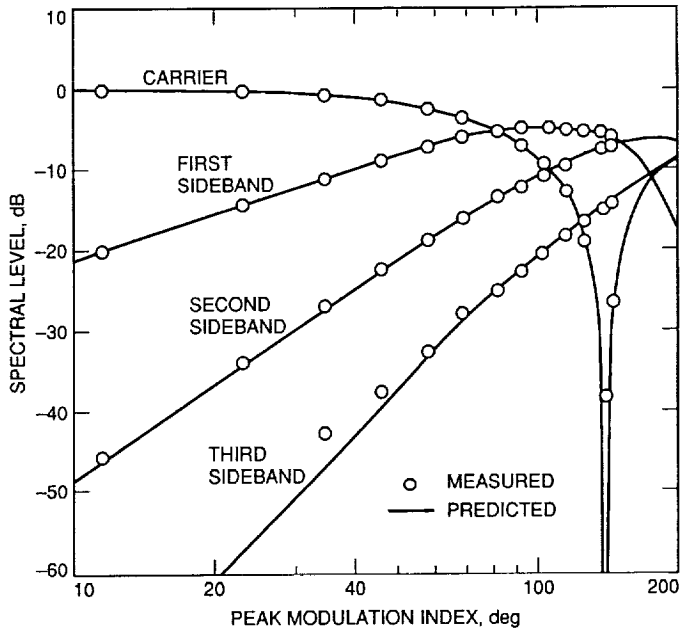


Fig. 7. Measured and predicted carrier, first, second, and third sideband levels versus peak modulation index for the case of a sinusoidal modulating wave at a frequency of 10 MHz.

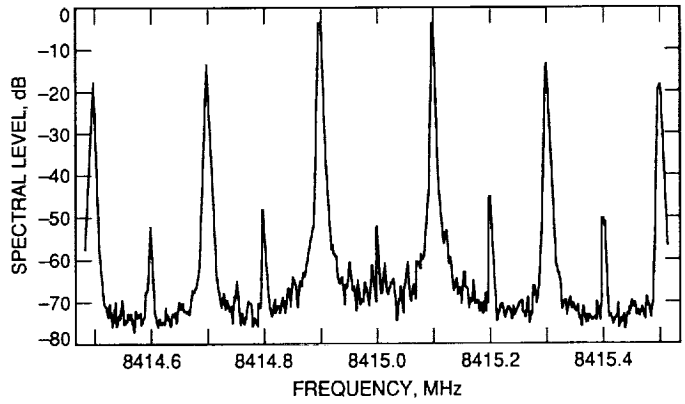


Fig. 8. Measured phase-modulated spectrum for the case of a square modulation wave at a frequency of 100 kHz and a peak modulation index of 1.571 rad.

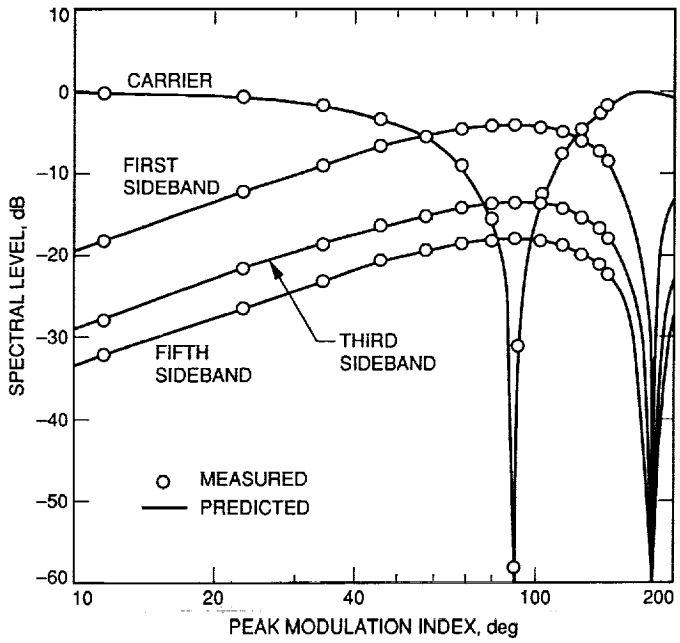


Fig. 9. Measured and predicted carrier, first, third, and fifth sideband levels versus peak modulation index for the case of a square modulating wave at a frequency of 100 kHz.

Management of Space Networks

R. W. Markley and B. F. Williams
Advanced Information Systems Section

NASA has proposed missions to the Moon and Mars that reflect three areas of emphasis: human presence, exploration, and space resource development for the benefit of Earth. A major requirement for such missions is a robust and reliable communications architecture. Network management—the ability to maintain some degree of human and automatic control over the span of the network from the space elements to the end users on Earth—is required to realize such robust and reliable communications. This article addresses several of the architectural issues associated with space network management.

Round-trip delays, such as the 5- to 40-min delays in the Mars case, introduce a host of problems that must be solved by delegating significant control authority to remote nodes. Therefore, management hierarchy is one of the important architectural issues.

The following article addresses these concerns, and proposes a network management approach based on emerging standards that covers the needs for fault, configuration, and performance management, delegated control authority, and hierarchical reporting of events. A relatively simple approach based on standards was demonstrated in the DSN 2000 Information Systems Laboratory, and the results are described.

I. Introduction

NASA has proposed missions to the Moon and Mars that reflect three areas of emphasis: human presence, exploration, and space resource development for the benefit of Earth. The Moon is a natural test-bed to prepare for missions to Mars through simulation, systems testing, operations, and studying human capabilities.

Communications with the Moon should be relatively straightforward with existing Deep Space Network (DSN) systems because lunar operations are initially planned to be centered at a main base on the near side of the Moon. However, a notable design issue will be communications from remote sites on the Moon. Such issues will increase the complexity of the space network.

The report on America's Space Exploration Initiative, *America at the Threshold* [1] describes the increased complexity of Martian communications as follows:

"Providing communications for the Martian missions is considerably more challenging than for lunar missions. Mars can be as much as 1,000 times more distant from Earth than the Moon, which results in a spatial signal loss one million times greater. In addition, Mars rotates at

about the same rate as the Earth, putting surface locations out of direct touch for over 12 hours at a time."

A tentative communications architecture for Mars (Fig. 1) proposes Mars-synchronous relay satellites to provide continuous coverage of surface elements as well as orbital elements [1]. The Martian main base may also have the ability to communicate directly with Earth when in view. Again in this case, the complexity of the space network will be increased as mobility and distribution of space elements are factored into the mission design.

A major requirement for such a lunar and Martian communications system is network management—the need to maintain some degree of human and automatic control over the network to assure highly reliable and robust communications from the space elements to the end users.

The following article addresses these issues, and proposes a network management architecture based on emerging commercial technology that covers the needs for fault, configuration, and performance management, delegated control authority, and hierarchical reporting of events. An approach based on standards was demonstrated in the DSN 2000 Information Systems Laboratory, and the results are described in this article.

II. Requirements

A. Space Management Network

Figure 2 illustrates a simplified schematic of the primary network to the Moon and Mars that will support science and human exploration. The figure illustrates the multiple paths between data sources and destinations, and the redundancy that is built into the primary architecture. For example, communications between one of the DSN antennas at a Deep Space Communications Complex (DSCC) and the Mars main base (Fig. 2(b)) may be direct or routed through a relay satellite. Data from the Mars habitat to a remote scientific instrument on Mars may be direct or routed through facilities on the relay satellite.

Round-trip delays, such as the 5- to 40-minute delays in the Mars case, introduce a host of problems that must be solved by delegating significant control authority to remote nodes.

This article proposes a Space Management Network (SMN) to support transmission of *management* data from all the network elements (from the space elements to the end users) to Earth-based operations centers. The SMN is a logical network that can be distinguished from the primary network because its chief function is to support transmission of management data. The SMN may use dedicated facilities or share facilities with the primary network.

Major nodes in the SMN include the mission and science support centers, the DSN, the lunar main base, and the Mars main base. The SMN may interface to each of these processing end points, as well as intermediate communication facilities.

In some ways space network management is similar to the management of complex Earth-based communications networks in which there are many types of interconnected networks, such as local area networks (LANs) and wide area networks (WANs). (In our unique network, however, there are also space segments.) Management of these complex configurations is an area of current research and development because of (1) the large numbers of nodes (in the thousands for some enterprises), (2) the geographic distances involved, (3) the remoteness of much of the equipment, (4) the need for human management of the systems, and (5) the vision that it is now an achievable goal because of the recent standardization of network management protocols.

B. Domains

The extent of the primary space network suggests a logical separation into four domains: Earth, Space, Moon, and Mars. Each domain may require varying degrees of security, performance, and availability. The networks in each domain are summarized in Table 1. This table identifies the types of subnets; the actual numbers of such subnets are a detailed design issue that will evolve with further mission planning.

1. Earth Domain. The Earth domain includes all the networks uniting the the network, mission, and science operations centers. The Deep Space Network, as the likely focus of network operations, would route video, voice, and data to the mission and science operations centers over domestic and international circuits. The network operations center could also be the primary location for integrated management of the end-to-end network.

Mission operations and science operations centers are responsible for management of the mission and monitoring and analyzing scientific data. A reasonable assumption is that thousands of network elements would be included in the Earth domain.

2. Space Domain. Moon and Mars main bases will be the destination end of the direct link from Earth. The main bases are likely to be the primary human interface for local network management activities.

Remote Moon rovers and science instruments may communicate directly with Earth if they are beyond the line of sight to the main base.

A second mode of Mars–Earth communication will be Mars-relay-satellite to Earth. This mode will be used when the main base is unable to communicate to the Earth because of Mars' planetary rotation. These links will require a complex management system with redundant equipment and redundant command channels.

It has been suggested that a solar activity warning system may be necessary to protect human explorers.[1] Such emergency data would be reported directly to the Moon and Mars over dedicated links. Depending on the Mars permanent outpost position, the data may go directly to the outpost or be routed through the communications relay. The data would provide timely warning of solar activity that may pose health hazards to humans on the surface. Such solar bursts induce extreme noise on the links and the links themselves will require substantial fault and error protection.

3. Moon Domain. Lunar surface-to-surface communications would be used for video, voice, and data communications. Nodes include a habitat, remote science instruments, and mobile rovers and humans involved in extravehicular activity (EVA). Locally, the links may be wire, optical fiber, or line-of-sight radio. Fault and configuration management are major issues with lunar communications.

Beyond the line of sight (approximately 6 km with a 10-m high antenna), lunar radio communications require a lunar surface path with intermediate radio relays, or a round-about path via direct Earth links.

4. Mars Domain. Surface-to-surface communications would be used for video, voice, and data communications. Nodes are similar to lunar nodes with similar communications options. Fault and configuration management are also major issues with Martian communications.

Beyond the line of sight (approximately 8 km with a 10-m high antenna), communications would likely be over intermediate surface radio relays or through an orbiting satellite relay. Surface-to-relay satellite communications would provide a path not only to locations over the horizon, but to the Earth and to other orbiting spacecraft.

Relay-satellite-to-relay-satellite communications extend the routing of voice and data to Earth beyond the limits of just one relay satellite. These links would also require a complex network management system with redundant equipment and redundant command channels to Mars and Earth. Manned and unmanned orbiting spacecraft at Mars will have network requirements similar to the communication relays. Manned orbiters will require intensive communication to operations centers on Earth and local communications to Mars.

[1] W. Kurth, personal communication, University of Iowa, January 5, 1992.

III. Network Management Technology

It is the premise of this article that end-to-end network management will be accomplished through a structured, evolvable management architecture based on standards because such an architecture is likely to minimize life-cycle costs. Hopefully, if the proper standards are chosen, a standards-based architecture will lead to utilization of low-cost commercial software products. The following elements are essential for the description of this architecture: (1) management model, (2) protocol architecture, (3) connectivity, and (4) human interfaces.

A. Management Model

In concept, network management usually involves application processes called "managers" on managing systems and "agents" on managed systems. Current commercial approaches generally focus on a management hierarchy with three layers of control (Fig. 3): element managers, network managers, and an integrated network manager.

An *element manager* performs management functions relative to that communication element and displays the data locally or makes the data available to a higher level management system. Typically network elements are managed through a *software agent*. The agent is devoted to monitoring the status and activity of the network element. The agent may be periodically polled by a higher level system or initiate urgent messages to a designated system if some threshold has been exceeded. If the concept were applied to the primary space network, examples would include agent software in network elements of the lunar main base or the DSN ground systems.

A *network manager* usually manages multiple elements on one type of network. A network manager is also concerned with managing the network circuits or channels, monitoring such parameters as link utilization and total packet rate over the medium. The area of surveillance could range in size from an area as small as a science data processing laboratory to as large as an antenna complex.

An *integrated network manager* is a layer of control at the highest layer that can incorporate information from

many types of networks. It typically interfaces to many network managers. An integrated network manager could be used to oversee end-to-end communications on the Earth, in space, and on the Moon and Mars. It is a "manager of managers." Such software systems are very complex; however, they are needed to make extended and complex systems humanly manageable. One of its greatest values is that the data are collected at a central location, and automatic fault management processes may be introduced, expanded, and modified as experience is gained with the network.

B. Protocol Architecture

A protocol architecture describes message formats for reporting management data and defines the managed objects. The managed objects are defined in terms of a management information base (MIB). The MIB includes a methodology for registering, identifying, and defining managed objects. There are numerous network management architectures and protocols available commercially; however, only two can be considered "standard." The first is the Simple Network Management Protocol (SNMP), developed by the Internet Activities Board (IAB) for use in the Internet, the world's largest public-access network; the second is the Common Management Information Service and Protocol (CMIS/CMIP) developed by the International Organization for Standardization (ISO) for use in Open Systems Interconnection (OSI) networks.

1. Internet Standards. The Internet has grown, especially in the last few years, as a result of the widespread availability of software and hardware supporting the Internet protocol suite. The suite includes such protocols as the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The Internet Simple Network Management Protocol was demonstrated in 1988 and has been implemented by a large (and increasing) number of vendors. Details about the protocol are described in Internet documents known as Requests for Comments (RFCs) that are referenced in the following section. At the present time, virtually every new commercial router and bridge incorporates an agent based on the SNMP protocol. Many computer companies, including Sun and Hewlett Packard, have included an SNMP agent in their operating system software, which, in addition to managing communications-related objects, also enables management of computer resources, such as disk storage availability and central processing unit utilization.

a. SNMP. The SNMP architectural model involves a *network management station* and a collection of *network elements* and remote *probes* (Fig. 4). The network management station monitors and controls network elements. Network elements are devices, such as workstations, routers and the like, which have software agents. Agents perform the functions requested by a remote network management station, and act solely on the elements they reside in. Probes passively monitor the network media and measure such characteristics as total throughput and efficiency. Management stations and network elements communicate using the SNMP message protocol.

Probes are a relatively new extension to the SNMP agent architecture. Currently these probes are available for Ethernets, and they passively collect statistics and historical information from the network.

Monitoring of the network state at any significant level of detail is normally accomplished by polling for appropriate information by the network management station. A spontaneous message (called a "Trap") is used by the agent to notify a network manager of abnormal conditions.

b. MIB. In the SNMP view, the objects to be managed are identified in a management information base. The MIB is a *virtual* store—that is, a concept that identifies all the objects that need to be managed in the network along with their parameters. In its actual implementation, the values of the MIB objects may be locally stored in the element and then reported upon request to the network manager and stored in its database. The organization and structure of the MIB is described in RFC 1155 [2].

The MIB defined by the Internet community has over 100 formal objects, called the *common* MIB [3]. The MIB has recently been extended (MIB II); the examples drawn in this article relate to the common MIB. A typical device may also have an additional 100 to 200 objects that have been defined in private or experimental MIB space. This extensibility could, for example, be applied to manage elements processing Consultive Committee for Space Data Systems (CCSDS) protocols.

The common MIB is organized into eight object groups (Table 2). While details of these groups are fully described in RFC 1213 [4], Table 3 presents an excerpt from the Interfaces Group to illustrate the concept. Many of the objects are similar to what would be managed in the Earth domain of the primary network.

c. SNMP messaging. The writing (setting) and reading (getting) of variables in an agent is accomplished through the use of the SNMP message. This protocol is described in RFC 1157 [3]. SNMP models all management agent functions as alterations or inspections of variables.

The SNMP message is contained in an SNMP protocol data unit (PDU). There are five types of SNMP PDUs:

GetRequest, GetResponse, GetNextRequest, SetRequest, and Trap, and they are described in Table 4. The User Datagram Protocol is used to deliver the SNMP PDUs.

d. Remote monitoring MIB. The remote monitoring (RMON) MIB is an extension of the SNMP MIB that applies to probes. The RMON MIB for Ethernet networks has been the first to be standardized [5]. It is intended that future versions of the RFC will define extensions for other network types.

2. International Standards. International standards have also been developed by ISO for the management of OSI-based networks. The OSI management framework is more elaborate than SNMP, but is similarly designed to control, coordinate, and monitor network resources. Although the development of the OSI network management architecture is close to completion, there are very few OSI agents implemented at the present time.

The United States Government has mandated the use of selected OSI protocols for Federal information systems in the Government OSI Profile [6]. A transition to these protocols is planned for NASA administrative systems [7]. Another OSI-related profile, the Government Network Management Profile [8], has been proposed for Federal systems that specifically focuses on network management. The impact of these profiles on stimulating development of OSI network management capability remains to be seen.

OSI management protocols are generic and may be used in any OSI command and control environment (not just network management). The management structure is defined in ISO 7498-4 [9]. In the OSI architecture, when an application process, such as an agent or manager, needs to exchange information and commands with another application process, it makes use of software known as the Common Management Information Service Element (CMISE). The CMIS standard [10] defines the service that the CMISE provides and CMIP [11] defines the protocols that it uses.

CMIP and CMIS standards were published in May 1990. They provide a flexible framework for the control and exchange of management information. Together, CMIS and CMIP define the bulk of the OSI network management protocol. Revised versions of CMIS and CMIP (CMIS/CMIP Version 2) were published in January 1992. With this update, CMIS and CMIP are expected to remain stable for a number of years.

a. OSI management structure. The requirements for OSI network management are grouped into the following five major functional areas: (1) fault management, (2) accounting management, (3) configuration management, (4) performance management, and (5) security management. While the scope is very impressive, a commercial product that implements these capabilities is not currently available.

In brief, fault management encompasses fault detection, isolation, and correction of abnormal operation of the OSI environment. Accounting management enables charges to be established for the use of resources in the OSI environment, and for costs to be identified. Configuration management includes functions to change the configuration of the system, set the parameters that control the routine operation of the system, and initialize and close down managed objects.

Performance management includes functions to determine system performance under natural and artificial conditions, to gather statistical information, and to maintain and examine historical logs. The purpose of security management is to support the application of security policies.

b. CMIS. The major difference between SNMP and OSI architectures is that while SNMP assumes an agent–manager relationship, OSI does not assume any management hierarchy. In fact, peers may communicate among themselves using CMIS/CMIP. Any necessary hierarchy is imposed by the management architects. This should simplify the development of an integrated network management (manager-to-manager) system.

CMIS management operations include a number of services that are summarized in Table 5. Note that "gets" and "sets" are similar in function to their SNMP counterparts.

c. CMIP. CMIP is a general-purpose protocol that supports the services defined by CMIS. CMIP, in turn, requires specific support from several relatively hidden OSI protocols. For example, the association services, M-INITIALIZE, M-TERMINATE, and M-ABORT, are supported by CMIP by invoking the Association Control Service Element defined in ISO 8649. The notification and operation services require use of the Remote Operations Service Element defined in ISO 9072-1. These complexities are characteristic of OSI implementations.

3. Summary of Data Architectures. The key words that summarize SNMP are "short-term" and "simple." The protocol was implemented within a few months. Today the protocol is almost universally implemented in network components, such as routers, and in certain computers. However additional extensions are being developed

in a newer version, SNMP Version 2, such as: (1) an authentication scheme to ensure security and filter out messages that may cause catastrophic errors and (2) the means to support an integrated network management system at a higher level.

As has been mentioned earlier, few, if any, OSI network management agents have been implemented in commercial equipment. However, there is strong bureaucratic interest in its adoption. The Federal Government promotes it and targets integrated network management systems as the key beneficiary.

IBM announced in March 1992 [12] that it plans to use CMIP to send information from its Advanced Peer-to-Peer Networking Network Nodes to Netview (the IBM network management system). This application of CMIP does not use OSI transport protocols; it runs over a traditional IBM Systems Network Architecture protocol stack and is thus a "hybrid" application.

The Internet community has also proposed a hybrid CMIP implementation. The implementation, called CMIP over TCP/IP or CMOT, has been implemented by a few vendors in their network manager software (not in any agents) and runs over Internet transport protocols [13].

As systems grow to include large numbers of monitored objects and subsystems, any standard evolvable management architecture remains a challenge. Fault management will be easier to scale because it usually operates on an exception basis. Performance and configuration management are more likely to initiate periodic reporting and create more traffic as the number of objects increases. The Space Management Network needs prototyping and analysis in this area.

The present application software interfaces have limitations. There is no sense of time other than "now." This makes it impossible to directly issue queries for historical information, or to issue scheduled command requests; these queries must be made through user-developed application software.

C. Connectivity

Connectivity between the element managers and the network management system may be over the primary data network or over dedicated links. An advantage of dedicated out-of-band circuits is that when the primary media becomes unavailable because of congestion or malfunctioning communications equipment, management systems can still determine and resolve the problem.

Path diversity is a major consideration in providing a wide range of options for the design of robust networks. Both SNMP and CMIP/CMIS have control messages that can be used to select alternate paths if the initial path is blocked or highly congested.

D. Human Interfaces

The human interface provides a location for management control of the primary network. It should include a standard presentation format at the user and the systems level. It should also include the availability of artificial intelligence to assist the managers.

A variety of technologies are expected to provide improved methods for allowing users to interface to computer systems. These interface technologies focus on improvement of the amount of information that the user can perceive from a given interface configuration. Graphic visualization and interactive displays are two particularly helpful technologies. Graphic visualization may be used to represent multidimensional data on computer graphics displays in images and in a form that allows people to perceive, amplify, and interpret the data. Animated models may be used for this interface.

Interaction between the user and the data will facilitate fault resolution. Transformations and algorithms may be used to explore the effects on the data.

Hypermedia software technology enables a user to retrieve data in various formats in one or more display windows. The formats may include text, graphics, animation, digital audio, and video. Hypermedia lends itself to browsing and searching knowledge bases.

Artificial intelligence (AI) techniques that help translate raw data into knowledge may be applied—such as the technique of context sensitivity to filter data using dynamic thresholding. Also, AI techniques may be used to abstract information and present summaries to the user. AI can provide techniques for knowing what state the entire system is in and how ongoing activities are expected to affect that state.

IV. Space Management Architecture

A. Management Model

A suggested architecture to support the First Lunar Outpost is illustrated in Fig. 5. Figure 5(a) illustrates the primary network and Fig. 5(b) illustrates the SMN, pointing out the operations centers. The SMN hierarchy has four, rather than three, tiers (Fig. 5(c)). At the highest level is an Integrated Operations Center (IOC) for the highest level overview of the status of the extended network. At the next lower level are two major operations

centers: the first is an Earth Operations Center (EOC), to support the Earth and Space domains, and the second is a Moon Operations Center (MOC), an on-site facility to support the Moon domain. At a later time, a Mars Operations Center will be required. At the next lower level are facility managers that provide network management for a limited number of large facilities, such as the DSN. At the lowest level are numerous element managers; these managers oversee individual network components, such as bridges, routers, and computers.

1. Integrated Network Manager. An integrated network manager will provide high-level coordination and security management. It will primarily communicate with the EOC and MOC and sometimes coordinate activities. The IOC may serve as an alternate EOC in the event of an emergency. This situation must be jointly reported for action. The action may be to jointly modify the forward error correction coding algorithm to increase the level of coding.

2. Operations Managers. Operations management systems will be required at the EOC and MOC where global fault, configuration, performance, and security issues must be reconciled. A possible role could be resolving an event, such as a severed fiber trunk that causes sudden communications outage between the DSN and Mission Control. Facility managers will report the problem to the EOC for action and the EOC may automatically reroute the signals over a diverse path.

3. Facility Managers. The facility managers oversee all the elements associated with their facilities. At this layer there is substantial fault tolerance—usually through the ability to manually and automatically reconfigure active elements. Potential facilities include the lunar main base, DSN, Mission Operations, Science Operations, and Network Operations. A possible role could be detection of a security threat, i.e., a persistent hacker. Such a problem would normally be elevated to the IOC for action.

4. Element Managers. The proposed SMN architecture has many element managers at each of the facilities. The element managers normally communicate with the following: (1) a facility manager to coordinate a response to local events, such as reboots and power outages; (2) a local human interface for local maintenance; and (3) a local database to store MIB parameters. The element should have fault tolerance through its own internal design.

B. Data Architecture

The primary architecture requirement is that the element managers incorporate the same standard management agents. The current popular standard is, of course, SNMP. Assuming that it will evolve into a more secure protocol with peer-to-peer communication capability, it is a leading candidate to prototype the higher management layers for the Space Management Network. A simple transition can be made later to CMIP, if such a transition is advantageous.

Network management can be implemented in the near term by specifying SNMP in all Earth-domain network elements—especially the DSN ground systems. The MIB can also be extended to include OSI and CCSDS protocol performance.

C. Connectivity

A major issue in the design of the SMN is the design of the physical network. The telecommunications industry has been migrating toward out-of-band or "common channel" signaling, a technique of putting management data into its own data channel separate from the primary data. In the case of LANs, this would be a separate LAN; in the case of WANs, it would be either a diverse circuit or a dedicated radio channel. In the space domain, the management data path may be a dedicated radio channel.

Another issue is the use of redundant communication paths for management data. A leased circuit may be identified for nominal conditions in the Earth domain. As a backup, an alternate path, such as a dial-up circuit, may be used to retrieve management data.

The bandwidth required for management data will normally be relatively low. To minimize cost, bandwidth-on-demand is an important technology for exploration missions for both the primary network and the management network.

D. Human Interfaces

Graphic visualization should be used to present multidimensional data in images and in a form that allows people to perceive, amplify, and interpret the data. Of particular concern is the configuration and status of the primary network. Animated models and interactive displays may also be used.

Hypermedia software technology should enable a user to retrieve data in various formats in one or more display windows. The formats may include text, graphics, animation, digital audio, and video.

Artificial intelligence techniques should also be applied in each of the operations centers to assist operations personnel to recognize and diagnose problems and develop alternative solutions.

V. Test-Bed Implementation

A prototype network management system for the DSN was configured in the DSN 2000 Information Systems Engineering Laboratory (Fig. 6). The management model was a simple two-tier hierarchy with a commercial network manager (SunNet Manager Version 1.2) in a dedicated Sun SparcStation IPX computer and element managers in each network element (routers and computers). The data architecture was based on SNMP, and connectivity of the network manager to the elements was through two Ethernets joined by a serial circuit through commercial routers.

SunNet has a graphical user interface based on OpenLook/X Windows. SunNet maintains a database of the network elements and it can be configured to periodically poll the elements for crucial information, such as the state of the interfaces.

The information that is obtained by polling can be examined with a textual browser or an elementary three-dimensional graphing package. The element database and polling capabilities of SunNet provide a low-level management station capability that can be extended by other vendors or by the user with additional effort.

The SunNet Manager software package included additional SNMP agents that could be installed on any Sun SparcStation. In addition to these agents, the laboratory had access to a public-domain SNMP agent, available from Carnegie Mellon University (CMU), which was useable (with minor modifications) on any Unix workstation. The CMU agent included source code, which allowed extension of the agent to monitor private variables on the Sun workstations. SunNet Manager can be configured to recognize and manage these extra variables.

Using SunNet Manager as a manager of the two-tier prototype network revealed several limitations in its use for Facility and Operations Center management; these limits may be addressed by additional application software or future SunNet Manager upgrades.

A relational database with number manipulation abilities would have been useful in analyzing the average traffic flowing over an interface and helpful in characterizing its behavior. A statistics program would have helped in analyzing data and determining what a "reasonable" number of errors on an interface would be. Ideally, a management station should incorporate an expert system with knowledge about the network's configuration, so that it can anticipate problems, suggest solutions, or automate this process. SunNet did not have these capabilities.

Another benefit that a more advanced network management station should offer is "intelligent" polling, which can reduce the bandwidth required for network management. Under normal conditions, the network manager would request only a small number of management variables. If something anomalous were to be detected, the station could check additional variables to determine if a real problem exists.

There were several conclusions:

(1) Inexpensive commercial software exists today for doing network management on Internet-standard networks. The managers are relatively inexpensive, ranging from $1,000 to $10,000, depending on the features and degree to which the software is bundled with a larger computer purchase. SunNet Manager was only $1,000. The element agents are inexpensive or come free; they are usually included in most routers and many computers. The SNMP agent software is also publicly available, and may be used with older Unix and MS-DOS workstations.

(2) Limited bandwidth over certain links, such as the DSN WAN circuits, require that the volume of management traffic be minimized, and this was possible to a certain extent with SunNet Manager by changing the polling rate. In the prototype, the agents were polled every 8 sec. A four-level hierarchical management structure, such as that proposed in this article, would tend to minimize traffic even more because each network manager would send only summary data to higher level managers.

(3) The element agents can be modified to support additional MIB objects, such as those that would be DSN specific. These potentially include CCSDS protocol objects. SunNet Manager was highly configurable and easily adopted the new MIB objects.

(4) The current SNMP standard does not explicitly include manager-to-manager communications; however, it would be possible for a user to develop this capability within the existing standard. SNMP Version 2, currently in the standards process, includes this ability.

(5) It is possible to send management information over alternate circuits under the control of SNMP. In the laboratory, the computers and routers had diverse connectivity, which allowed management information to be transmitted over alternate Ethernets even if one specific network were down. This is an important design consideration for the Space Management Network. A robust, reliable network must have such alternate connections available. With a network management system, failures can be detected and data can be transmitted over alternate routes either automatically or with operator consent.

VI. Conclusions

A network management architecture was proposed for future exploration missions. The architecture has four layers of management: (1) element management for local monitor and control of communications nodes, (2) facility management for managing major sites, (3) two global operations centers to oversee Earth and Moon operations, and (4) an Integrated Operations Center to oversee end-to-end (e.g., lunar instrument-to-principal researcher) communications.

The most widely available standard in commercial networking hardware is the SNMP Internet standard. The standard is being upgraded by the IAB to include much of the functionality of its OSI counterpart, CMIP. Substantial cost savings will result from using the standard in the Earth domain that has the widest implementation and lowest cost, and this appears to be SNMP at the present time.

In the context of the end-to-end data system, the DSN is one of several facilities in the architecture. Although the DSN is one of the most important elements, it currently lacks a standard network management capability.

The following relevant issues remain:

(1) DSN Network Management. The DSN facility has strategic importance as the primary Earth- and space-domain interface. A DSN management architecture, when implemented, could be used as a reference implementation for other facilities in the SMN.

(2) Messaging Standard. Manager-to-manager communications need to be prototyped with a messaging technique, such as SNMP 2 or CMIP.

(3) MIB Development. CCSDS standards are to be applied to most future space missions. In order to manage network processing and distribution of CCSDS packets, a CCSDS MIB is required. Based on the experience of the Internet, the CCSDS MIB could be developed and tested in a CCSDS testbed, and then be permitted to evolve based on experience. Standardization of the MIB with the CCSDS would be a necessary step.

(4) First Lunar Outpost. A space management network for a specific proposed mission, such as the First Lunar Outpost, needs to be baselined in which specific roles for element, facility, operations, and integrated management are defined. The requirements for fault, performance, configuration, and security management need to be defined.

(5) Automated Functions. Automated functions need to be identified for selected remote activities, and then prototyped. This capability would demonstrate configuration management responsibilities of the SMN. An example could be to simulate alternate paths between major nodes (path diversity) and include automation as a means to select an optimal path.

References

[1] *America at the Threshold: Report of the Synthesis Group on America's Space Exploration Initiative*, Arlington, Virginia: The Synthesis Group, May 1991.

[2] *Structure and Identification of Management Information for TCP/IP-Based Internets*, RFC 1155, Menlo Park, California: SCI International, Network Information Center, May 1990.

[3] *A Simple Network Management Protocol (SNMP)*, RFC 1157, Menlo Park, California: SCI International, Network Information Center, May 1990.

[4] *Management Information Base for Network Management of TCP/IP-Based Internets: MIB II*, RFC 1213, Menlo Park, California: SCI International, Network Information Center, March 1991.

[5] *Remote Network Monitoring Management Information Base*, RFC 1271, Menlo Park, California: SCI International, Network Information Center, November 1991.

[6] *U.S. Government Open Systems Interconnection Profile (GOSIP), Version 2.0*, Washington, D.C.: U.S. Department of Commerce, National Institute of Standards and Technology, October 1990.

[7] *NASA Management Plan for Government Open Systems Interconnection Profile (GOSIP) Implementation*, Washington D.C.: National Aeronautics and Space Administration, OSI Management Steering Group, February 1992.

[8] *Proposed Government Network Management Profile (GNMP), Version 1.0*, Washington, D.C.: U.S. Department of Commerce, National Institute of Standards and Technology, March 8, 1991.

[9] *Information Processing Systems—Open Systems Interconnection—Basic Reference Model—Part 4: Management Framework*, ISO 7498-4, New York: American National Standards Institute, November 15, 1989.

[10] *Information Technology—Open Systems Interconnection—Common Management Information Service (CMIS) Definition*, ISO/IEC 9595, New York: American National Standards Institute, April 1, 1991.

[11] *Information Technology—Open Systems Interconnection—Common Management Information Protocol (CMIP)—Part 1: Specification—Part 2: Protocol Implementation Conformance Statement (PICS) Proforma*, ISO/IEC DIS 9596, New York: American National Standards Institute, December 22, 1988.

[12] S. Gibson, "Products to Get APPN; Mgnt Will Use CMIP," *Communication Week*, no. 396, p. 1, March 30, 1992.

[13] *The Common Information Services and Protocol over TCP/IP (CMOT)*, RFC 1095, Menlo Park, California: SCI International, Network Information Center, April 1990.

Table 1. Domains and potential subnetworks.

Domains	Types of subnets
Earth	Earth surface-to-surface via wire/fiber Earth surface-to-surface via radio Earth surface-to-surface via satellite
Space (Moon)	Earth surface-to-Moon surface (main base) Earth surface-to-Moon surface (remote)
Space (Mars)	Earth surface-to-Mars surface (main base) Earth surface-to-Mars surface (remote) Earth surface to Mars relay satellite
Moon	Moon surface-to-surface via wire/fiber (local) Moon surface-to-surface via radio (remote)
Mars	Mars surface-to-surface via wire/fiber (local) Mars surface-to-surface via radio (remote) Mars surface-to-relay satellite Mars relay satellite-to-relay satellite

Table 2. SNMP managed object groups.

Object group	Description
System	Objects that describe high-level characteristics of this network element
Interfaces	Objects associated with the network interfaces to which this network element can communicate with IP datagrams
Address translation	Translation table for converting an IP address into a subnetwork-specific (physical) address
Internet Protocol (IP)	Subgroup of objects associated with IP
Internet Control Message Protocol (ICMP)	Subgroup of objects associated with ICMP
Transmission Control Protocol (TCP)	Subgroup of objects associated with TCP
User Datagram Protocol (UDP)	Subgroup of objects associated with UDP
Exterior Gateway Protocol (EGP)	Subgroup of objects associated with EGP

Table 3. Representative objects in the SNMP Interfaces Group.

Object	Definition	Access
ifNumber	The number of network interfaces (regardless of their current state) present on this system.	Read-only
ifIndex	A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant at least from one re-initialization of the entity's network management system to the next re-initialization.	Read-only
ifDescr	A textual string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the hardware interface.	Read-only
ifType	The type of interface, distinguished according to the physical/ link protocol(s) immediately "below" the network layer in the protocol stack.	Read-only
ifMtu	The size of the largest datagram which can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.	Read-only
ifSpeed	An estimate of the interface's current bandwidth in bits per second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.	Read-only
ifAdminStatus	The desired state of the interface.	Read-Write
ifOperStatus	The current operational state of the interface.	Read-only

Table 4. SNMP PDUs.

SNMP PDUs	Description
GetRequest	At the initiation of this PDU by the sender, this PDU requests the current status of objects from the destination system.
GetResponse	In response to the GetRequest-PDU, the destination system returns the name and value of each object requested using this PDU.
GetNextRequest	This PDU is used to simplify the retrieval of successive variables in the MIB that are ordered in the form of a table (such as routing table data).
SetRequest	Upon receipt of this PDU, for each object named in the PDU, the corresponding value is assigned to the variable. The receiving station returns a GetResponse-PDU of identical form as an acknowledgment.
Trap	Upon receipt of the Trap-PDU, the data contents are passed to the application-level software for appropriate processing. Several generic traps include: (1) ColdStart Trap—the sending entity is reinitializing itself; its implementation may be altered. (2) WarmStart Trap—the sending entity is reinitializing itself; its implementation will not be altered. (3) LinkDown Trap—the sending entity recognizes a failure in one of the communication links represented in its configuration. (4) LinkUp Trap—the sending entity recognizes that one of the communication links represented in its configuration has come up.

Table 5. Management Operation Services.

Service primitive	Description
M-GET	Request the retrieval of management information from a peer CMISE-service-user. The service may be requested only in a confirmed mode, and a reply is expected.
M-SET	Request the modification of management information by a peer CMISE-service-user. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode a reply is requested.
M-ACTION	Request a peer CMISE-service-user to perform an action. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode, a reply is expected.
M-CREATE	Request a peer CMISE-service-user to create another instance of a managed object. The service may be requested only in a confirmed mode, and a reply is expected.
M-DELETE	Request a peer CMISE-service-user to delete an instance of a managed object. The service may be requested only in a confirmed mode, and a reply is expected.

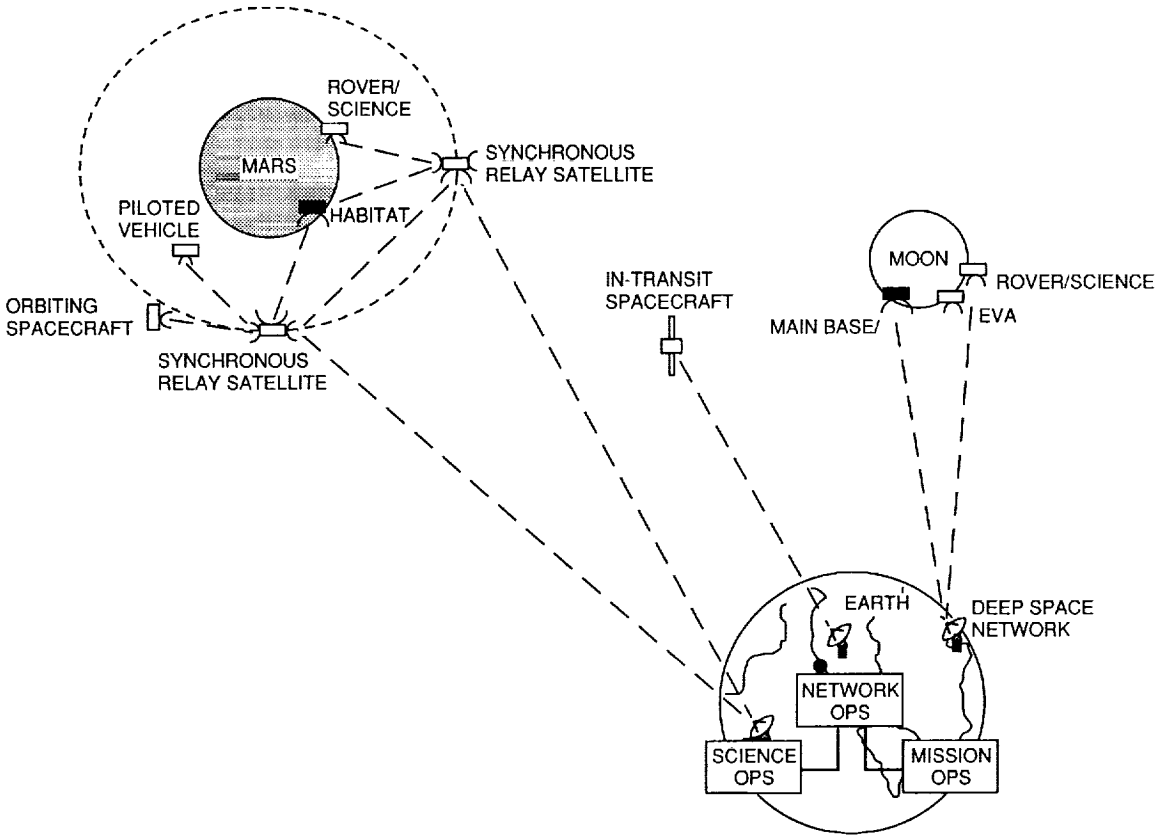


Fig. 1. Space Exploration Initiative communications architecture.

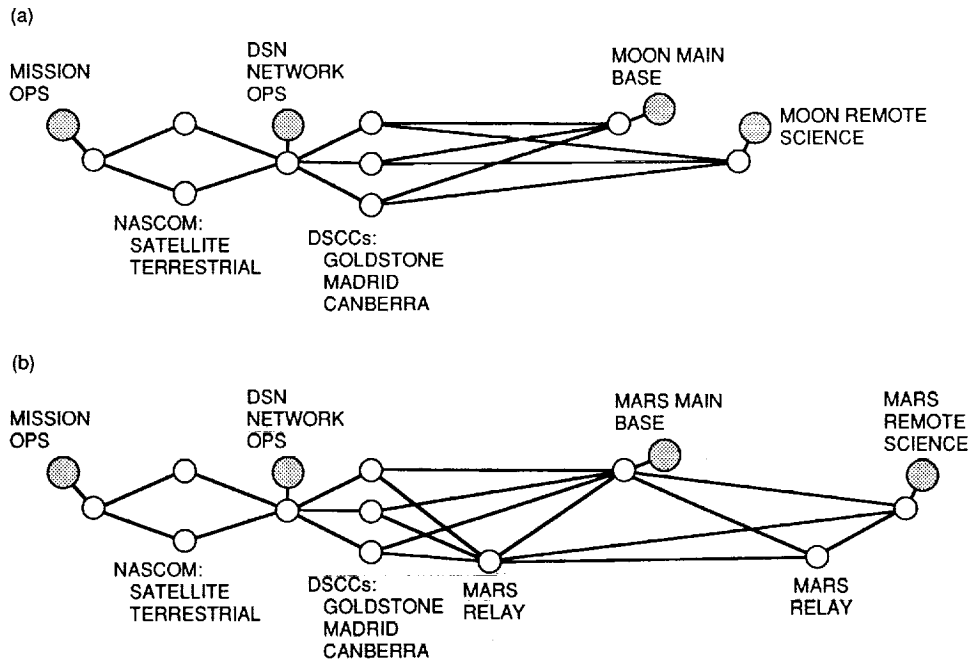


Fig. 2. Simplified schematic of primary network: (a) Moon primary space network and (b) Mars primary space network.

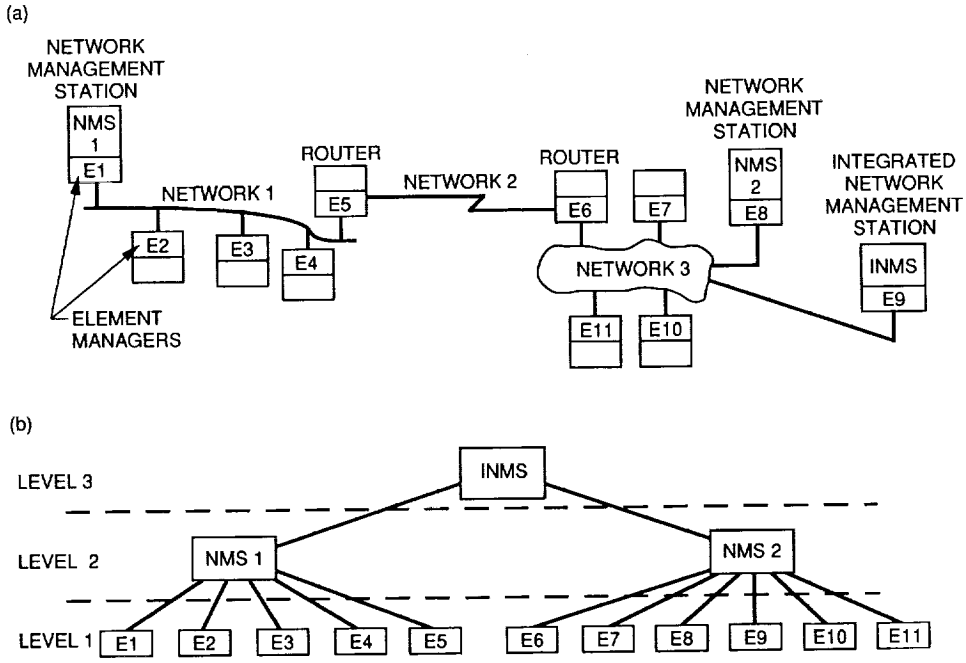


Fig. 3. Network management hierarchy: (a) physical network and (b) management hierarchy.

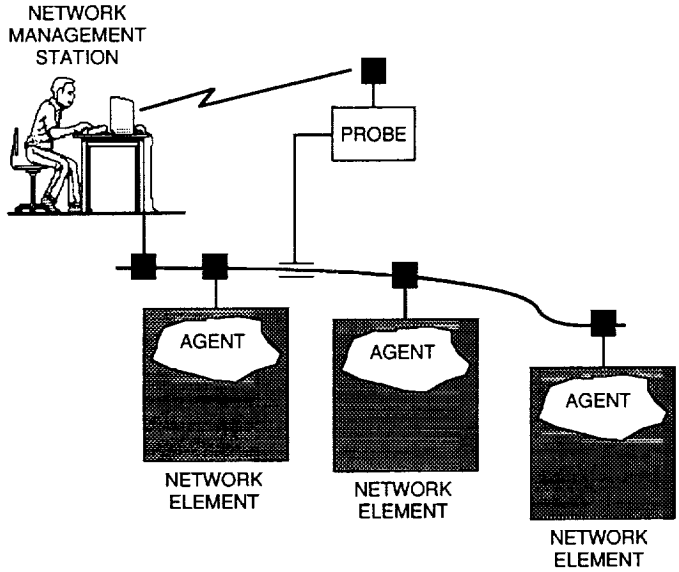


Fig. 4. SNMP architectural model.

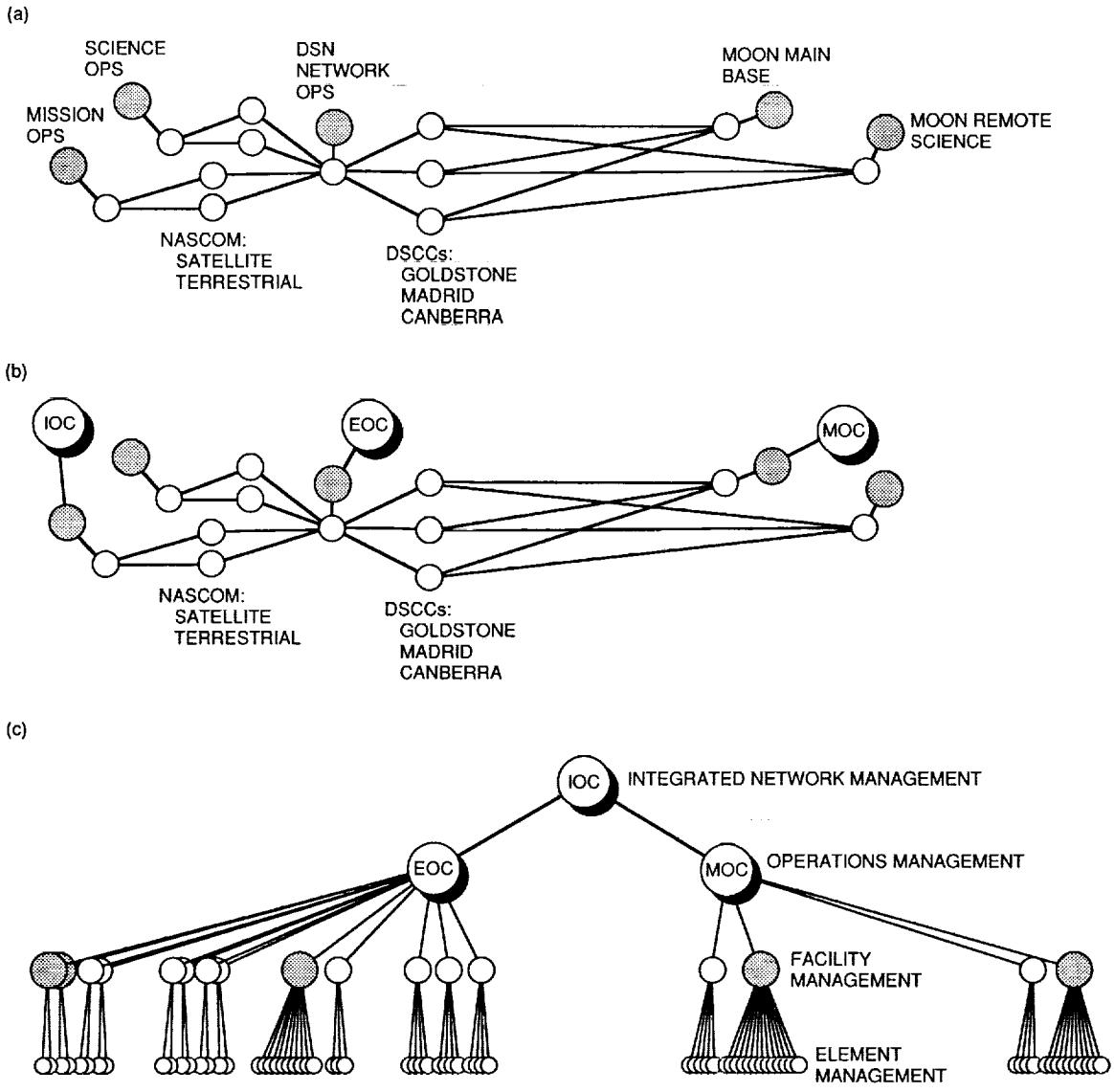


Fig. 5. Network management architecture for the Moon: (a) Moon primary space network; (b) space management network; (c) hierarchical connectivity.

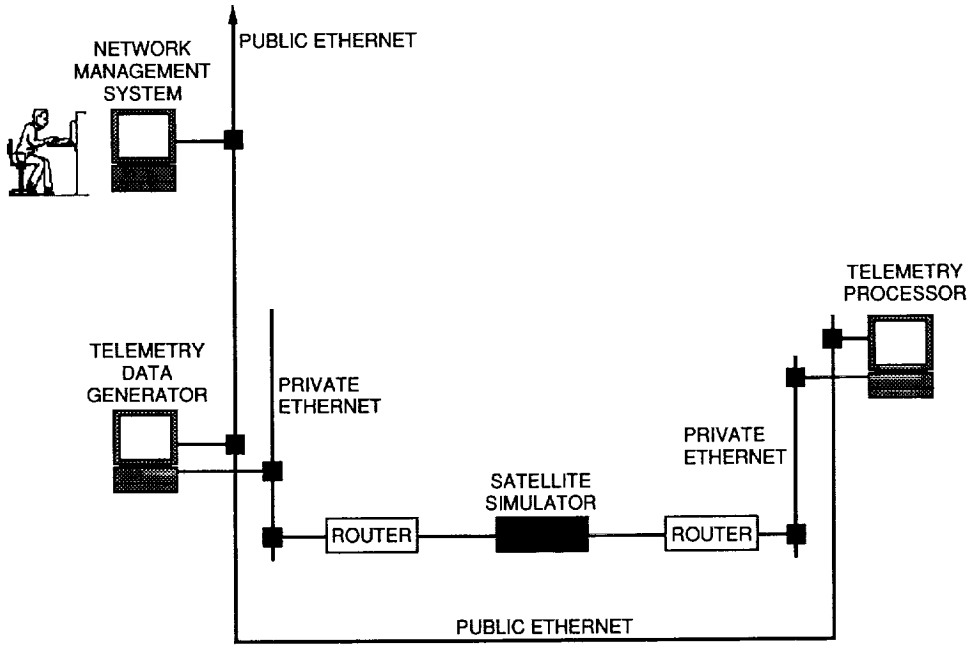


Fig. 6. DSN 2000 Information Systems Engineering Laboratory.

OMIT TO END

Errata

T. Y. Otoshi and M. M. Franco have submitted the following errata to "DSS-13 Beam Waveguide Antenna Frequency Stability" that appeared in *The Telecommunications and Data Acquisition Progress Report 42-110*, vol. April–June 1992, pp. 151–162.

In the discussion of wind direction on pages 152 and 153, the terms "front side" and "back side" of the antenna should be interchanged. Also, the phrase "blowing into the face" should be replaced with "blowing into the back."

The changes are made after discovering that the instruments at DSS 13 give wind direction as *coming from* instead of *going towards* an indicated direction. Figures 7 and 14 of the original article should be replaced by the ones that follow.

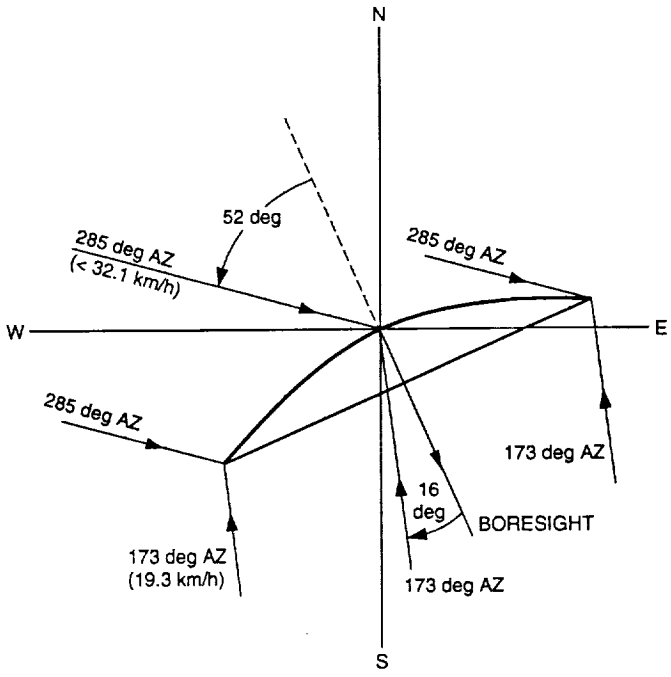


Fig. 7. Wind direction relative to the BWS antenna main axis during test with the GSTAR I geotstationary satellite.

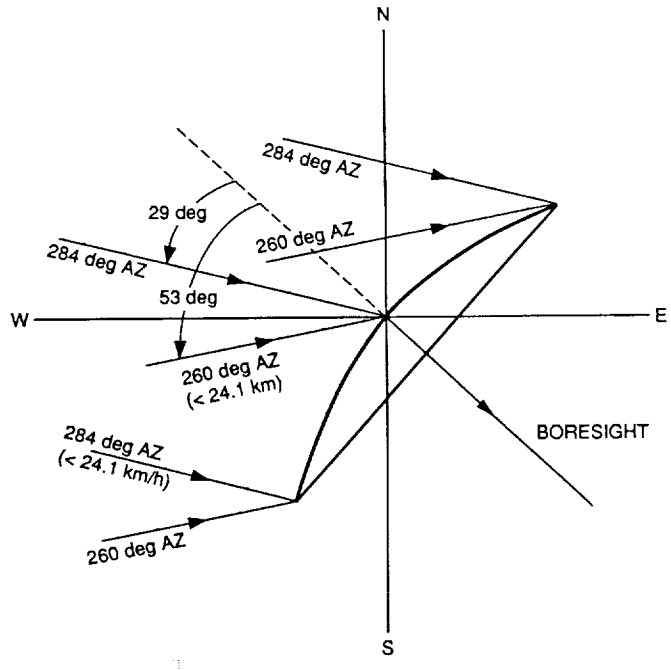


Fig. 14. Wind direction relative to the BWS antenna main axis during test with the SATCOM KI geotstationary satellite.

Author Index,[1] 1992

The Telecommunications and Data Acquisition Progress Report

42-109, January–March, May 15, 1992
42-110, April–June, August 15, 1992
42-111, July–September, November 15, 1992
42-112, October–December, February 15, 1993

[1] In the case of joint authorship, the reader is referred to the citation under the first author, where all the authors of the article are listed.