The Telecommunications and Data Acquisition Progress Report 42-121

January–March 1995

Joseph H. Yuen
Editor

May 15, 1995

NASA

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The Telecommunications and Data Acquisition Progress Report 42-121

January–March 1995

Joseph H. Yuen
Editor

May 15, 1995

National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

Note From the Editor

Since issue 42-118, published on August 15, 1994, *The Telecommunications and Data Acquisition Progress Report* has been available to readers at JPL in both printed and electronic form as a pilot program, with the goal of ultimately publishing the *TDA Progress Report* electronically. Now produced through the use of newly available software that has proven user friendly, the electronic *TDA Progress Report* has received quite favorable comments from its JPL readers. Consequently, beginning with this issue, the *TDA Progress Report* will be available electronically to all its readers on the World Wide Web at http://tda.jpl.nasa.gov/progress_report. Printed copies are also being produced, but we are considering the possibility of publishing the *TDA Progress Report* solely in electronic form sometime in the future. Readers with questions or concerns regarding this change are welcome to contact the editor.

Preface

This quarterly publication provides archival reports on developments in programs managed by JPL's Telecommunications and Mission Operations Directorate (TMOD), which now includes the former Telecommunications and Data Acquisition (TDA) Office. 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).

TMOD also performs work funded by other NASA program offices through and with the cooperation of OSC. The first of these is the Orbital Debris Radar Program funded by the Office of Space Systems Development. It exists at Goldstone only and makes use of the planetary radar capability when the antennas are configured as science instruments making direct observations of the planets, their satellites, and asteroids of our solar system. The Office of Space Sciences funds the data reduction and science analyses of data obtained by the Goldstone Solar System Radar. The antennas at all three complexes are also configured for radio astronomy research and, as such, conduct experiments funded by the National Science Foundation in the U.S. and other agencies at the overseas complexes. These experiments are either in microwave spectroscopy or very long baseline interferometry.

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

This and each succeeding issue of *The Telecommunications and Data Acquisition Progress Report* will present material in some, but not necessarily all, of the aforementioned programs.

Contents

OSC TASKS
DSN Advanced Systems
TRACKING AND GROUND-BASED NAVIGATION

COMMUNICATIONS, SPACECRAFT–GROUND

DSN Systems Implementation
NETWORK UPGRADE AND SUSTAINING

N95- 32222

Determination of the Position of Jupiter From Radio Metric Tracking of Voyager 1

W. M. Folkner
Tracking Systems and Application Section

R. J. Haw
Navigation Systems Section

The Voyager 1 spacecraft flew by Jupiter on March 5, 1979. Spacecraft navigation was performed with radio tracking data from NASA's Deep Space Network. In the years since then, there has been a great deal of progress in the definition of celestial reference frames and in determining the orbit and orientation of the Earth. Using these improvements, the radio metric range and Doppler data acquired from the Voyager 1 spacecraft near its encounter with Jupiter have been reanalyzed to determine the plane-of-sky position of Jupiter with much greater accuracy than was possible at the time of the encounter. The position of Jupiter at the time of encounter has been determined with an accuracy of 40 nrad in right ascension and 140 nrad in declination with respect to the celestial reference frame defined by the International Earth Rotation Service. This position estimate has been done to improve the ephemeris of Jupiter prior to the upcoming encounter of the Galileo spacecraft with Jupiter.

I. Introduction

Radio metric tracking data have been used since the inception of interplanetary space exploration to determine the trajectory of the robotic probes. Several analyses have been written that describe the ability of radio metric data to determine the position of interplanetary spacecraft [1–3]. The ability to determine the plane-of-sky position of spacecraft comes from the signature imposed on the spacecraft radio signal by the rotation and orbital motion of the Earth. This signature can be analyzed to determine the right ascension and declination of the spacecraft. There is also a signature in the spacecraft radio signal due to the acceleration caused by a nearby planetary body, which can be used to determine the position of the spacecraft with respect to the planetary body. The combined signatures can be used to determine the position of the planet at the time of the spacecraft encounter.

The diurnal signature in the radio metric data gives information about the spacecraft right ascension and declination with respect to the direction of the Earth's spin axis at the time of the measurement. The direction of the Earth's spin axis and the orbit of the Earth with respect to a desired inertial celestial coordinate system must be known in order to use the radio metric data to deduce the inertial coordinates of the spacecraft.

The determination of the orbit and orientation of the Earth has been a field of intensive study. The introduction of routine very long baseline interferometry (VLBI) observations in the early 1980's has enabled the definition of a celestial reference frame, defined by the positions of extragalactic radio sources, with internal consistency of about 5 nrad (e.g., see [4]). This is about a factor of 100 better than optical star catalogs previously used to define the celestial reference frame (e.g., see [5]). The orientation of the Earth is measured by VLBI with an accuracy of about 5 nrad with respect to the extragalactic radio sources. Beginning in 1988, the International Earth Rotation Service (IERS) was formed to facilitate reporting Earth orientation in a standard way. The IERS adopted a conventional celestial reference frame defined by the positions of extragalactic radio sources. Earth orientation measurements with respect to the IERS celestial reference frame are regularly distributed [6]. Since about 1970, the orbits of the Earth, Moon, and Mars have been determined with an internal accuracy of about 5 nrad from the analysis of ranging data to the Viking landers and lunar laser ranging (LLR) [7]. The LLR data can also be used to determine the orientation of the Earth with respect to the Earth's orbit. Comparison of LLR and VLBI Earth orientation has been used to determine the orientation of the Earth's orbit with respect to the IERS celestial reference frame with an accuracy of about 15 nrad [8].

The ephemerides of the outer planets have been heavily dependent on optical astrometric measurements due to a scarcity of more accurate measurements. The limited accuracy of the ground-based optical astrometric data, and the uncertainty in orientation of the optical reference frame with respect to the radio reference frame, contributed to an apparent discrepancy in the position of Jupiter of 400 km during the Ulysses spacecraft Jupiter encounter in February 1992 [9]. This discrepancy and the upcoming encounter of the Galileo spacecraft with Jupiter in December 1995 prompted a reanalysis of radio tracking data from the Voyager 1 encounter with Jupiter to provide a radio metric position of Jupiter referred to the IERS celestial reference frame.

The closest approach of the Voyager 1 spacecraft to Jupiter occurred on March 5, 1979. Shortly after the closest approach to Jupiter, the spacecraft flew within 21,000 km of Io and then within 150,000 km of Ganymede and Callisto. Navigation of Voyager 1 was performed using radio range and Doppler measurements by the Deep Space Network and by using images of the satellites of Jupiter against background stars taken by the onboard camera [10,11]. The Voyager 1 navigation provided a determination of the Earth–Jupiter range at the time of encounter[1] and data for the improvement of the ephemerides of the satellites of Jupiter [12]. However, the large uncertainty of the orientation of the Earth with respect to the Earth's orbit at that time prevented a useful improvement in the plane-of-sky position of Jupiter. A reanalysis of the Voyager 1 radio tracking data, based on the previous work of the Voyager 1 navigation team and with updated models for the orbit and orientation of the Earth, has been performed to determine the right ascension and declination of Jupiter at the time of the Voyager 1 encounter.

II. Method

Two-way Voyager 1 tracking data were acquired by an antenna from the Deep Space Network transmitting a signal to the spacecraft at a frequency near 2.1 GHz (S-band) with the spacecraft receiving and coherently retransmitting the signal to Earth at 2.3 GHz or 8.4 GHz (X-band). The data employed for the reanalysis spanned 32 days, ending a few hours after the closest approach to Jupiter and before the encounter with Io. Doppler measurements were made by comparing the frequency of the received carrier with the transmitted carrier at the DSN antenna. Range measurements were made by determining the delay between the time of transmission of a range code (a set of coherent tones about the carrier) and the time of reception of the retransmitted range code. The dominant noise on the measurements was due to variations in the charged particle distribution between Earth and the spacecraft, mostly due to solar plasma. For much of the time, Voyager 1 transmitted coherent signals at both 2.3 and 8.4 GHz. For the reanalysis, only dual-band downlink data were used. Because the charged particle effects are proportional

[1] J. K. Campbell, "Earth–Jupiter Range Fixes From Voyager," JPL Interoffice Memorandum 314.8-351 (internal document), Jet Propulsion Laboratory, Pasadena, California, 1982.

to the inverse of the square of the carrier frequency, the dual-band downlink provides a measure of the charged particle effects on the downlink signal. By interpolating the charged particle effects to the time of the uplink, it was possible to remove most of the effect on the tracking data. At the beginning of a tracking pass, there are no dual-band downlink measurements near the time of the uplink signal, so larger residuals are expected for the first 75 minutes (one round-trip light time) of each tracking pass.

The spacecraft trajectory was integrated from initial position and velocity conditions using models for the dynamic forces on the spacecraft. The modeled gravitational forces on the spacecraft were due to the masses of the Sun and planets, the Galilean satellites, and the oblateness of Jupiter. The relative locations of the Sun and planets were based on the JPL ephemeris labeled DE200 [13] but rotated so that the orbit of the Earth had the correct orientation with respect to the IERS celestial reference frame at the time of encounter [8]. The positions of the Galilean satellites were given by Lieske [12]. The masses of the Jovian system and the oblateness of Jupiter are given by Campbell and Synnott [11]. Other modeled forces were solar radiation pressure and thruster firings.

The Voyager 1 spacecraft is three-axis stabilized using unbalanced thrusters. Because of torques acting on the spacecraft (mainly due to solar pressure), the thrusters repeatedly fire to maintain a specified orientation. These thruster firings produce small velocity changes to the spacecraft trajectory. Changes in the orientation of the spacecraft caused a change in the torque on the spacecraft and a change in the pattern of the thruster firings. Information about the thruster firings was encoded in the spacecraft telemetry stream, but this information was imperfect. Instead of relying on the incomplete telemetry information, the magnitudes of the thruster firings were estimated using two models. Constant accelerations were estimated while the spacecraft was in a fixed attitude, to approximate the nearly constant thruster firings needed to maintain the attitude. Impulsive maneuvers were estimated for larger events associated with changes in the spacecraft orientation. In addition, there was one larger impulsive maneuver 12.5 days before Jupiter encounter to correct the spacecraft trajectory. Table 1 gives the acceleration and maneuver times included in the reanalysis. Some information about the history of the spacecraft orientation is no longer available, so some of the events in Table 1 were inferred from an examination of the tracking data. In principle, the only consequence of estimating too many maneuvers and accelerations is to weaken the solution.

Table 1. Modeled thruster firing times.

Maneuver time, 1979	Acceleration start time, 1979
February 4, 00:00	February 1, 00:00
February 5, 12:00	February 4, 08:30
February 9, 04:02	February 5, 12:00
February 17, 00:00	February 9, 04:00
February 18, 18:00	February 11, 02:00
February 19, 00:00	February 15, 00:00
February 21, 03:58	February 17, 15:00
March 1, 23:00	February 19, 05:00
March 3, 20:00	February 21, 18:00
	March 4, 00:00

Computed values for the tracking measurements were derived from nominal values for the spacecraft epoch state, force models, inertial Deep Space Station locations, and calibration for propagation delays due to Earth troposphere [14]. A least-squares fit to the observed minus computed measurement values was made to estimate model parameters. The estimated parameters included the spacecraft initial state, the position of Jupiter, the direction of Jupiter's spin axis, a range bias for each DSN antenna, and parameters to describe the thruster firings. Locations for the stations of the DSN were consistent with the IERS terrestrial reference frame [15]. The station locations were mapped from Earth-fixed locations to inertial space using models for precession, nutation, and solid Earth tides, and calibrations for polar motion and length-of-day variations and corrections to the standard nutation model in the manner defined by the IERS.

The estimated uncertainty for the spacecraft trajectory depended on assumed a priori uncertainties for the estimated parameters, the assumed data arc and data weights, and a priori uncertainties for model parameters that are not estimated. The effect of uncertainties of nonestimated model parameters is included through the use of consider analysis [16]. The assumed a priori information for estimated and consider parameters is summarized in Table 2. The a priori uncertainties for spacecraft initial state were large enough to leave it essentially unconstrained. The thruster firing uncertainty levels were based on the level of variation as recorded by the telemetry information [10] and by checking that the estimated corrections to the acceleration were significantly smaller than the a priori uncertainty. The uncertainties in the position of Jupiter and in the Jupiter spin axis direction were set large enough to not influence the solution. Because range calibrations were not recovered for the reanalysis, the DSN range bias uncertainties were set to a value corresponding to the total delay through the ground station. DSN station locations are currently known with about a 3-cm accuracy [15], but because of uncertainty in the rate of change of station locations due to plate tectonics, this was increased to a 10-cm uncertainty for the 1979 encounter data (and was large enough to include uncertainties in Earth orientation). The uncertainty in the orientation of the Earth's orbit comes from the comparison of VLBI and LLR Earth orientation [8]. The uncertainty in the troposphere calibration is taken from Robinson.[2] The uncertainties in the mass and oblateness of Jupiter's gravity field are given by Campbell and Synnott [11].

III. Results

Figures 1 and 2 show the post-fit data residuals. Some small signatures can be seen in the Doppler data in Fig. 1. These are most apparent at the beginning of tracking passes and are probably due to residual solar plasma effects. The Doppler residuals have a root-mean-square (rms) of 0.1 mm/s. Most of the data points have averaging times much longer than the standard 60 s. If the data noise is assumed to be white-frequency noise, then the Doppler data residuals correspond to an rms of 0.3 mm/s for a 60-s averaging time. The solar plasma is known to impose more noise on the Doppler data at low frequencies [17], so for the final estimate, the Doppler data were conservatively weighted at 1-mm/s uncertainty for a 60-s count time, even though the solar plasma was partially calibrated. The conservative weighting of the Doppler data prevents the small signatures in the Doppler data from excessively influencing the solution estimates and increases the formal uncertainty. The range data have an rms of 3.2 m and were weighted at 4 m in the solution.

Tables 3 and 4 give the estimated position of the barycenter of the Jupiter system at a time near the closest approach of the Voyager 1 spacecraft in Cartesian and spherical coordinates. Because Jupiter is within the solar system, the light time significantly affects the apparent position of Jupiter. To avoid complications of light-time calculation, time transformations, and other effects, Tables 3 and 4 give the instantaneous Earth–Jupiter vector in the IERS celestial reference frame. That is, the Earth–Jupiter vector is the difference between the position of Jupiter at the specified solar-system barycentric coordinate time (TDB) and the position of the Earth at the same coordinate time. For reference, the Earth–Jupiter vector is also given in the widely available ephemeris DE200.

[2] S. E. Robinson, "Errors in Surface Model Estimates of Zenith Wet Path Delays Near DSN Stations," JPL Interoffice Memorandum 335.4-594 (internal document), Jet Propulsion Laboratory, Pasadena, California, 1986.

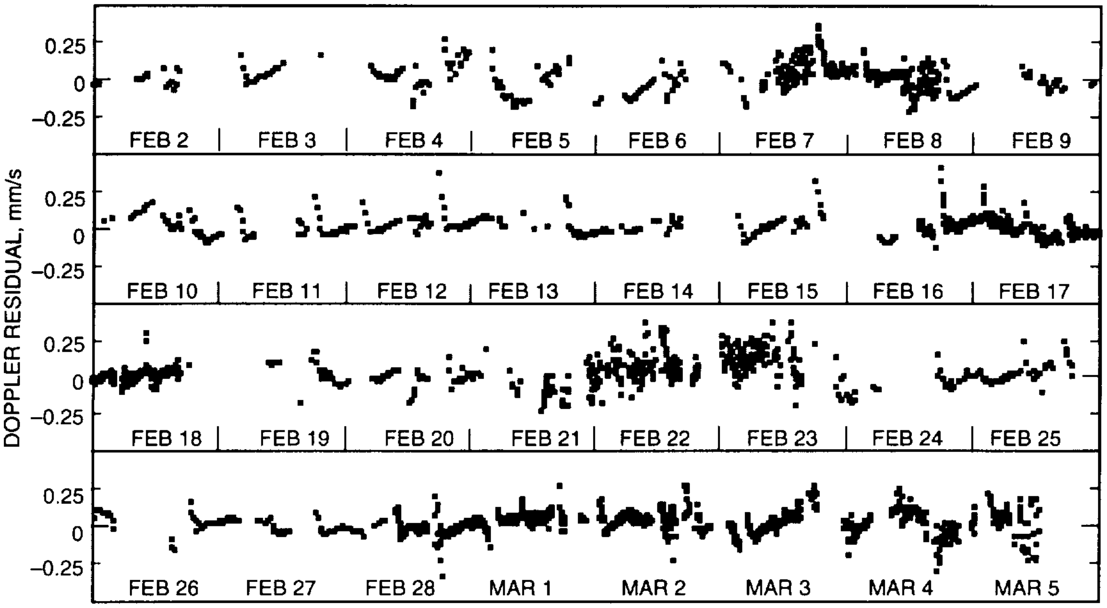


Fig. 1. Voyager 1 S-band Doppler data residuals.

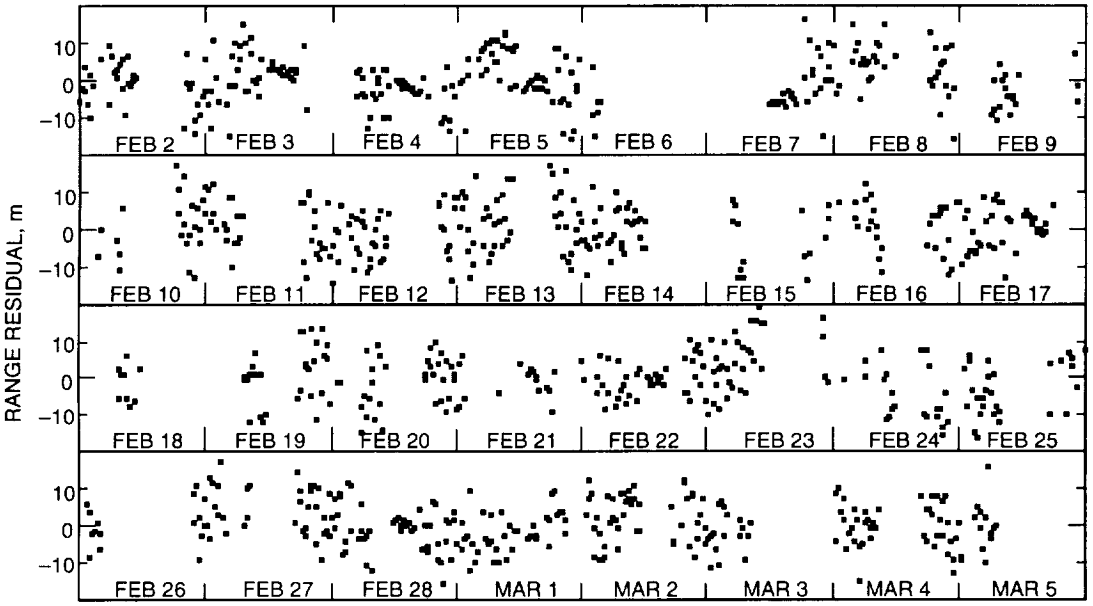


Fig. 2. Voyager 1 S-band range residuals.

The uncertainties in Table 4 correspond to 40 nrad in right ascension and 140 nrad in declination. The given uncertainties are expected to reflect the actual uncertainties as realistically as possible. The actual uncertainties are dependent on the spacecraft thruster firing history, which cannot be easily reconstructed at this late date. As a check for errors in modeling assumptions, separate fits were made using only the first 16 days of data within the arc and with only the last 16 days of data. In each case, the estimated position of Jupiter agreed with the value given in Table 3 within 1 sigma. The uncertainty in the Earth–Jupiter range is due to not having the ranging system calibrations available for the reanalysis. The

Table 2. Estimated and considered parameters and their uncertainties.

Estimated parameters	Uncertainty
Spacecraft initial position	10^5 km
Spacecraft initial velocity	100 km/s
Impulsive maneuvers (each component)	1 cm/s
Thruster accelerations (each component)	10^{-11} km/s^2
Jupiter right ascension	500 nrad
Jupiter declination	500 nrad
Earth–Jupiter range	100 km
Jupiter spin axis, right ascension	0.1 deg
Jupiter spin axis, declination	0.1 deg
DSN range biases	3 km
Consider parameters	**Uncertainty**
DSN station locations	10 cm
Earth orbit orientation with respect to IERS frame	15 nrad
Troposphere zenith delay	4 cm
Jupiter mass (GM)	100 km^3/s^2
Jupiter oblateness (J2)	0.01 percent

Table 3. Cartesian coordinates of Jupiter on March 5, 1979, 12:00:00.000 TDB.

Position	x, km	y, km	z, km
Estimated position	−339109994	536319388	241482423
Position in DE200	−339110282	536319389	241481691

Table 4. Spherical coordinates of Jupiter on March 5, 1979, 12:00:00.000 TDB.

Position	Range, km	Right ascension	Declination
Estimated position	678931392 ± 3	8 h 9 min 13.1531 s ± 0.0005 s	20° 50′ 6.487″ ± 0.028″
Position in DE200	678931276	8 h 9 min 13.1584 s	20° 50′ 6.262″

right ascension and declination estimated for Jupiter are more accurate than any other measurements except for the VLBI data taken from the Ulysses spacecraft [18]. The only other position measurement with comparable accuracy is from observations of the satellites of Jupiter with the Very Large Array, which determined the position of Jupiter with an accuracy of 125 nrad in right ascension and declination [19]. The Voyager 1 position determination will make a significant contribution to determining the ephemeris of Jupiter prior to Galileo's encounter in December 1995.

Acknowledgments

This work was made possible by the diligent efforts of George Lewis in recovering and archiving the Voyager tracking data and by the efforts of the Voyager Navigation Team, especially Jim Campbell. The authors would like to thank Myles Standish, Tony Taylor, and Jim Border for helpful discussions.

References

[1] D. W. Curkendall and S. R. McReynolds, "A Simplified Approach for Determining the Information Content of Radio Tracking Data," *J. Spacecraft*, vol. 6, pp. 520–525, 1969.

[2] T. W. Hamilton and W. G. Melbourne, "Information Content of a Single Pass of Doppler Data From a Distant Spacecraft," *Space Programs Summary 37-39*, vol. III, Jet Propulsion Laboratory, Pasadena, California, pp. 18–23, May 31, 1966.

[3] J. O. Light, "An Investigation of the Orbit Redetermination Process Following the First Midcourse Maneuver," *Space Programs Summary 37-33*, vol. IV, Jet Propulsion Laboratory, Pasadena, California, pp. 8–16, June 30, 1965.

[4] J. L. Fanselow, O. J. Sovers, J. B. Thomas, G. H. Purcell, Jr., E. J. Cohen, D. H. Rogstad, L. J. Skjerve, and D. J. Spitzmesser, "Radio Interferometric Determination of Source Positions Utilizing Deep Space Network Antennas—1971 to 1980," *Astron. J.*, vol. 89, pp. 987–998, 1984.

[5] C. Ma, D. B. Shaffer, C. de Vegt, K. J. Johnston, and J. L. Russell, "A Radio Optical Reference Frame, I. Precise Radio Source Positions Determined by Mark III VLBI: Observations From 1979 to 1988 and a Tie to the FK5," *Astron. J.*, vol. 99, pp. 1284–1298, 1990.

[6] International Earth Rotation Service, *Annual Report for 1988*, Observatoire de Paris, Paris, France, 1989.

[7] E. M. Standish, Jr. and J. G. Williams, "Dynamical Reference Frames in the Planetary and Earth–Moon Systems," *Inertial Coordinate Systems on the Sky*, edited by J. H. Lieske and V. K. Abalakin, Dordrecht, Netherlands: Kluwer Academic, pp. 173–181, 1990.

[8] W. M. Folkner, P. Charlot, M. H. Finger, J. G. Williams, O. J. Sovers, XX Newhall, and E. M. Standish, "Determination of the Extragalactic-Planetary Frame Tie From Joint Analysis of Radio Interferometric and Lunar Laser Ranging Measurements," *Astron. Astrophys.*, vol. 287, pp. 279–289, 1993.

[9] T. McElrath, B. Tucker, P. Menon, E. Higa, and K. Criddle, "Ulysses Navigation at Jupiter Encounter," AIAA Paper 92-4524, presented at the AIAA/AAS Astrodynamics Conference, Hilton Head, South Carolina, 1992.

[10] J. K. Campbell, S. P. Synnott, J. E. Riedel, S. Mandell, L. A. Morabito, and G. C. Rinker, "Voyager 1 and Voyager 2 Jupiter Encounter Orbit Determination," AIAA Paper 80-0241, presented at the AIAA 18th Aerospace Sciences Meeting, Pasadena, California, 1980.

[11] J. K. Campbell and S. P. Synnott, "Gravity Field of the Jovian System from Pioneer and Voyager Tracking Data," *Astron. J.*, vol. 90, pp. 364–372, 1985.

[12] J. H. Lieske, "Improved Ephemerides of the Galilean Satellites," *Astron. Astrophys.*, vol. 82, pp. 340–348, 1980.

[13] E. M. Standish, Jr., "Orientation of the JPL Ephemerides DE200/LE200 to the Dynamical Equinox of J2000," *Astron. Astrophys*, vol. 114, pp. 297–302, 1982.

[14] C. C. Chao, *The Troposphere Calibration Model for Mariner Mars 1971*, JPL Technical Report 32-1587, Jet Propulsion Laboratory, Pasadena, California, pp. 61–76, 1974.

[15] C. Boucher, Z. Altamimi, and L. Duhem, *Results and Analysis of the ITRF93*, IERS Technical Note 18, Observatoire de Paris, France, 1994.

[16] G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation*, New York: Academic Press, 1977.

[17] J. W. Armstrong, R. Woo, and F. B. Estabrook, "Interplanetary Phase Scintillation and the Search for Very Low Frequency Gravitational Radiation," *Astrophys. J.*, vol. 230, pp. 570–574, 1979.

[18] W. M. Folkner and T. P. McElrath, "Determination of Radio-Frame Position for Earth and Jupiter From Ulysses Encounter Tracking," AAS Paper 93-167, AAS/AIAA Spaceflight Mechanics Meeting, Pasadena, California, 1993.

[19] D. O. Muhleman, G. L. Berge, D. J. Rudy, A. E. Niell, R. P. Linfield, and E. M. Standish, "Precise Position Measurements of Jupiter, Saturn, and Uranus Systems With the Very Large Array," *Celestial Mechanics*, vol. 37, pp. 329–337, 1985.

Rate Considerations in Deep Space Telemetry

M. Costa, M. Belongie, and F. Pollara
Communications Systems Research Section

The relationship between transmission rate and source and channel signal-to-noise ratios (SNRs) is discussed for the transmission of a Gaussian source over a binary input, additive Gaussian channel, with a mean-squared distortion criterion. We point out that for any finite rate, and sufficiently high channel SNR, the fidelity criterion (reproduction SNR) is upper bounded by a function of the transmission rate. Thus, the performance becomes rate limited rather than power limited. This effect is not observed with the binary symmetric source, the binary-input Gaussian channel combination, or the Gaussian source, unconstrained-input Gaussian channel combination.

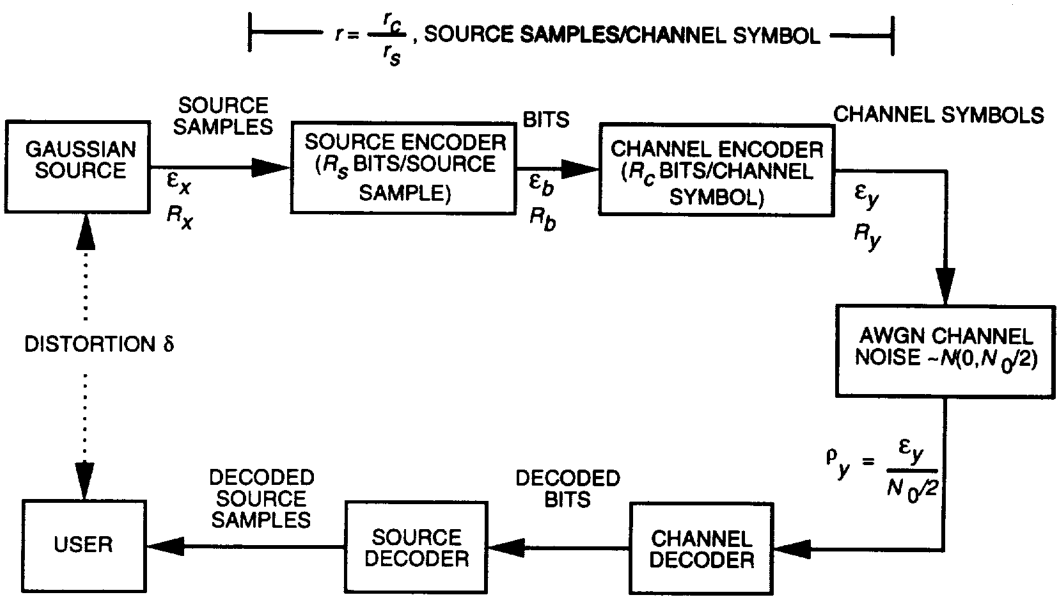
I. Introduction

The deep space communication channel uses binary phase shift keying (BPSK) modulation and is well modeled as a binary input, additive white Gaussian noise (AWGN) channel model. It is usually accepted that there is no bandwidth constraint in deep space communication application and that, for sufficiently wide bandwidth usage, the full benefit of unconstrained bandwidth is essentially realized. While these notions are correct, they must be viewed with caution. It does not necessarily follow that, for sufficiently low overall transmission rate, there is little to be gained by further decreasing the rate. The interplay between source and channel coding and the issue of coding complexity need to be considered. Depending on the telemetry source and the available channel signal-to-noise ratio (SNR), there may be a significant advantage in further decreasing the rate.

In this article, we review these notions in the context of a deep space communication system with an independent identically distributed (i.i.d.) Gaussian source and a conventional BPSK, power-limited channel, using mean-squared error (MSE) as a distortion criterion. While not an accurate model for most deep space telemetry sources, the white Gaussian source is a useful reference model. Typical telemetry data can be transformed by an (approximately) decorrelating orthogonal transformation, such as the discrete cosine transform, producing data that can be approximated by parallel sources with white (generalized) Gaussian distributions of different variances, one for each transform coefficient. Thus, the combined source and channel coding of a white Gaussian source for transmission over the deep space channel is a relevant exercise.

II. Preliminaries

The well-known equations governing transmission rate and source and channel SNRs were established by Shannon in his seminal 1948 articles [1]. We refer to [2] as a source of notation. Figure 1 shows the system under consideration.



$r = \frac{r_c}{r_s}$, SOURCE SAMPLES/CHANNEL SYMBOL
GAUSSIAN SOURCE
SOURCE SAMPLES
$\mathcal{E}_x$
R_x
SOURCE ENCODER (R_s BITS/SOURCE SAMPLE)
BITS
$\mathcal{E}_b$
R_b
CHANNEL ENCODER (R_c BITS/CHANNEL SYMBOL)
CHANNEL SYMBOLS
$\mathcal{E}_y$
R_y
AWGN CHANNEL NOISE ~$N(0,N_0/2)$
$\rho_y = \frac{\mathcal{E}_y}{N_0/2}$
DISTORTION δ
USER
DECODED SOURCE SAMPLES
SOURCE DECODER
DECODED BITS
CHANNEL DECODER

Fig. 1. Communication system model.

The capacity of a binary-input AWGN channel is given by

$$C(\rho_y) = 1 - E_u\left[\log_2(1 + e^{-2u})\right] \tag{1}$$

where $\rho_y = 2\mathcal{E}_y/N_0$, $\mathcal{E}_y$ is the available energy per channel symbol, $N_0/2$ is the two-sided noise spectral density, and E_u denotes expectation over u, a random variable with distribution $N(\rho_y, \rho_y)$.

The rate distortion function for an i.i.d. Gaussian source is given by

$$R(\delta) = \frac{1}{2}\log_2\left(\frac{1}{\delta}\right) \tag{2}$$

where δ is the normalized MSE distortion. The reproduction SNR (RSNR) is given by $1/\delta$.

III. Discussion

There are three variables of interest in this communication problem. They are

(1) δ, the normalized MSE distortion of reproduction at the receiver

(2) ρ_x, the available channel SNR, given by $\rho_x = 2\mathcal{E}_x/N_0$

(3) r, the overall transmission rate, measured in source samples per channel use

These quantities must satisfy the inequality

$$C(r\rho_x) \geq rR(\delta) \tag{3}$$

If the coding procedure is divided into a cascade of source and channel encoders, where the source is first converted into a string of binary symbols, the rate r satisfies

$$r = \frac{r_c}{r_s} = \frac{R_x}{R_y} \tag{4}$$

where r_s is the source code rate measured in bits per source sample, r_c is the channel code rate in information bits per channel use, R_x is the source rate in samples per second, and R_y is the channel rate in channel uses per second. Considering that each bandwidth unit (Hertz) corresponds, by the Nyquist sampling theorem, to two dimensions (channel uses) per second, we relate the bandwidth B to R_y by $B = R_y/2$.

Other channel SNRs of interest are ρ_b and ρ_y, the signal-to-noise ratios available per information bit and per channel use, respectively. We have selected ρ_x for our considerations because it is desirable to compare transmission schemes that use the same power and time to transmit each source sample. These three SNRs are related by $r\rho_x = r_c\rho_b = \rho_y$.

Substituting Eqs. (1) and (2) in Eq. (3), we can obtain the fundamental bound on RSNR given r and ρ_x:

$$\frac{r}{2}\log_2\left(\frac{1}{\delta}\right) \le 1 - E_u\left[\log_2(1+e^{-2u})\right] \tag{5}$$

where the distribution of u is now expressed as $N(r\rho_x, r\rho_x)$. This bound is depicted in Fig. 2, where we present plots of RSNR versus $\mathcal{E}_x/N_0$ for different values of overall rate r. (We use $\mathcal{E}_x/N_0$ instead of ρ_x in all the figures for consistency with [2] and other articles.)

In the limit as $r \to 0$, Eq. (5) becomes

$$\frac{1}{\delta} \le e^{\rho_x} \tag{6}$$

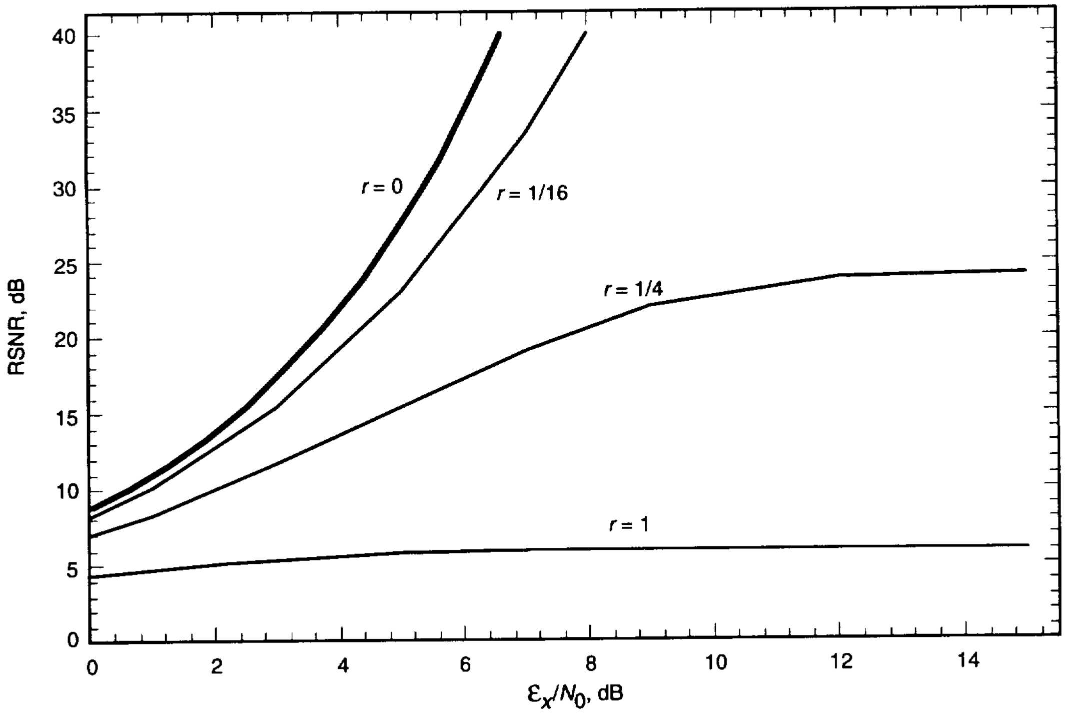


Fig. 2. Bounds on performance for a binary input channel with fixed *r*.

Thus, as ρ_x increases without bound, RSNR also may increase without bound. To increase ρ_x, one needs to alter the source transmission rate or the available power P. We have $\rho_x = P/R_x$. Thus, ρ_x can be increased by reducing the source rate R_x. This in turn affects the overall rate, since $r = R_x/R_y = R_x/2B$. Alternatively, ρ_x can be increased with an increase in P.

The noted unbounded growth in RSNR only occurs in the limit as $r \rightarrow 0$. For any positive value of r, the upper bound on RSNR approaches a finite limit as ρ_x increases. This occurs when ρ_x is large enough to make the channel essentially noiseless. Since the channel is restricted to binary input, its capacity is upper bounded by 1-bit-per-channel use. Thus, the RSNR is upper bounded by a function of the overall rate: $1/\delta < 2^{(2/r)}$. Since this bound can be arbitrarily smaller than the bound that prevails in the limit as $r \rightarrow 0$, Eq. (6), it is clear that the performance can greatly benefit from a decrease in overall rate (or an increase in bandwidth when R_x is held constant).

As shown in [3], the binary input AWGN channel has essentially the same performance as the unconstrained power-limited AWGN channel for low enough overall rates (e.g., less than 0.3 bit/channel use) when used to communicate a binary symmetric source. Interestingly, the same observation cannot be made for the case of communicating a Gaussian random variable, except in the limit as $r \rightarrow 0$. For any positive value of r, which suggests a finite level of complexity, and sufficiently high ρ_x, the binary input channel will have its performance (RSNR) limited by rate rather than by power. This effect is not observed in the unconstrained input AWGN case, where, for a fixed arbitrary rate, the upper bound on RSNR grows to ∞ as $\rho_x \rightarrow \infty$. Figure 3 compares, for various values of r, the unconstrained input and binary input cases. (The dotted lines are asymptotes.)

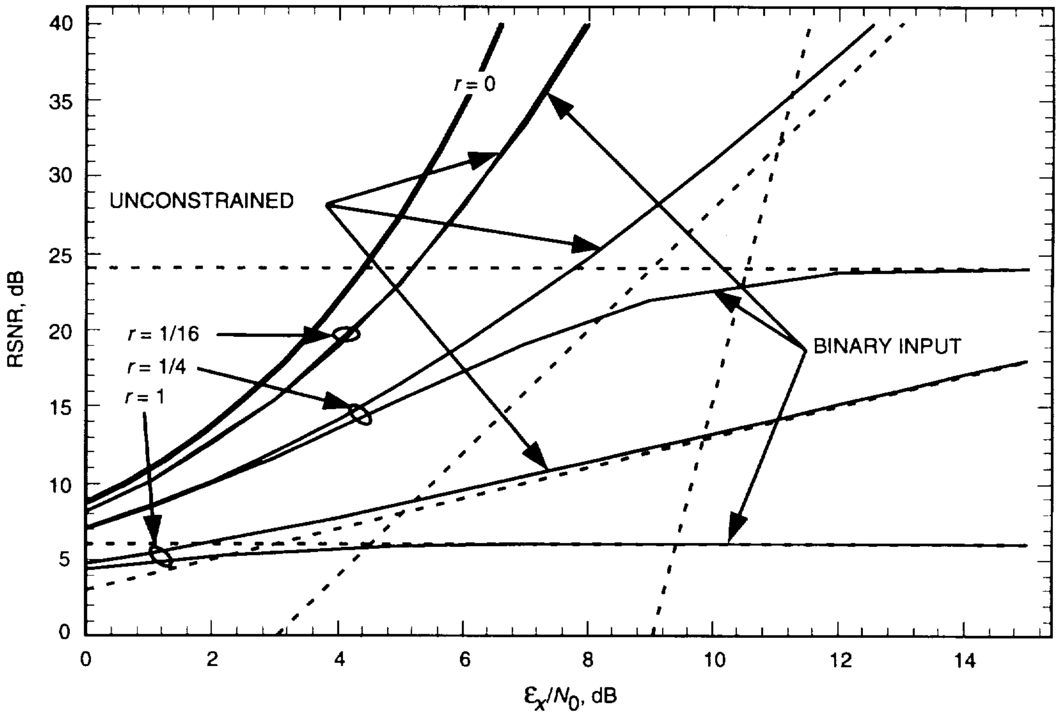


Fig. 3. Comparison of binary input channel and unconstrained channel for fixed *r*.

IV. Applicability

Under what circumstances might there be a lower bound on the overall rate r? This is a complicated issue, but we can make a few observations. First, any real system must have some nonzero value of r. Second, r clearly has some relationship to complexity, because $r = r_c/r_s$, and both lower-rate channel codes and higher-rate source codes generally imply higher complexity. Thus, a constraint on r can be seen as a constraint on overall complexity. However, we can also consider the two components, r_s and r_c, separately. Fixing r_s explicitly puts an upper bound on RSNR, resulting in the bounds shown in Fig. 4. For this case, there is no difference between the unconstrained and binary input channels. Fixing r_c results in curves as shown in Fig. 5. Although a difference is seen between the unconstrained and binary input channels, the curves all have the same exponential shape. So, the interesting phenomenon described for fixed values of r (i.e., the different limiting behavior for binary input and unconstrained channels) depends on a simultaneous bound on r_s and r_c by fixing their ratio.

To see what implications this phenomenon might have, we must consider for which combinations of r, RSNR, and ρ_x it occurs. For a fixed value of r, the intercept of the asymptotes, as illustrated in Fig. 3, is approximately where the effect becomes significant. This intercept occurs at $\delta = 2^{-2/r}$ and $\rho_x = 4/r$. So, for instance, if $r = 1/4$, the effect becomes significant for RSNR > 24 dB and $\rho_x > 9$ dB. While these SNRs are certainly within the range of interest, it is hard to imagine reasonable circumstances requiring $r \geq 1/4$. For $r = 1/16$, which is known to be quite feasible for deep space communication, the effect becomes significant for RSNR > 96 dB and $\rho_x > 15$ dB. These SNRs are probably outside the range of interest of most missions.

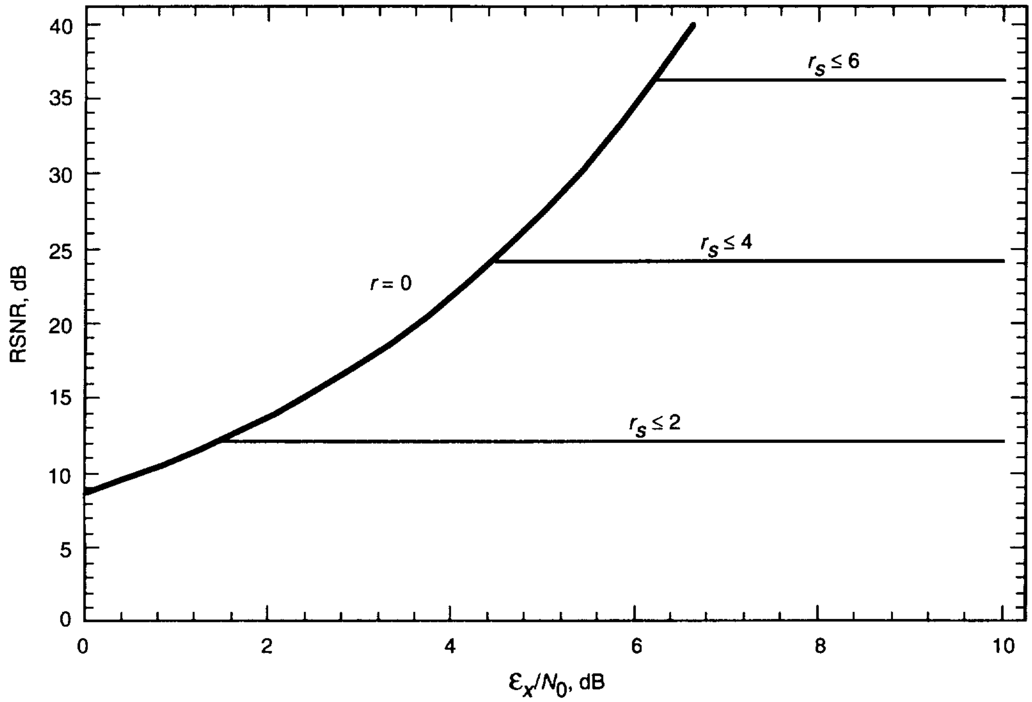


Fig. 4. Bounds on performance for binary input channel or unconstrained channel with r_s limited.

V. Performance Bounds With Fixed Channel SNR

Complexity is not the only reason that $r = 0$ is impossible. For a fixed ρ_x, $r \rightarrow 0$ implies $\rho_y \rightarrow 0$. Thus, even if the computational complexity of a very low-rate channel code or very high-rate source code is not a concern, the low SNR of the channel symbols might be. Although in theory ρ_y can be arbitrarily small as long as $C(\rho_y) > rR(\delta)$, in practice there is a lower bound on ρ_y below which any given receiver cannot perform symbol synchronization. Performance curves at constant ρ_y are shown in Fig. 6 for both the unconstrained and binary input channels. Since the curves are all exponential, we see that the differing behavior between the unconstrained and binary input channels for fixed values of r is not due to a bound on ρ_y. It can also be seen from Fig. 6 that the performance difference between the unconstrained and binary input channels is negligible for $\rho_y < 0$ dB.

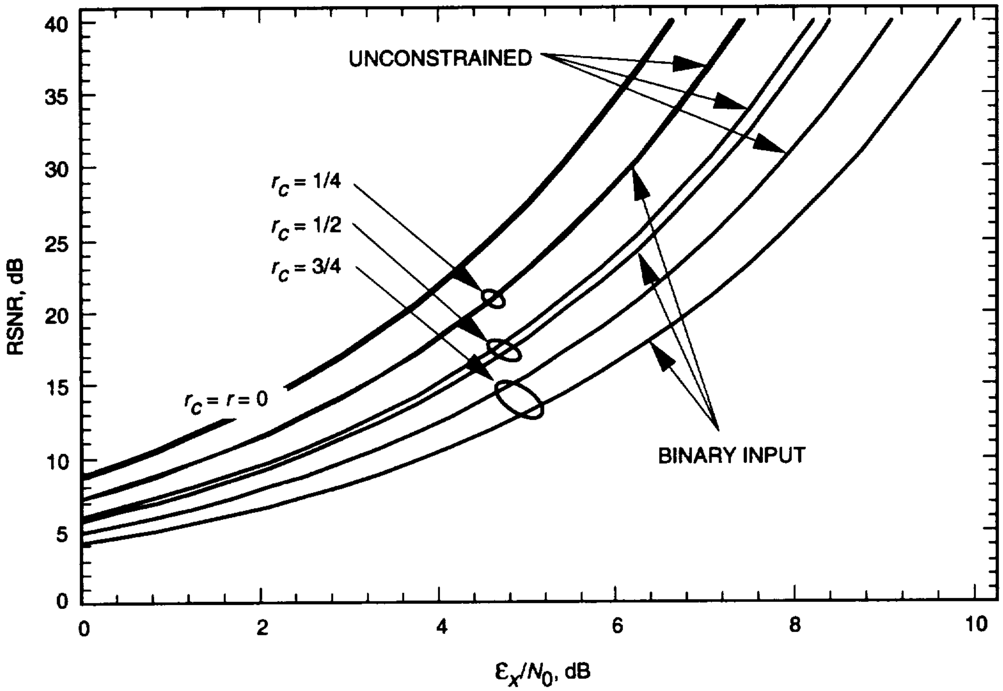


Fig. 5. Bounds on performance for binary input channel and unconstrained channel with fixed r_c.

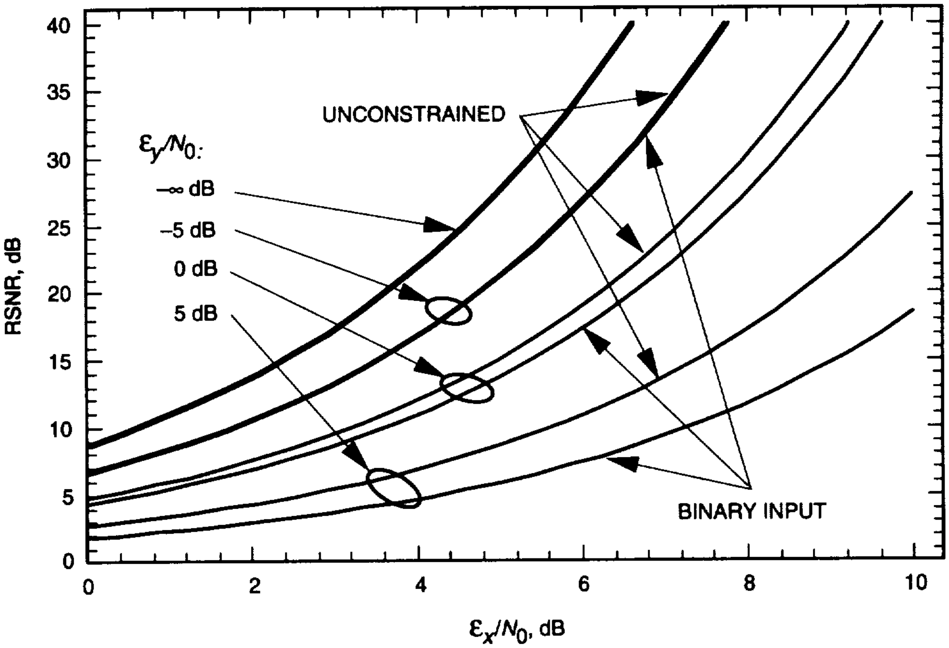


Fig. 6. Bounds on performance for binary input channel and unconstrained channel with fixed $\mathcal{E}_y/N_0$.

References

[1] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, 1948.

[2] S. J. Dolinar and F. Pollara, "The Theoretical Limits of Source and Channel Coding," *The Telecommunications and Data Acquisition Progress Report 42-102, April–June 1990*, Jet Propulsion Laboratory, Pasadena, California, pp. 62–72, August 15, 1990.

[3] S. A. Butman and R. J. McEliece, "The Ultimate Limits of Binary Coding for a Wideband Gaussian Channel," *The Deep Space Network Progress Report 42-22, May–June 1974*, Jet Propulsion Laboratory, Pasadena, California, pp. 78–80, August 15, 1974.

\ \

An Efficient Implementation of Forward–Backward Least-Mean-Square Adaptive Line Enhancers

H.-G. Yeh
Spacecraft Telecommunications Equipment Section

T. M. Nguyen
Communications Systems Research Section

An efficient implementation of the forward-backward least-mean-square (FBLMS) adaptive line enhancer is presented in this article. Without changing the characteristics of the FBLMS adaptive line enhancer, the proposed implementation technique reduces multiplications by 25 percent and additions by 12.5 percent in two successive time samples in comparison with those operations of direct implementation in both prediction and weight control. The proposed FBLMS architecture and algorithm can be applied to digital receivers for enhancing signal-to-noise ratio to allow fast carrier acquisition and tracking in both stationary and nonstationary environments.

I. Introduction

Adaptive line enhancers (ALEs) are useful in many areas, including time-domain spectral estimation for fast carrier acquisition [2–4]. For example, a fast carrier acquisition technique [2],[1] as shown in Fig. 1, will be very useful for a deep-space mission, especially in a nonstationary environment or emergencies. Figure 1 is the block diagram of an ALE in a digital receiver used for both acquisition and tracking. First, the receiver is in the acquisition mode. Second, when the uplink carrier is acquired as indicated by the lock detector, the switch is shifted to the tracking position and the tracking process takes over immediately. With this acquisition scheme, the uplink carrier can be acquired by a transponder in seconds (as opposed to minutes for the Cassini transponder). Although devised to support a space mission, the architecture of the forward–backward least-mean-square (FBLMS) ALE and the associated algorithm proposed in this article are also applicable to other systems, including fixed-ground and mobile communication systems. Note that this proposed ALE scheme in the receiver needs a residual carrier, and does not work directly in suppressed-carrier cases.

A conventional ALE system using a least-mean-square (LMS) algorithm is depicted in Fig. 2, where z^{-1} represents a delay. The analysis of the ALE for enhancing the signal-to-noise ratio (SNR) to allow fast acquisition is given in [2]. The block diagram of a FBLMS adaptive line enhancer is shown in Fig. 3. The performance analysis of the FBLMS adaptive line enhancer is provided in [1]. The FBLMS adaptive line enhancer algorithm enjoys approximately half the misadjustment of that of the LMS algorithm [1].

[1] T. M. Nguyen, H. G. Yeh, and L. V. Lam, "A New Carrier Frequency Acquisition Technique for Future Digital Transponders," to be published in a future issue of *The Telecommunications and Data Acquisition Progress Report.*

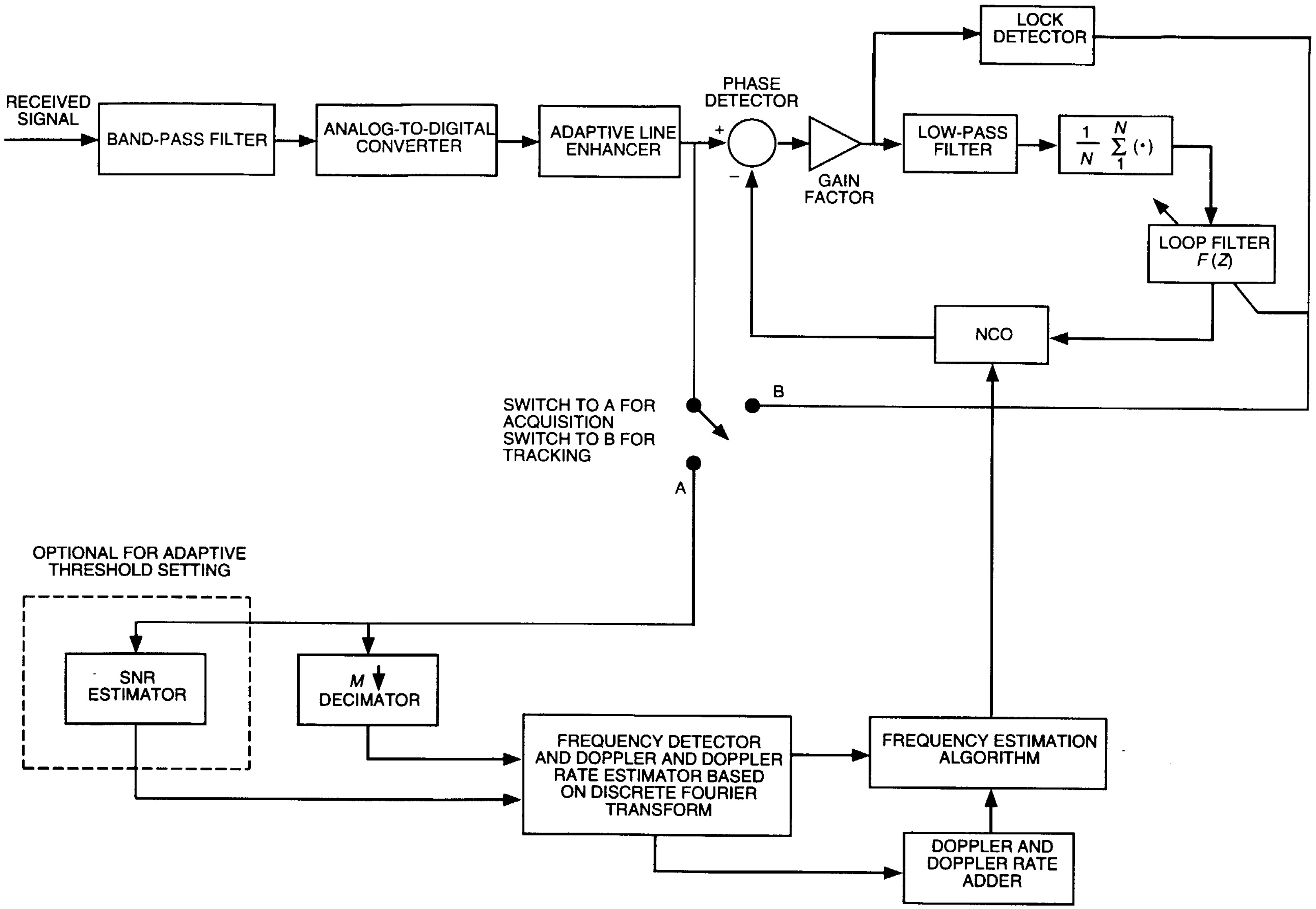

RECEIVED SIGNAL
BAND-PASS FILTER
ANALOG-TO-DIGITAL CONVERTER
ADAPTIVE LINE ENHANCER
PHASE DETECTOR
+
−
GAIN FACTOR
LOCK DETECTOR
LOW-PASS FILTER
$\frac{1}{N}\sum_{1}^{N}(\cdot)$
LOOP FILTER F(Z)
NCO
B
SWITCH TO A FOR ACQUISITION SWITCH TO B FOR TRACKING
A
OPTIONAL FOR ADAPTIVE THRESHOLD SETTING
SNR ESTIMATOR
M ↓ DECIMATOR
FREQUENCY DETECTOR AND DOPPLER AND DOPPLER RATE ESTIMATOR BASED ON DISCRETE FOURIER TRANSFORM
FREQUENCY ESTIMATION ALGORITHM
DOPPLER AND DOPPLER RATE ADDER

Fig. 1. The ALE in the digital receiver for both acqusition and tracking.

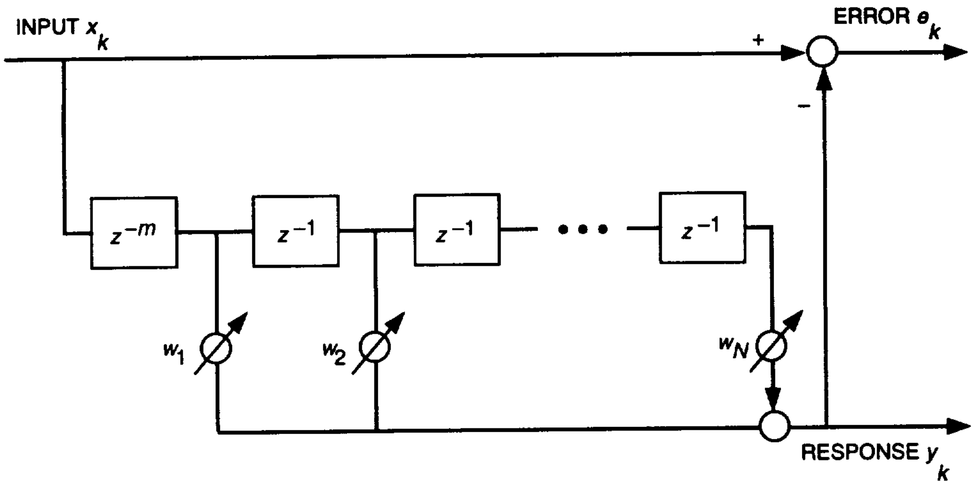


Fig. 2. The architecture of the conventional ALE.

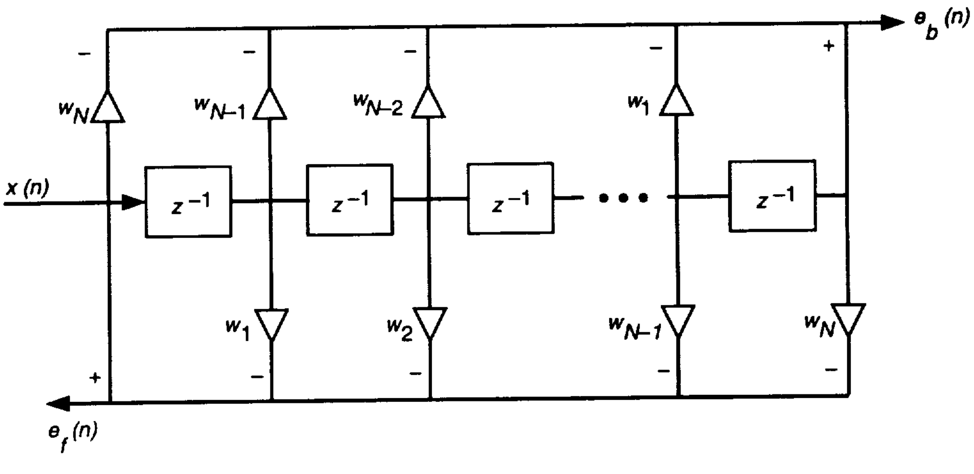


Fig. 3. The structure of the FBLMS adaptive line enhancer.

However, it requires about twice the number of multiplications and additions of the LMS algorithm. In this article, an efficient implementation of the fast FBLMS algorithm is presented. This fast algorithm provides the same speed of convergence as that of the LMS algorithm and provides the same misadjustment as that of the FBLMS adaptive line enhancer, but requires fewer multiplications and additions. The computational reduction is achieved by grouping two successive predictor computations together and computing weight adaption at every other sampling time [5]. By using a radix-2 structure to manipulate time samples, redundant computations embedded in two successive time samples can be removed via a new structure of the fast FBLMS algorithm.

This article is organized as follows. The FBLMS algorithm is reviewed in Section II. The fast FBLMS algorithm is derived and proposed in Section III. The fast FBLMS algorithm implementation is given in Section IV and simulation results are presented in Section V. Finally, the conclusion is given in Section VI.

II. Forward–Backward LMS Adaptive Line Enhancer Algorithm

The structure of the forward–backward LMS adaptive line enhancer [1] is shown in Fig. 3. The forward and backward prediction errors are then defined, respectively, as follows:

$$e_f(n) = x(n) - \mathbf{X}^T(n)\mathbf{W}(n) \tag{1a}$$

$$e_b(n) = x(n-N) - \mathbf{X}_b^T(n)\mathbf{W}(n) \tag{1b}$$

where the superscript T denotes the transpose of a vector, and

$$\mathbf{X}^T(n) = [x(n-1), x(n-2), \cdots, x(n-N)] \tag{1c}$$

$$\mathbf{X}_b^T(n) = [x(n-N+1), x(n-N+2), \cdots, x(n)] \tag{1d}$$

$$\mathbf{W}^T(n) = [w_1(n), w_2(n), \cdots, w_N(n)] \tag{1e}$$

In any gradient algorithm, the coefficient vector $\mathbf{W}(n)$ is updated using

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu\hat{\nabla}\{e(n)^2\} \tag{2a}$$

where μ is the adaptive step size and the $\hat{\nabla}\{e(n)^2\}$ is the estimated gradient of the surface of $E\{e(n)^2\}$. Note that $E\{\cdot\}$ denotes the expected value. In the forward–backward algorithm, $e(n)^2 = e_f(n)^2 + e_b(n)^2$, and the gradient estimate is chosen as

$$\hat{\nabla}\{e(n)^2\} = -[e_f(n)\mathbf{X}(n) + e_b(n)\mathbf{X}_b(n)] \tag{2b}$$

It is shown in [1] that Eq. (2b) is an unbiased estimator of the gradient. This leads to the coefficient update

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu[e_f(n)\mathbf{X}(n) + e_b(n)\mathbf{X}_b(n)] \tag{2c}$$

This means that $\mathbf{W}(n+1) \cong \mathbf{W}(n)$ in steady state when both forward and backward errors are approaching zero.

III. The Fast Forward–Backward LMS Algorithm

The fast FBLMS algorithm is derived in this section by using the radix-2 algorithm on time samples. Both predictor and weight update sections are provided in detail.

A. Predictor Section

We consider the computation of two successive predictions in both forward and backward directions with the fixed weight coefficient $\mathbf{W}(n-1)$. After regrouping even and odd terms, the forward predictor is obtained [5] and given in Eq. (3):

$$\begin{bmatrix} \hat{d}_f(n-1) \\ \hat{d}_f(n) \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T(n-1) \\ \mathbf{X}^T(n) \end{bmatrix} \mathbf{W}(n-1) = \begin{bmatrix} \mathbf{A}^T & \mathbf{B}^T \\ \mathbf{C}^T & \mathbf{A}^T \end{bmatrix}_n \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \end{bmatrix}_{n-1} \tag{3a}$$

where

$$\mathbf{A}^T = [x(n-2), x(n-4), \cdots, x(n-N+2), x(n-N)] \tag{3b}$$

$$\mathbf{B}^T = [x(n-3), x(n-5), \cdots, x(n-N+1), x(n-N-1)] \tag{3c}$$

$$\mathbf{C}^T = [x(n-1), x(n-3), \cdots, x(n-N+3), x(n-N+1)] \tag{3d}$$

$$\mathbf{W}_0 = [w_0(n-1), w_2(n-1), \cdots, w_{N-2}(n-1)]^T \tag{3e}$$

$$\mathbf{W}_1 = [w_1(n-1), w_3(n-1), \cdots, w_{N-1}(n-1)]^T \tag{3f}$$

Similarly, the backward predictor is obtained and given as follows:

$$\begin{bmatrix} \hat{d}_b(n-1) \\ \hat{d}_b(n) \end{bmatrix} = \begin{bmatrix} \mathbf{X}_b^T(n-1) \\ \mathbf{X}_b^T(n) \end{bmatrix} \mathbf{W}(n-1) = \begin{bmatrix} \mathbf{F}^T & \mathbf{G}^T \\ \mathbf{G}^T & \mathbf{H}^T \end{bmatrix}_n \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \end{bmatrix}_{n-1} \tag{4a}$$

where

$$\mathbf{F}^T = [x(n-N), x(n-N+2), \cdots, x(n-4), x(n-2)] \tag{4b}$$

$$\mathbf{G}^T = [x(n-N+1), x(n-N+3), \cdots, x(n-3), x(n-1)] \tag{4c}$$

$$\mathbf{H}^T = [x(n-N+2), x(n-N+4), \cdots, x(n-2), x(n)] \tag{4d}$$

Equations (3a) and (4a) are approximations by virtue of updating the weight vector only once every two cycles. The relationship between the two sequence sets $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ and $\{\mathbf{F}, \mathbf{G}, \mathbf{H}\}$ is given as follows:

$$\mathbf{F} = \mathbf{A}_r \tag{5}$$

$$\mathbf{G} = \mathbf{C}_r \tag{6}$$

$$z^{-1}\mathbf{H} = \mathbf{A}_r \tag{7}$$

where subscript r means the reversed order of the sequence and the z^{-1} means one delay unit of the corresponding sequence and is equivalent to two time sample delays. Furthermore, we observe the following relationships between $\mathbf{G}$, $\mathbf{B}$, $\mathbf{C}$:

$$z^{-1}\mathbf{G} = \mathbf{B}_r \tag{8}$$

$$z^{-1}\mathbf{C} = \mathbf{B} \tag{9}$$

After performing the appropriate computation, Eq. (4a) can be rewritten as follows:

$$\begin{bmatrix} \hat{d}_b(n-1) \\ \hat{d}_b(n) \end{bmatrix} = \begin{bmatrix} \mathbf{G}^T(\mathbf{W}_0 + \mathbf{W}_1) + (\mathbf{F} - \mathbf{G})^T \mathbf{W}_0 \\ \mathbf{G}^T(\mathbf{W}_0 + \mathbf{W}_1) - (\mathbf{G} - \mathbf{H})^T \mathbf{W}_1 \end{bmatrix} \tag{10}$$

The computation of Eq. (4a) requires two inner products of length N, while that of Eq. (10) requires only three inner products of length $N/2$ and $N/2$ additions to perform $\mathbf{W}_0 + \mathbf{W}_1$. Similarly, by combining Eqs. (5) through (9), Eq. (3a) can be rewritten as follows:

$$\begin{bmatrix} \hat{d}_f(n-1) \\ \hat{d}_f(n) \end{bmatrix} = \begin{bmatrix} \mathbf{A}^T(\mathbf{W}_0 + \mathbf{W}_1) + (\mathbf{B} - \mathbf{A})^T \mathbf{W}_1 \\ \mathbf{A}^T(\mathbf{W}_0 + \mathbf{W}_1) - (\mathbf{A} - \mathbf{C})^T \mathbf{W}_0 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{A}^T(\mathbf{W}_0 + \mathbf{W}_1) + z^{-1}(\mathbf{G} - \mathbf{H})_r^T \mathbf{W}_1 \\ \mathbf{A}^T(\mathbf{W}_0 + \mathbf{W}_1) - (\mathbf{F} - \mathbf{G})_r^T \mathbf{W}_0 \end{bmatrix} \tag{11}$$

Clearly, the sequences $(\mathbf{G} - \mathbf{H})$ and $(\mathbf{F} - \mathbf{G})$ of Eq. (10) are reused again in Eq. (11), but in reverse order. The computation of Eq. (11) requires only three inner products of length $N/2$. The total number of multiplications and additions required in both forward and backward predictor sections for two successive computations is about $3N$ and $3.5N$, respectively. The total number of multiplications and additions required in Eqs. (1a) and (1b) for two successive prediction sections is $4N$ and $4(N-1)$. Consequently, there are about 25 percent and 12.5 percent savings in multiplications and additions, respectively.

B. Weight Update Section

We consider the weight coefficient updates now. Since weights are explicitly computed at every other time update using the look-ahead approach [6], the weight update of Eq. (2c) can be rewritten as follows:

$$\mathbf{W}(n+1) = \mathbf{W}(n-1) + \mu\left[e_f(n-1)\mathbf{X}(n-1) + e_b(n-1)\mathbf{X}_b(n-1)\right] + \mu\left[e_f(n)\mathbf{X}(n) + e_b(n)\mathbf{X}_b(n)\right]$$

$$= \mathbf{W}(n-1) + [\mathbf{X}(n) \quad \mathbf{X}(n-1)] \begin{bmatrix} \mu e_f(n) \\ \mu e_f(n-1) \end{bmatrix} + [\mathbf{X}_b(n) \quad \mathbf{X}_b(n-1)] \begin{bmatrix} \mu e_b(n) \\ \mu e_b(n-1) \end{bmatrix} \tag{12}$$

By combining Eqs. (5) through (9), Eq. (12) is rewritten as follows:

$$\begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \end{bmatrix}_{n+1} = \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \end{bmatrix}_{n-1} + \begin{bmatrix} \mathbf{C} & \mathbf{A} \\ \mathbf{A} & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mu e_f(n) \\ \mu e_f(n-1) \end{bmatrix} + \begin{bmatrix} \mathbf{G} & \mathbf{F} \\ \mathbf{H} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mu e_b(n) \\ \mu e_b(n-1) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{W}_0 \\ \mathbf{W}_1 \end{bmatrix}_{n-1} + \mu \begin{bmatrix} \mathbf{A}(e_f(n) + e_f(n-1)) - (\mathbf{F} - \mathbf{G})_r e_f(n) \\ \mathbf{A}(e_f(n) + e_f(n-1)) + z^{-1}(\mathbf{G} - \mathbf{H})_r e_f(n-1) \end{bmatrix}$$

$$+ \mu \begin{bmatrix} \mathbf{G}(e_b(n) + e_b(n-1)) + (\mathbf{F} - \mathbf{G}) e_b(n-1) \\ \mathbf{G}(e_b(n) + e_b(n-1)) - (\mathbf{G} - \mathbf{H}) e_b(n) \end{bmatrix} \tag{13}$$

The vectors $(\mathbf{F}-\mathbf{G})$ and $(\mathbf{G}-\mathbf{H})$ are once more employed in Eq. (13). Notice that the term $\mu[\mathbf{A}(e_f(n) + e_f(n-1)) + \mathbf{G}(e_b(n) + e_b(n-1))]$ is computed only once, and the sum is applied to both $\mathbf{W}_0$ and $\mathbf{W}_1$ for updates. The total numbers of multiplications and additions in Eq. (13) are about $3N$ and $3.5N$, respectively. However, the total numbers of multiplications and additions of Eq. (2c) for two adaptations are $4N$ and $4(N-1)$. Consequently, 25 percent of multiplications and 12.5 percent of additions are saved by using Eq. (13) in comparison with those operations of Eq. (2c).

IV. Implementation

The architecture of the fast FBLMS algorithm is depicted in Fig. 4. A switching circuit is employed after the adaptive line enhancer, and the switch rate (from **C** to **A** or from **A** to **C**) is the same as the sampling rate. The switching circuit is switched between points **C** and **A** alternately. Sequences **C** and **A** are generated at a rate of $1/(2T)$ accordingly. The sequence **B** is a delayed version of the sequence **C**. By using a radix-2 structure, sequences $\{\mathbf{B}-\mathbf{A}\}$ and $\{\mathbf{A}-\mathbf{C}\}$ are then generated at the upper and lower lag, respectively. By using the sequence $\{\mathbf{B}-\mathbf{A}\}$, inner products $(\mathbf{B}-\mathbf{A})^T\mathbf{W}_1$ and $z^{-1}(\mathbf{G}-\mathbf{H})^T\mathbf{W}_1$ are generated at the upper and lower lag, respectively, of the upper forward–backward tapped-delay-line structure. Similarly, by using the sequence $\{\mathbf{A}-\mathbf{C}\}$, inner products $(\mathbf{A}-\mathbf{C})^T\mathbf{W}_0$ and $(\mathbf{F}-\mathbf{G})^T\mathbf{W}_0$ are generated at the upper and lower lag, respectively, of the lower forward–backward tapped-delay-line structure. Note that vectors **F**, **G**, and **H** are defined in Eqs. (5), (6), and (7), respectively. Inner products of $\mathbf{A}^T(\mathbf{W}_0 + \mathbf{W}_1)$ and $\mathbf{G}^T(\mathbf{W}_0 + \mathbf{W}_1)$ are computed at the top and bottom portions, respectively, of the fast FBLMS architecture. Finally, forward errors $\{e_f(n)$ and $e_f(n-1)\}$ and backward errors $\{e_b(n-1)$ and $e_b(n-2)\}$ are computed at the right-hand side of Fig. 4. In order to subtract the term of $z^{-1}(\mathbf{G}-\mathbf{H})^T\mathbf{W}_1$ and form the backward error, a delay unit is applied to the output branch of the inner product of $\mathbf{G}^T(\mathbf{W}_0 + \mathbf{W}_1)$. Consequently, the corresponding backward error is delayed from $e_b(n)$ to $e_b(n-2)$. Notice that this radix-2 structure concept can be applied again to the upper and lower forward–backward taped-delay-line portion of the fast FBLMS algorithm to further reduce the number of multiplications and additions.

Although the fast FBLMS architecture shown in Fig. 4 appears more complex than the FBLMS shown in Fig. 3, the structure is still very simple. In fact, the fast FBLMS architecture consists of radix-2, forward LMS, and FBLMS structures. The increased data flow complexity over the FBLMS algorithm is limited; therefore, the fast FBLMS algorithm can be easily implemented with digital signal processors.

V. Simulation Results

An adaptive line enhancer with 6-weight ($N = 6$) is chosen as an example. The input signal is a sinusoid of frequency f_0 contaminated by white noise. Computer simulations are conducted for the misadjustment calculation by using forward LMS, FBLMS, and fast FBLMS algorithms. The misadjustment [1] is computed after convergence as follows:

$$\mathrm{M} = \frac{\text{extra output power due to weight jittering}}{\text{minimum output power}}$$

$$= \frac{E[\Delta(n)^T\phi(x,x)\Delta(n)]}{E[e(n)^2]_{opt}} \tag{14}$$

where

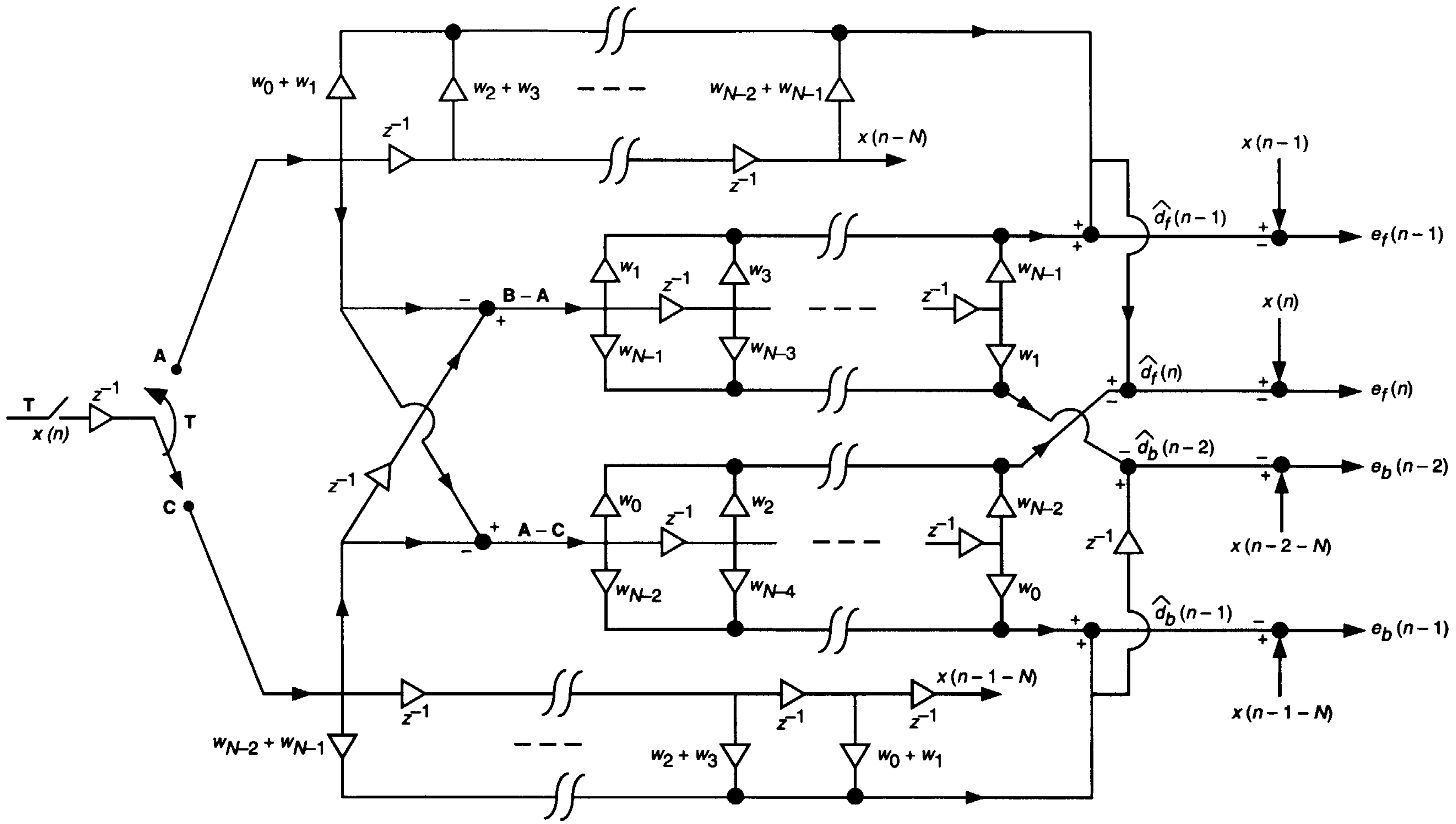


Fig. 4. The architecture of the fast FBLMS algorithm.

$$\Delta(n) = \mathbf{W}(n) - \mathbf{W}_{opt} \tag{15}$$

$$\phi(x,x) = E[\mathbf{X}(n)\mathbf{X}^T(n)] \tag{16}$$

$$E[e(n)^2]_{opt} = E[x(n)^2] - \mathbf{W}_{opt}^T E[x(n)\mathbf{X}(n)] \tag{17}$$

Table 1 shows the measured misadjustments for various values of SNR at step size $\mu = 2^{-8}$. Apparently, the excess error power for both the FBLMS and the fast FBLMS algorithms is approximately half that of the forward LMS algorithm at the 10-dB SNR. The improvement of the misadjustment by using both the FBLMS and the fast FBLMS algorithms over that of the forward LMS algorithm is limited at an SNR around 0 dB. However, the misadjustment of the fast FBLMS algorithm is about the same as that of the FBLMS algorithm. Furthermore, it is observed in Table 1 that, at a higher SNR, the misadjustment increases (for a given step size $\mu = 2^{-8}$). This is because the minimum output error power decreases much more rapidly than the extra output power due to weight jittering, as depicted by Eq. (14). This high misadjustment is significantly reduced when the step size μ is cut to 2^{-10}, as shown in Table 2.

Table 2 shows the measured misadjustments for various values of the step size and the frequency f_0 at SNR = 10 dB. Apparently, the excess error power for both the FBLMS and the fast FBLMS algorithms is approximately half that of the forward LMS algorithm at the step size $\mu = 2^{-8}$ and $\mu = 2^{-10}$. The misadjustment is much reduced when the step size is small (2^{-10}) by using any one of the three algorithms. Again, the misadjustment of the fast FBLMS algorithm is about the same as that of the FBLMS. The $E[e(n)^2]_{opt}$ used to derive the misadjustment is computed by using 500 samples in each run. The misadjustment results listed in Tables 1 and 2 were obtained by averaging 100 runs of the excess error power curves after convergence had been achieved.

Table 1. A comparison between the misadjustment powers of three algorithms at $\mu = 2^{-8}$.

SNR	f_0	Percent misadjustment		
		Forward LMS	FBLMS	Fast FBLMS
0	0.1	3.04	2.75	2.75
3	0.1	3.74	2.84	2.93
10	0.1	32.50	13.77	16.95

Table 2. A comparison between the misadjustment powers of three algorithms using fixed SNR = 10 dB with different μ.

μ	f_0	Percent misadjustment		
		Forward LMS	FBLMS	Fast FBLMS
2^{-8}	0.1667	31.34	14.47	16.03
2^{-8}	0.1	32.5	13.77	16.95
2^{-10}	0.1667	3.06	2.05	1.99
2^{-10}	0.1	2.33	1.24	1.30

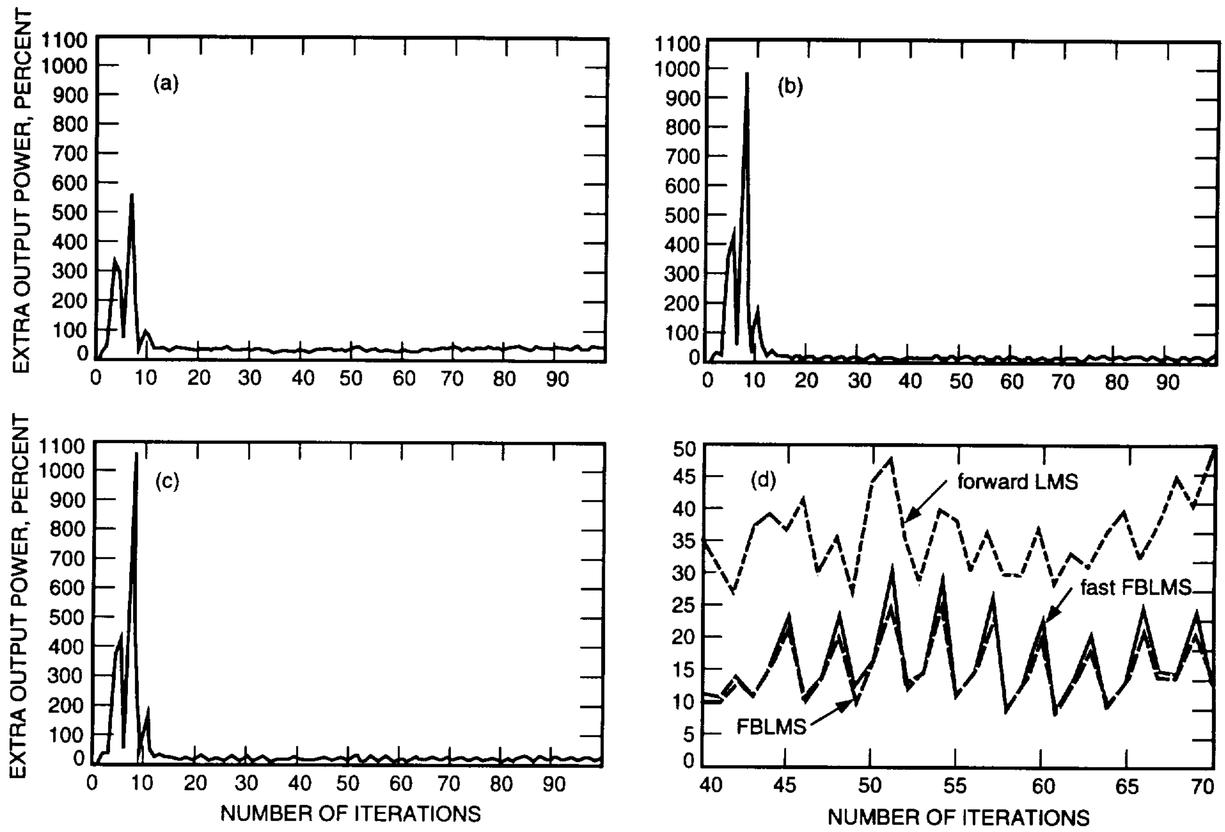


Fig. 5. A typical excess error power versus *n* plot by using the (a) forward LMS, (b) FBLMS, (c) fast FBLMS algorithm, and (d) the steady-state comparison.

Figures 5(a), (b), and (c) show a typical excess error power versus n plot at $f_0 = 1/6$, step size $= 2^{-8}$, and SNR = 10 dB for the forward LMS, FBLMS, and fast FBLMS algorithms, respectively. Figure 5(d) shows the excess error power at the steady state. It is clear that the performance of the fast FBLMS algorithm is about the same as that of the FBLMS algorithm.

VI. Conclusion

The fast forward–backward LMS algorithm presented in this article shows that the number of arithmetic operations in [1] can be reduced without degrading performance. In the forward–backward predictor section, 25 percent of multiplications and 12.5 percent of additions are saved in each of two successive operations. Similarly, in the weight control section, 25 percent of multiplications and 12.5 percent of additions are saved in each of two adaptations. Simulation results indicate that improvements in misadjustment for both the FBLMS and the fast FBLMS algorithms over the conventional LMS algorithm are about 50 percent at a high SNR. When the SNR is low, the misadjustment improvement for both the FBLMS and the fast FBLMS algorithms over the conventional LMS algorithm is less than 50 percent. Notice that this fast forward–backward LMS algorithm is well suited for implementation on application-specific integrated circuits and digital signal processors. This implementation method can be generalized by using higher than two steps of look-ahead. Further computational savings are possible with limited cost on controlling appropriate data flow. This fast FBLMS adaptive line enhancer can be easily integrated together with either a conventional voltage-controlled oscillator in a closed loop for acquisition/tracking, as used in the present deep-space transponder, or a numerically controlled oscillator in an open-loop scheme for acquiring and tracking the carrier signal, as will be used in future deep-space transponders.

Acknowledgments

The authors wish to acknowledge the valuable support of Arthur W. Kermode, Lance A. Riley, and Sami M. Hinedi. Special thanks to Victor A. Vilnrotter for his review, correction, and discussion.

References

[1] Y. C. Lim and C. C. Ko, "Forward-Backward LMS Adaptive Line Enhancer," *IEEE Trans. Circuits Syst.*, vol. 37, no. 7, pp. 936–940, July 1990.

[2] H. G. Yeh and T. M. Nguyen, "Adaptive Line Enhancers for Fast Acquisition," *The Telecommunications and Data Acquisition Progress Report 42-119, July–September 1994*, Jet Propulsion Laboratory, Pasadena, California, pp. 140–159, November 15, 1994.

[3] L. Griffiths, "Rapid Measurement of Digital Instantaneous Frequency," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, no. 2, pp. 207–222, April 1975.

[4] J. T. Rickard, J. R. Zeibler, N. J. Dentino, and M. Shensa, "A Performance Analysis of Adaptive Line Enhancer—Augmented Spectral Detectors," *IEEE Trans. Trans. ASSP*, vol. ASSP-29, pp. 694–701, June 1981.

[5] Z. J. Mou and P. Duhamel, "Short-Length FIR Filters and Their Use in Fast Nonrecursive Filtering," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-39, pp. 1322–1332, June 1991.

[6] K. K. Parhi and D. G. Messerschmitt, "Pipeline Interleaving and Parallelism in Recursive Digital Filters—Part 1: Pipelining Using Scattered Look-Ahead and Decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 1099–1117, July 1989.

TDA Progress Report 42-121

May 15, 1995

Computer Simulation Results for PCM/PM/NRZ Receivers in Nonideal Channels

A. Anabtawi, T. M. Nguyen, and S. Million
Communications Systems Research Section

This article studies, by computer simulations, the performance of deep-space telemetry signals that employ the pulse code modulation/phase modulation (PCM/PM) technique, using nonreturn-to-zero data, under the separate and combined effects of unbalanced data, data asymmetry, and a band-limited channel. The study is based on measuring the symbol error rate performance and comparing the results to the theoretical results presented in previous articles. Only the effects of imperfect carrier tracking due to an imperfect data stream are considered. The presence of an imperfect data stream (unbalanced and/or asymmetric) produces undesirable spectral components at the carrier frequency, creating an imperfect carrier reference that will degrade the performance of the telemetry system. Further disturbance to the carrier reference is caused by the intersymbol interference created by the band-limited channel.

I. Introduction

There is considerable interest among international space agencies in searching for a bandwidth-efficient modulation scheme that can be used for future space missions without major modifications to their ground stations [1–4]. The Consultative Committee for Space Data Systems (CCSDS) has undertaken the task of investigating a modulation scheme that offers both of these features (bandwidth efficiency and no major hardware modifications to the current systems).

Currently, the space telemetry systems employ residual carrier modulation with subcarriers that are used to separate the data from the RF residual carrier. This was necessary to avoid interference because most of the data power fell within the bandwidth of the carrier phase-locked loop (PLL), as shown in Fig. 1(a). The CCSDS has recommended that square-wave and sine-wave subcarriers be used for the deep-space and near-Earth missions, respectively [5]. This modulation scheme is called pulse code modulation/phase-shift keying/phase modulation (PCM/PSK/PM), and it was developed at a time when weak signals and low data rates dominated [6]. With the development of technology and the evolvement of the Deep Space Network (DSN), a significant increase in the signal power can result in higher data rates. Using subcarriers in this case causes the occupied bandwidth to increase significantly. This is prohibitive because the space telemetry systems often operate under imposed bandwidth constraints. A natural solution is to eliminate the subcarrier and modulate the nonreturn-to-zero (NRZ) data directly on the RF carrier. This modulation scheme is referred to as PCM/PM/NRZ, and not only does it require

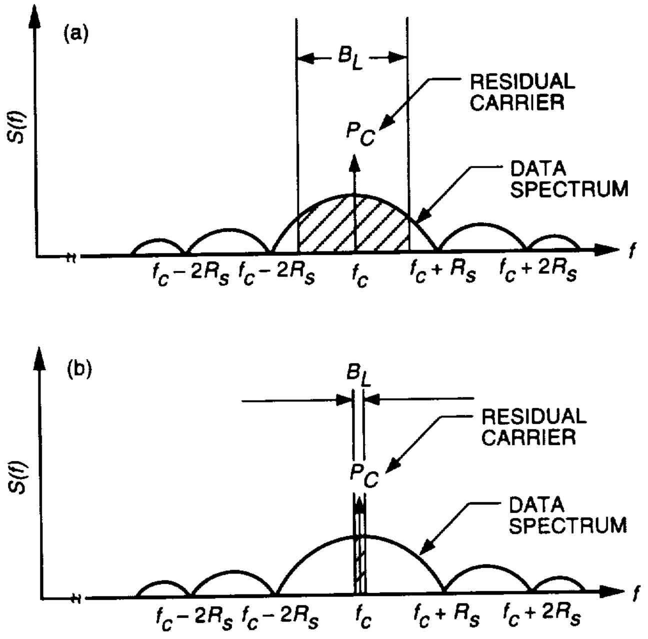


Fig. 1. PCM/PM/NRZ modulation: (a) low data rate: high ratio of loop bandwidth to data rate and (b) high data rate: low ratio of loop bandwidth to data rate.

minimum hardware modifications to the current systems, but it also achieves the bandwidth efficiency [4,7]. In this modulation technique, the part of the data spectrum that falls within the narrow carrier loop bandwidth seems flat and appears as white noise, as shown in Fig. 1(b), and, since the ratio of loop bandwidth to data rate is very small, the carrier tracking performance degradation due to this white noise component is negligible.

Recently, Nguyen has investigated and analyzed the behavior of PCM/PM receivers in nonideal channels [1,2]. The imbalance between +1's and −1's and/or data asymmetry in the data stream produce undesirable spectral components that degrade the performance of the system. Further degradation is caused by the intersymbol interference (ISI) created by the band-limited channel. This article verifies, by computer simulations, the theoretical results presented in [1,2] for the NRZ data stream. The Signal Processing Worksystem (SPW) was used for implementing and simulating the system. The separate effects of unbalanced data, data asymmetry, and band-limited channel on the symbol error rate (SER) performance of PCM/PM/NRZ receivers were simulated and then compared to the theoretical results presented in [1]. In reality, however, the receivers operate in the aggregate presence of these three effects, and the symbol signal-to-noise ratio (SSNR) degradation due to the three effects is not the algebraic sum of the SSNR degradation due to each separate effect. The second part of this article presents the simulation results for the degradation due to the combined effects on the SER performance, and these results are compared to the theoretical results presented in [2].

The organization of this article is as follows: Section II describes the separate effects on PCM/PM/NRZ receivers of perfect, unbalanced, asymmetric, and band-limited data streams. Section III describes the combined effects of these streams on PCM/PM/NRZ receivers. Section IV gives a brief description of the PCM/PM receiver blocks that were used to build the system in the SPW, Section V discusses the simulation results and compares them to theory, and, finally, Section VI presents the conclusion.

II. Separate Effects on PCM/PM/NRZ Receivers

The deep-space received telemetry signal, in the absence of a subcarrier, is given by [1]

$$s_r(t) = \sqrt{2P}\,\{\cos(m_T)\cos(\omega_c t + \theta_0) - d(t)\,\sin(m_T)\,\sin(\omega_c t + \theta_0)\} + n(t) \tag{1}$$

where P is the transmitted power, m_T is the telemetry modulation index in rad, $\omega_c = 2\pi f_c$ is the angular carrier center frequency in rad/s, θ_0 is the initial carrier phase, $n(t)$ is an additive white Gaussian noise (AWGN), and $d(t)$ is the data stream (NRZ) defined as

$$d(t) = \sum_{k=-\infty}^{\infty} d_k p(t + kT_s) \tag{2}$$

where $d_k = \pm 1$ with transition density p_t, $p(t)$ is the baseband pulse, and T_s is the symbol period in seconds. The first and second terms of Eq. (1) are the residual carrier and data components, respectively.

The undesired spectral components caused by the imperfect data stream (unbalanced data and/or data asymmetry) can degrade the carrier tracking performance. If $\hat{\theta}$ denotes the carrier loop estimate of θ_0, the phase error due to the thermal noise and interference caused by the imperfect data stream is defined as

$$\theta_e = \theta_0 - \hat{\theta} = \theta_e(noise) + \theta_e(data) + \theta_e(spike) \tag{3}$$

where $\theta_e(noise)$, $\theta_e(data)$, and $\theta_e(spike)$ are the phase error caused by the noise, data interference, and the spike caused by the imperfect data stream, respectively.

The carrier loop tracks the residual carrier component in Eq. (1) to provide an imperfect reference given by

$$r(t) = \sqrt{2}\,\cos(\omega_c t + \hat{\theta}) \tag{4}$$

The average probability of error due to the imperfect carrier tracking is given by

$$P_e = \int_{\theta_e} P_e(\theta_e) P(\theta_e)\, d\theta_e \tag{5}$$

where $P_e(\theta_e)$ is the conditional probability of error, and $P(\theta_e)$ is the probability density function (pdf) of the carrier tracking phase error θ_e. Assuming that this pdf has a Tikhonov distribution that is entirely characterized by the mean (assumed 0) and variance σ^2 of θ_e, and when the loop signal-to-noise ratio (SNR) is high, $P(\theta_e)$ may be approximated as Gaussian distribution, namely,

$$P(\theta_e) \approx \frac{\exp(-\theta_e^2/(2\sigma^2))}{[2\pi\sigma^2]^{-1/2}}, -\infty < \theta_e < \infty \tag{6}$$

As mentioned above, this expression was derived assuming the mean of the phase error θ_e to be zero. This assumption, however, is not true for an imperfect data stream, as will be shown in the subsequent sections.

The expressions for $P_e(\theta_e)$ and the carrier tracking phase error variance σ^2 have been evaluated in [1,2] for all the different cases studied in this article. The final results will be presented here for completeness.

A. Perfect Data Stream

In a perfect purely random data stream, the probability of transmitting a +1 pulse (or probability of mark), p, is equal to the probability of transmitting a -1, q, with transition density, p_t, given by

$$p_t = 2pq = \frac{1}{2} \tag{7}$$

where $q = 1 - p$.

The carrier term of Eq. (1) generates a residual carrier at f_c with power, P_c, given by

$$P_c = P\cos^2(m_T) \tag{8}$$

Combining the carrier and data terms, the one-sided power spectrum of a PCM/PM/NRZ perfect data stream is given by

$$S(f) = S_c(f) + S_D(f) \tag{9}$$

where

$$S_c(f) = P_c\delta(f) \tag{10}$$

and

$$S_D(f) = P\sin^2(m_T)S_{cont}(f) \tag{11}$$

is the data spectrum with power P_D defined as

$$P_D = P\sin^2(m_T) \tag{12}$$

For a perfect NRZ data stream, $S_{cont}(f)$ is defined as the power spectral density (PSD) for an ideal NRZ data stream and is given by

$$S_{cont}(f) = T_s\left\{\frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2}\right\} \tag{13}$$

Figure 2(a) shows the power spectrum of a perfect NRZ data stream (generated using SPW for symbol rate $R_s = 1/T_s = 10^4$ kbits/s).

For a perfect data stream and ideal channel, the conditional probability of error is given by

$$P_e(\theta_e) = \frac{1}{2}\,\mathrm{erfc}\left\{\sqrt{\frac{E_s}{N_0}}\cos(\theta_e)\right\} \tag{14}$$

where E_s/N_0 denotes the SSNR, that is,

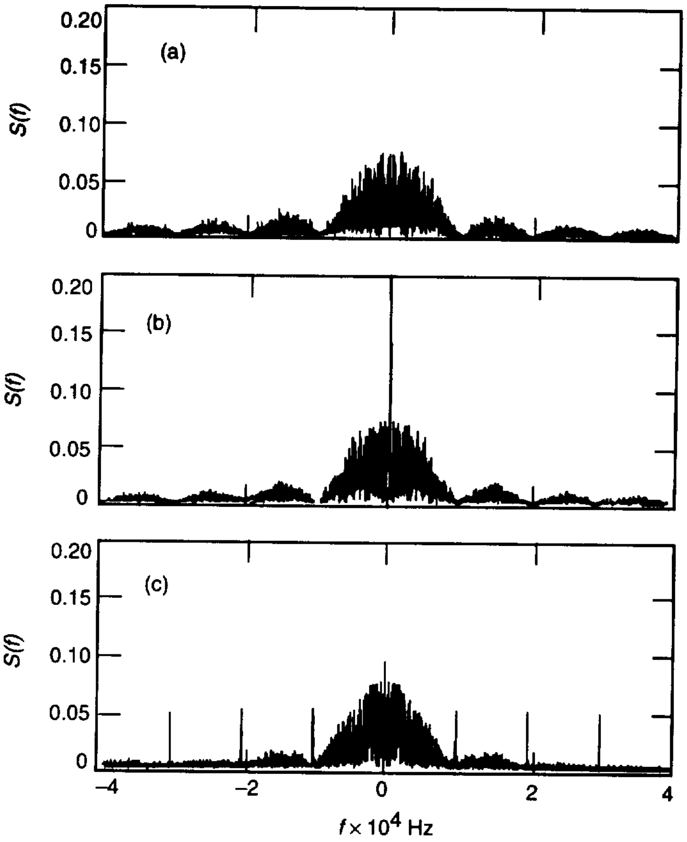


Fig. 2. Spectrums of different NRZ data streams: (a) balanced data stream (p = 0.5), (b) unbalanced data stream (p = 0.4), and (c) asymmetric data stream (ξ = 14 percent).

$$\frac{E_s}{N_0} = \frac{PT_s \sin^2(m_T)}{N_0} = \frac{P_D T_s}{N_0} \tag{15}$$

and erfc (x) is defined as the complementary error function given by

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x \exp(-v^2)\, dv \tag{16}$$

Note that for this case, the mean of the phase error θ_e in the steady state is zero. This, however, is not true for an imperfect data stream, as will be shown in the subsequent sections.

For the high data rate case ($B_L/R_s << 0.1$, where B_L denotes the one-sided loop bandwidth), the variance of the carrier tracking phase error is given as [1]

$$\sigma^2 = \frac{1}{\rho_0} + \frac{B_L}{R_s} \tan^2(m_T) \tag{17}$$

where

$$\rho_0 = \frac{(E_s/N_0)}{(B_L/R_s)\tan^2(m_T)} \tag{18}$$

By substituting Eqs. (6) and (14) into Eq. (5) and performing the numerical integration, the curve for the probability of error versus SSNR was obtained and is shown in Figs. 3 through 9 for comparison purposes.

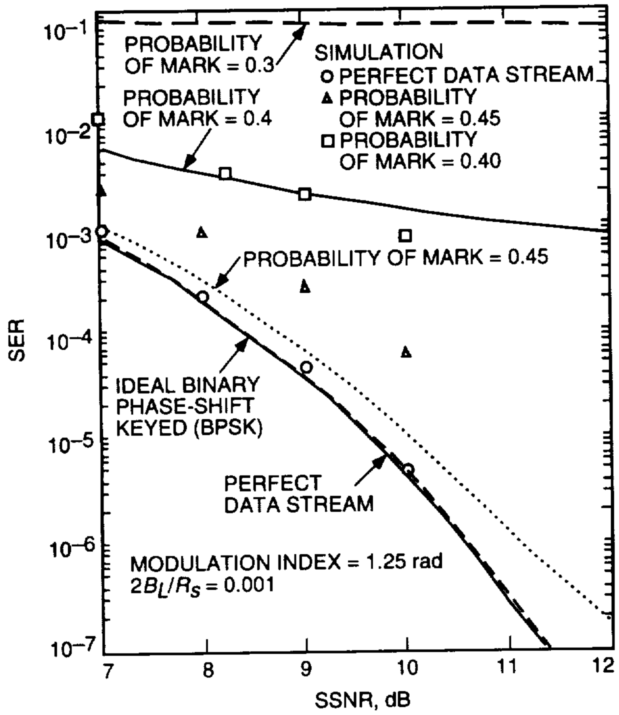


Fig. 3. Theory and simulation versus SSNR for unbalanced data without modification to the phase error.

B. Unbalanced Data Stream

The imbalance between +1's and −1's in the data stream causes an additional corruption to the received signal in Eq. (1), generating undesirable spectral components that can potentially degrade the performance of the telemetry system. When p is not equal to 0.5 (and, therefore, $p_t < 0.5$), the data component will be affected and Eq. (11) now becomes

$$S_D(f) = P\sin^2(m_T)\{S_{dc}(f) + S_{cont}(f)\} \tag{19}$$

where $S_{dc}(f)$ is the dc (or harmonic) component caused by the imperfect data stream that falls on the RF carrier.

The spectrum of an unbalanced NRZ data stream for $p = 0.4$, generated using the SPW, is shown in Fig. 2(b). For a PCM/PM/NRZ unbalanced data stream, the dc and continuous PSD components are found to be [1]

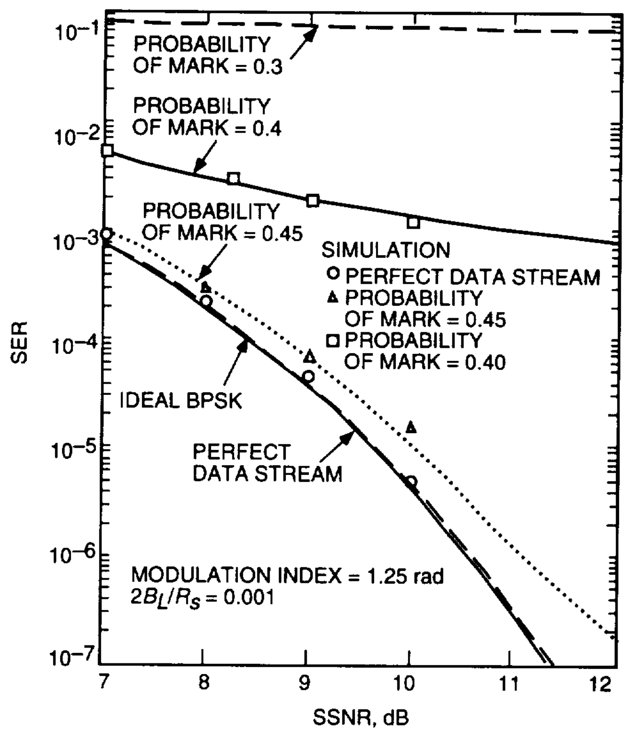


Fig. 4. Theory and simulation SER versus SSNR for unbalanced data.

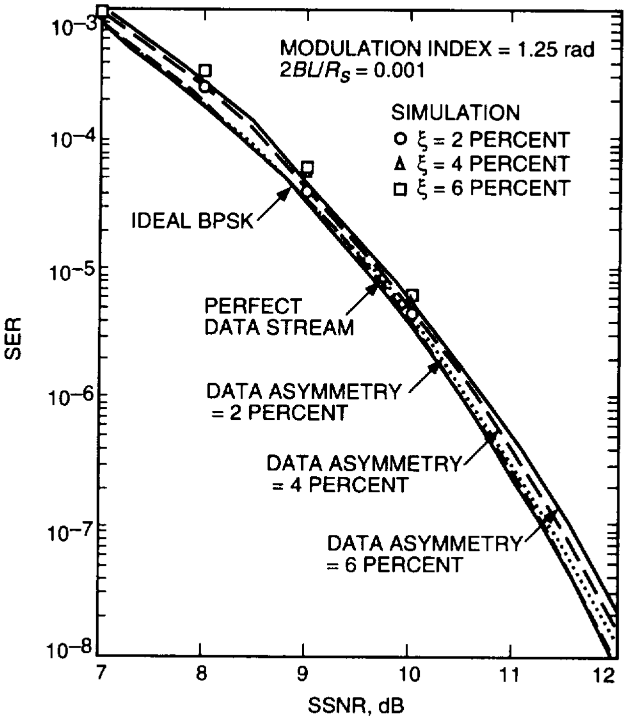


Fig. 5. Theory and simulation versus SSNR for data asymmetry.

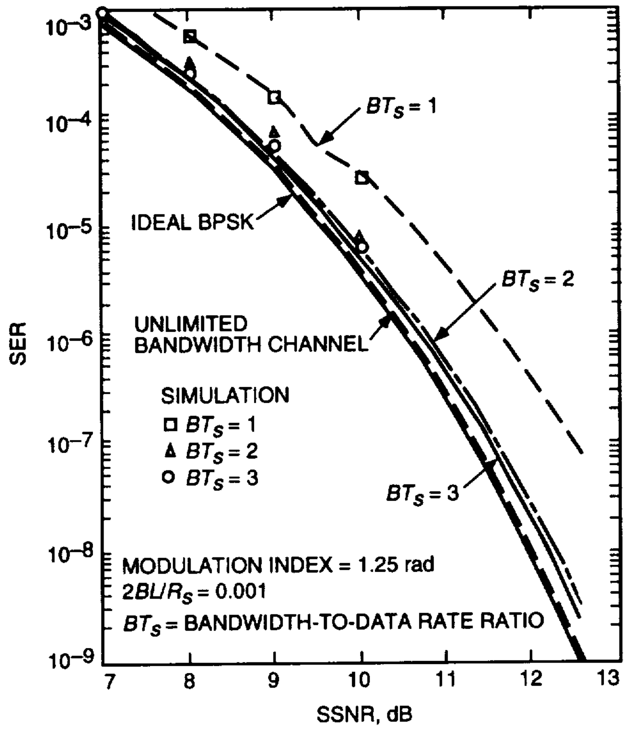


Fig. 6. Theory and simulation SER versus SSNR for band-limited channel.

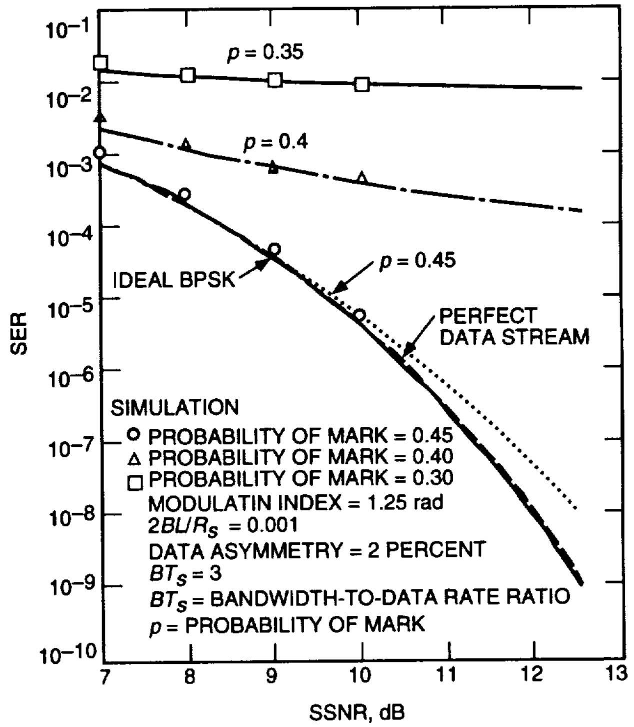


Fig. 7. Theory and simulation SER versus SSNR for various values of unbalanced data.

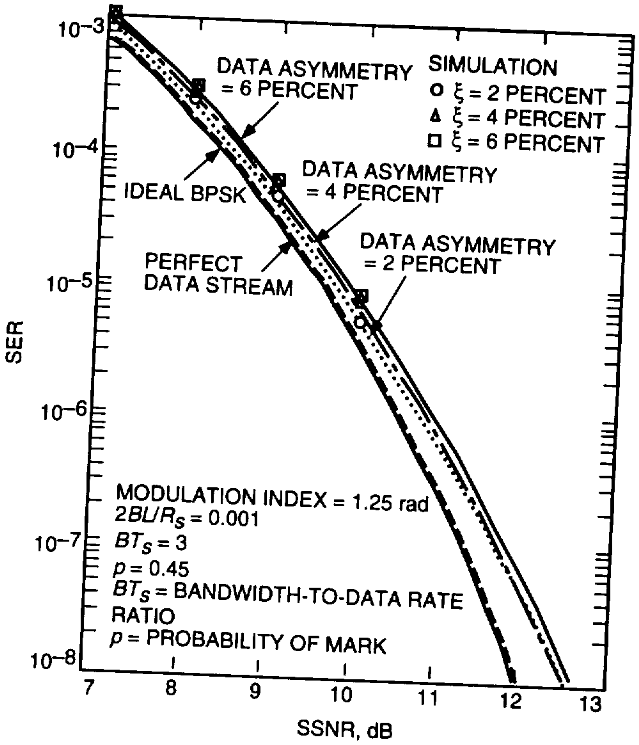


Fig. 8. Theory and simulation SER versus SSNR for various values of data asymmetry.

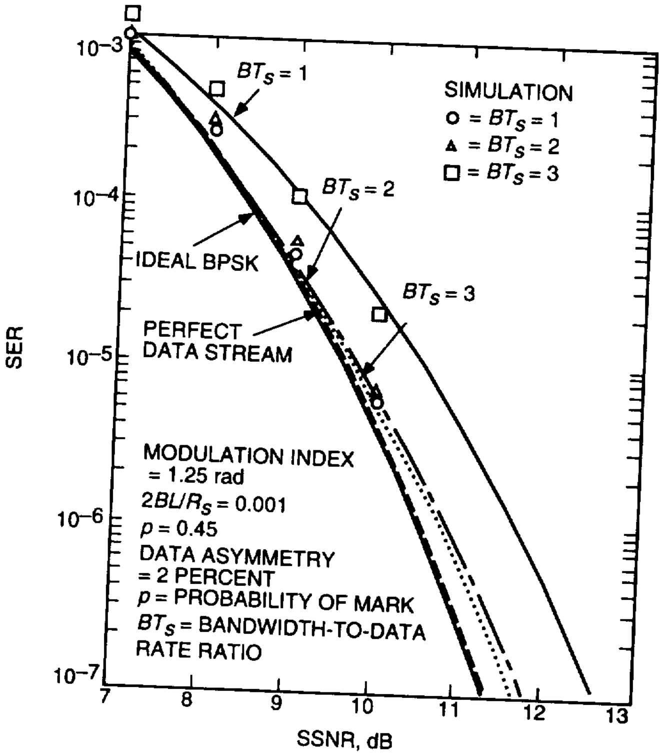


Fig. 9. Theory and simulation SER versus SSNR for various values of band-limited channel.

$$S_{dc}(f) = (1-2p)^2\delta(f) \tag{20}$$

$$S_{cont}(f) = 4T_s pq\left\{\frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2}\right\} \tag{21}$$

with power given by

$$P_{dc} = (1-2p)^2 P \sin^2(m_T) \tag{22}$$

$$P_{cont} = 4pqP\sin^2(m_T) \tag{23}$$

respectively, and where

$$P_D = P_{dc} + P_{cont} \tag{24}$$

Therefore, in addition to the tone generated at f_c by the residual carrier component in Eq. (1) with power given by Eq. (8), the spectrum of unbalanced PCM/PM/NRZ will include another tone at f_c generated by the imbalance between +1's and −1's with power given by Eq. (22). However, these two tones at f_c have noncoherent phases, causing the mean of the carrier tracking phase error in the steady state to deviate away from zero. This deviation is defined as $\theta_e(mean)$, which is a function of p and the modulation index m_T and is given by

$$\theta_e(mean) = -\tan^{-1}\{(\tan m_T)(2p-1)\} \tag{25}$$

Note that when $p = 0.5$, then $\tan^{-1} 0 = 0$, independent of m_T, as one would expect.

Figure 10 shows the $\theta_e(mean)$ of balanced and unbalanced data streams as generated by an SPW for $\theta_0 = 0$. Note that for the case of a balanced data stream, the mean of the phase error is centered at zero, whereas for an unbalanced data stream, the mean is at a negative value which, using the above equation, is calculated to be about −0.54 rad (−31 deg) for $p = 0.6$ and $m_T = 1.25$.

The conditional probability of error $P_e(\theta_e)$ is the same as the one given by Eq. (14). Recall, however, that Eq. (6) for the pdf of the carrier tracking phase error $P(\theta_e)$ was derived assuming the mean of θ_e to be zero. Therefore, the simulations will have to compensate for the phase difference $(\theta_0 - \hat{\theta})$ (Eq. (3)) by adding the value of $\theta_e(mean)$ (Eq. (25)) to the phase of Eq. (4), $(\hat{\theta})$, which results in a zero-mean phase error. In that case, $P(\theta_e)$ is given by Eq. (6) with the tracking variance given by

$$\sigma^2 = \frac{1}{\rho_0} + \frac{\alpha}{2}\tan^2(m_T) + \frac{1}{2}\frac{I}{C} \tag{26}$$

where ρ_0 is defined as before, α is the interference due to the continuous spectrum, and I/C is the interference caused by the dc component-to-carrier power ratio given, respectively, by

$$\alpha = 4T_s p(1-p)\int_{-\infty}^{\infty} |H(2\pi f)|^2 \frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2}\, df \tag{27}$$

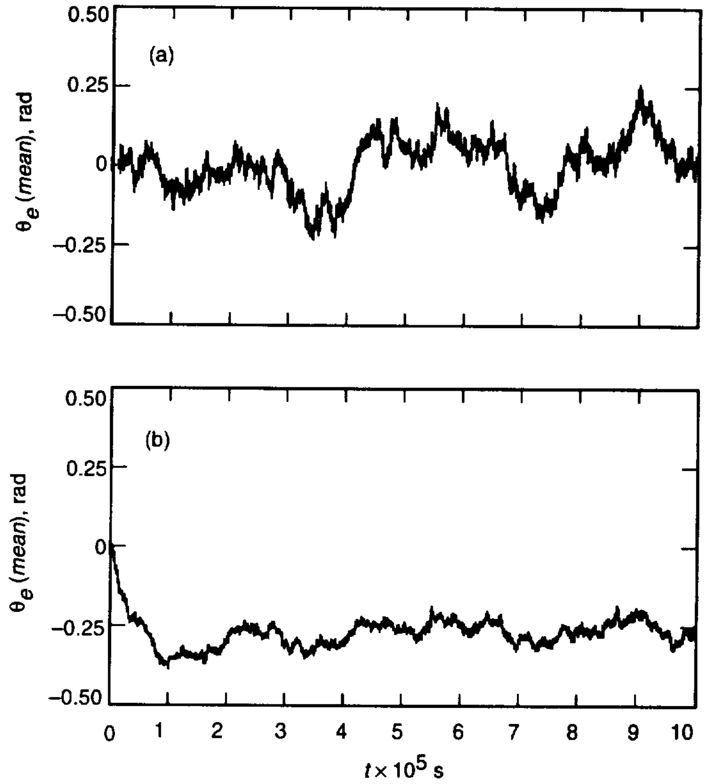


Fig. 10. θe (*mean*) of balanced and unbalanced data streams: (a) balanced data stream (*p* = 0.5): mean of phase error is centered at zero and (b) unbalanced data stream (*p* = 0.6): mean of the phase error deviates away from zero.

$$\frac{I}{C} = (1 - 2p)^2 \tan^2 (m_T) \tag{28}$$

where $H(j2\pi f)$ denotes the carrier loop transfer function, and for a second-order PLL is given by

$$|H(j2\pi f)|^2 = \frac{1 + 2\,(f/f_n)}{1 + (f/f_n)} \tag{29}$$

where f_n is the loop natural frequency.

The plot of the SER versus SSNR is shown in Figs. 3 and 4.

C. Data Asymmetry

Data asymmetry, due to rising and falling voltage transitions, causes undesirable spectral components that degrade the performance of the space telemetry system. The data asymmetry model adopted in this article assumes that +1 symbols are elongated by $(\Delta T_s)/2$ (relative to their nominal value of T_s seconds) when a negative-going data transition occurs, and -1 symbols are shortened by the same amount when a positive-going data transition occurs. Otherwise (when no transitions occur), the symbols maintain their nominal T_s seconds width. This model is illustrated in Fig. 11 for a purely random NRZ data stream.

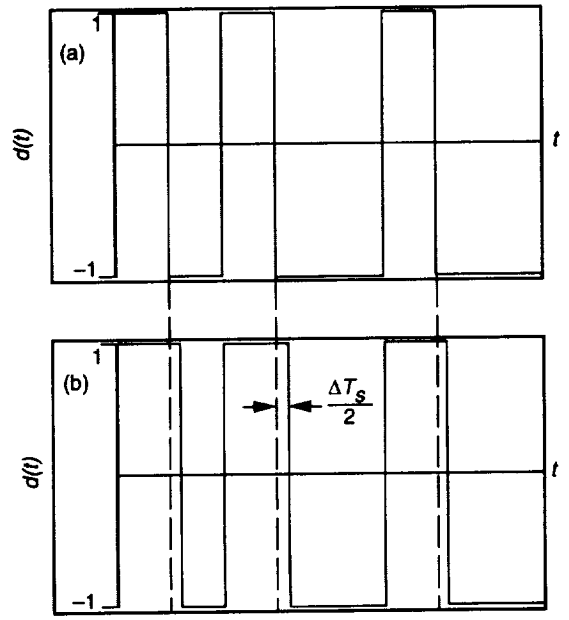


Fig. 11. Perfect and asymmetric NRZ data streams: (a) ξ = 0 (perfect data stream) and (b) , ξ = 0 (asymmetric data stream).

The power spectrum of an asymmetric NRZ random data stream with equiprobable symbols (that is, $p = p_t = 0.5$) and symbol rate R_s of 10^4 kbits/s is shown in Fig. 2(c). The dc, continuous, and harmonics PSD components are given by [1,3]

$$S_{dc}(f) = \frac{1}{4}\xi^2\delta(f) \tag{30}$$

$$S_{cont}(f) = \frac{T_s}{8}\left[\frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2}\right][3 + 5\cos^2(\pi f T_s \xi)] + \frac{T_s}{8}\left[\frac{\sin^2(2\pi f T_s \xi)}{(\pi f T_s)^2}\right][3\cos^2(\pi f T_s) + \cos^2(2\pi f T_s \xi)] \tag{31}$$

$$S_h(f) = \frac{1}{2\pi^2}\sum_{m=1}^{\infty}\frac{1}{m^2}C(m, \frac{1}{2}, \xi)\delta(f - mR_s) \tag{32}$$

respectively, where ξ denotes data asymmetry and is defined as

$$\xi = \frac{\Delta}{2} \tag{33}$$

and where

$$C\left(m, \frac{1}{2}, \xi\right) = \frac{1}{4}\sin^2(2m\pi\xi) \tag{34}$$

Hence, the data spectrum can be written as

$$S_D(f) = P\sin^2(m_T)\left\{S_{dc}(f) + S_{cont}(f) + S_h(f)\right\} \tag{35}$$

which, in addition to the tone at f_c caused by the carrier component, generates a spike at f_c due to the dc component $S_{dc}(f)$, and a spike at integer multiples of the symbol rate R_s due to the harmonics component $S_h(f)$. The continuous spectrum $S_{cont}(f)$ is plotted in Fig. 12 for various values of ξ. Note that when $\xi = 0$, the above equation reduces to the perfect NRZ random data case given by Eq. (11).

Similar to the unbalanced data case, the phase of the dc component at f_c caused by asymmetry and the phase of the carrier tone are noncoherent. The mean of the phase error θ_e for a perfectly balanced asymmetric data stream was derived to be

$$\theta_e(mean) = -\tan^{-1}\left\{(\tan m_T)\left(\frac{\xi}{2}\right)\right\} \tag{36}$$

Note that when $\xi = 0$, $\theta_e(mean) = 0$, as expected. Again, the simulations may have to compensate for the phase difference (Eq. (3)) to make the mean of θ_e zero at steady state.

Recall that in order to calculate the average probability of error, the conditional probability of error $P_e(\theta_e)$ and the tracking variance σ^2 must be determined. For the data asymmetry model used in this article and for a purely random and equiprobable (perfectly balanced) NRZ data stream, the conditional probability of error has the following form:

$$P_e(\theta_e) = \frac{5}{16}\,\text{erfc}\left\{\sqrt{\frac{E_s}{N_0}}\cos(\theta_e)\right\} + \frac{1}{8}\,\text{erfc}\left\{\sqrt{\frac{E_s}{N_0}}(1-\xi)\cos(\theta_e)\right\} + \frac{1}{16}\,\text{erfc}\left\{\sqrt{\frac{E_s}{N_0}}(1-2\xi)\cos(\theta_e)\right\} \tag{37}$$

and the variance of the tracking phase error σ^2 is given by Eq. (26), where ρ_0 is defined in Eq. (18) with

$$\alpha = \int_{-\infty}^{\infty} |H(2\pi f)|^2 S_{cont}(f)\, df \tag{38}$$

and

$$\frac{I}{C} = \frac{1}{4}\xi^2\tan^2(m_T) \tag{39}$$

Figure 5 shows the curves for SER versus SSNR when data asymmetry is present.

D. Band-Limited Channel

An additional impairment that contributes to the degradation of the overall performance of the system is the ISI caused by the band-limited channel. Band limiting causes interference between successive pulses producing the ISI effect, which behaves like an additional random noise.

If $p(t)$ denotes the pulse shape of the data, and $h'(t)$ denotes the impulse response of the equivalent low-pass filter of the RF band-pass filter with bandwidth B, then the received data can be expressed as

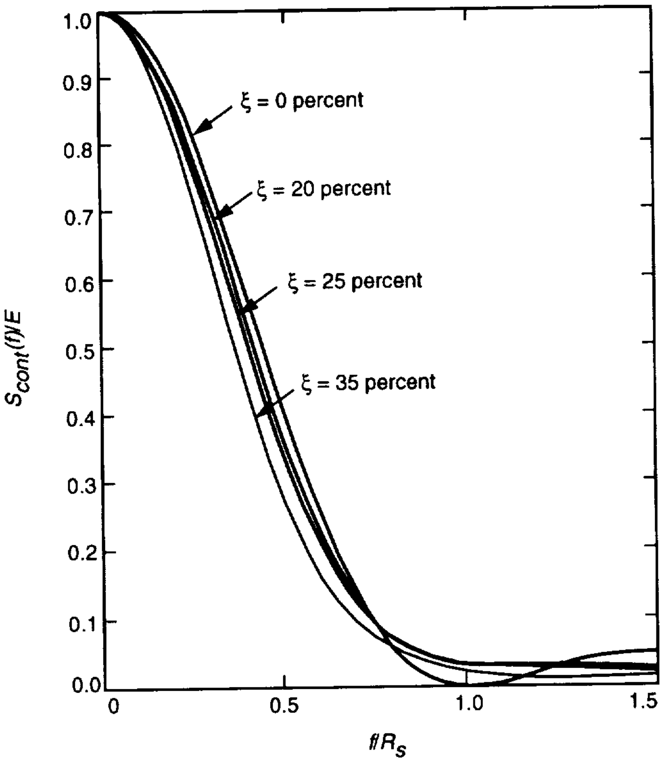


Fig. 12. Normalized power spectrum for various values of data asymmetry.

$$d(t) = \sum_{k=-\infty}^{\infty} d_k g(t + kT_s) \tag{40}$$

where $d_k = \pm 1$ with $p = q = 0.5$, and $g(t)$ is given by

$$g(t) = p(t) * h'(t) \tag{41}$$

where $*$ denotes convolution.

The impulse response of an ideal channel $h'(t)$ is given by the inverse Fourier transform of the transfer function $H'(f)$:

$$H'(f) = \begin{cases} 1 & -B < \mathrm{f} < B \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

resulting in

$$h'(t) = 2B\frac{\sin(2\pi Bt)}{2\pi Bt} \tag{43}$$

For an ideal filter and a perfect data stream, $g(t + kT)$ can be found to be [1]

$$g(t+kT_s) = \frac{1}{\pi}\left[Si\left\{2\pi B\left(t+T_s\left(k+\frac{1}{2}\right)\right)\right\} - Si\left\{2\pi B\left(t+T_s\left(k-\frac{1}{2}\right)\right)\right\}\right] \tag{44}$$

where

$$Si(x) = \int_0^x \frac{\sin(u)}{u}\, du \tag{45}$$

Figure 13 shows a plot of $g(t)$ versus the normalized time t/T_s. Note that the shape of the output is dependent on the time-bandwidth product BT_s. For $BT_s >> 1$, the degradation due to band limiting becomes negligible. As BT_s approaches 1, the rise and fall times of the output are significant when compared to the input, and the output signal is further spread in time.

To calculate the average probability of error, $P_e(\theta_e)$ and σ^2 need to be determined. Calculating $P_e(\theta_e)$ exactly is very difficult because one has to take into account all possible combinations of the digits $d_k = \pm 1$, $1 \leq |k| \leq \infty$. It is assumed that only a finite number of M pulses before and after d_0, $d_0 = 0$, are taken into account. That is, only the ISI effects of the M preceding and M subsequent bits are considered on the bit under detection. For $BT_s \geq 1$, the value of $1 \leq M \leq 2$ is sufficient. The conditional error probability may be determined using [1]

$$P_e(\theta_e) = \frac{1}{2}\frac{1}{2^{2M}} \sum_{\substack{k=2^{2M} \\ combinations}} \text{erfc}\left\{\sqrt{\frac{E_s}{N_0}}\left[1+\sum_{k\neq 0} d_k\lambda_k\right]\cos(\theta_e)\right\} \tag{46}$$

where the SSNR for this case is given by

$$\frac{E_s}{N_0} = \frac{P\sin^2(m_T)}{N_0}\int_0^{T_s} |g(t)|^2\, dt \tag{47}$$

and

$$\lambda_k = \frac{\int_0^{T_s} g(t)g(t+kT_s)\, dt}{\int_0^{T_s} |g(t)|^2\, dt} \tag{48}$$

The variance of the carrier tracking phase error is given by Eq. (17). Therefore, for $1 \leq M \leq 2$, the average error probability can be obtained by substituting Eqs. (6) and (46) into Eq. (5) and performing the numerical integration. The results are shown in Fig. 6.

III. Combined Effects on PCM/PM/NRZ Receivers

The practical PCM/PM/NRZ receivers operate in the presence of both an imperfect data stream and a band-limited channel. This part of the article studies the combined effects of an unbalanced data stream, data asymmetry, and ISI on the SER. The total SSNR degradation of the receivers due to these three

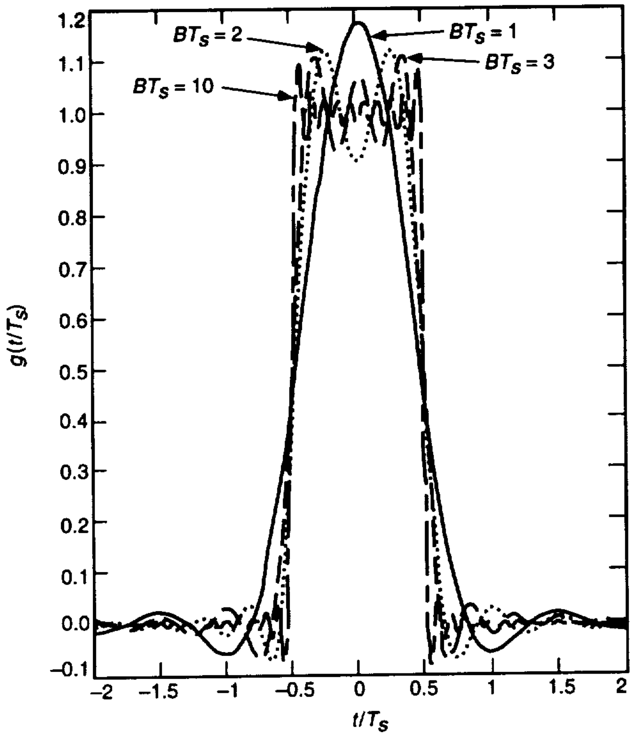


Fig. 13. Output response of the ideal filter to NRZ pulse for various values of BT_S.

undesirable effects is not the algebraic sum of the SSNR degradation due to each separate effect found in Section II. Therefore, it is necessary to study the combined effects of these three sources on the error probability performance.

For an unbalanced and asymmetric data stream, the dc, continuous, and harmonics-PSD components are given by [2]

$$S_{dc}(f) = [2p - (1 - 2\xi p_t)]^2 \delta(f) \tag{49}$$

$$\begin{aligned} S_{cont}(f) &= T_s \left[\frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2}\right] [a_1(p_t) + a_2(p, p_t, \xi)] + T_s \left[\frac{\sin^2(\pi f T_s \xi)}{(\pi f T_s)^2}\right] [a_3(p_t, \xi)] \\ &+ T_s \left[\frac{\sin^2(\pi f T_s)}{(\pi f T_s)^2}\right] [a_4(p, p_t, \xi) - a_5(p, p_t)] \end{aligned} \tag{50}$$

$$S_h(f) = 2\frac{p_t^2}{\pi^2} \sum_{m=1}^{\infty} \frac{1}{m^2} C(m, p, \xi) \delta(f - mR_s) \tag{51}$$

respectively, where

$$a_1(p_t) = p_t(1 - p_t)[1 + 2(1 - p_t)] - p_t^3 \tag{52}$$

$$a_2(p, p_t, \xi) = \left\{3p_t^3 + p_t(1-p_t)[1+2(1-2p)]\right\} \cos^2(pfT_s\xi) \tag{53}$$

$$a_3(p_t, \xi) = p_t(1 + p_t^2 - p_t) \cos^2(\pi f T_s) + p_t^3 \cos(2\pi f T_s \xi) \tag{54}$$

$$a_4(p, p_t, \xi) = p_t(1-p_t)(1-2p)[0.5 \cos(2\pi f T_s \xi) - p \sin(2\pi f T_s \xi)] \tag{55}$$

$$a_5(p, p_t) = 0.5 p_t(1-p_t)(1-2p) \tag{56}$$

$$C(m, p, \xi) = \sin^2(m\pi\xi)\left[\cos^2(m\pi\xi) - (1-2p)^2 \sin^2(m\pi\xi)\right] \tag{57}$$

Note that when $p = p_t = 1/2$ and $\xi = 0$, that is, a perfect data stream, Eqs. (49) through (51) all reduce to the result for a perfect NRZ random data stream, Eq. (13).

Once again, the presence of the two noncoherent tones (the dc component due to the imperfect data stream and the carrier tone), both at f_c, causes the mean of the phase difference $(\theta_0 - \hat{\theta})$ to deviate away from zero. The expression for the mean of this phase difference was derived to be

$$\theta_e(mean) = -\tan^{-1}\{(\tan m_T)[(2p-1) + 2\xi p(1-p)]\} \tag{58}$$

Note that this equation reduces to Eq. (25) and Eq. (36) by setting $\xi = 0$ and $p = 0$, respectively.

The same approach used in Section II will be used here to determine the average SER. Therefore, the simulations will again have to compensate for the phase difference $(\theta_0 - \hat{\theta})$. The average probability of error is given by Eq. (5), and therefore, the expressions for $P_e(\theta_e)$ and $P(\theta_e)$ must be determined to evaluate P_e. The conditional error probability in the presence of an imperfect data stream and band-limited channel is given by [2]

$$P_e(\theta_e) = p\Pr\overline{\left\{Z(T_s) < \frac{0}{\theta_e}, d_0 = +1\right\}} + q\Pr\overline{\left\{Z(T_s) > \frac{0}{\theta_e}, d_0 = -1\right\}} \tag{59}$$

where the overbar denotes statistical averaging over the joint distribution of the double infinite data sequence d_k, and the test statistic $Z(T_s)$ is given by

$$Z(T_s) = E_s\left[\pm 1 + \sum_{\substack{k=-\infty \\ k\neq 0}}^{k=\infty} d_k \lambda_k(i)\right] \cos(\theta_e) + n(T_s) \tag{60}$$

where ±1 corresponds to $d_0 = \pm 1$. It is assumed that the corrupting noise process $n(T_s)$ is a zero-mean Gaussian random variable with a variance $N_0 T_s/2$. The parameter $\lambda_k(i)$ is defined as

$$\lambda_k(i) = \frac{\displaystyle\int_0^{T_s} g(t) g_i(t + kT_s)\, dt}{\displaystyle\int_0^{T_s} |g(t)|^2\, dt}, \quad i = 1, 2, 3, 4 \tag{61}$$

where $g(t)$ is the output of the ideal filter for a perfect data stream given by Eq. (44), and $g_i(t)$ for $i = 1, 2, 3, 4$ is defined as [2]

$$g_1(t + kT_s) = \frac{1}{\pi}\left[Si\left\{2\pi B\left(t + T_s\left(k + \frac{1}{2}\right)\right)\right\} - Si\left\{2\pi B\left(t + T_s\left(k - \frac{1}{2} - \xi\right)\right)\right\}\right] \tag{62}$$

$$g_2(t + kT_s) = -\frac{1}{\pi}\left[Si\left\{2\pi B\left(t + T_s\left(k + \frac{1}{2}\right)\right)\right\} - Si\left\{2\pi B\left(t + T_s\left(k - \frac{1}{2} + \xi\right)\right)\right\}\right] \tag{63}$$

$$g_3(t + kT_s) = \frac{1}{\pi}\left[Si\left\{2\pi B\left(t + T_s\left(k + \frac{1}{2}\right)\right)\right\} - Si\left\{2\pi B\left(t + T_s\left(k - \frac{1}{2}\right)\right)\right\}\right] \tag{64}$$

$$g_4(t + kT_s) = -g_3(t + kT_s) \tag{65}$$

The variance of the carrier tracking phase error σ^2 can be obtained using Eq. (26) where

$$\alpha = \int_{-\infty}^{\infty} |H(2\pi f)|^2 S_{cont}(f)\ df \tag{66}$$

and

$$\frac{I}{C} = [2p - (1 - 2\xi p_t)]^2 \tan^2(m_T) \tag{67}$$

Again, α is the interference due to the continuous spectrum component, and I/C is the interference caused by the dc component-to-carrier power ratio. The harmonic components caused by asymmetry do not interfere with the carrier tracking because of the assumption that $2B_L << R_s$.

The average probability of error can be found by substituting Eqs. (6) and (59) into Eq. (5) and performing the numerical integration. The results are shown in Figs. 7 through 9.

IV. Description of PCM/PM Receiver Blocks

Figure 14 shows the block diagram of a PCM/PM receiver. This receiver consists of the test signal generator (TSG), the advanced receiver (ARX), and the error counter. The TSG, shown in Fig. 15, generates the deep-space spacecraft signal at an intermediate frequency (IF). The TSG's random data block controls the parameter p, and depending on this value, a balanced or unbalanced data stream is generated. The data asymmetry block controls the parameter ξ, producing an asymmetric data stream. The Appendix gives a brief description of this block. Setting $p = 0.5$ and $\xi = 0$ will produce a perfect purely random data stream. Setting $p \neq 0.5$, $\xi \neq 0$, or the combination will result in an unbalanced data stream, asymmetric data stream, or a data stream with the combined imperfections, respectively.

INPUT PARAMETERS	
SAMPLING RATE, f_S:	5×10^5 Hz
CARRIER FREQUENCY, f_C:	1×10^5 Hz
INITIAL CARRIER PHASE, θ_0:	0.0 deg
SYMBOL RATE, R_S:	1×10^4 Hz
MODULATION INDEX, m_T:	71.62 deg
TOTAL POWER/NOISE RATIO, P/N_0:	47.455 dB-Hz

CARRIER PARAMETERS	
NUMERICALLY CONTROLLED OSCILLATOR (NCO) FREQUENCY:	1×10^5 Hz
INITIAL NCO PHASE:	0.0 deg
CARRIER UPDATE RATE:	5×10^5 Hz
ONE-SIDED LOOP BANDWIDTH, B_L, CARRIER:	5.0 Hz

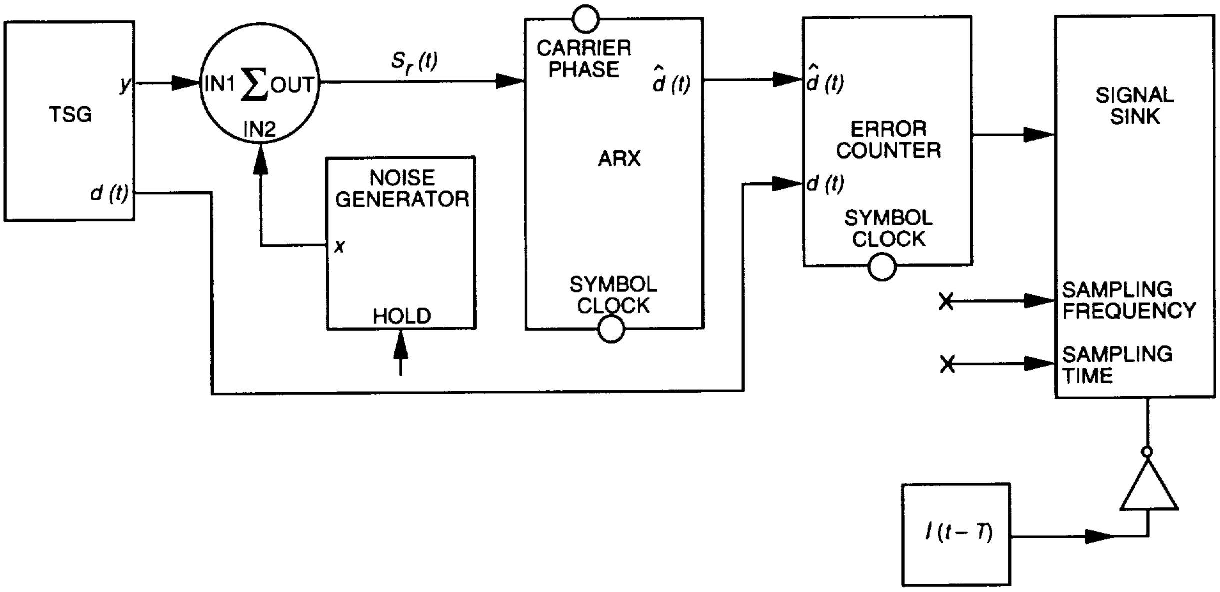


Fig. 14. PCM/PM/NRZ system block diagram as implemented in SPW.

Other TSG parameters include the following (the values shown are the ones used in simulations):

500×10^3 Hz	=	sampling rate, f_s
10×10^3 Hz	=	symbol rate, R_s
100×10^3 Hz	=	carrier frequency, f_c
0 deg	=	initial carrier phase, θ_0
71.62 deg	=	modulation index, m_T (corresponding to 1.25 rad)

and the total power-to-noise ratio P/N_0 is calculated using

$$\frac{P}{N_0} = \frac{E_s}{N_0} - 10\log_{10}(\sin^2 m_T) + 10\ \log_{10} R_s \tag{68}$$

where E_s/N_0 is the SSNR in dB.

The ARX, shown in Fig. 16, consists of the following blocks:

TSG BLOCK PARAMETERS	
SAMPLING RATE, f_S:	5×10^5 Hz
CARRIER FREQUENCY, f_C:	1×10^5 Hz
INITIAL CARRIER PHASE, θ_0:	0.0 deg
SYMBOL RATE, R_S:	1×10^4 Hz
MODULATION INDEX, m_T:	71.62 deg
TOTAL POWER/NOISE RATIO, P/N_0:	47.455 dB-Hz
PROBABILITY OF ZERO, p:	0.5
NO. OF SAMPLES/SYMBOL	50.0
DATA ASYMMETRY, ξ:	0.0

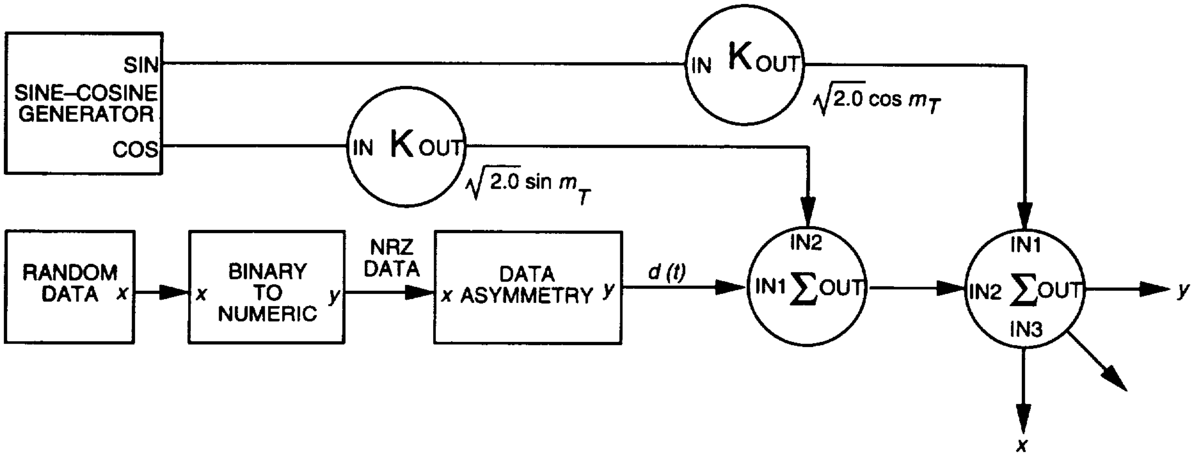


Fig. 15. Test signal generator (TSG) block diagram.

(1) The carrier PLL block estimates the incoming carrier phase and frequency and mixes it with the input signal.

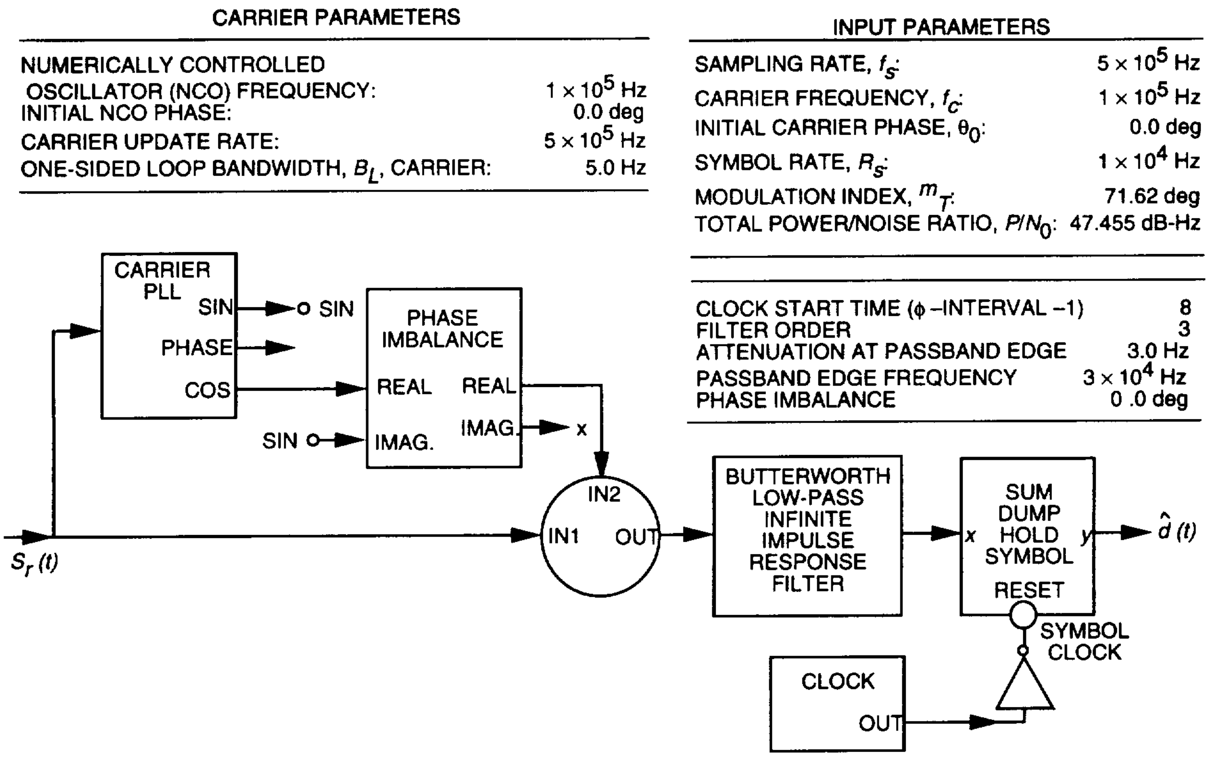
(2) The phase imbalance block adds (or subtracts) a phase to its input according to the value of the phase imbalance parameter. The input to this block in simulations is given by Eq. (4); therefore, depending on the kind of imperfect data stream present, this parameter is set to $\theta_e(mean)$, as given by Eqs. (25), (36), or (58), so that by adding this phase to the incoming phase $\hat{\theta}$, the output of the block will have a zero-mean phase error. The phase imbalance parameter is set to zero when no phase compensation is required, that is, when no unbalanced and/or asymmetric data streams are present.

(3) The Butterworth low-pass filter controls the presence of the band-limiting effect by setting the filter bandwidth B to a value that depends on the product BT_s. If no band limiting is present, the filter bandwidth B is set to 100 kHz.

(4) The ideal clock generates the timing for the sum-dump-hold symbol block. The use of an ideal clock to produce the timing instead of the digital data transition tracking loop block was for the purpose of matching the assumption made in theory, and therefore, eliminating the loss due to symbol synchronization.

(5) The sum-dump-hold block outputs the soft symbols.

Finally, the error counter block compares the soft symbols of the ARX to the transmitted symbols and outputs the number of errors N.



CARRIER PARAMETERS	
NUMERICALLY CONTROLLED OSCILLATOR (NCO) FREQUENCY:	1×10^5 Hz
INITIAL NCO PHASE:	0.0 deg
CARRIER UPDATE RATE:	5×10^5 Hz
ONE-SIDED LOOP BANDWIDTH, B_L, CARRIER:	5.0 Hz

INPUT PARAMETERS	
SAMPLING RATE, f_S:	5×10^5 Hz
CARRIER FREQUENCY, f_C:	1×10^5 Hz
INITIAL CARRIER PHASE, θ_0:	0.0 deg
SYMBOL RATE, R_S:	1×10^4 Hz
MODULATION INDEX, m_T:	71.62 deg
TOTAL POWER/NOISE RATIO, P/N_0:	47.455 dB-Hz

CLOCK START TIME (ϕ-INTERVAL -1)	8
FILTER ORDER	3
ATTENUATION AT PASSBAND EDGE	3.0 Hz
PASSBAND EDGE FREQUENCY	3×10^4 Hz
PHASE IMBALANCE	0 .0 deg

Fig. 16. Advanced receiver block diagram.

V. Discussion and Simulation Results

By substituting the expressions for the conditional probability of error $P_e(\theta_e)$ and the probability density function for the phase error $P(\theta_e)$ into Eq. (5), the SER as a function of SSNR was plotted in [1,2] for each of the cases discussed above. Using typical operating conditions of $m_T = 1.25$ rad and $2B_L/R_S = 0.001$, these theoretical plots are shown in Figs. 3 through 9 as the continuous curves. The computer simulation results are shown as the triangular, circular, and square points for variables shown therein.

Using the SPW, simulations were performed at 7-, 8-, 9- and 10-dB SSNR (E_s/N_0), and the corresponding P/N_0 was calculated. The result of each simulation was the number of errors N (produced by the error counter as a result of comparing the soft symbols to the transmitted ones). The average error probability P_e was then calculated using

$$P_e = \frac{N}{\text{number of iterations}/(f_s/R_s)} \tag{69}$$

where f_s is the sampling frequency in Hz and the fraction (f_s/R_s) is the number of samples per symbol. The number of iterations must be chosen large enough so that the simulation results have sufficient statistics. That is,

$$\text{number of iterations} = \frac{100}{\text{SER}}\left(\frac{f_s}{R_s}\right) \tag{70}$$

where SER is the symbol error rate as given by the theory. Finally, P_e was plotted versus SSNR and the results were compared to the theoretical curves presented in [1,2].

A. Unbalanced Data

To verify the performance of the receiver in the presence of an unbalanced data stream, simulations were performed for $p = 0.5$, 0.45, and 0.4.

As mentioned in Section II.B, when $p \neq 0.5$, the phase of the tone caused by the unbalanced data is noncoherent with the carrier phase, which results in a nonzero-mean phase error θ_e. In order to overcome this problem, the phase error was calculated using Eq. (25), checked by simulations, and then modified so that the resultant phase error is of zero mean.

Figure 3 shows the theoretical and simulation results when no phase modification is made to the phase error. When $p \neq 0.5$, the simulations are in disagreement with theory, and the SSNR degradation in some cases exceeds 1 dB. On the other hand, when the phase error is modified, the theoretical and simulation results are in good agreement. These results are presented in Table 1 and Fig. 4. It is obvious that as p deviates from 0.5, the performance of PCM/PM/NRZ degrades significantly, and that the degradation becomes unacceptable when $p < 0.45$. This is due to the presence of a strong dc component caused by the unbalanced data stream at the carrier frequency. The higher the deviation from 0.5, the stronger the dc component, and as a result, the worse the degradation.

Table 1. Simulation data and results for unbalanced data, separate effects.[a]

Probability of mark	E_s/N_0, dB	P/N_0, dB	No. of iterations in millions	No. of errors	P_e	P_e theory
0.5	7	47.455	6	105	8.75×10^{-4}	7.727×10^{-4}
0.5	8	48.455	28.2	126	2.23×10^{-4}	1.909×10^{-4}
0.5	9	49.455	150	116	3.87×10^{-5}	3.363×10^{-5}
0.5	10	50.455	1300	132	5.08×10^{-6}	3.872×10^{-6}
0.45	7	47.455	6.01	115	9.57×10^{-4}	1.100×10^{-3}
0.45	8	48.455	25.2	129	2.56×10^{-4}	3.100×10^{-4}
0.45	9	49.455	80.2	112	6.98×10^{-5}	6.600×10^{-5}
0.45	10	50.455	500.2	134	1.34×10^{-5}	1.100×10^{-5}
0.4	7	47.455	7	764	5.79×10^{-3}	5.400×10^{-3}
0.4	8	48.455	10.0	704	3.52×10^{-3}	3.250×10^{-3}
0.4	9	49.455	15	612	2.04×10^{-3}	2.000×10^{-3}
0.4	10	50.455	25	686	1.37×10^{-3}	1.500×10^{-3}

[a] $m = 1.25$ rad, $R_s = 1 \times 10^4$ Hz, $f_s = 5 \times 10^5$ Hz, $B_L = 5$ Hz, $2B_L/R_s = 0.001$.

B. Data Asymmetry

Since the power of the dc component generated by the asymmetric data at f_c is much less than the power of the carrier tone, the mean of the phase error will be small. The mean was calculated (Eq. (36)) and measured to be between -1.7 and -5.2 deg for ξ between 2 and 6 percent, respectively, which are the minimum and maximum values for ξ used in the simulations. The degradation due to this nonzero-mean phase error is negligible and, hence, no compensation was done to the phase error. Simulations were performed for $\xi = 2, 4$, and 6 percent. The results are shown in Table 2 and Fig. 5. Again the simulation results are in good agreement with the theoretical results (within 0.2 dB). The numerical results show that, for data asymmetry less than or equal to 2 percent, the SSNR degradation is on the order of 0.1 dB or less, and that this degradation is between 0.2 dB and 0.25 dB for data asymmetry of 6 percent and for $10^{-7} \leq \text{SER} \leq 10^{-5}$.

Table 2. Simulation data and results for data asymmetry, separate effects.[a]

Data asymmetry, percent	E_s/N_0, dB	P_t/N_0, dB	No. of iterations in millions	No. of errors	P_e	P_e theory
2	7	47.455	6.6	124	9.39×10^{-4}	8.75×10^{-4}
2	8	48.455	23	119	2.59×10^{-4}	2.20×10^{-4}
2	9	49.455	130	139	5.35×10^{-5}	3.85×10^{-5}
2	10	50.455	2100	203	4.83×10^{-6}	4.80×10^{-6}
4	7	47.455	6.6	144	1.09×10^{-3}	9.60×10^{-4}
4	8	48.455	20	137	3.43×10^{-4}	2.60×10^{-4}
4	9	49.455	115	136	5.91×10^{-5}	4.40×10^{-5}
4	10	50.455	1900	253	6.66×10^{-6}	5.50×10^{-6}
6	7	47.455	6.6	146	1.11×10^{-3}	1.30×10^{-3}
6	8	48.455	18	125	3.47×10^{-4}	2.85×10^{-4}
6	9	49.455	108	132	6.11×10^{-5}	4.70×10^{-5}
6	10	50.455	800	138	8.63×10^{-6}	6.30×10^{-6}

[a] $p = 0.5$, $m = 1.25$ rad, $R_s = 1 \times 10^4$ Hz, $f_s = 5 \times 10^5$ Hz, $B_L = 5$ Hz, $2B_L/R_s = 0.001$.

C. Band-Limited Channel

In order to test the effect of the band-limited channel on the overall performance of the system, simulations were performed for different values of the time-bandwidth product $BT_s = 1, 2$, and 3. As expected, the higher the value of the product BT_s, the better the performance of the system. The simulation results are shown in Table 3 and Fig. 6. The numerical results show that for $10^{-7} \leq \text{SER} \leq 10^{-5}$, the SSNR degradation is in the range of 1 to 1.2 dB for $BT_s = 1$, and less than 0.3 for $BT_s = 2$. The theoretical and simulation results are in good agreement. However, the simulations are a little worse than the theoretical results. This is because the theoretical results were obtained for the case when the ISI is caused by two adjacent pulses, that is, two pulses before and two pulses after the current pulse is considered in the SER calculation.

D. Combined Effects

To test the behavior of PCM/PM/NRZ receivers in the presence of the combination of the three undesirable effects, simulations were performed for different values of p, ξ, and BT_s. One of the parameters was varied as the other two remained constant. Since data imbalance and asymmetry were always present, all simulations required compensation for the phase error $\theta_e(mean)$ (Eq. (58)) so that the result is a zero-mean phase error.

Figure 7 plots the SER as a function of SSNR for a fixed data asymmetry ξ of 2 percent and $BT_s = 3$ with p, probability of mark, as a parameter. The simulation results are also shown in Table 4, and are in good agreement with the theory. The results indicate that, for $m_T = 1.25$ rad and $2B_L/R_S = 0.001$, the SER degrades seriously as p deviates from 0.45.

Table 3. Simulation data and results for a band-limited channel, separate effects.[a]

BT_s	E_s/N_0, dB	P/N_0, dB	No. of iterations in millions	No. of errors	P_e	P_e theory
1	7	47.455	6	226	1.88×10^{-3}	1.80×10^{-3}
1	8	48.455	11	141	6.41×10^{-4}	5.80×10^{-4}
1	9	49.455	30	94	1.57×10^{-4}	1.70×10^{-4}
1	10	50.455	160	88	2.75×10^{-5}	3.30×10^{-5}
2	7	47.455	6	150	1.25×10^{-3}	9.20×10^{-4}
2	8	48.455	21	146	3.48×10^{-4}	2.50×10^{-4}
2	9	49.455	105	143	6.81×10^{-5}	4.83×10^{-5}
2	10	50.455	800	137	8.56×10^{-6}	6.50×10^{-6}
3	7	47.455	6	125	1.04×10^{-3}	8.60×10^{-4}
3	8	48.455	22.6	131	2.90×10^{-4}	2.30×10^{-4}
3	9	49.455	114	126	5.53×10^{-5}	4.40×10^{-5}
3	10	50.455	800	124	7.75×10^{-6}	5.60×10^{-6}

[a] $m = 1.25$ rad, probability of mark = 0.5, $R_s = 1 \times 10^4$ Hz, $f_s = 5 \times 10^5$ Hz, $B_L = 5$ Hz, $2B_L/R_s = 0.001$.

Table 4. Simulation data and results for various probabilities of mark, combined effects.[a]

Probability of mark	E_s/N_0, dB	P/N_0, dB	No. of iterations in millions	No. of errors	P_e	P_e theory (approximate)
0.45	7	47.455	7.2	149	1.03×10^{-3}	7.80×10^{-4}
0.45	8	48.455	25.2	137	2.72×10^{-4}	2.00×10^{-4}
0.45	9	49.455	80.2	78	4.86×10^{-5}	3.60×10^{-5}
0.45	10	50.455	500.2	57	5.70×10^{-6}	5.40×10^{-6}
0.4	7	47.455	7.2	467	3.24×10^{-3}	2.20×10^{-3}
0.4	8	48.455	25.2	711	1.41×10^{-3}	1.25×10^{-3}
0.4	9	49.455	80.2	983	6.13×10^{-4}	6.50×10^{-4}
0.4	10	50.455	500.2	4.23×10^{3}	4.23×10^{-4}	3.85×10^{-4}
0.35	7	47.455	7.2	2.80×10^{3}	1.94×10^{-2}	1.50×10^{-2}
0.35	8	48.455	25.2	6.41×10^{3}	1.27×10^{-2}	1.25×10^{-2}
0.35	9	49.455	80.2	1.89×10^{4}	1.18×10^{-2}	1.00×10^{-2}
0.35	10	50.455	500.2	8.72×10^{4}	8.72×10^{-3}	9.00×10^{-3}

[a] Data asymmetry = 2 percent, $BT_s = 3$, $m = 1.25$ rad, $R_s = 1 \times 10^4$ Hz, $f_s = 5 \times 10^5$ Hz, $B_L = 5$ Hz, $2B_L/R_s = 0.001$.

Table 5 shows the simulation results obtained for various values of data asymmetry ξ with $BT_s = 3$ and $p = 0.45$. As shown in Fig. 8, the simulations are in good agreement with the theory. It is also obvious that PCM/PM/NRZ is not sensitive to data asymmetry since the SSNR degradation is between 0.1 and 0.5 dB when ξ varies between 2 and 6 percent and the SER is between 10^{-4} and 10^{-7}.

Table 5. Simulation data and results for various values of data asymmetry, combined effects.[a]

Data asymmetry, percent	E_s/N_0, dB	P/N_0, dB	No. of iterations in millions	No. of errors	P_e	P_e theory (approximate)
2	7	47.455	7.2	149	1.03×10^{-3}	7.80×10^{-4}
2	8	48.455	25.2	137	2.72×10^{-4}	2.00×10^{-4}
2	9	49.455	80.2	78	4.86×10^{-5}	3.60×10^{-5}
2	10	50.455	500.2	57	5.70×10^{-6}	5.40×10^{-6}
4	7	47.455	7.2	160	1.11×10^{-3}	9.70×10^{-4}
4	8	48.455	25.2	155	3.08×10^{-4}	2.70×10^{-4}
4	9	49.455	80.2	95	5.92×10^{-5}	4.80×10^{-5}
4	10	50.455	500.2	78	7.80×10^{-6}	6.50×10^{-6}
6	7	47.455	7.2	168	1.17×10^{-3}	1.10×10^{-3}
6	8	48.455	25.2	168	3.33×10^{-4}	2.95×10^{-4}
6	9	49.455	80.2	98	6.11×10^{-5}	5.50×10^{-5}
6	10	50.455	500.2	82	8.20×10^{-6}	7.75×10^{-6}

[a] Probability of mark = 0.45, $BT_s = 3$, $m = 1.25$ rad, $R_s = 1 \times 10^4$ Hz, $f_s = 5 \times 10^5$ Hz, $B_L = 5$ Hz, $2B_L/R_s = 0.001$.

Table 6 and Fig. 9 illustrate the SER performance in the presence of a band-limiting channel for $p = 0.45$ and $\xi = 2$ percent with BT_s as a parameter. As shown, the simulations are in good agreement with the theory, and for $BT_s = 3$, the SSNR degradation is on the order of 0.4 dB or less when the SER is between 10^{-4} and 10^{-7}.

The numerical results prove that the total SSNR degradation due to the three undesirable effects is not the algebraic sum of the SSNR degradation due to each separate effect. As an example, when the SER is 10^{-5}, the SSNR degradation when $p = 0.45$, $\xi = 2$ percent, and $BT_s = 3$ (Fig. 7) is about 0.1 dB, whereas, the algebraic sum of the SSNR degradations due to each separate effect (Figs. 4 through 6) is about 0.6 dB.

VI. Conclusion

This article studied, by computer simulations, the separate and combined effects of unbalanced data, data asymmetry, and a band-limited channel on the performance of a PCM/PM/NRZ receiver. All the simulation results were in good agreement with the theoretical results presented in [1,2]. Hence, the mathematical models presented in [1,2] can be used to predict the performance of the PCM/PM/NRZ receivers. PCM/PM/NRZ was shown to be most sensitive to the imbalance between +1's and −1's in the data stream, as the performance degradation became unacceptable when $p < 0.45$, and least sensitive to data asymmetry. For $BT_S = 3$, the SER performance was shown to be acceptable for both near-Earth and deep-space missions.

Table 6. Simulation data and results for various values of BT_s, combined effects.[a]

BT_s	E_s/N_0, dB	P/N_0, dB	No. of iterations in millions	No. of errors	P_e	P_e theory (approximate)
3	7	47.455	7.2	149	1.03×10^{-3}	7.80×10^{-4}
3	8	48.455	25.2	137	2.72×10^{-4}	2.00×10^{-4}
3	9	49.455	80.2	78	4.86×10^{-5}	3.60×10^{-5}
3	10	50.455	500.2	57	5.70×10^{-6}	5.40×10^{-6}
2	7	47.455	7.2	168	1.17×10^{-3}	8.50×10^{-4}
2	8	48.455	25.2	158	3.13×10^{-4}	2.20×10^{-4}
2	9	49.455	80.2	92	5.74×10^{-5}	3.90×10^{-5}
2	10	50.455	500.2	70	7.00×10^{-6}	6.60×10^{-6}
1	7	47.455	7.2	222	1.54×10^{-3}	1.25×10^{-3}
1	8	48.455	25.2	244	4.84×10^{-4}	3.75×10^{-4}
1	9	49.455	80.2	177	1.10×10^{-4}	1.20×10^{-4}
1	10	50.455	500.2	207	2.07×10^{-5}	2.60×10^{-5}

[a] Probability of mark = 0.45, data asymmetry = 2 percent, $m = 1.25$ rad, $R_s = 1 \times 10^4$ Hz, $f_s = 5 \times 10^6$ Hz, $B_L = 5$ Hz, $2B_L/R_s = 0.001$.

Another modulation scheme that is of interest to CCSDS is PCM/PM/Bi-ϕ, which is also known to be one of the most efficient modulation schemes in terms of bandwidth occupancy as compared to PCM/PSK/PM [4]. Mathematical models have been developed to predict the performance of PCM/PM/Bi-ϕ receivers [1,2], and these models are currently being verified by members of CCSDS and the results will be reported later.

Acknowledgments

The authors would like to thank Warren L. Martin for his continuous support and encouragement. Thanks also to Sami M. Hinedi for his invaluable comments and suggestions, and to John Gevargiz for reviewing this article.

References

[1] T. M. Nguyen, "Behavior of PCM/PM Receivers in Non-Ideal Channels, Part I: Separate Effects of Imperfect Data Streams and Bandlimiting Channels on Performances," *Report of the Proceedings of the RF and Modulation Subpanel 1E Meeting at the German Space Operations Centre, September 20–24, 1993*, CCSDS B20.0-Y-1, Consultative Committee for Space Data Systems, February 1994.

[2] T. M. Nguyen, "Behavior of PCM/PM Receivers in Non-Ideal Channels, Part II: Combined Effects of Imperfect Data Streams and Bandlimiting Channels on Performances," *Report of the Proceedings of the RF and Modulation Subpanel 1E Meeting at the German Space Operations Centre, September 20–24, 1993*, CCSDS B20.0-Y-1, Consultative Committee for Space Data Systems, February 1994.

[3] T. M. Nguyen, "The Impact of NRZ Data Asymmetry on the Performance of a Space Telemetry System," *IEEE Transactions on Electromagnetic Compatibility*, vol. 33, no. 4, November 1991.

[4] M. M. Shihabi, T. M. Nguyen, and S. M. Hinedi, "A Comparison of Telemetry Signals in the Presence and Absence of a Subcarrier," *IEEE Transactions on Electromagnetic Compatibility*, vol. 36, no. 1, February 1994.

[5] Consultative Committee for Space Data Systems, *Recommendations for Space Data System Standards, Radio Frequency and Modulation Systems, Part I, Earth Stations and Spacecraft*, CCSDS 401.0-B, Blue Book, Washington D.C.: CCSDS Secretariat, Communications and Data Systems Division (Code OS), NASA.

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

[7] J. F. Pelayo and J.-L. Gerner, "PCM/PSK/PM and PCM-SPL/PM Signals—Occupied Bandwidth and Bit Error Rate," *Report of the Proceedings of the RF and Modulation Subpanel 1E Meeting at the German Space Operations Centre, September 20–24, 1993*, CCSDS B20.0-Y-1, Consultative Committee for Space Data Systems, February 1994.

Appendix
Data Asymmetry Block

The data asymmetry block outputs NRZ asymmetric data stream y when the input x is a purely random NRZ data stream. This block was implemented in SPW using mostly delays, switches, and decision blocks. It first detects the transition that occurs at the end of every symbol using

$$trans = \frac{d_k - d_{k-1}}{2} \tag{A-1}$$

where d_k is the present symbol value and d_{k-1} is the previous symbol value, and therefore, this yields a -1 when a $+1$ to -1 transition occurs, a $+1$ when a -1 to $+1$ transition occurs, and a 0 when no transition occurs. The block then determines a threshold value T_1, which is 0 if $trans = +1$ or 0, and ξ if $trans = -1$, where ξ denotes data asymmetry. If there are N samples per symbol for the input x, P denotes the past sample value, C the current, and i the ith sample in the symbol, then, for $i = 0$, $i < N$; if $i < T_1$, then $y = P$; otherwise, if $T_1 < i < N$, then $y = C$.

The Application of Noncoherent Doppler Data Types for Deep Space Navigation

S. Bhaskaran
Navigation Systems Section

Recent improvements in computational capability and DSN technology have renewed interest in examining the possibility of using one-way Doppler data alone to navigate interplanetary spacecraft. The one-way data can be formulated as the standard differenced-count Doppler or as phase measurements, and the data can be received at a single station or differenced if obtained simultaneously at two stations. A covariance analysis, which analyzes the accuracy obtainable by combinations of one-way Doppler data, is performed and compared with similar results using standard two-way Doppler and range. The sample interplanetary trajectory used was that of the Mars Pathfinder mission to Mars. It is shown that differenced one-way data are capable of determining the angular position of the spacecraft to fairly high accuracy, but have relatively poor sensitivity to the range. When combined with single-station data, the position dispersions are roughly an order of magnitude larger in range and comparable in angular position as compared to dispersions obtained with standard two-way data types. It was also found that the phase formulation is less sensitive to data weight variations and data coverage than the differenced-count Doppler formulation.

I. Introduction

With increasing emphasis on controlling the costs of deep space missions, several options are being examined that decrease the costs of the spacecraft itself. One such option is to fly spacecraft in a noncoherent mode; that is, the spacecraft does not carry a transponder capable of coherently returning a carrier signal. Historically, one-way Doppler data have not been used as the sole data type due to the instability of spaceborne oscillators, the use of S-band (2.3-GHz) frequencies, and the corresponding error sources that could not be adequately modeled. However, with the advent of high-speed workstations and more sophisticated modeling ability, the possibility of using one-way Doppler is being reexamined. This article assesses the navigation performance of various one-way Doppler data types for use in interplanetary missions. As a representative interplanetary mission, the Mars Pathfinder spacecraft model and trajectory were used to perform the analysis. Comparisons are given between results employing Doppler data formulated as standard differenced-count Doppler (which yields a frequency measurement) as well as accumulated carrier phase (which yields a distance measurement, usually given in terms of cycles). Combinations of one-way data obtained simultaneously at two different stations and then differenced (to produce an angular type measurement) and single-station one-way data are shown to produce results that may satisfy future mission requirements.

II. Spacecraft Trajectory

In order to perform the analysis, a representative interplanetary trajectory was needed. The one used in this study is the Mars Pathfinder cruise from Earth to Mars. The spacecraft is injected into its trans-Mars trajectory on January 3, 1997, and reaches Mars on July 4, 1997. A schematic of this trajectory is shown in Fig. 1.[1] In between, there are four trajectory correction maneuvers (TCMs) (on February 2, March 3, May 5, and June 24), with mean magnitudes of 22.1, 1.4, 0.2, and 0.1 m/s, respectively. The first two are to remove an injection targeting bias that the initial interplanetary trajectory contains in order to satisfy planetary quarantine requirements. The final two are used to precisely target the spacecraft for its final approach and entry into the Martian atmosphere. Since Pathfinder goes directly from its interplanetary trajectory to atmospheric entry, the aim point of the targeting maneuvers is chosen such that the entry flight path angle is between 14.5 and 16.5 deg.[2] This corresponds to an entry corridor in the B-plane about 50-km wide in the cross-track direction. The down-track and normal direction constraints are chosen to ensure that the spacecraft reaches the landing site with a 99-percent probability of being within a 200-km down-track by 100-km cross-track ellipse.[3]

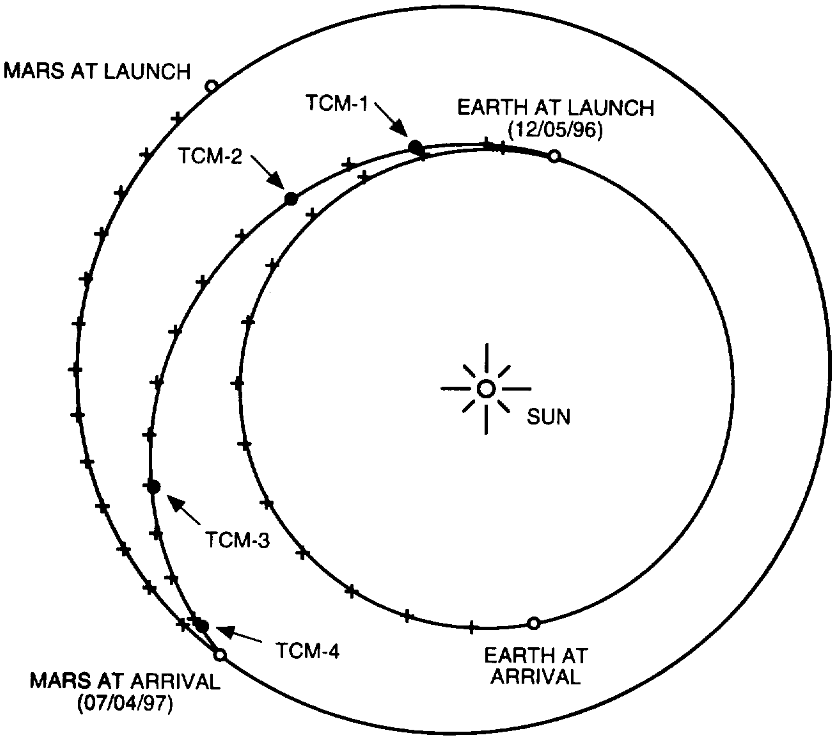


Fig. 1. Mars Pathfinder trajectory.

III. Doppler Measurement Model

When operating in one-way mode, the DSN measures the Doppler frequency of the carrier signal received from a spacecraft by comparing it with a reference frequency generated by a local oscillator. The two signals are differenced, and a counter measures the accumulated phase of the resultant signal

[1] Provided by P. H. Kallemeyn, Mars Pathfinder Navigation, Jet Propulsion Laboratory, Pasadena, California, January 1995.

[2] P. H. Kallemeyn, *Mars Pathfinder Navigation Plan*, JPL D-11349 (internal document), Jet Propulsion Laboratory, Pasadena, California, July 1994.

[3] Ibid.

over set periods of time, called the count time. The total phase change over the count time, divided by the count time, produces a measure of the Doppler shift of the incoming signal, with which the range rate of the spacecraft can be inferred. This is referred to as differenced-count Doppler, the standard measurement used for all deep space missions thus far. If, instead, the original phase data themselves are used, a measure of the change in the range of the spacecraft over the length of the pass is obtained, with the initial range at the start of the pass being an unknown. Although in principle this is a fairly powerful data type, it has not been used in the past due to operational problems associated with cycle slips, whereby the receiver momentarily loses lock with the incoming signal. Advances in technology over the years, however, have made cycle slips less frequent and, thus, there is renewed interest in examining the possibility of using the phase measurement directly as a data type.

The four data types investigated in this study were one-way Doppler, one-way differenced Doppler, one-way phase, and one-way differenced phase. In order to obtain a qualitative understanding of what information is available with these data, some simple equations will be presented. Neglecting error sources and relativistic effects for the moment, one-way Doppler data are approximately proportional to the topocentric range rate of a spacecraft:

$$f \approx f_T \frac{\dot{\rho}}{c} \tag{1}$$

where

f = the observed Doppler shift of the carrier signal

f_T = the carrier frequency transmitted by the spacecraft

$\dot{\rho}$ = the station–spacecraft range rate

c = the speed of light

Hamilton and Melbourne [1] derived a simple approximation for the topocentric range rate seen at a tracking station in terms of the cylindrical coordinates of the station and the geocentric range rate, right ascension, and declination of the spacecraft:

$$\dot{\rho} \approx \dot{r} + \omega r_s \cos\delta \sin(\omega t + \alpha_s + \lambda_s - \alpha) \tag{2}$$

where

$\dot{r}$ = the geocentric range rate of the spacecraft

α, δ = the geocentric right ascension and declination of the spacecraft

ω = the rotation rate of the Earth

α_s = the right ascension of the Sun

r_s, λ_s = the spin radius and longitude of the station

Thus, the signal seen at the station represents the sum of the geocentric velocity of the spacecraft and short term sinusoidal variations due to the rotation of the Earth. The amplitude of the sinusoidal variation is proportional to the cosine of the declination of the spacecraft, and its phase includes information about the right ascension. Now, if the signals received simultaneously at two stations are differenced, the geocentric range rate drops out of the equation and only the periodic variations are left. This implies that differenced Doppler data are incapable of directly measuring the range of the spacecraft, but can better resolve its angular position than the undifferenced data. In addition, the differenced data are nearly insensitive to short-term variations in the velocity, such as those due to short thruster firings.

If Eq. (1) is now integrated over the interval from t_0 to t, the following expression for the Doppler phase is obtained:

$$\phi_t - \phi_{t_0} \approx f_T \frac{\rho_t - \rho_{t_0}}{c} \tag{3}$$

where

$\rho =$ the topocentric range of the spacecraft at times t and t_0

$\phi =$ the measured phase of the carrier signal at times t and t_0

Thus, the phase of the received carrier signal at a given time measures the change in range from the previous time. At the beginning of the pass, there will be an unknown bias representing the initial range to the spacecraft. An analytical approximation for the difference of two range measurements received simultaneously at two stations can be written in terms of the baseline vector between them as [2]

$$\Delta\rho \approx r_B \cos\delta \cos(\alpha_B - \alpha) + z_B \sin\delta \tag{4}$$

where

$r_B =$ the baseline component normal to the Earth's spin axis

$z_B =$ the baseline component parallel to the Earth's spin axis

$\alpha_B =$ the baseline right ascension

$\alpha =$ the spacecraft right ascension

$\delta =$ the spacecraft declination

Once again, it can be seen that differencing the data removes direct information about the radial distance to the spacecraft and the result is given in terms of its angular position.

All data used in this analysis were assumed to be obtained at X-band frequencies (7.2–8.4 GHz). The differenced data types were taken when the spacecraft was visible simultaneously from two DSN stations above an elevation cutoff of 15 deg. This resulted in overlaps of roughly 4 hours in length occurring over the Goldstone–Madrid and Goldstone–Canberra baselines throughout the data arc. No data over the Canberra–Madrid baseline could be obtained.

Data scheduling was set as follows: Single-station one-way data were taken during every other pass at all three DSN sites, starting at the beginning of the Mars Pathfinder trajectory (January 3, 1997) and ending at the data cutoff on June 19, 1997. This results in roughly 14,000 points (at 10-min intervals). Two-station differenced data were scheduled at every overlap until the data cutoff date, resulting in approximately 6000 points. The assumed noise levels used were 0.1 and 1.0 cycle for phase data and 0.05 and 0.5 mm/s for the Doppler data.

IV. Orbit Determination Error Analysis

Orbit determination is composed of several steps: generation of a reference trajectory, computation of observational partial derivatives with respect to the reference trajectory, and correction of the trajectory and error model parameters using an estimation algorithm or filter. The associated error covariance of the estimated parameters is also obtained as part of this procedure. The error covariance analysis was performed using a modified version of JPL's DPTRAJ/ODP software called MIRAGE [3]. MIRAGE offers an improvement over the ODP in that it is capable of modeling time-varying stochastic parameters that have different "batch" lengths, that is, time steps over which the parameters are piecewise continuous.

In order to obtain a realistic estimate of the covariance, the dynamic forces affecting the spacecraft and the error sources affecting the data must be modeled properly. A detailed analysis of these model parameters has already been performed for the Mars Pathfinder mission;[4] the results will be summarized here. In the filter model, all known dynamic parameters and significant Doppler error sources are modeled and explicitly estimated. The dynamic parameters included the spacecraft state (position and velocity), coefficients for solar radiation pressure, random nongravitational accelerations, and spacecraft maneuvers. The solar radiation pressure and random accelerations both have three components: a radial one along the Earth line and two cross-line-of-sight ones that are mutually orthogonal to the radial direction. These are modeled as stochastic Gaussian colored noise parameters; that is, an estimate is made for the parameters within each batch, and their values from one batch to another are statistically correlated with a characteristic decorrelation time input by the user. The solar radiation pressure coefficients vary slowly over the course of the mission as the reflectivity of the spacecraft changes, so the decorrelation time of these parameters was set to 60 days. The uncertainties are roughly 5 percent of the nominal values of the coefficients. Stochastic accelerations are needed to model small thruster firings, such as those used for attitude updates. The size and frequency of these firings result in accelerations with decorrelation times of 5 to 6 days and an rms magnitude of about 2×10^{-12} km/s^2 in the radial direction and 1×10^{-12} km/s^2 in the cross-track directions. Spacecraft maneuvers are deterministic in nature and, in general, can be modeled as impulsive velocity changes placed at the midpoint of the maneuver time. Experience on previous missions has shown that the maneuver magnitude can be controlled to around 1-percent accuracy, so the a priori uncertainty in the maneuver parameters was set to 1 percent of the expected size of the change in velocity (ΔV) for each midcourse maneuver. No constraints were placed on the direction. Table 1 summarizes all of the statistical values used in the filter.

Table 1. A priori 1-σ uncertainties of filter parameters.

Parameter	A priori uncertainty	Correlation time
Position (x, y, z)	100.0 km	—
Velocity $(\dot{x}, \dot{y}, \dot{z})$	1.0 m/s	—
Solar radiation pressure coefficient (radial)	0.07	60 days
Solar radiation pressure coefficient (cross-line-of-sight)	0.02	60 days
Stochastic acceleration (radial)	2.4×10^{-12} mm/s^2	5 days
Stochastic acceleration (cross-line-of-sight)	0.8×10^{-12} mm/s^2	5 days
Maneuvers	1% of nominal value	—
Station locations (spin radius, z-height, longitude)	0.1 m	—
Troposphere (wet)	5 cm	2 hours
Troposphere (dry)	5 cm	2 hours
Ionosphere (day)	3 cm	4 hours
Ionosphere (night)	1 cm	1 hour
Pole X and Y	0.1 m	2 days
Earth rotation (UTC)	0.15 m	1 day

[4] S. W. Thurman, "Orbit Determination Filter and Modeling Assumptions for MESUR Pathfinder Guidance and Navigation Analysis," JPL Interoffice Memorandum 314.3-1075 (internal document), Jet Propulsion Laboratory, Pasadena, California, October 15, 1993.

Error sources that affect the data include media calibration errors (wet and dry troposphere, day and night ionosphere), solar plasma effects, Earth platform calibration errors (station location in cylindrical coordinates, pole location in Cartesian x- and y-coordinates), and Earth rotation (UTC). The delays in the signal caused by its path through the troposphere and ionosphere are modeled, but errors still remain. Currently, the troposphere model is good to 5 cm and the ionosphere to 3 cm.[5] The errors vary at a relatively high frequency, and so the decorrelation time is set to a few hours. The station location set and its associated uncertainties are the DE234 coordinates developed for use by the Mars Observer (MO) mission.[6] The station location uncertainties were modified to approximately account for precession and nutation modeling errors as well. These values are assumed fixed for the duration of the Pathfinder trajectory. The polar motion and UTC variations can be predicted by the DSN to a level of around 10 to 15 cm, and they vary on the order of 1 to 2 days. The a priori uncertainties of these error model parameters, along with their characteristic decorrelation time if they are stochastic variables, are also shown in Table 1. One point to note is that the Mars ephemeris uncertainties were not included in the filter. This was done so that the computed dispersions reflect only the strengths and weaknesses of the data in determining the spacecraft trajectory.

When one-way Doppler data are used, several additional error sources must also be taken into account. For single-station data, the largest error source is the frequency drift of the spacecraft oscillator. Ultrastable oscillators of the class used by the Galileo and Mars Observer spacecraft are expected to be stable to around 1 part in 10^{12} over time spans of around a day. Over longer time spans, however, the frequency will wander and must be modeled. The method used to model this error source is to treat the bias as a random walk parameter. Qualitatively, the random walk model allows the parameter to move away from its value at the previous batch time step by an amount constrained by its given a priori uncertainty. It differs from a Gaussian white or colored noise stochastic parameter in that the parameter does not simply oscillate around its mean value, but is allowed to wander from one time step to the next. This model was also intended to approximately account for solar plasma fluctuations, which induce frequency variations on the order of 1 part in 10^{14} over 1 day. For this study, a fairly modest stability of 1 part in 10^9 over the course of a day was assumed to be the nominal. The value for the oscillator bias is updated every hour, and its a priori sigma corresponds to the change in frequency over an hour expected for the given stability.

The one-way Doppler phase formulation requires six additional parameters in the estimate list. Phase data is measured by counting the integer number of zero crossings of the signal; a resolver then determines the fractional portion of the phase at a given time. Initially, however, there will be an ambiguity in the number of cycles it took for the signal to reach the ground and the phase when the receiver locks onto the signal. To account for this, a phase bias at all three DSN stations is included in the filter. The a priori uncertainty of the bias is set to 1000 cycles (essentially infinity), and the parameter is reset at the beginning of each pass. Also, during data acquisition, the station clocks have small drifts relative to a time standard, which cause the phase count to drift as well. The drift is calibrated at the stations using data from the Global Positioning System, but residual errors remain. The magnitude with which the drift manifests itself in the phase count is about 6×10^{-4} cycles/s, so a phase drift parameter with this value for the a priori uncertainty is also included in the filter. Once again, the parameter is reset at the beginning of each pass.

The primary advantage of using differenced data is that the spacecraft oscillator drift is effectively canceled out when the single-station Doppler data are differenced, thus removing a major error source. However, an additional error source will appear: the asynchronicity of the clocks at the two receiving stations. Currently, the clocks are calibrated to about the 5-ns level (based on examination of frequency

[5] *Deep Space Network System Functional Requirements and Design: Tracking System (1988 Through 1993)*, JPL D-1662, Rev. C (internal document), Jet Propulsion Laboratory, Pasadena, California, pp. 3-4, April 15, 1993.

[6] W. M. Folkner, "DE234 Station Locations and Covariance for Mars Observer," JPL Interoffice Memorandum 335.1-92-013 (internal document), Jet Propulsion Laboratory, Pasadena, California, May 26, 1992.

and timing standard reports distributed weekly by the DSN) between each pair of stations. Thus, a parameter that represents this timing mismatch is added to the filter estimate list. In addition, the differenced phase data still require parameters to model the phase bias and drift which, in this case, are errors in the differenced phase measurement due to relative clock drifts between the two station pairs. The magnitudes of the uncertainties are kept the same as before. All one-way measurement error parameters and uncertainties are summarized in Table 2.

Table 2. A priori 1-σ uncertainties of one-way measurement error parameters.

Parameter	A priori uncertainty	Correlation time
Frequency bias	0.366 Hz	Random walk, value reset every hour
Phase bias	1000 cycles	White noise, value reset at each pass
Phase drift	6.0×10^{-4} cycles/s	White noise, value reset at each pass
Clock offset	5 ns	White noise, value reset at each pass

V. Results

Although normally the results of a covariance analysis of an interplanetary trajectory are given in terms of encounter coordinates, the so-called B-plane system, it is more instructive in this case to present the uncertainties in radial–transverse–normal (RTN) coordinates. In RTN coordinates, the radial direction is along the Earth–spacecraft vector, the transverse direction is in the plane defined by the radius and the velocity vector, and the normal direction is perpendicular to both, forming an orthogonal triad. When viewed in this frame, it is easier to see in which direction the various data types have their greatest strength.

Table 3 shows the results of the covariance analysis in RTN coordinates for all combinations of data tried thus far. The first row in the table is a "nominal" result using a standard tracking schedule for Pathfinder that includes standard two-way Doppler and range. It can be seen that the radial uncertainty is best determined, with the cross-line-of-sight directions being marginally worse with a maximum uncertainty of 7.2 km. These results when mapped to the Mars B-plane are sufficient to meet the requirements of Pathfinder.

The second and third rows in the table were obtained using only one-way phase data, weighted at 0.1 and 1.0 cycle, respectively. The result clearly shows the ability of the differential data type to determine the angular position of the spacecraft as seen from the Earth. Using a data weight of 0.1 cycle, the normal direction is determined to 11.6 km, which compares fairly well with the 7.2-km result using Doppler and range. The uncertainty in the transverse direction does not compare quite as well, about a factor of three times worse than the nominal, but is still at a reasonable magnitude. The radial direction, however, is very poorly determined, with the uncertainty using differenced-phase data being about two orders of magnitude worse than the standard case. Changing the data weight from 0.1 to 1.0 cycle has little effect in the transverse and normal directions but degrades the radial sigma by around 30 percent.

For comparison, the uncertainties using differenced one-way data formulated as Doppler frequency measurements were also examined (rows 4 and 5 in Table 3). The results are fairly similar to those of differenced-phase data in the transverse and normal directions when the tighter data weight was used on the differenced Doppler. With the data weighted at 0.5 mm/s, however, the numbers are degraded considerably, especially in the radial direction.

Due to its inability to effectively discern the range to the spacecraft, it is highly unlikely that one-way differenced data alone would be sufficient to satisfy the navigation requirements of any realistic missions. It is desirable, therefore, to augment the differenced data with another data type, the obvious choice

Table 3. 1-σ dispersion ellipses in RTN coordinates.

No.	Data type(s) used	Data weight	$\sigma(R \times T \times N)$, km
1	2-way Doppler + 2-way range	0.05 mm/s 2.0 m	3.9 x 6.4 x 7.2
2	Differenced 1-way phase	0.1 cycle	360.9 x 20.3 x 11.6
3	Differenced 1-way phase	1.0 cycle	476.8 x 23.9 x 12.1
4	Differenced 1-way Doppler	0.05 mm/s	428.5 x 23.7 x 11.3
5	Differenced 1-way Doppler	0.5 mm/s	1307.0 x 63.3 x 19.3
6	Differenced 1-way phase + 1-way phase	0.1 cycle 0.1 cycle	66.4 x 10.8 x 11.5
7	Differenced 1-way phase + 1-way phase	1.0 cycle 1.0 cycle	68.7 x 12.1 x 12.1
8	Differenced 1-way Doppler + 1-way Doppler	0.05 mm/s 0.05 mm/s	76.9 x 12.7 x 11.1
9	Differenced 1-way Doppler + 1-way Doppler	0.5 mm/s 0.5 mm/s	254.1 x 33.7 x 18.7
10	Differenced 1-way phase + 2-way Doppler	0.1 cycle 0.05 mm/s	6.7 x 8.3 x 11.1
11	Differenced 1-way Doppler + 2-way Doppler	0.05 mm/s 0.05 mm/s	6.8 x 8.4 x 10.8
12	2-way Doppler	0.05 mm/s	14.4 x 14.4 x 23.7

being single-station one-way data. Rows 6 and 7 in Table 3 show the results of combining one-way phase with differenced phase at the two data weights. The effect is quite dramatic in the radial direction, with the uncertainty brought down from 360.9 and 476.8 km to 66.4 and 68.7 km. This is still over an order of magnitude larger than the nominal case, but it is now at a level that could satisfy mission requirements. In the transverse direction, the uncertainties were brought down to very near the values of the nominal. The additional data had almost no effect in the normal direction. It is interesting to note that, with the additional data, the data weight made very little difference in the final results.

The same effect is seen when one-way Doppler data are added to differenced one-way Doppler at the tight data weight (row 8 of Table 3). The uncertainty values in the transverse and normal directions are now fairly close to those obtained with the phase data, and the radial sigma is only worse by around 15 percent. The case with the lower data weight (row 9 of Table 3), however, does not show similar behavior. The radial sigma has been brought down by an order of magnitude, but its value is still too large to be of use in many missions.

Rows 10 and 11 in Table 3 show the results of using differenced phase and Doppler augmented by standard two-way Doppler data at a rate of one pass per week. This result is included to show what to expect if a spacecraft has a transponder on board but with no ranging capability. These values indicate that navigation performance is only slightly degraded if two-way range is replaced by the differenced one-way data types. Comparison with the final row in the table (2-way Doppler only) shows that the differenced data type improves the solution by a factor of two in all three components.

The results so far using one-way data assume a spacecraft oscillator stability of one in 10^9 over the course of a day. The question can then be raised as to how a better or worse oscillator would affect the orbit determination accuracies. The effect would be negligible if only the differenced data types were used, but it will make a difference when single-station data are added. Figures 2 and 3 present

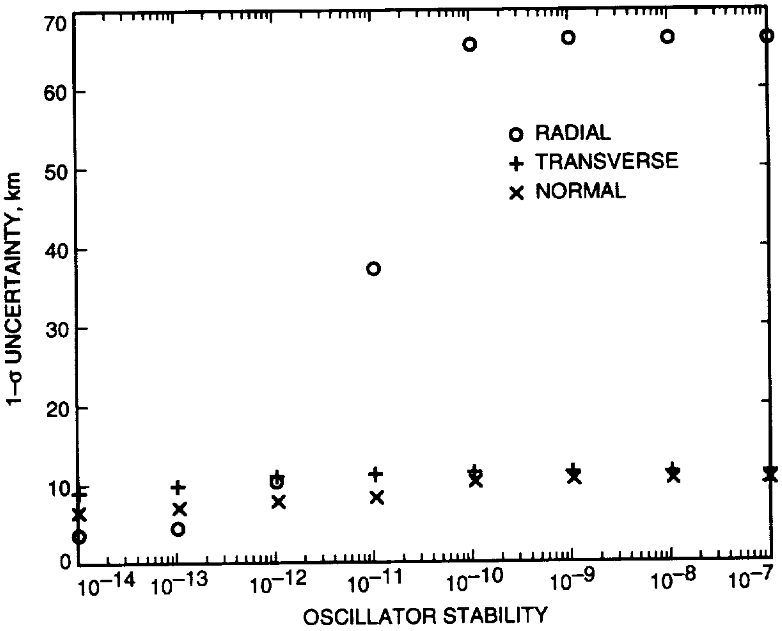


Fig. 2. Sensitivity of position uncertainty to oscillator stability for differenced-phase plus phase data.

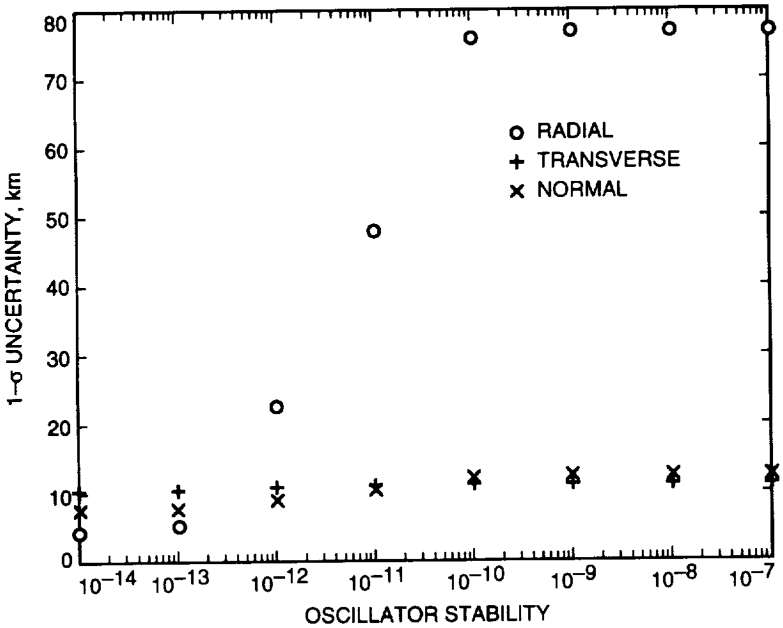


Fig. 3. Sensitivity of position uncertainty to oscillator stability for differenced-Doppler plus Doppler data.

the results when the oscillator stability varies from one part in 10^7 to one in 10^{14} over 1 day for the differenced-phase plus phase and differenced-Doppler plus Doppler cases, respectively. In both cases, the tighter data weight was assumed. As can be seen from these plots, there is a sharp knee in the curve that takes place at around the 10^{10} value in the radial directions for both phase and Doppler. The transverse and normal sigmas change very little as a function of oscillator stability. At a stability level of 10^{12}, the phase formulation case is now quite comparable in all three components to the standard two-way Doppler and range results, and the Doppler formulation is only slightly worse. Further improvements in stability do not seem to make much difference. This implies that a spacecraft carrying an ultrastable oscillator (USO) of the class used by Galileo or Mars Observer can conceivably approach the navigation accuracies achieved with two-way data types.

Another useful figure of merit is the amount of single-station one-way data employed. The nominal results are based on a dense tracking schedule of using every other available pass. Figures 4 and 5 present the results if the amount of single-station data is reduced to one pass per day, one pass per week, and one pass per month (the differenced data are assumed to remain at the nominal schedule, and the tight data weight was used). Once again, it can be seen that the transverse and normal sigmas are affected very little. The radial sigmas, however, show small changes when the data are thinned to once per day, and then a marked degradation when thinned further. The effect is more pronounced in the case of the differenced-phase Doppler formulation, with the radial sigma dropping from its nominal value of around 80 km to a worst case of nearly 200 km. The phase formulation does not suffer as much, as the decrease is only from 65 to 120 km.

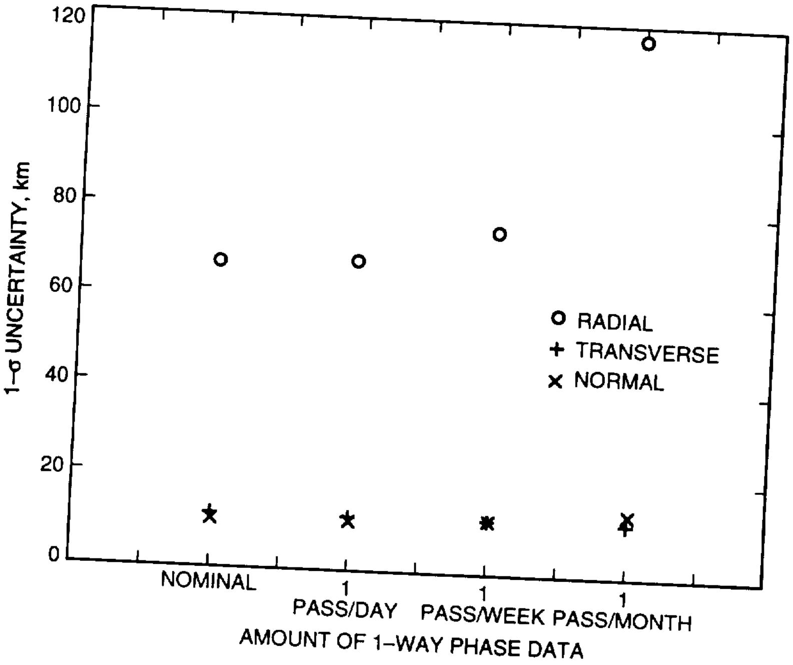


Fig. 4. Sensitivity of position uncertainty to amount of single-station data coverage for differenced-phase plus phase data.

VI. Conclusions

The results of this study suggest that a combination of single-station and two-station differenced one-way data types may be a realistic option for some interplanetary missions. This may be somewhat surprising because it has long been assumed that a very stable frequency is needed to render one-way data usable. However, it has been shown here that, with a modest oscillator and the proper mathematical formulation of the data and filter, reasonable results can be obtained by combining data that have different strengths. In particular, the estimation of the spacecraft's angular position in the sky can be nearly as good as with standard data types, although the spacecraft's radial position is relatively poorly determined. If a very good oscillator (stability of 1 part in 10^{12} over a day, or better) is available, then the accuracy in all three components may approach those obtained with standard navigation data types. One point to note, though, is that the oscillator stabilities were measured over a day. For a noncoherent system to be

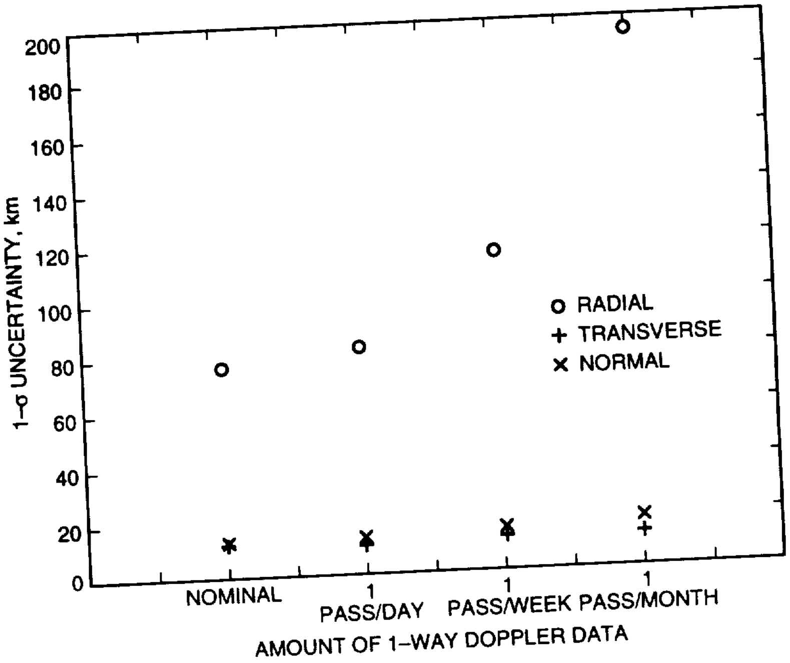


Fig. 5. Sensitivity of position uncertainty to the amount of single-station data coverage for differenced-Doppler plus Doppler data.

confidently used would require preflight testing of the oscillator over these time periods, something that has not generally been done in the past. Also, the results indicate that the phase formulation of Doppler data is superior in some respects to the differenced-phase Doppler formulation in terms of navigation accuracies. At the tight data weights and with good data coverage, the values are comparable, but the phase data show less sensitivity to decreasing data weights or coverage.

In practice, the choice of using noncoherent data types for navigation depends on the particular mission scenario and its requirements. In the case of the Mars Pathfinder mission, the geometry of the trajectory is such that the radial uncertainty maps almost completely into the time-of-flight direction (parallel to the incoming asymptote of the trajectory) in the Mars B-plane. Since the critical requirement is to maintain the proper entry angle (determined by the components perpendicular to the incoming asymptote), the degradation in performance is not severe. For example, if the entire Earth–Mars transfer were navigated using only differenced and single-station one-way phase, the probability of successful entry is still approximately 70 percent[7] (the probability is over 99 percent using two-way Doppler data). This value is obviously too low for Pathfinder to use noncoherent data as its baseline, but it is acceptable as a backup if the transponder fails. If the spacecraft were to go into orbit, however, the navigation accuracies using noncoherent data might be adequate, depending on other factors, such as propellant constraints, orbit maintenance requirements, etc. For missions whose geometry results in the radial sigma being of primary importance though, the switch to a noncoherent navigation system may not be advisable. Ultimately, the trade-off between cost and performance must be evaluated on a mission-by-mission basis, and no one answer is applicable to all cases.

[7] P. H. Kallemeyn, personal communication, Navigation Systems Section, Jet Propulsion Laboratory, Pasadena, California, January 1995.

Acknowledgments

The author would like to thank Vince Pollmeier, Sam Thurman, and Pieter Kallemeyn for their valuable suggestions and insights for this study. Also, Peter Wolff's help in using the MIRAGE software set is deeply appreciated.

References

[1] T. W. Hamilton and W. G. Melbourne, "Information Content of a Single Pass of Doppler Data From a Distant Spacecraft," *JPL Space Programs Summary 37-39, March-April 1966*, vol. III, pp. 18–23, May 31, 1966.

[2] S. W. Thurman, "Deep-Space Navigation With Differenced Data Types Part I: Differenced Range Information Content," *The Telecommunications and Data Acquisition Progress Report 42-103, July-September 1990*, Jet Propulsion Laboratory, Pasadena, California, pp. 47–60, November 15, 1990.

[3] J. R. Guinn and P. J. Wolff, "TOPEX/Poseidon Operational Orbit Determination Results Using Global Positioning Satellites," AAS Paper 93-573, presented at the AAS/AIAA Astrodynamics Specialist Conference, Victoria, British Columbia, Canada, August 16–19, 1993.

Multiple Turbo Codes for Deep-Space Communications

D. Divsalar and F. Pollara
Communications Systems Research Section

In this article, we introduce multiple turbo codes and a suitable decoder structure derived from an approximation to the maximum a posteriori probability (MAP) decision rule, which is substantially different from the decoder for two-code-based encoders. We analyze the effect of interleaver choice on the weight distribution of the code, and we describe simulation results on the improved performance of these new codes.

I. Introduction

Coding theorists have traditionally attacked the problem of designing good codes by developing codes with a lot of structure, which lends itself to feasible decoders, although coding theory suggests that codes chosen "at random" should perform well if their block size is large enough. The challenge to find practical decoders for "almost" random, large codes has not been seriously considered until recently. Perhaps the most exciting and potentially important development in coding theory in recent years has been the dramatic announcement of "turbo codes" by Berrou et al. in 1993 [1]. The announced performance of these codes was so good that the initial reaction of the coding establishment was deep skepticism, but recently researchers around the world have been able to reproduce those results [3,4]. The introduction of turbo codes has opened a whole new way of looking at the problem of constructing good codes and decoding them with low complexity.

It is claimed these codes achieve near-Shannon-limit error correction performance with relatively simple component codes and large interleavers. A required E_b/N_o of 0.7 dB was reported for a bit error rate (BER) of 10^{-5} [1]. However, some important details that are necessary to reproduce these results were omitted. The purpose of this article is to shed some light on the accuracy of these claims and to extend these results to multiple turbo codes with more than two component codes.

The original turbo decoder scheme, for two component codes, operates in serial mode. For multiple-code turbo codes, we found that the decoder, based on the optimum maximum a posteriori (MAP) rule, must operate in parallel mode, and we derived the appropriate metric, as illustrated in Section III.

II. Parallel Concatenation of Convolutional Codes

The codes considered in this article consist of the parallel concatenation of multiple convolutional codes with random interleavers (permutations) at the input of each encoder. This extends the analysis reported in [4], which considered turbo codes formed from just two constituent codes. Figure 1 illustrates

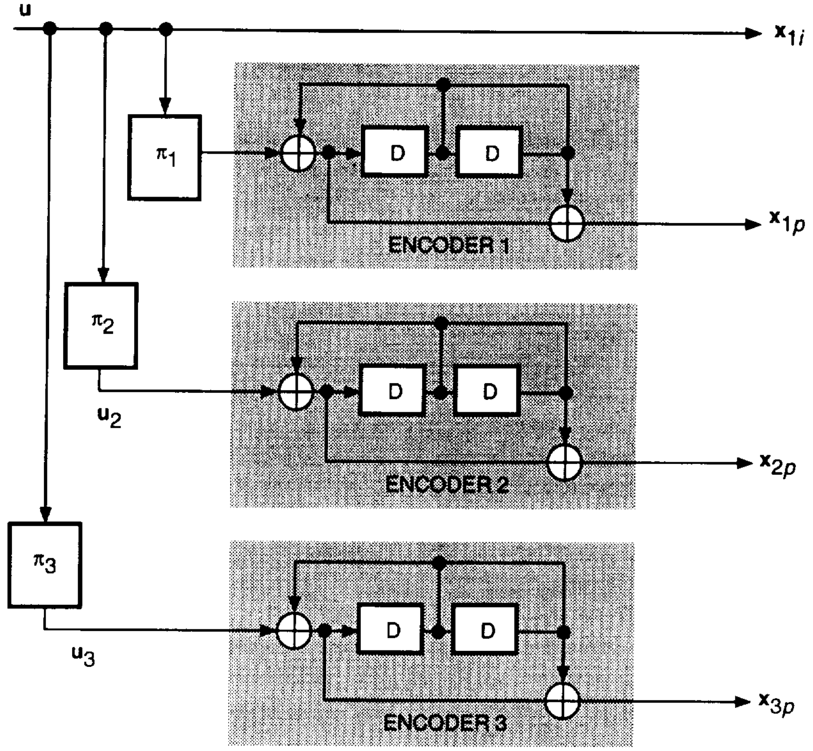


Fig. 1. Example of encoder with three codes.

a particular example that will be used in this article to verify the performance of these codes. The encoder contains three recursive binary convolutional encoders, with M_1, M_2 and M_3 memory cells, respectively. In general, the three component encoders may not be identical and may not have identical code rates. The first component encoder operates directly (or through π_1) on the information bit sequence $\mathbf{u} = (u_1, \cdots, u_N)$ of length N, producing the two output sequences $\mathbf{x}_{1i}$ and $\mathbf{x}_{1p}$. The second component encoder operates on a reordered sequence of information bits, $\mathbf{u}_2$, produced by an interleaver, π_2, of length N, and outputs the sequence $\mathbf{x}_{2p}$. Similarly, subsequent component encoders operate on a reordered sequence of information bits, $\mathbf{u}_j$, produced by interleaver π_j, and output the sequence $\mathbf{x}_{jp}$. The interleaver is a pseudorandom block scrambler defined by a permutation of N elements with no repetitions: A complete block is read into the the interleaver and read out in a specified (fixed) random order. The same interleaver is used repeatedly for all subsequent blocks. Figure 1 shows an example where a rate $r = 1/n = 1/4$ code is generated by three component codes with $M_1 = M_2 = M_3 = M = 2$, producing the outputs $\mathbf{x}_{1i} = \mathbf{u}$, $\mathbf{x}_{1p} = \mathbf{u} \cdot g_b/g_a$, $\mathbf{x}_{2p} = \mathbf{u}_2 \cdot g_b/g_a$, and $\mathbf{x}_{3p} = \mathbf{u}_3 \cdot g_b/g_a$ (here π_1 is assumed to be an identity, i.e., no permutation), where the generator polynomials g_a and g_b have octal representation $(7)_{octal}$ and $(5)_{octal}$, respectively. Note that various code rates can be obtained by proper puncturing of $\mathbf{x}_{1p}$, $\mathbf{x}_{2p}$, $\mathbf{x}_{3p}$, and even $\mathbf{x}_{1i}$ if the decoder works (for an example, see Section IV). The design of the constituent convolutional codes, which are not necessarily optimum convolutional codes, is still under investigation. It was suggested in [5] that good codes are obtained if g_a is a primitive polynomial.

We use the encoder in Fig. 1 to generate an $(n(N+M), N)$ block code, where the M tail bits of code 2 and code 3 are not transmitted. Since the component encoders are recursive, it is not sufficient to set the last M information bits to zero in order to drive the encoder to the all-zero state, i.e., to *terminate* the trellis. The termination (tail) sequence depends on the state of each component encoder after N bits, which makes it impossible to terminate all component encoders with M predetermined tail bits. This issue, which had not been resolved in previously proposed turbo code implementations, can be dealt with by applying the method described in [4], which is valid for any number of component codes.

A. Weight Distribution

In order to estimate the performance of a code, it is necessary to have information about its minimum distance, weight distribution, or actual code geometry, depending on the accuracy required for the bounds or approximations. The challenge is in finding the pairing of codewords from each individual encoder, induced by a particular set of interleavers. Intuitively, we would like to avoid joining low-weight codewords from one encoder with low-weight words from the other encoders. In the example of Fig. 1, the component codes have minimum distances 5, 2, and 2. This will produce a worst-case minimum distance of 9 for the overall code. Note that this would be unavoidable if the encoders were not recursive since, in this case, the minimum weight word for all three encoders is generated by the input sequence $\mathbf{u} = (00\cdots0000100\cdots000)$ with a single "1," which will appear again in the other encoders, for any choice of interleavers. This motivates the use of recursive encoders, where the key ingredient is the recursiveness and not the fact that the encoders are systematic. For our example, the input sequence $\mathbf{u} = (00\cdots00100100\cdots000)$ generates a low-weight codeword with weight 6 for the first encoder. If the interleavers do not "break" this input pattern, the resulting codeword's weight will be 14. In general, weight-2 sequences with $2 + 3t$ zeros separating the 1's would result in a total weight of $14 + 6t$ if there were no permutations. By contrast, if the number of zeros between the ones is not of this form, the encoded output is nonterminating until the end of the block, and its encoded weight is very large unless the sequence occurs near the end of the block.

With permutations before the second and third encoders, a weight-2 sequence with its 1's separated by $2 + 3t_1$ zeros will be permuted into two other weight-2 sequences with 1's separated by $2 + 3t_i$ zeros, $i = 2, 3$, where each t_i is defined as a multiple of $1/3$. If any t_i is not an integer, the corresponding encoded output will have a high weight because then the convolutional code output is nonterminating (until the end of the block). If all t_i's are integers, the total encoded weight will be $14 + 2\sum_{i=1}^{3} t_i$. Thus, one of the considerations in designing the interleaver is to avoid integer triplets (t_1, t_2, t_3) that are simultaneously small in all three components. In fact, it would be nice to design an interleaver to guarantee that the smallest value of $\sum_{i=1}^{3} t_i$ (for integer t_i) grows with the block size N.

For comparison, we consider the same encoder structure in Fig. 1, except with the roles of g_a and g_b reversed. Now the minimum distances of the three component codes are 5, 3, and 3, producing an overall minimum distance of 11 for the total code without any permutations. This is apparently a better code, but it turns out to be inferior as a turbo code. This paradox is explained by again considering the critical weight-2 data sequences. For this code, weight-2 sequences with $1 + 2t_1$ zeros separating the two 1's produce self-terminating output and, hence, low-weight encoded words. In the turbo encoder, such sequences will be permuted to have separations $1 + 2t_i, i = 2, 3$, for the second and third encoders, where now each t_i is defined as a multiple of $1/2$. But now the total encoded weight for integer triplets (t_1, t_2, t_3) is $11 + \sum_{i=1}^{3} t_i$. Notice how this weight grows only half as fast with $\sum_{i=1}^{3} t_i$ as the previously calculated weight for the original code. If $\sum_{i=1}^{3} t_i$ can be made to grow with block size by the proper choice of an interleaver, then clearly it is important to choose component codes that cause the overall weight to grow as fast as possible with the individual separations t_i. This consideration outweighs the criterion of selecting component codes that would produce the highest minimum distance if unpermuted.

There are also many weight-n, $n = 3, 4, 5, \cdots$, data sequences that produce self-terminating output and, hence, low encoded weight. However, as argued below, these sequences are much more likely to be broken up by the random interleavers than the weight-2 sequences and are, therefore, likely to produce nonterminating output from at least one of the encoders. Thus, turbo code structures that would have low minimum distances if unpermuted can still perform well if the low-weight codewords of the component codes are produced by input sequences with weight higher than two.

B. Random Interleavers

Now we briefly examine the issue of whether one or more random interleavers can avoid matching small separations between the 1's of a weight-2 data sequence with equally small separations between the 1's of its permuted version(s). Consider, for example, a particular weight-2 data sequence $(\cdots 001001000 \cdots)$, which corresponds to a low-weight codeword in each of the encoders of Fig. 1. If we randomly select an interleaver of size N, the probability that this sequence will be permuted into another sequence of the same form is roughly $2/N$ (assuming that N is large and ignoring minor edge effects). The probability that such an unfortunate pairing happens for at least one possible position of the original sequence $(\cdots 001001000 \cdots)$ within the block size of N is approximately $1 - (1 - 2/N)^N \approx 1 - e^{-2}$. This implies that the minimum distance of a two-code turbo code constructed with a random permutation is not likely to be much higher than the encoded weight of such an unpermuted weight-2 data sequence, e.g., 14 for the code in Fig. 1. (For the worst-case permutations, the d_{min} of the code is still 9, but these permutations are highly unlikely if chosen randomly.) By contrast, if we use three codes and two different interleavers, the probability that a particular sequence $(\cdots 001001000 \cdots)$ will be reproduced by both interleavers is only $(2/N)^2$. Now the probability of finding such an unfortunate data sequence somewhere within the block of size N is roughly $1 - \left[1 - (2/N)^2\right]^N \approx 4/N$. Thus, it is probable that a three-code turbo code using two random interleavers will see an increase in its minimum distance beyond the encoded weight of an unpermuted weight-2 data sequence. This argument can be extended to account for other weight-2 data sequences that may also produce low-weight codewords, e.g., $(\cdots 00100(000)^t 1000 \cdots)$, for the code in Fig. 1. For comparison, let us consider a weight-3 data sequence such as $(\cdots 0011100 \cdots)$, which for our example corresponds to the minimum distance of the code (using no permutations). The probability that this sequence is reproduced with one random interleaver is roughly $6/N^2$, and the probability that some sequence of the form $(\cdots 0011100 \cdots)$ is paired with another of the same form is $1 - (1 - 6/N^2)^N \approx 6/N$. Thus, for large block sizes, the bad weight-3 data sequences have a small probability of being matched with bad weight-3 permuted data sequences, even in a two-code system. For a turbo code using three codes and two random interleavers, this probability is even smaller, $1 - \left[1 - (6/N^2)^2\right]^N \approx 36/N^3$. This implies that the minimum distance codeword of the turbo code in Fig. 1 is more likely to result from a weight-2 data sequence of the form $(\cdots 001001000 \cdots)$ than from the weight-3 sequence $(\cdots 0011100 \cdots)$ that produces the minimum distance in the unpermuted version of the same code. Higher weight sequences have an even smaller probability of reproducing themselves after being passed through the random interleavers.

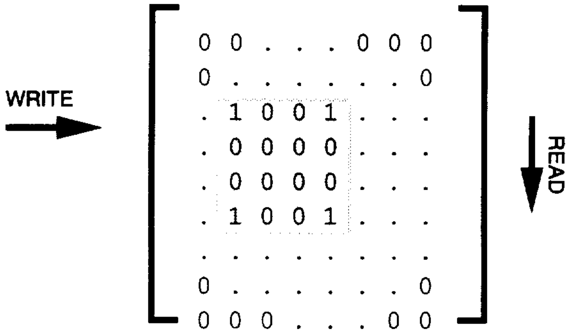
For a turbo code using q codes and $q-1$ interleavers, the probability that a weight-n data sequence will be reproduced somewhere within the block by all $q-1$ permutations is of the form $1 - \left[1 - (\beta/N^{n-1})^{q-1}\right]^N$, where β is a number that depends on the weight-n data sequence but does not increase with block size N. For large N, this probability is proportional to $(1/N)^{nq-n-q}$, which falls off rapidly with N, when n and q are greater than two. Furthermore, the symmetry of this expression indicates that increasing either the weight of the data sequence n or the number of codes q has roughly the same effect on lowering this probability.

In summary, from the above arguments, we conclude that weight-2 data sequences are an important factor in the design of the component codes, and that higher weight sequences have successively decreasing importance. Also, increasing the number of codes and, correspondingly, the number of interleavers, makes it more and more likely that the bad input sequences will be broken up by one or more of the permutations.

The minimum distance is not the most important characteristic of the turbo code, except for its asymptotic performance, at very high E_b/N_o. At moderate signal-to-noise ratios (SNRs), the weight distribution for the first several possible weights is necessary to compute the code performance. Estimating the complete weight distribution of these codes for large N and fixed interleavers is still an open problem. However, it is possible to estimate the weight distribution for large N for random interleavers by using probabilistic arguments. (See [4] for further considerations on the weight distribution).

C. Design of Nonrandom and Partially Random Interleavers

Interleavers should be capable of spreading low-weight input sequences so that the resulting codeword has high weight. Block interleavers, defined by a matrix with ν_r rows and ν_c columns such that $N = \nu_r \times \nu_c$, may fail to spread certain sequences. For example, the weight-4 sequence shown in Fig. 2 cannot be broken



$$\begin{bmatrix} 0 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & . & . & . & . & . & . & 0 \\ . & 1 & 0 & 0 & 1 & . & . & . \\ . & 0 & 0 & 0 & 0 & . & . & . \\ . & 0 & 0 & 0 & 0 & . & . & . \\ . & 1 & 0 & 0 & 1 & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & . & 0 \\ 0 & 0 & 0 & . & . & . & 0 & 0 \end{bmatrix}$$

Fig. 2. Example where a block interleaver fails to "break" the input sequence.

by a block interleaver. In order to break such sequences, random interleavers are desirable, as discussed above. (A method for the design of nonrandom interleavers is discussed in [3]). Block interleavers are effective if the low-weight sequence is confined to a row. If low-weight sequences (which can be regarded as the combination of lower-weight sequences) are confined to several consecutive rows, then the ν_c columns of the interleaver should be sent in a specified order to spread as much as possible the low-weight sequence. A method for reordering the columns is given in [7]. This method guarantees that for any number of columns $\nu_c = aq + r$, $(r \leq a - 1)$, the minimum separation between data entries is $q - 1$, where a is the number of columns affected by a burst. However, as can be observed in the example in Fig. 2, the sequence 1001 will still appear at the input of the encoders for any possible column permutation. Only if we permute the rows of the interleaver in addition to its columns is it possible to break the low-weight sequences. The method in [7] can be used again for the permutation of rows. Appropriate selection of a and q for rows and columns depends on the particular set of codes used and on the specific low-weight sequences that we would like to break.

We have also designed semirandom permutations (interleavers) by generating random integers i, $1 \leq i \leq N$, without replacement. We define an "S-random" permutation as follows: Each randomly selected integer is compared to S previously selected integers. If the current selection is equal to any S previous selections within a distance of $\pm S$, then the current selection is rejected. This process is repeated until all N integers are selected. The searching time for this algorithm increases with S and is not guaranteed to finish successfully. However, we have observed that choosing $S < \sqrt{N/2}$ usually produces a solution in a reasonable time. Note that for $S = 1$, we have a purely random interleaver. In the simulations, we used $S = 31$ with block size $N = 4096$.

III. Turbo Decoding for Multiple Codes

In this section, we consider decoding algorithms for multiple-code turbo codes. In general, the advantage of using three or more constituent codes is that the corresponding two or more interleavers have a better chance to break sequences that were not broken by another interleaver. The disadvantage is that, for an overall desired code rate, each code must be punctured more, resulting in weaker constituent codes. In our experiments, we have used randomly selected interleavers and interleavers based on the row–column permutation described above.

A. Turbo Decoding Configurations

The turbo decoding configuration proposed in [1] for two codes is shown schematically in Fig. 3. This configuration operates in serial mode, i.e., "Dec 1" processes data before "Dec 2" starts its operation, and so on. An obvious extension of this configuration to three codes is shown in Fig. 4(a), which also operates in serial mode. But, with more than two codes, there are other possible configurations, such as that shown in Fig. 4(b), where "Dec 1" communicates with the other decoders, but these decoders do

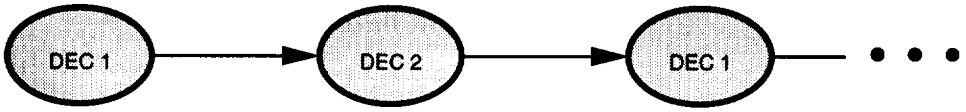


Fig. 3. Decoding structure for two codes.

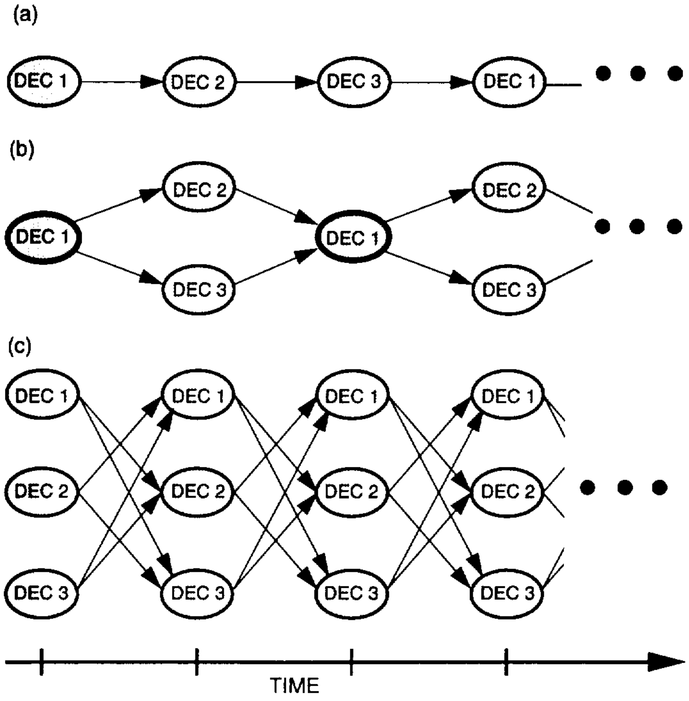


Fig. 4. Different decoding structures for three codes: (a) serial, (b) master and slave, and (c) parallel.

not exchange information between each other. This "master and slave" configuration operates in a mixed serial-parallel mode, since all other decoders except the first operate in parallel. Another possibility, shown in Fig. 4(c), is that all decoders operate in parallel at any given time. Note that self loops are not allowed in these structures since they cause degradation or divergence in the decoding process (positive feedback). We are not considering other possible hybrid configurations. Which configuration performs better? Our selection of the best configuration and its associated decoding rule is based on a detailed analysis of the minimum-bit-error decoding rule (MAP algorithm), as described below.

B. Turbo Decoding for Multiple Codes

Let u_k be a binary random variable taking values in $\{0, 1\}$, representing the sequence of information bits $\mathbf{u} = (u_1, \cdots, u_N)$. The MAP algorithm [6] provides the log likelihood ratio L_k, given the received symbols $\mathbf{y}$:

$$L_k = \log \frac{P(u_k = 1|\mathbf{y})}{P(u_k = 0|\mathbf{y})} \tag{1}$$

$$= \log \frac{\sum_{\mathbf{u}:u_k=1} P(\mathbf{y}|\mathbf{u}) \prod_{j\neq k} P(u_j)}{\sum_{\mathbf{u}:u_k=0} P(\mathbf{y}|\mathbf{u}) \prod_{j\neq k} P(u_j)} + \log \frac{P(u_k = 1)}{P(u_k = 0)} \tag{2}$$

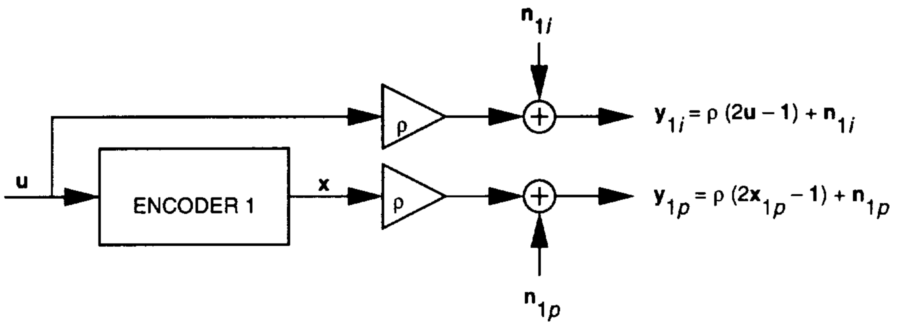


Fig. 5. Channel model.

For efficient computation of Eq. (2) when the a priori probabilities $P(u_j)$ are nonuniform, the modified MAP algorithm in [2] is simpler to use than the version considered in [1]. Therefore, in this article, we use the modified MAP algorithm of [2], as we did in [4].

The channel model is shown in Fig. 5, where the n_{ik}'s and the n_{pk}'s are independent identically distributed (i.i.d.) zero-mean Gaussian random variables with unit variance, and $\rho = \sqrt{2rE_b/N_o}$ is the SNR. The same model is used for each encoder. To explain the basic decoding concept, we restrict ourselves to three codes, but extension to several codes is straightforward. In order to simplify the notation, consider the combination of permuter and encoder as a block code with input $\mathbf{u}$ and outputs $\mathbf{x}_i$, $i = 0, 1, 2, 3 (\mathbf{x}_0 = \mathbf{u})$ and the corresponding received sequences $\mathbf{y}_i$, $i = 0, 1, 2, 3$. The optimum bit decision metric on each bit is (for data with uniform a priori probabilities)

$$L_k = \log \frac{\sum_{\mathbf{u}:u_k=1} P(\mathbf{y}_0|\mathbf{u})P(\mathbf{y}_1|\mathbf{u})P(\mathbf{y}_2|\mathbf{u})P(\mathbf{y}_3|\mathbf{u})}{\sum_{\mathbf{u}:u_k=0} P(\mathbf{y}_0|\mathbf{u})P(\mathbf{y}_1|\mathbf{u})P(\mathbf{y}_2|\mathbf{u})P(\mathbf{y}_3|\mathbf{u})} \tag{3}$$

but in practice, we cannot compute Eq. (3) for large N because the permutations π_2, π_3 imply that $\mathbf{y}_2$ and $\mathbf{y}_3$ are no longer simple convolutional encodings of $\mathbf{u}$. Suppose that we evaluate $P(\mathbf{y}_i|\mathbf{u})$, $i = 0, 2, 3$ in Eq. (3) using Bayes' rule and using the following approximation:

$$P(\mathbf{u}|\mathbf{y}_i) \approx \prod_{k=1}^{N} \tilde{P}_i(u_k) \tag{4}$$

Note that $P(\mathbf{u}|\mathbf{y}_i)$ is not separable in general. However, for $i = 0$, $P(\mathbf{u}|\mathbf{y}_0)$ is separable; hence, Eq. (4) holds with equality. If such an approximation, i.e., Eq. (4), can be obtained, we can use it in Eq. (3) for $i = 2$ and $i = 3$ (by Bayes' rule) to complete the algorithm. A reasonable criterion for this approximation is to choose $\prod_{k=1}^{N} \tilde{P}_i(u_k)$ such that it minimizes the Kullback distance or free energy [8,9]. Define $\tilde{L}_{ik}$ by

$$\tilde{P}_i(u_k) = \frac{e^{u_k \tilde{L}_{ik}}}{1 + e^{\tilde{L}_{ik}}} \tag{5}$$

where $u_k \in \{0, 1\}$. Then the Kullback distance is given by

$$F(\tilde{\mathbf{L}}_i) = \sum_{\mathbf{u}} \frac{e^{\sum_{k=1}^{N} u_k \tilde{L}_{ik}}}{\prod_{k=1}^{N}(1 + e^{\tilde{L}_{ik}})} \log \frac{e^{\sum_{k=1}^{N} u_k \tilde{L}_{ik}}}{\prod_{k=1}^{N}(1 + e^{\tilde{L}_{ik}}) P(\mathbf{u}|\mathbf{y}_i)} \tag{6}$$

Minimizing $F(\tilde{\mathbf{L}}_i)$ involves forward and backward recursions analogous to the MAP decoding algorithm, but we have not attempted this approach in this work. Instead of using Eq. (6) to obtain $\{\tilde{P}_i\}$ or, equivalently, $\{\tilde{L}_{ik}\}$, we use Eqs. (4) and (5) for $i = 0, 2, 3$ (by Bayes' rule) to express Eq. (3) as

$$L_k = f(\mathbf{y}_1, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_2, \tilde{\mathbf{L}}_3, k) + \tilde{L}_{0k} + \tilde{L}_{2k} + \tilde{L}_{3k} \tag{7}$$

where $\tilde{L}_{0k} = 2\rho \mathbf{y}_{0k}$ and

$$f(\mathbf{y}_1, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_2, \tilde{\mathbf{L}}_3, k) = \log \frac{\sum_{\mathbf{u}:u_k=1} P(\mathbf{y}_1|\mathbf{u}) \prod_{j \neq k} e^{u_j(\tilde{L}_{0j} + \tilde{L}_{2j} + \tilde{L}_{3j})}}{\sum_{\mathbf{u}:u_k=0} P(\mathbf{y}_1|\mathbf{u}) \prod_{j \neq k} e^{u_j(\tilde{L}_{0j} + \tilde{L}_{2j} + \tilde{L}_{3j})}} \tag{8}$$

We can use Eqs. (4) and (5) again, but this time for $i = 0, 1, 3$, to express Eq. (3) as

$$L_k = f(\mathbf{y}_2, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_1, \tilde{\mathbf{L}}_3, k) + \tilde{L}_{0k} + \tilde{L}_{1k} + \tilde{L}_{3k} \tag{9}$$

and similarly,

$$L_k = f(\mathbf{y}_3, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_1, \tilde{\mathbf{L}}_2, k) + \tilde{L}_{0k} + \tilde{L}_{1k} + \tilde{L}_{2k} \tag{10}$$

A solution to Eqs. (7), (9), and (10) is

$$\tilde{L}_{1k} = f(\mathbf{y}_1, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_2, \tilde{\mathbf{L}}_3, k); \; \tilde{L}_{2k} = f(\mathbf{y}_2, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_1, \tilde{\mathbf{L}}_3, k); \; \tilde{L}_{3k} = f(\mathbf{y}_3, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_1, \tilde{\mathbf{L}}_2, k) \tag{11}$$

for $k = 1, 2, \cdots, N$, provided that a solution to Eq. (11) does indeed exist. The final decision is then based on

$$L_k = \tilde{L}_{0k} + \tilde{L}_{1k} + \tilde{L}_{2k} + \tilde{L}_{3k} \tag{12}$$

which is passed through a hard limiter with zero threshold. We attempted to solve the nonlinear equations in Eq. (11) for $\tilde{\mathbf{L}}_1$, $\tilde{\mathbf{L}}_2$, and $\tilde{\mathbf{L}}_3$ by using the iterative procedure

$$\tilde{L}_{1k}^{(m+1)} = \alpha_1^{(m)} f(\mathbf{y}_1, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_2^{(m)}, \tilde{\mathbf{L}}_3^{(m)}, k) \tag{13}$$

for $k = 1, 2, \cdots, N$, iterating on m. Similar recursions hold for $\tilde{L}_{2k}^{(m)}$ and $\tilde{L}_{3k}^{(m)}$. The gain $\alpha_1^{(m)}$ should be equal to one, but we noticed experimentally that better convergence can be obtained by optimizing this gain for each iteration, starting from a value slightly less than one and increasing toward one with the iterations, as is often done in simulated annealing methods. We start the recursion with the initial condition[1] $\tilde{\mathbf{L}}_1^{(0)} = \tilde{\mathbf{L}}_2^{(0)} = \tilde{\mathbf{L}}_3^{(0)} = \tilde{\mathbf{L}}_0$. For the computation of $f(\cdot)$, we use the modified MAP algorithm as described in [4] with permuters (direct and inverse) where needed, as shown in Fig. 6 for block decoder 2. The MAP algorithm always starts and ends at the all-zero state since we always terminate the trellis as described in [4]. Similar structures apply for block decoder 1 (we assumed $\pi_1 = I$ identity; however, any π_1 can be used) and block decoder 3. The overall decoder is composed of block decoders

[1] Note that the components of the $\tilde{\mathbf{L}}_i$'s corresponding to the tail bits, i.e., $\tilde{L}_{ik}$, for $k = N + 1, \cdots, N + M$, are set to zero for all iterations.

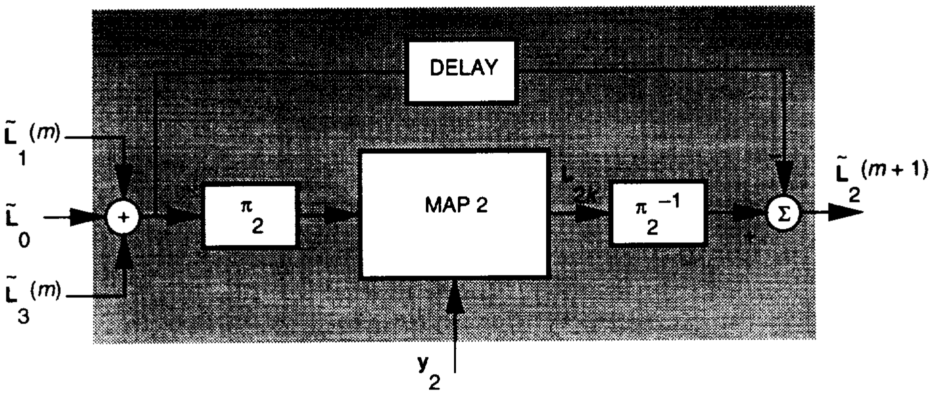


Fig. 6. Structure of block decoder 2.

connected as in Fig. 4(c), which can be implemented as a pipeline or by feedback. We proposed an alternative version of the above decoder in [10]. At this point, further approximation for turbo decoding is possible if one term corresponding to a sequence $\mathbf{u}$ dominates other terms in the summation in the numerator and denominator of Eq. (8). Then the summations in Eq. (8) can be replaced by "maximum" operations with the same indices, i.e., replacing $\sum_{\mathbf{u}:u_k=i}$ with $\max_{\mathbf{u}:u_k=i}$ for $i = 0, 1$. A similar approximation can be used for $\tilde{L}_{2k}$ and $\tilde{L}_{3k}$ in Eq. (11). This suboptimum decoder then corresponds to a turbo decoder that uses soft output Viterbi (SOVA)-type decoders rather than MAP decoders.

C. Multiple-Code Algorithm Applied to Two Codes

For turbo codes with only two constituent codes, Eq. (13) reduces to

$$\tilde{L}_{1k}^{(m+1)} = \alpha_1^{(m)} f(\mathbf{y}_1, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_2^{(m)}, k)$$

$$\tilde{L}_{2k}^{(m+1)} = \alpha_2^{(m)} f(\mathbf{y}_2, \tilde{\mathbf{L}}_0, \tilde{\mathbf{L}}_1^{(m)}, k)$$

for $k = 1, 2, \cdots, N$ and $m = 1, 2, \cdots$, where, for each iteration, $\alpha_1^{(m)}$ and $\alpha_2^{(m)}$ can be optimized (simulated annealing) or set to 1 for simplicity. The decoding configuration for two codes, according to the previous section, is shown in Fig. 7. In this special case, since the two paths in Fig. 7 are disjoint, the decoder structure reduces to duplicate copies of the structure in Fig. 3 (i.e., to the serial mode).

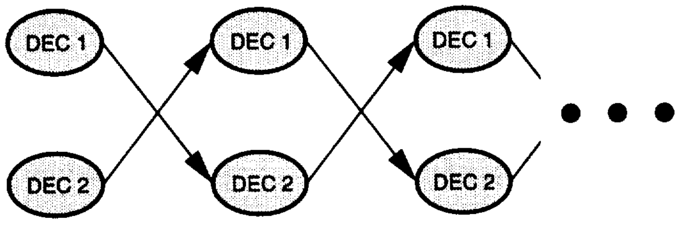


Fig. 7. Parallel structure for two codes.

If we optimize $\alpha_1^{(m)}$ and $\alpha_2^{(m)}$, our method for two codes is similar to the decoding method proposed in [1], which requires estimates of the variances of $\tilde{L}_{1k}$ and $\tilde{L}_{2k}$ for each iteration in the presence of errors. In the method proposed in [2], the received "systematic" observation was subtracted from $\tilde{L}_{1k}$, which results in performance degradation. In [3] the method proposed in [2] was used but the received "systematic" observation was interleaved and provided to decoder 2. In [4], we argued that there is no need to interleave the received "systematic" observation and provide it to decoder 2, since $\tilde{L}_{0k}$ does this job. It seems that our proposed method with $\alpha_1^{(m)}$ and $\alpha_2^{(m)}$ equal to 1 is the simplest and achieves the same performance reported in [3] for rate 1/2 codes.

D. Terminated Parallel Convolutional Codes as Block Codes

Consider the combination of permuter and encoder as a linear block code. Define P_i as the parity matrix of the terminated convolutional code i. Then the overall generator matrix for three parallel codes is

$$G = [I \;\; \pi_1 P_1 \;\; \pi_2 P_2 \;\; \pi_3 P_3]$$

where π_i are the permutations (interleavers). In order to maximize the minimum distance of the code given by G, we should maximize the number of linearly independent columns of the corresponding parity check matrix H. This suggests that the design of P_i (code) and π_i (permutation) are closely related, and it does not necessarily follow that optimum component codes (maximum d_{min}) yield optimum parallel concatenated codes. For very small N, we used this concept to design jointly the permuter and the component convolutional codes.

IV. Performance and Simulation Results

For comparison with the new results on three-code turbo codes, we reproduce in Fig. 8 the performance obtained in [4] by using two-code $K = 5$ turbo codes with generators $(1, g_b/g_a)$, where $g_a = (37)_{octal}$ and $g_b = (21)_{octal}$, and with random permutations of lengths $N = 4096$ and $N = 16384$. The best performance curve in Fig. 8 is approximately 0.7 dB from the Shannon limit at BER $= 10^{-4}$. We also repeat for comparison in Fig. 8 the results obtained in [4] by using encoders with unequal rates with two $K = 5$ constituent codes $(1, g_b/g_a, g_c/g_a)$ and (g_b/g_a), where $g_a = (37)_{octal}$, $g_b = (33)_{octal}$, and $g_c = (25)_{octal}$. To show that it is possible not to send uncoded information for both codes, we used an overall rate 1/2 turbo code using two codes with $K = 2$ (differential encoder) with generator (g_b/g_a), where $g_a = (3)_{octal}$ and $g_b = (1)_{octal}$, and a $K = 5$ code with generator (g_b/g_a), where $g_a = (23)_{octal}$ and $g_b = (33)_{octal}$. A bit error rate of 10^{-5} was achieved at BSNR = 0.85 dB using an S-random permutation of length N = 16,384 with $S = 40$.

A. Three Codes

The performance of two different three-code turbo codes with random interleavers is shown in Fig. 9 for $N = 4096$. The first code uses three recursive codes shown in Fig. 1 with constraint length $K = 3$. The second code uses three recursive codes with $K = 4$, $g_a = (13)_{octal}$, and $g_b = (11)_{octal}$. Note that the nonsystematic version of the second encoder is catastrophic, but the recursive systematic version is noncatastrophic. We found that this $K = 4$ code has better performance than several others.

As seen in Fig. 9, the performance of the $K = 4$ code was improved by going from 20 to 30 iterations. We found that the performance could also be improved by using an S-random interleaver with $S = 31$.

V. Conclusions

We have shown how three-code turbo codes and decoders can be used to further improve the coding gain for deep-space applications as compared with the codes studied in [4]. These are just preliminary results that require extensive further analysis. In particular, we need to improve our understanding of the influence of the interleaver design on the code performance and to analyze how close the proposed decoding algorithm is to maximum-likelihood or MAP decoding.

These new codes offer better performance than the large constraint-length convolutional codes employed by current missions and, most importantly, achieve these gains with much lower decoding complexity.

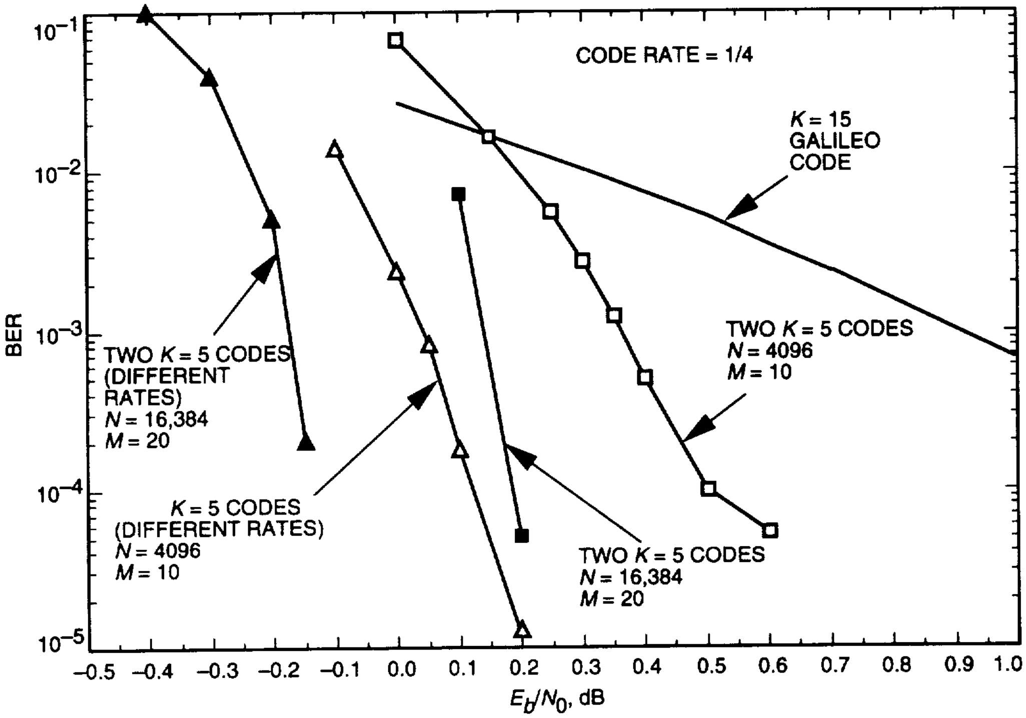


Fig. 8. Two-code performance, $r = 1/4$.

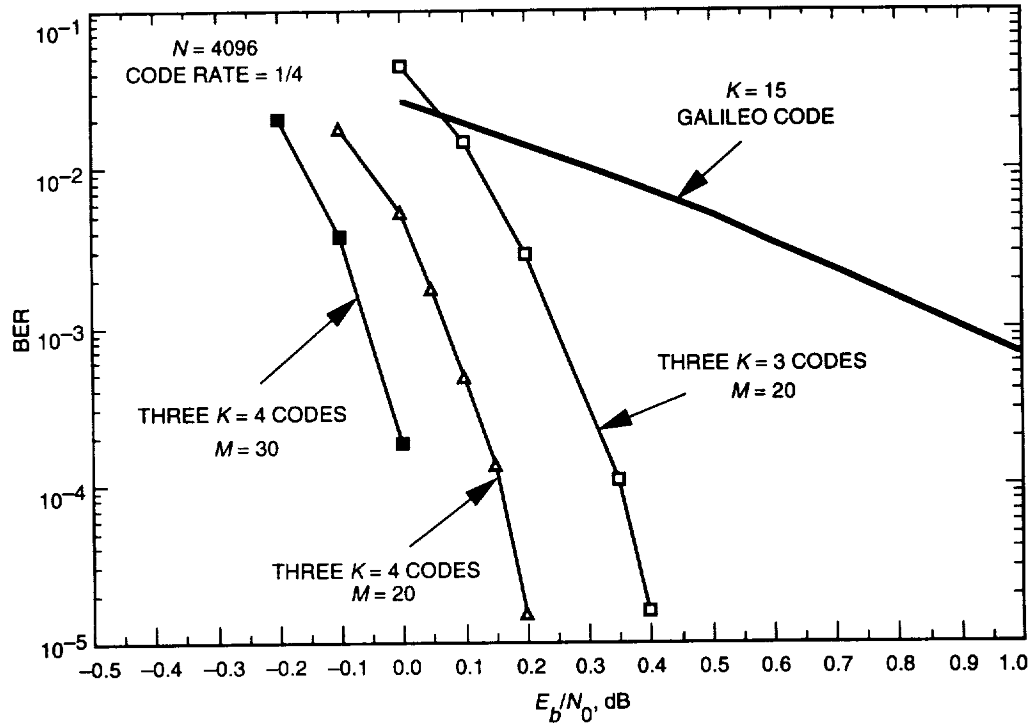


Fig. 9. Three-code performance, $r = 1/4$.

Acknowledgments

The authors are grateful to S. Dolinar for his contributions to the study of the weight distribution and interleavers[2] and to R. J. McEliece for helpful comments throughout this study.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding: Turbo Codes,"*Proc. 1993 IEEE International Conference on Communications*, Geneva, Switzerland, pp. 1064–1070, May 1993.

[2] J. Hagenauer and P. Robertson, "Iterative (Turbo) Decoding of Systematic Convolutional Codes With the MAP and SOVA Algorithms," *Proc. of the ITG Conference on Source and Channel Coding*, Frankfurt, Germany, October 1994.

[3] P. Robertson, "Illuminating the Structure of Code and Decoder of Parallel Concatenated Recursive Systematic (Turbo) Codes, *Proceedings GLOBECOM '94*, San Francisco, California, pp. 1298–1303, December 1994.

[4] D. Divsalar and F. Pollara, "Turbo Codes for Deep-Space Communications," *The Telecommunications and Data Acquisition Progress Report 42-120, October–December 1994*, Jet Propulsion Laboratory, Pasadena, California, pp. 29–39, February 15, 1995.

[5] G. Battail, C. Berrou, and A. Glavieux, "Pseudo-Random Recursive Convolutional Coding for Near-Capacity Performance," *Comm. Theory Mini-Conference, GLOBECOM '93*, Houston, Texas, December 1993.

[6] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, 1974.

[7] E. Dunscombe and F. C. Piper, " Optimal Interleaving Scheme for Convolutional Codes," *Electronic Letters*, vol. 25, no. 22, pp. 1517–1518, October 26, 1989.

[8] M. Moher, "Decoding Via Cross-Entropy Minimization," *Proceedings GLOBECOM '93*, pp. 809–813, December 1993.

[9] G. Battail and R. Sfez, "Suboptimum Decoding Using the Kullback Principle," *Lecture Notes in Computer Science*, vol. 313, pp. 93–101, 1988.

[10] D. Divsalar and F. Pollara, "Turbo Codes for PCS Applications," *Proceedings of IEEE ICC'95*, Seattle, Washington, June 1995.

[2] More detailed results are given in S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Non-Random Permutations," JPL Interoffice Memorandum 331-95.2-016 (internal document), Jet Propulsion Laboratory, Pasadena, California, March 15, 1995.

TDA Progress Report 42-121

May 15, 1995

Degradation in Finite-Harmonic Subcarrier Demodulation

Y. Feria and S. Townes
Communications Systems Research Section

T. Pham
Telecommunications Systems Section

Previous estimates on the degradations due to a subcarrier loop assume a square-wave subcarrier. This article provides a closed-form expression for the degradations due to the subcarrier loop when a finite number of harmonics are used to demodulate the subcarrier, as in the case of the buffered telemetry demodulator. We compared the degradations using a square wave and using finite harmonics in the subcarrier demodulation and found that, for a low loop signal-to-noise ratio, using finite harmonics leads to a lower degradation. The analysis is under the assumption that the phase noise in the subcarrier (SC) loop has a Tikhonov distribution. This assumption is valid for first-order loops.

I. Introduction

In an imperfect subcarrier demodulation, the difference between the phase of the reference signal and that of the subcarrier of the received signal causes the signal power to degrade while the noise power remains the same. This degradation is measured as the ratio of the reduced symbol energy-to-noise density ratio (E_s/N_0), or symbol signal-to-noise ratio (SNR), to the symbol SNR of an ideal demodulation where the phase difference is zero. The degradations due to the subcarrier loop were previously computed assuming a square wave [3]. This assumption is inappropriate in the case where only a finite number of harmonics of the subcarrier are there to be demodulated, as in the buffered telemetry demodulator (BTD) [2]. This article provides a closed-form expression for computing the degradation due to a finite-harmonic subcarrier tracking loop. Numerically, we found that, for low loop SNR cases, we actually have less degradation using a finite number of harmonics than using "all" the harmonics, namely, the square wave. The degradation due solely to the subcarrier loop using four harmonics is 0.15 to 0.3 dB lower than that using a square wave for loop SNRs in the range of 14 to 30 dB.

At first glance, the above may seem to contradict the intuition that the more harmonics we use, the higher the SNR we should get. This intuition is correct when the loop SNR is high, that is, when the jitter of the phase difference (between the true and the reference phases) is low. At low loop SNRs, however, we have a different scenario.

To explain this, let us first take a look at how the subcarriers are demodulated. A square-wave subcarrier is demodulated by multiplying the received signal by a square-wave reference signal. When we only have a finite number of harmonics of the square-wave subcarrier, the current design for the BTD [2] demodulates the subcarrier by multiplying each received harmonic by its reference signal and combining the resulting harmonics with the weights of $1/n$, where n indicates the nth harmonic. In the case of square-wave subcarrier demodulation, we are implicitly combining the harmonics the same way, only now we have an infinite number of harmonics (see the Appendix for the proof). The reference signals are generated by using the phase of the fundamental frequency component or the first harmonic.

Therefore, if the first harmonic has a phase noise with a standard deviation of σ, then the nth harmonic will have a phase noise with a standard deviation of $n\sigma$, which implies that the nth harmonic will suffer a higher degradation than the first one. At low loop SNRs, the degradations in higher harmonics can be even higher than the SNR that they contribute. In such cases, higher harmonics should not be used in the subcarrier demodulation.

In the full spectrum combining case [1], more harmonics means that more data need to be transmitted to the combining location or stored locally. In the case of intercontinental arraying, where data transmission becomes expensive, suppressing higher harmonics becomes an important issue. Later in this article, we will show that, for a given loop SNR, there is an optimum number of harmonics that should be used, and in the region of the operating loop SNRs, these numbers are mostly finite.

To compare the degradations when using finite harmonics and a square wave, we first give an expression to compute the degradations using a square wave, assuming that the phase noise has a Tikhonov distribution. This assumption is valid for first-order loops only [1]. For higher loop SNRs, the degradation due to the phase noise with a Tikhonov distribution is very close to that due to a phase noise with a Gaussian distribution. In the range of the operating loop SNRs, the two distribution assumptions lead to similar results. We then give an expression of degradation for finite-harmonic subcarrier demodulation, assuming that the phase noise has a Tikhonov distribution.

II. Square-Wave Case

When the subcarrier is a square wave, the degradation due to the subcarrier loop has the form [1]

$$\overline{C^2_{sc_{sq}}} = 1 - \frac{4}{\pi}\overline{|\phi_{sc}|} + \frac{4}{\pi^2}\overline{\phi^2_{sc}} \qquad (1)$$

where ϕ_{sc} is the phase noise in the square-wave subcarrier tracking loop. If the phase noise, ϕ_{sc}, is assumed to have a Gaussian distribution with zero mean and a variance of σ^2, then [1]

$$\overline{|\phi_{sc}|} = \sqrt{\frac{2}{\pi}}\sigma$$

$$\overline{\phi^2_{sc}} = \sigma^2$$

The degradation due to the subcarrier loop is [1]

$$\overline{C^2_{scG_{sq}}} = 1 - \sqrt{\frac{32}{\pi^3}}\sigma + \frac{4}{\pi^2}\sigma^2 \qquad (2)$$

While the Gaussian assumption is accurate for high loop SNR cases, Tikhonov distribution is a better assumption for low loop SNR cases. Note that the Tikhonov assumption is valid for first-order loops. If the phase noise ϕ_{sc} in a Costas loop is assumed to have a Tikhonov distribution, then we can show that

$$\overline{|\phi_{sc}|} = \int_{-\pi/2}^{\pi/2} \frac{\exp\left[(1/4)\rho_{sc}\cos 2\phi_{sc}\right]}{\pi I_0(\rho_{sc}/4)} |\phi_{sc}| d\phi_{sc}$$

$$= \frac{\pi}{4} + \frac{1}{\pi I_0(\rho_{sc}/4)} \sum_{k=1}^{\infty} I_k(\rho_{sc}/4) \frac{(-1)^k - 1}{k^2} \tag{3}$$

$$\overline{\phi_{sc}^2} = \int_{-\pi/2}^{\pi/2} \frac{\exp\left[(1/4)\rho_{sc}\cos 2\phi_{sc}\right]}{\pi I_0(\rho_{sc}/4)} \phi_{sc}^2 \, d\phi_{sc}$$

$$= \frac{\pi^2}{12} + \frac{1}{I_0(\rho_{sc}/4)} \sum_{k=1}^{\infty} I_k(\rho_{sc}/4) \frac{(-1)^k}{k^2} \tag{4}$$

where I_k is the modified Bessel function of order k, and ρ_{sc} is the subcarrier-loop SNR, which can be computed using

$$\rho_{sc} = \frac{4}{\pi^2} \frac{1}{B_{sc} W_{sc}} \frac{P_d}{N_0} \left[1 + \frac{1}{2E_s/N_0}\right]^{-1} \tag{5}$$

Here B_{sc} denotes the one-sided subcarrier loop bandwidth, W_{sc} denotes the subcarrier window size [2], P_d/N_0 denotes the total data power over the one-sided noise density, and E_s/N_0 denotes the symbol energy-to-noise density ratio.

Note that Eqs. (3) and (4) are different from Eqs. (22) and (23) in [1] in that the former are for Costas loops and the latter are for phase-locked loops. Assuming a Tikhonov distribution, the degradation due to the subcarrier loop is

$$\overline{C_{sc_{sq}}^2} = \frac{1}{3} + \frac{4}{\pi^2} \frac{1}{I_0(\rho_{sc}/4)} \sum_{k=1}^{\infty} I_k(\rho_{sc}/4) \frac{1}{k^2} \tag{6}$$

III. Finite Number of Harmonics Case

When a finite number of harmonics are used to track and demodulate the subcarrier, as in the BTD, the signal amplitude has the form [2]

$$S_{sc} = \frac{8}{\pi^2} \sum_{m=0}^{L-1} \frac{\cos\left[(2m+1)\phi_{sc}\right]}{(2m+1)^2} \tag{7}$$

where L is the number of harmonics and ϕ_{sc} is the phase noise resulting from the subcarrier tracking loop. Clearly, when $\phi_{sc} = 0$, we have the ideal case,

$$S_{sc_{ideal}} = \frac{8}{\pi^2} \sum_{m=0}^{L-1} \frac{1}{(2m+1)^2} \tag{8}$$

Taking the ratio of Eq. (7) and Eq. (8), we obtain the signal-amplitude degradation,

$$\begin{aligned} C_{sc} &= \frac{S_{sc}}{S_{sc_{ideal}}} \\ &= \frac{\sum_{m=0}^{L-1} \left(\cos\left[(2m+1)\phi_{sc}\right]/(2m+1)^2\right)}{\sum_{m=0}^{L-1} \left(1/(2m+1)^2\right)} \end{aligned} \tag{9}$$

Squaring Eq. (9) and taking the expectation, we have the signal power degradation,

$$\overline{C_{sc}^2} = \frac{1}{\left(\sum_{m=0}^{L-1} \left[1/(2m+1)^2\right]\right)^2} \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} \frac{\overline{\cos 2(m-n)\phi_{sc}} + \overline{\cos 2(m+n+1)\phi_{sc}}}{2(2m+1)^2(2n+1)^2} \tag{10}$$

The noise power after the subcarrier demodulation is not affected by the phase noise in the subcarrier loop. This can be observed from the noise power expressions in Eqs. (A-28) and (A-29) of [2]. This implies that the degradation in the symbol SNR is the same as the signal-power degradation as given in Eq. (10).

For first-order Costas loops, the phase noise ϕ_{sc} has a Tikhonov distribution:

$$p(\phi_{sc}) = \begin{cases} \dfrac{\exp[(1/4)\rho_{sc}\cos(2\phi_{sc})]}{\pi I_0(\rho_{sc}/4)}, & |\phi_{sc}| \le \dfrac{\pi}{2} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

Hence we have,

$$\overline{\cos(n\phi_{sc})} = \frac{I_{n/2}(\rho_{sc}/4)}{I_0(\rho_{sc}/4)} \tag{12}$$

where n is an even number, I_n is the modified Bessel function of order n, and ρ_{sc} is the subcarrier-loop SNR.

Plugging Eq. (12) in Eq. (10), we have

$$\overline{C_{sc}^2} = \frac{1}{\left(\sum_{m=0}^{L-1} \left[1/(2m+1)^2\right]\right)^2} \frac{1}{I_0(\rho_{sc}/4)} \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} \frac{I_{m-n}(\rho_{sc}/4) + I_{m+n+1}(\rho_{sc}/4)}{2(2m+1)^2(2n+1)^2} \tag{13}$$

As L approaches infinity, Eq. (13) becomes identical to Eq. (6) (see the Appendix for the proof).

For $L = 4$, we have the SNR degradation due to the subcarrier loop,

$$\begin{aligned}\overline{C_{sc}^2} &= \frac{1}{\left(\sum_{m=0}^{L-1}\left[1/(2m+1)^2\right]\right)^2}\left[0.507181 + 0.616372\frac{I_1(\rho_{sc}/4)}{I_0(\rho_{sc}/4)} + 0.153379\frac{I_2(\rho_{sc}/4)}{I_0(\rho_{sc}/4)}\right. \\ &\quad + 0.066581\frac{I_3(\rho_{sc}/4)}{I_0(\rho_{sc}/4)} + 0.0248526\frac{I_4(\rho_{sc}/4)}{I_0(\rho_{sc}/4)} + 0.00306757\frac{I_5(\rho_{sc}/4)}{I_0(\rho_{sc}/4)} \\ &\quad \left. +0.000816327\frac{I_6(\rho_{sc}/4)}{I_0(\rho_{sc}/4)} + 0.000208247\frac{I_7(\rho_{sc}/4)}{I_0(\rho_{sc}/4)}\right] \end{aligned} \qquad (14)$$

The subcarrier-loop SNR, ρ_{sc}, can be computed using the following equations:[1]

$$\rho_{sc} = \frac{\alpha\beta^2}{\gamma B_{sc}}\frac{P_d}{N_0}\left(\alpha + \frac{1}{2E_s/N_0}\right)^{-1}$$

where

$$\alpha = \frac{8}{\pi^2}\sum_{m=0}^{L-1}\frac{1}{(2m+1)^2}$$

$$\beta = \frac{8}{\pi^2}\sum_{n=0}^{L-1} w_n$$

$$\gamma = \frac{8}{\pi^2}\sum_{n=0}^{L-1} w_n^2$$

and

$$w_n = \frac{\sin[(2n+1)(\pi/2)W_{sc}]}{2n+1}$$

For different loop SNRs, the degradations $\overline{C_{scGsq}^2}$, $\overline{C_{sc_{sq}}^2}$, and $\overline{C_{sc}^2}$ in Eqs. (2), (6), and (14), respectively, are plotted in Fig. 1. Figure 2 shows the achievable subcarrier-loop SNR for both square wave and four harmonics for $P_d/N_0 = 15$ dB-Hz at a symbol rate of 100 sym/s with a suppressed carrier. The window sizes in the subcarrier loops for the square wave and the four harmonics are $W_{sc} = 1/4$ and $W_{sc} = 1/16$, respectively. For the above parameters, the achievable subcarrier-loop SNRs are almost the same.

IV. Optimum Number of Harmonics

To make a fair comparison among the square wave and different numbers of harmonics in the subcarrier, we should compare the losses due to all three loops (carrier, subcarrier, and symbol) and the harmonic cutoffs, since the harmonic cutoffs also affect the carrier and symbol loop SNRs. The degradation due

[1] H. Tsou, personal communication, Communications Systems Research Section, Jet Propulsion Laboratory, Pasadena, California, October 1994.

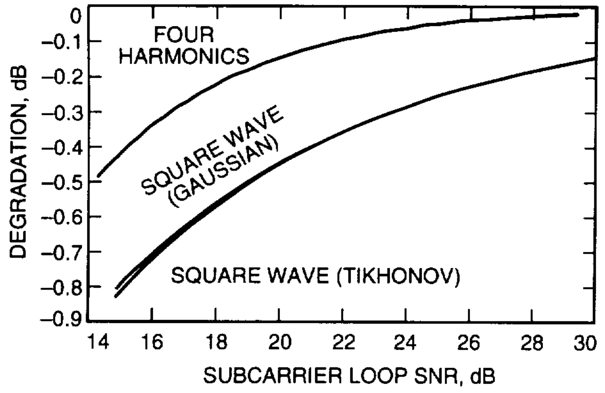


Fig. 1. Degradation due to the subcarrier loop versus loop SNR.

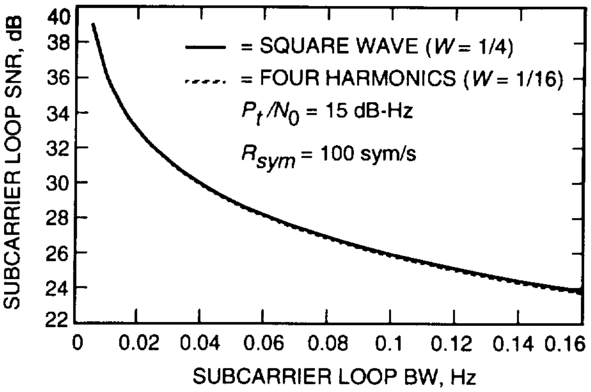


Fig. 2. Subcarrier loop SNR versus loop bandwidth.

to the suppressed-carrier loop can be found in [3], while the degradation due to the symbol loop can be found in [1]. Finally, the degradation due to the harmonic cutoffs can be found in [4].

With the number of harmonics limited to less than or equal to four, we compare the degradations in all loops, including the loss due to using a finite number of harmonics. For a particular set of parameters, the comparison is shown in Fig. 3. It can be observed that for a subcarrier-loop SNR below 16 dB, adding the fourth harmonic does not increase symbol SNR. On the other hand, the loop may lose lock for a subcarrier-loop SNR below 16 dB, so the region of operation has to be greater than 16 dB. For this region, using four harmonics will lead to a lower degradation than will using fewer harmonics.

Without any limitation on the number of harmonics, we computed the degradations due to all three loops and to the harmonic cutoffs. For the same set of parameters, we plotted the degradation versus the subcarrier-loop SNR for different numbers of harmonics, as shown in Fig. 4. We found the optimum numbers of harmonics for three regions of subcarrier-loop SNR and tabulated them in Table 1. By the optimum number of harmonics, we mean that, using more harmonics than the optimum will result in a higher degradation in symbol SNR. Note that this table only applies to the set of parameters listed in Fig. 4.

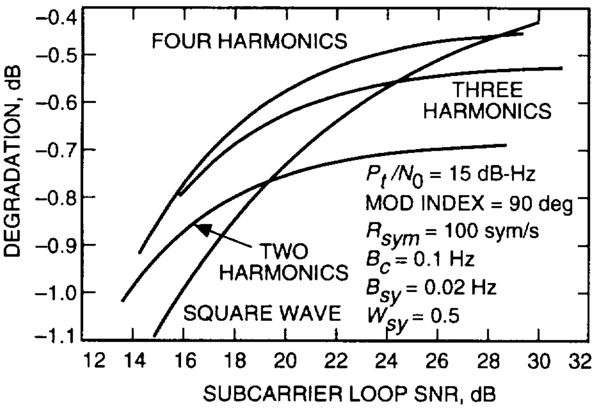


Fig. 3. Comparison using two, three, and four harmonics in terms of degradations due to all loops and harmonic cutoffs.

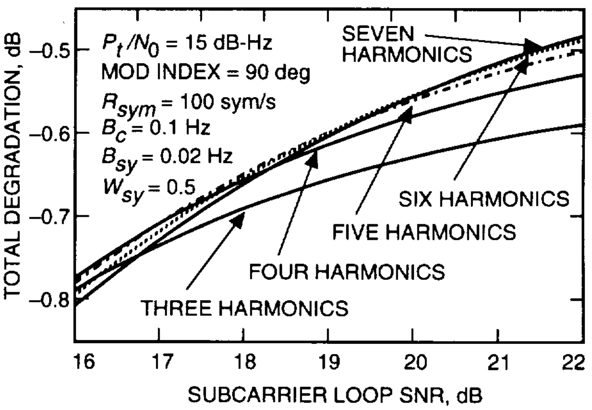


Fig. 4. Optimum number of harmonics in terms of degradations due to all loops and harmonic cutoffs.

C-2

Table 1. Optimum number of harmonics.

SC loop SNR, dB	Optimum number of harmonics
16.0 to 17.0	4
17.0 to 18.8	5
18.8 to 20.3	6

V. Conclusion

In this article, we presented a closed-form expression to compute the degradation due to the subcarrier loop when only a finite number of harmonics are used to demodulate the subcarrier. This expression assumes that the phase noise has a Tikhonov distribution, which is valid for first-order loops. Using this expression, we computed the degradations in the subcarrier loop for different numbers of harmonics in the subcarrier and found that, in certain regions of the subcarrier-loop SNRs, using a finite number of harmonics leads to a lower degradation in symbol SNR than does using all harmonics or a square wave.

Acknowledgments

The authors wish to thank David Rogstad, Kar-Ming Cheung, and Sam Dolinar for their helpful suggestions and discussions.

References

[1] A. Mileant and S. Hinedi, "Overview of Arraying Techniques in the Deep Space Network," *The Telecommunications and Data Acquisition Progress Report 42-104, October–December 1990*, Jet Propulsion Laboratory, Pasadena, California, pp. 109–138, February 15, 1991.

[2] H. Tsou, B. Shah, R. Lee, and S. Hinedi, "A Functional Description of the Buffered Telemetry Demodulation (BTD)," *The Telecommunications and Data Acquisition Progress Report 42-112, October–December 1992*, Jet Propulsion Laboratory, Pasadena, California, pp. 50–73, February 15, 1993.

[3] S. Million, B. Shah, and S. Hinedi, "A Comparison of Full-Spectrum and Complex Symbol Combining Techniques for the Galileo S-Band Mission," *The Telecommunications and Data Acquisition Progress Report 42-116, October–December 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 128–162, February 15, 1994.

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

Appendix
As the Number of Harmonics Approaches Infinity

To prove that Eq. (13) approaches Eq. (6) as the number of harmonics approaches infinity, it suffices to prove that

$$\frac{1}{\left(\sum_{m=0}^{\infty}[1/(2m+1)^2]\right)^2}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\frac{\cos 2(m-n)\phi+\cos 2(m+n+1)\phi}{2(2m+1)^2(2n+1)^2}=\frac{1}{3}+\frac{4}{\pi^2}\sum_{k=1}^{\infty}\frac{\cos(2k\phi)}{k^2} \qquad \text{(A-1)}$$

Expanding the left side of the above equation and ignoring the coefficient before the summations, which has the value $(8/\pi^2)^2$, we have

$$\text{left side}=\sum_{m=0}^{\infty}\frac{1}{2(2m+1)^4}+\sum_{m=0}^{\infty}\sum_{n=0,n\neq m}^{\infty}\frac{\cos(2(m-n)\phi)}{2(2m+1)^2(2n+1)^2}+\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\frac{\cos[2(m+n+1)\phi]}{2(2m+1)^2(2n+1)^2} \qquad \text{(A-2)}$$

The first term of "left side" is

$$\sum_{m=0}^{\infty}\frac{1}{2(2m+1)^4}=\frac{1}{3}\left(\frac{\pi^2}{8}\right)^2 \qquad \text{(A-3)}$$

For the second term of "left side," let $k=m-n$. For a fixed n, k runs from $-n$ to infinity. The second term becomes

$$\begin{aligned}\sum_{m=0}^{\infty}\sum_{n=0,n\neq m}^{\infty}\frac{\cos(2(m-n)\phi)}{2(2m+1)^2(2n+1)^2}&=\sum_{n=0}^{\infty}\sum_{k=-n,k\neq 0}^{\infty}\frac{\cos 2k\phi}{2(2n+1+2k)^2(2n+1)^2}\\&=\sum_{k=1}^{\infty}\frac{\cos 2k\phi}{2}\sum_{n=0}^{\infty}\frac{1}{(2n+1+2k)^2(2n+1)^2}+\sum_{n=0}^{\infty}\sum_{k=1}^{n}\frac{\cos 2k\phi}{2(2n+1-2k)^2}\end{aligned} \qquad \text{(A-4)}$$

The inner sum of the first term in the above equation is

$$\begin{aligned}\sum_{n=0}^{\infty}\frac{1}{(2n+1+2k)^2(2n+1)^2}&=\frac{1}{(2k)^2}\sum_{n=0}^{\infty}\frac{1}{(2n+1+2k)^2}+\frac{4}{(2k)^3}\sum_{n=0}^{\infty}\frac{1}{2n+1+2k}\\&\quad+\frac{1}{(2k)^2}\sum_{n=0}^{\infty}\frac{1}{(2n+1)^2}-\frac{4}{(2k)^3}\sum_{n=0}^{\infty}\frac{1}{2n+1}\\&=\frac{1}{(2k)^2}\left(\frac{\pi^2}{4}-\sum_{q=0}^{k-1}\frac{1}{2q+1}\right)-\frac{4}{(2k)^3}\sum_{q=0}^{k-1}\frac{1}{2q+1}\end{aligned} \qquad \text{(A-5)}$$

For the third term of "left side," let $p = m + n + 1$. Then, for a fixed m, p runs from $m + 1$ to infinity. The third term becomes

$$\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\frac{\cos(2(m+n+1)\phi)}{2(2m+1)^2(2n+1)^2} = \sum_{m=0}^{\infty}\sum_{p=m+1}^{\infty}\frac{\cos 2p\phi}{2(2m+1)^2(2m+1-2p)^2}$$

$$= \sum_{p=1}^{\infty}\frac{\cos 2p\phi}{2}\sum_{m=0}^{\infty}\frac{1}{(2m+1-2p)^2(2m+1)^2} - \sum_{m=0}^{\infty}\sum_{p=1}^{m}\frac{\cos 2p\phi}{2(2m+1-2p)^2} \tag{A-6}$$

The inner sum of the first term in the above equation is

$$\sum_{m=0}^{\infty}\frac{1}{(2m+1-2p)^2(2m+1)^2} = \frac{1}{(2p)^2}\sum_{m=0}^{\infty}\frac{1}{(2m+1-2p)^2} - \frac{4}{(2k)^3}\sum_{m=0}^{\infty}\frac{1}{2m+1-2p}$$

$$+ \frac{1}{(2p)^2}\sum_{m=0}^{\infty}\frac{1}{(2m+1)^2} + \frac{4}{(2k)^3}\sum_{m=0}^{\infty}\frac{1}{2m+1}$$

$$= \frac{1}{(2p)^2}\left(\frac{\pi^2}{4} + \sum_{q=0}^{p-1}\frac{1}{2q+1}\right) + \frac{4}{(2p)^3}\sum_{q=0}^{p-1}\frac{1}{2q+1} \tag{A-7}$$

Substitute Eq.(A-5) in Eq. (A-4), and Eq. (A-7) in Eq. (A-6), and then, adding the results, we have the sum of the second and third terms of "left side":

$$\sum_{m=0}^{\infty}\sum_{n=0,n\neq m}^{\infty}\frac{\cos(2(m-n)\phi)}{2(2m+1)^2(2n+1)^2} + \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\frac{\cos(2(m+n+1)\phi)}{2(2m+1)^2(2n+1)^2} = \sum_{k=1}^{\infty}\frac{\cos 2k\phi}{(2k)^2}\frac{\pi^2}{4} \tag{A-8}$$

Finally, adding the first term to the above, and multiplying the coefficient $(8/\pi^2)^2$, we have "left side" equal to

$$\text{left side} = \frac{1}{3} + \frac{4}{\pi^2}\sum_{k=1}^{\infty}\frac{\cos 2k\phi}{k^2} \tag{A-9}$$

Towards Optimum Demodulation of Bandwidth-Limited and Low SNR Square-Wave Subcarrier Signals

Y. Feria
Communications Systems Research Section

W. Hurd
Radio Frequency and Microwave Subsystems Section

The optimum phase detector is presented for tracking square-wave subcarriers that have been bandwidth limited to a finite number of harmonics. The phase detector is optimum in the sense that the loop signal-to-noise ratio (SNR) is maximized and, hence, the rms phase tracking error is minimized. The optimum phase detector is easy to implement and achieves substantial improvement. Also presented are the optimum weights to combine the signals demodulated from each of the harmonics. The optimum weighting provides SNR improvement of 0.1 to 0.15 dB when the subcarrier loop SNR is low (15 dB) and the number of harmonics is high (8 to 16).

I. Introduction

This work was motivated by the need for near-optimum demodulation of the extremely weak signal received from the Galileo spacecraft. This demonstration is accomplished in the buffered telemetry demodulator (BTD). Since the BTD is a software demodulator, it is practical to tailor the processing more closely to the Galileo signal conditions than would be practical in other systems, such as the Block V Receiver.

A limitation of the BTD is that the input signal has been recorded by the full spectrum recorder and contains only the first four harmonics of the originally transmitted square-wave subcarrier. The subcarrier phase detector initially implemented in the BTD uses a windowing technique similar to that used in the Advanced Receiver II and the Block V Receiver [1] but modified for the four-harmonic case [3]. There is a parameter, W_{sc}, that is analogous to the fractional window width in a square-wave subcarrier phase detector. As shown in Fig. 1, this phase detector results in a degradation (loss in symbol signal-to-noise ratio (SNR) due to harmonic truncation and phase tracking error), which does not monotonically decrease as the number of harmonics is increased.[1] In fact, when the tracking error is large, and when the harmonics are combined using the usual $1/n$ weighting for the nth harmonic, it is sometimes better to use only four harmonics than to use all harmonics. This suggests two things: First, it tells us that

[1] Based on work by D. Rogstad, Tracking Systems and Applications Section, and Y. Feria, Communications Systems Research Section, Jet Propulsion Laboratory, Pasadena, California, October 1994.

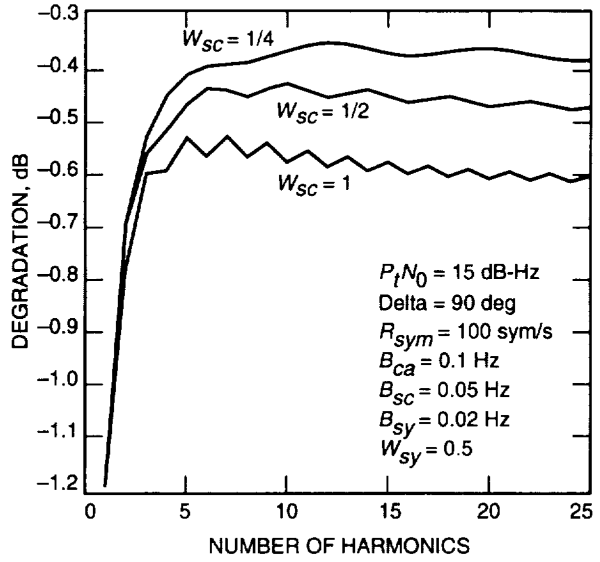


Fig. 1. Degradation as a function of the number of harmonics, using the current BTD.

the phase detector may not be using the harmonics optimally. Second, it indicates that the demodulated harmonics may not be optimally combined.

The phase detector used in [3] is derived from a window used on a square-wave subcarrier loop. This phase detector may not be the optimum for a finite-harmonic subcarrier. As a previous work [2] indicates, the higher harmonics get larger phase noise jitters. Therefore, the effective signal amplitude on the nth harmonic is no longer $1/n$ but some number smaller than that. The optimum weights to combine the demodulated harmonics should account for the SNR losses due to the loop.

II. Optimum Phase Detector

Here we derive a phase detector (PD) that is optimum in the sense that the loop SNR is maximized. To show the derivation, let us first take a look at the current phase detector used in the BTD. The current phase detector is the product of the combined in-phase signals $\sqrt{P_d}d_k\cos\phi_c(8/\pi^2)\sum_{n=0}^{L-1}(1/(2n+1)^2)\cos[(2n+1)\phi_{sc}]$ and the combined quadrature signals $\sqrt{P_d}d_k\cos\phi_c(8/\pi^2)\sum_{n=0}^{L-1}w_n(1/(2n+1))\sin[(2n+1)\phi_{sc}]$ where the w_n are the weights used to combine the quadrature signals and, in the current BTD, these weights are

$$w_n = \frac{\sin[(2n+1)(\pi/2)W_{sc}]}{2n+1}$$

The loop SNR using the current BTD is derived as[2]

$$\rho_{sc} = \frac{\alpha\beta^2}{\gamma B_{sc}}\frac{P_d}{N_0}\left(\alpha + \frac{1}{2E_s/N_0}\right)^{-1}$$

where

[2] H. Tsou, personal communication, Communications Systems Research Section, Jet Propulsion Laboratory, Pasadena, California, October 1994.

$$\alpha = \frac{8}{\pi^2}\sum_{n=0}^{L-1}\frac{1}{(2n+1)^2}$$

$$\beta = \frac{8}{\pi^2}\sum_{n=0}^{L-1} w_n$$

$$\gamma = \frac{8}{\pi^2}\sum_{n=0}^{L-1} w_n^2$$

where L is the total number of harmonics used in the phase detector, P_d/N_0 is the data power-to-noise ratio, E_s/N_0 is the symbol SNR, and B_{sc} is the subcarrier loop bandwidth.

Now in order to maximize the subcarrier loop SNR, ρ_{sc}, let w_k, $k = 0, \cdots, L-1$, be unknown and α be the same as before, and differentiate the loop SNR, ρ_{sc}, with respect to w_k and set the expression to zero. We then have

$$\frac{\partial \rho_{sc}}{\partial w_k} = \frac{2\beta\gamma - 2\beta^2 w_k}{\gamma^2}\,\frac{1}{B_{sc}}\,\frac{P_d}{N_0}\,\frac{\alpha}{\alpha + 1/(2E_s/N_0)}$$

$$= 0 \qquad (1)$$

Since $P_d/N_o \neq 0$, $\alpha \neq 0$, and γ, B_{sc} are finite, the above is zero if and only if

$$\gamma - \beta w_k = 0$$

That is,

$$\sum_{n=0}^{L-1} w_n^2 - \sum_{n=0}^{L-1} w_n w_k = 0$$

or

$$\sum_{n=0}^{L-1} w_n(w_n - w_k) = 0, \text{ for all } k$$

which implies that

$$w_n = w_k, \text{ for all } n \text{ and } k$$

The conclusion is that the optimum weights to combine the quadrature signals in the phase detector are a constant for all (finite) harmonics. Note that, for infinite harmonics, the parameters β and γ do not converge; therefore, the above weights cannot be used for square waves. When the optimum weights are used in the phase detector, the loop SNR becomes

$$\rho_{sc} = \frac{L}{B_{sc}} \frac{P_d}{N_0} \frac{\alpha}{\alpha + 1/(2E_s/N_0)} \tag{2}$$

Using the optimum weights in the phase detector (called the optimum phase detector), we can improve the loop SNR by 9.5 dB over the current BTD with window size = 1, and by 1.1 dB over the current BTD with window size = 1/4 (see Fig. 2). The same figure also shows that, using the optimum phase detector, the loop SNR obtained by using only one harmonic is higher than that using the current BTD with the window size being either 1 or 1/2. Note that when we use only one harmonic in the optimum phase detector, we may still use all the available harmonics to demodulate the subcarrier.

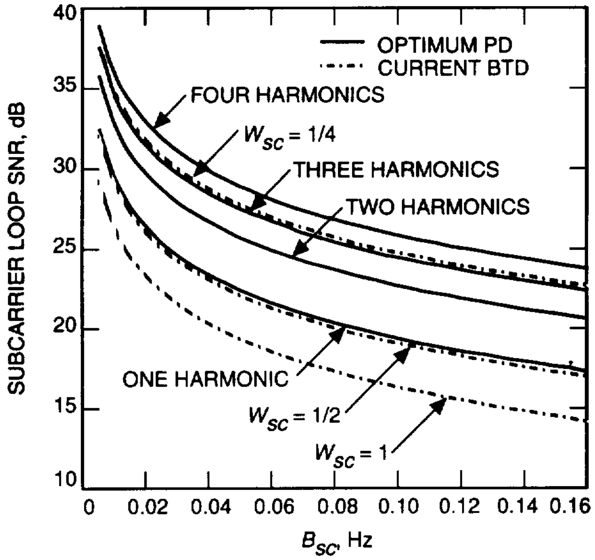


Fig. 2. Comparison in loop SNR using the optimum phase detector and the current BTD.

Degradations due to a finite-harmonic subcarrier loop can be computed using the expressions given in [2]. Degradations as a function of the number of harmonics are shown in Fig. 3. Clearly, we can observe that, using the optimum phase detector, we obtained a lower degradation with more harmonics. This agrees with our intuition.

With the increase of the loop SNR, that is, with the increase of the number of harmonics, the linear region shrinks. See the normalized S-curves shown in Fig. 4. As the number of harmonics approaches infinity, the linear region of the S-curve approaches zero. In other words, this optimum phase detector is only for a finite number of harmonics.

III. Optimum Combining Weights in Demodulation

The demodulated harmonics are currently combined with the weight $1/n$ for the nth harmonic. These weights are optimum if each of the harmonics of the subcarrier is demodulated with the same phase jitter. In our case, however, we know that if the first harmonic has a phase jitter with a variance of σ^2, then the nth harmonic would have a variance of $(n\sigma)^2$. The weight $1/n$ is no longer optimum.

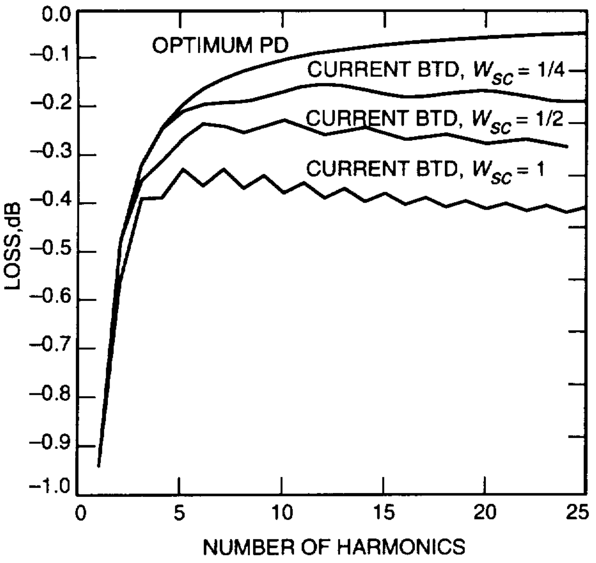


Fig. 3. Degradation as a function of the number of harmonics, using the optimum weights.

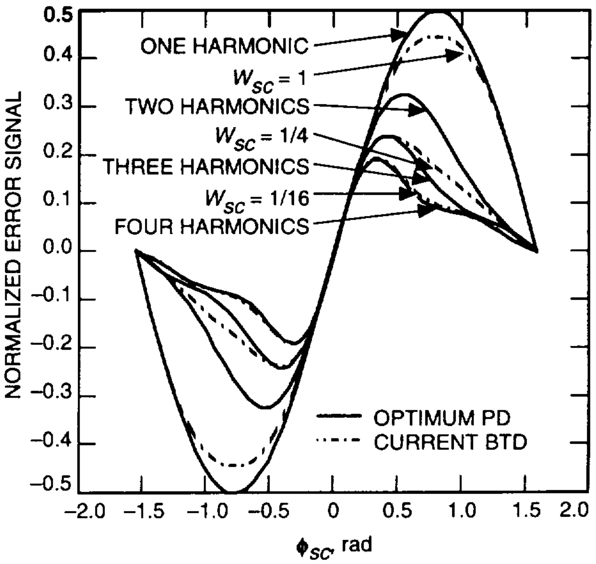


Fig. 4. Normalized S-curves.

To derive the optimum combining weights, we assume that the harmonics are combined using unknown weights b_n. We then express the SNR in terms of the weights. Differentiating the SNR with respect to the weights and setting it to zero, we should obtain the optimum weights.

The optimum weight to combine the demodulated $(2n+1)$th harmonics is derived in the Appendix as

$$b_n = \frac{\overline{\cos[(2n+1)\phi_{sc}]}}{\overline{\cos\phi_{sc}}}\,\frac{1}{2n+1} \tag{3}$$

When ϕ_{sc} is assumed to have a Tikhonov distribution,

$$\overline{\cos(2n+1)\phi_{sc}} = \int_0^\pi \frac{\exp[(1/4)\rho_{sc}\cos\phi_{sc}]}{\pi I_0(\rho_{sc}/4)} \cos\left[\frac{2n+1}{2}\phi_{sc}\right] d\phi_{sc}$$

Assuming that we have 4, 8, and 16 harmonics, the degradations in symbol SNR versus the subcarrier loop SNR, using the optimum weights and the usual $1/n$ weights, are compared in Figs. 5 through 7.

IV. Approximated Optimum Combining Weights in Demodulation

Since the cosine function is "smooth" in the vicinity of zero, for small phase jitters, $n\phi_{sc}$, the expected value of $\cos(n\phi_{sc})$ can be approximated by

$$\mathcal{E}\{\cos(n\phi_{sc})\} \approx 1 - n^2\frac{\sigma^2}{2} \tag{4}$$

The approximated optimum weights are

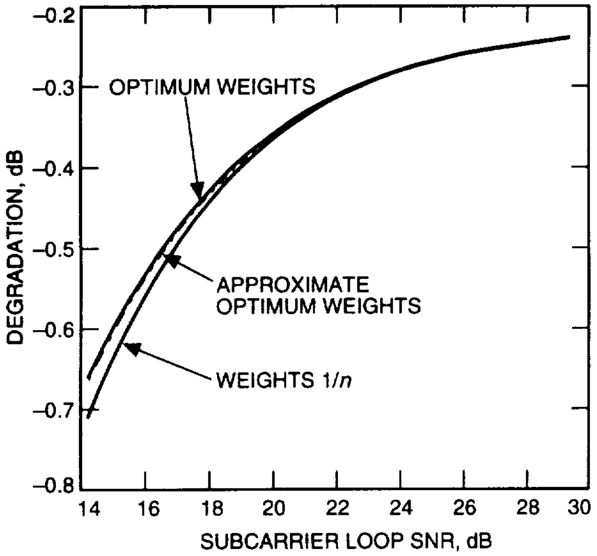


Fig. 5. Symbol SNR degradation when using optimum weights (four harmonics).

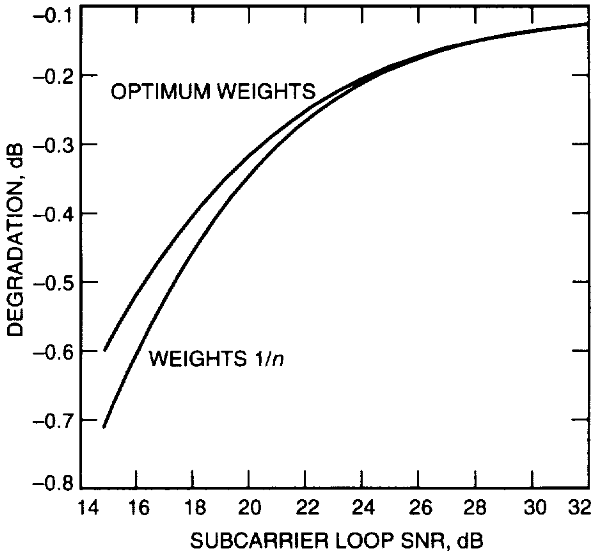


Fig. 6. Symbol SNR degradation when using optimum weights (eight harmonics).

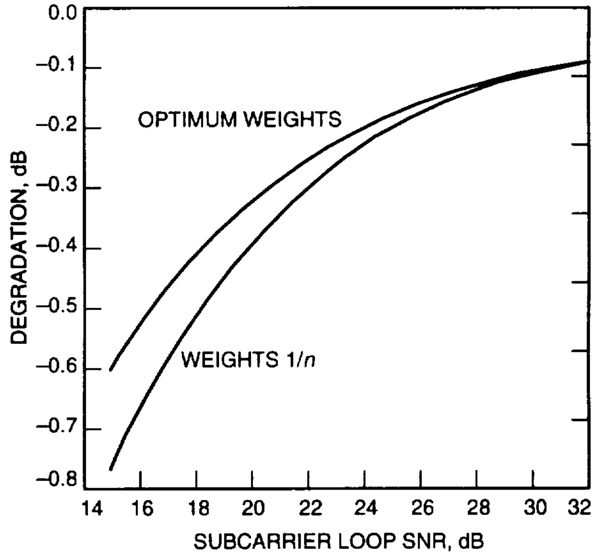


Fig. 7. Symbol SNR degradation when using optimum weights (16 harmonics).

$$b_n \approx \frac{1-(2n+1)^2\sigma^2}{1-\sigma^2/2}\,\frac{1}{2n+1} \tag{5}$$

Note that this approximation is valid only when $n\phi_{sc}$ is small. Using the approximated optimum weights for four harmonics, the symbol SNR degradation is only slightly more than that using the optimum weight as shown in Fig. 5.

V. Conclusion

We presented an optimum way of tracking and demodulating a finite-harmonic subcarrier. We found an optimum phase detector in the sense that the loop SNR is maximized. The more harmonics used, the higher the loop SNR we obtain. However, the linear region of the phase error signal shrinks with the increase of the number of harmonics. Therefore, this optimum phase detector is only appropriate for a finite number of harmonics. Using the optimum phase detector, the loop SNR is about 9.5 dB higher than that of the current BTD using window size 1, and is about 1 dB higher than that of the current BTD with window size 1/4.

For demodulation, we found the optimum combining weights that account for the losses due to the phase jitter. Compared to using the usual $1/n$ combining weights, the use of the optimum combining weights can improve the symbol SNR by 0.1 to 0.15 dB at a low loop SNR (15 dB) and a high number of harmonics (8 to 16).

References

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

[2] Y. Feria, T. Pham, and S. Townes, "Degradation in Finite-Harmonic Subcarrier Demodulation," *The Telecommunications and Data Acquisition Progress Report 42-121, January–March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 78–86, May 15, 1995.

[3] H. Tsou, B. Shah, R. Lee, and S. Hinedi, "A Functional Description of the Buffered Telemetry Demodulation (BTD)," *The Telecommunications and Data Acquisition Progress Report 42-112, October–December 1992*, Jet Propulsion Laboratory, Pasadena, California, pp. 50–73, February 15, 1993.

Appendix

Derivation of the Optimum Combining Weights in Demodulation

After each of the harmonics of the subcarrier is demodulated, the signals from each harmonic demodulation need to be combined. Assume that the combining weight for the $(2n+1)$th harmonic is b_n; the signal amplitude at the lth symbol is

$$s = \sqrt{P_d} d_l \cos\phi_c \frac{2}{\pi} \sum_{n=0}^{L-1} b_n \frac{1}{2n+1} \cos[(2n+1)\phi_{sc}] \tag{A-1}$$

where P_d is the data power, and ϕ_c and ϕ_{sc} are the phase offsets of the carrier and subcarrier, respectively. The noise variance is

$$\sigma^2 = \sum_{n=0}^{L-1} b_n^2 \frac{N_0}{2} R_{sym} \tag{A-2}$$

Taking the ratio of the average signal power and the noise variance, we have the average symbol SNR of the combined signal:

$$\begin{aligned} SNR &= \frac{\mathcal{E}\{s^2\}}{2\sigma^2} \\ &= \frac{\mathcal{E}\{(4/\pi^2)P_d \cos^2\phi_c (\sum_{n=0}^{L-1} b_n \cos[(2n+1)\phi_{sc}]/(2n+1))^2\}}{\sum_{n=0}^{L-1} b_n^2 N_0 R_{sym}} \end{aligned} \tag{A-3}$$

Differentiating the symbol SNR with respect to b_k, $k = 0, \cdots, L-1$, we have

$$\begin{aligned} \frac{\partial(SNR)}{\partial b_k} &= \frac{P_d \overline{\cos^2\phi_c}(4/\pi^2)}{(\sum_{n=0}^{L-1} b_n^2 N_0 R_{sym})^2} \mathcal{E}\left\{ \left[2\sum_{n=0}^{L-1} b_n \frac{\cos[(2n+1)\phi_{sc}]}{2n+1} \; \frac{\cos[(2k+1)\phi_{sc}]}{2k+1} \right] \sum_{n=1}^{L-1} b_n^2 N_0 R_{sym} \right. \\ &\qquad \left. - \left[\sum_{n=0}^{L-1} b_n \frac{\cos[(2n+1)\phi_{sc}]}{2n+1} \right]^2 2 b_k N_0 R_{sym} \right\} \\ &= 0 \end{aligned} \tag{A-4}$$

Simplifying the above equation, we have

$$\mathcal{E}\left\{ \frac{\cos[(2k+1)\phi_{sc}]}{2k+1} \sum_{n=0}^{L-1} b_n^2 - \sum_{n=0}^{L-1} b_n \frac{\cos[(2n+1)\phi_{sc}]}{2n+1} b_k \right\} = 0 \tag{A-5}$$

Let $k = 0$ and $b_0 = 1$; we have

$$\overline{\cos\phi_{sc}} \sum_{n=0}^{L-1} b_n^2 - \sum_{n=0}^{L-1} b_n \frac{\overline{\cos[(2n+1)\phi_{sc}]}}{2n+1} = 0 \tag{A-6}$$

That is,

$$\sum_{n=0}^{L-1} b_n \left[\overline{\cos\phi_{sc}} b_n - \frac{\overline{\cos[(2n+1)\phi_{sc}]}}{2n+1} \right] = 0 \tag{A-7}$$

Finally, solving for b_n, we have the optimum combining weights,

$$b_n = \frac{\overline{\cos[(2n+1)\phi_{sc}]}}{\overline{\cos\phi_{sc}}} \frac{1}{2n+1} \tag{A-8}$$

Enhanced Decoding for the Galileo Low-Gain Antenna Mission: Viterbi Redecoding With Four Decoding Stages

S. Dolinar and M. Belongie
Communications Systems Research Section

The Galileo low-gain antenna mission will be supported by a coding system that uses a (14,1/4) inner convolutional code concatenated with Reed–Solomon codes of four different redundancies. Decoding for this code is designed to proceed in four distinct stages of Viterbi decoding followed by Reed–Solomon decoding. In each successive stage, the Reed–Solomon decoder only tries to decode the highest redundancy codewords not yet decoded in previous stages, and the Viterbi decoder redecodes its data utilizing the known symbols from previously decoded Reed–Solomon codewords.

A previous article [1] analyzed a two-stage decoding option that was not selected by Galileo. The present article analyzes the four-stage decoding scheme and derives the near-optimum set of redundancies selected for use by Galileo. The performance improvements relative to one- and two-stage decoding systems are evaluated.

I. Introduction

This article is a follow-on to [1], which analyzed two enhanced decoding options planned for the Galileo low-gain antenna (LGA) mission: Reed–Solomon redecoding using erasure declarations and Viterbi redecoding using Reed–Solomon corrected symbols. The analysis in [1] produced tables of gains achievable from enhanced decoding under an assumption of infinite interleaving for one, two, or four stages of Viterbi decoding, but no Reed–Solomon redecoding, and for one or two stages of Viterbi decoding, with or without Reed-Solomon redecoding, under the actual Galileo conditions of depth-8 interleaving. The present article looks at the case of four stages of Viterbi decoding and depth-8 interleaving. The four-stage coding system has been selected for implementation to support the Galileo LGA mission.

II. Block Diagram of Coding Options

A block diagram of the various coding options is shown in Fig. 1. A Reed–Solomon encoded data block is interleaved to depth 8 and then encoded by the (14,1/4) convolutional encoder. The Reed–Solomon codewords can have four different levels of redundancies, as depicted by the lightly shaded areas at the bottom of the code block in Fig. 1. The encoded data are modulated, passed over an additive white Gaussian noise (AWGN) channel, demodulated, and presented to a Viterbi decoder. After

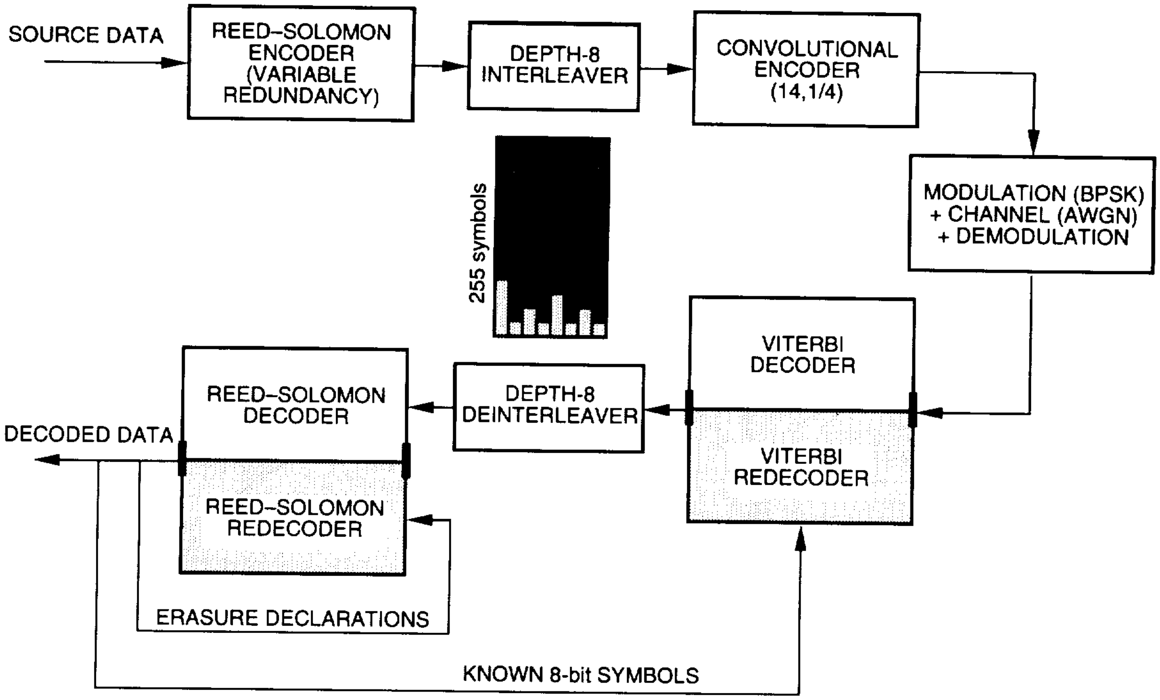


Fig. 1. Coding options.

deinterleaving, the codeword or set of codewords with the highest redundancy is decoded by the Reed–Solomon decoder. The symbols in the codeword(s) decoded by the Reed–Solomon decoder are fed back to assist the Viterbi decoder in *redecoding* the symbols in weaker codewords. The output of the Viterbi redecoder is deinterleaved, and the set of codewords with the next highest redundancy is then decoded by the Reed–Solomon decoder. The newly decoded symbols are fed back to further assist the Viterbi redecoder, and the process is repeated for two more decoding stages until the codewords in all four redundancy classes are successfully decoded.

Figure 1 also shows an option for a shorter feedback loop entirely within the Reed–Solomon decoder using erasure declarations. As shown in [1], Reed–Solomon redecoding using erasure declarations based on error forecasting was worth around 0.19 dB when used in conjunction with one-stage decoding of the Galileo LGA convolutional code. However, the extra gain from using erasure declarations shrinks to a minuscule 0.02 dB when combined with two-stage Viterbi decoding. For four-stage decoding, the marginal improvements gained from erasure declarations are almost nil. Therefore, in the present article, Reed–Solomon redecoding using erasure declarations has not been considered in analyzing four-stage decoding performance. However, the Galileo LGA coding system will still incorporate the capability to perform this type of redecoding, as it may prove helpful in overcoming decoding difficulties not caused by AWGN, such as closing data gaps caused by unsynchronized symbols.

III. The Simulation Data

Figures 2 through 5 are improved and expanded versions of Figs. 1 through 4 of [1], obtained by accumulating many more millions and billions of simulated decoded bits during the interim. Figure 2 shows the bit error rate (BER) and symbol error rate (SER) (for 8-bit Reed–Solomon symbols) for convolutionally encoded symbols decoded by either the Big Viterbi Decoder (BVD) or a software (S/W) Viterbi decoder. The software decoding algorithm is a close approximation to the software decoder that is actually being designed to support the Galileo LGA mission. Figure 3 shows the decoded symbol error

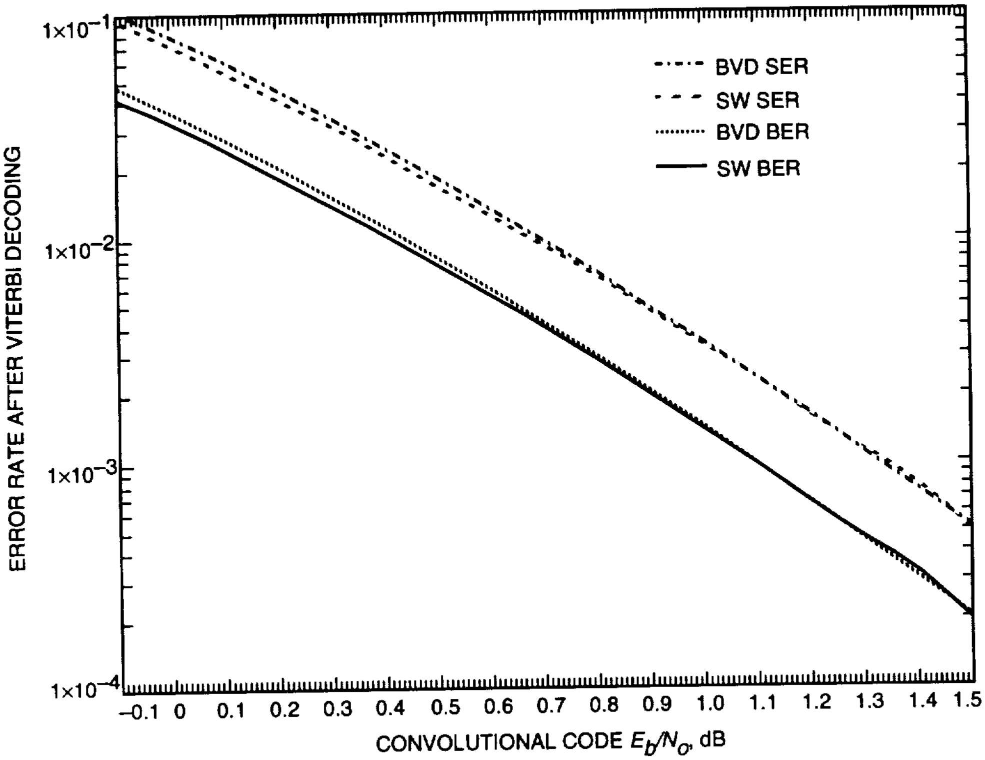


Fig. 2. BER and SER after Viterbi decoding of the (14,1/4) Galileo LGA convolutional code by the BVD and by a software decoder.

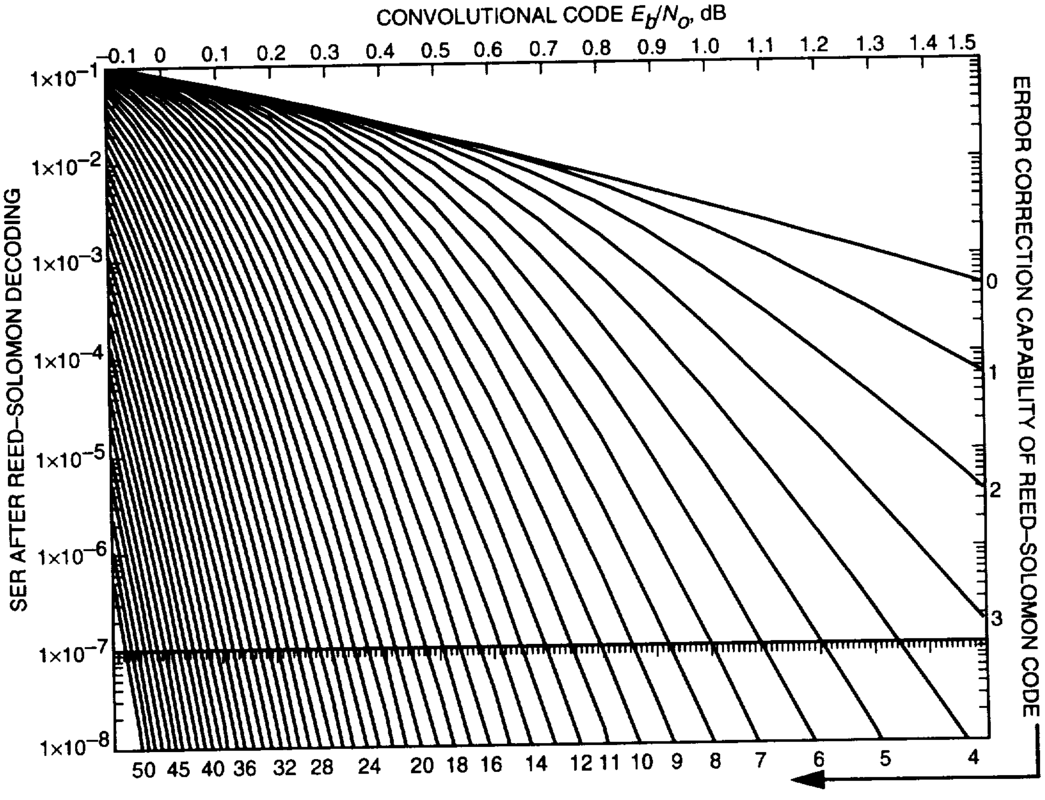


Fig. 3. SER after decoding inner convolutional code with BVD followed by Reed–Solomon decoding with infinite interleaving.

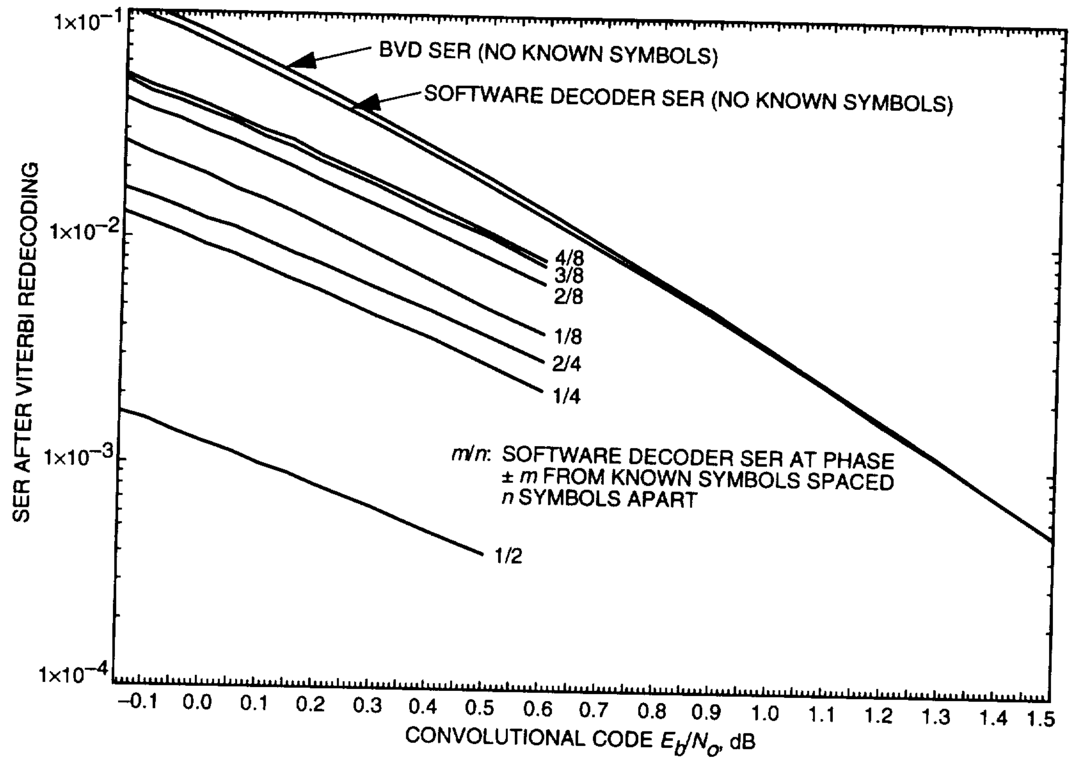


Fig. 4. SER after Viterbi redecoding with the software decoder for various spacings of known symbols and relative phases of unknown symbols.

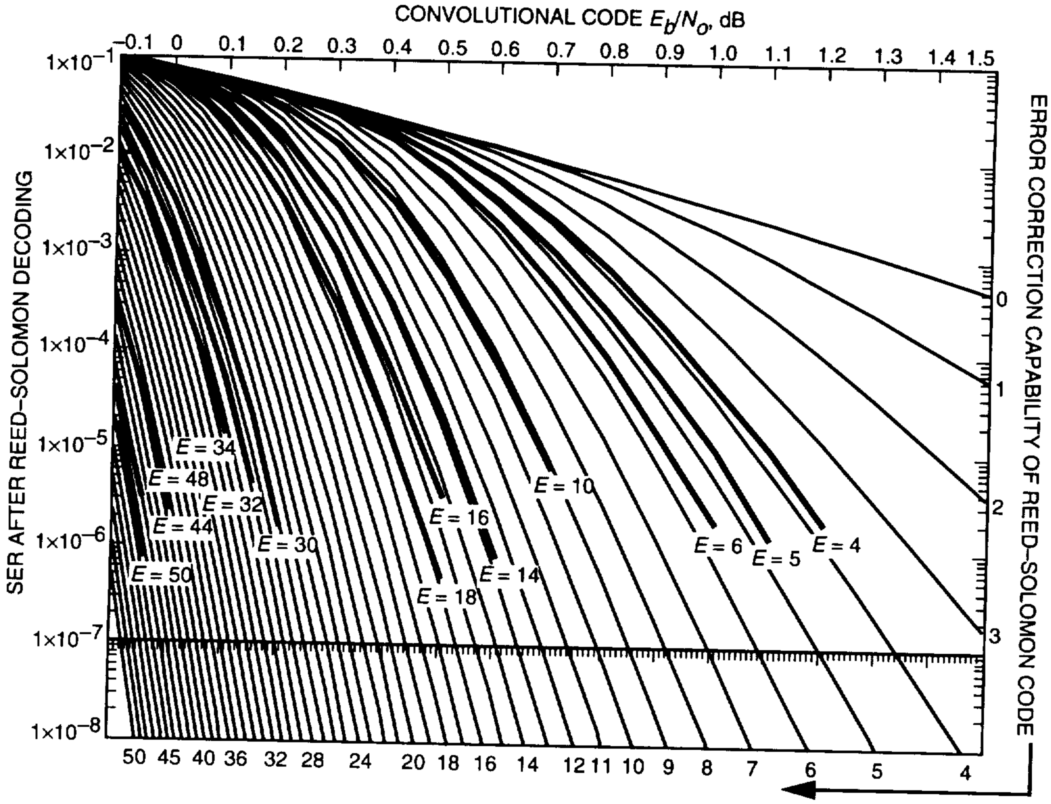


Fig. 5. SER after decoding inner convolutional code with BVD followed by Reed–Solomon decoding with depth-8 interleaving.

rate for a Reed–Solomon decoder receiving convolutionally decoded bits from the BVD; the x-axis of Fig. 3 is the convolutional code signal-to-noise ratio (SNR) E_b/N_o. Figure 4 shows the decoder symbol error rate for the software Viterbi decoder presented with known symbols repeating once every eight, four, or two symbols; as discussed in [1], these SERs depend on the phase of the decoded symbols relative to the locations of the known symbols. The baseline SER curves from Fig. 2 for no known symbols are also included in this figure for reference. Figure 5 repeats the infinite interleaving performance curves from Fig. 3 and overlays curves representing Reed–Solomon decoded SER when the codewords are interleaved to depth 8. As in Fig. 3, the Reed–Solomon decoder for Fig. 5 receives its symbols from the output of the BVD, and the x-axis measures the signal-to-noise ratio at the output of the BVD, not the overall signal-to-noise ratio at the output of the concatenated Reed–Solomon and convolutional codes. SER estimates in Figs. 2 through 5 were taken at spacings of 0.05 or 0.10 dB, and each estimate was based on about 2 Gbits of decoded data from the BVD or 25 to 100 Mbits of data from the software decoder.

As is evident from Figs. 2 and 4, the software decoder performs a few hundredths of a dB better than the BVD (due to a longer truncation length and other factors). The analysis of four-stage decoding requires the use of both Figs. 4 and 5; proper calibration is important between the software-decoder-based curves in Fig. 4 and BVD-based curves in Figure 5. In [1], no distinction was made between the performance of the two decoders, because the software decoder at that time resembled the BVD more closely than the ultimate Galileo LGA decoder. From Fig. 4 is deduced a table of SER-equivalent E_b/N_o operating points for the BVD operating with no known symbols. Whenever the software decoder is decoding data at a value of E_b/N_o in the leftmost column of Table 1, the BVD achieves the same average SER at the "equivalent" E_b/N_o in the columns to the right. There is one BVD-equivalent E_b/N_o column for each of the software-decoder-based curves in Fig. 4. For the case of no known symbols, this really is a near equivalence, and the decoded bit errors from the software decoder and the BVD have very similar burst statistics, not just average SER. For the various cases of known symbols presented to the software decoder, this equivalence is only in terms of average SER. As noted in [1], the error bursts from a decoder presented with known symbols are more benign than those for a decoder operating at the same average SER without any known symbols, as measured by their effects on Reed–Solomon decoding with finite interleaving. Thus, use of the BVD-equivalent signal-to-noise ratios in Table 1 will give slightly conservative predictions of performance in decoding stages 2 through 4.

Table 1. BVD-equivalent signal-to-noise ratios E_b/N_o, dB.

Software decoder E_b/N_o, dB	Known symbol phase/spacing input to software decoder							
	None	4/8	3/8	2/8	1/8	2/4	1/4	1/2
−0.15	—	0.16	0.18	0.24	0.39	0.54	0.62	1.20
−0.10	−0.06	0.20	0.22	0.28	0.43	0.57	0.65	1.22
−0.05	−0.01	0.23	0.25	0.32	0.46	0.61	0.69	1.25
0.00	0.04	0.27	0.29	0.36	0.50	0.64	0.72	1.28
0.05	0.09	0.31	0.33	0.39	0.54	0.67	0.75	1.30
0.10	0.14	0.35	0.37	0.43	0.58	0.70	0.78	1.33
0.15	0.18	0.38	0.41	0.47	0.61	0.74	0.82	1.36
0.20	0.23	0.43	0.45	0.51	0.66	0.77	0.85	1.39
0.30	0.33	0.51	0.53	0.59	0.74	0.84	0.92	1.44
0.40	0.43	0.59	0.61	0.67	0.82	0.91	0.99	—
0.50	0.53	0.67	0.69	0.75	0.90	0.98	1.06	—
0.60	0.62	0.75	0.77	0.82	0.97	1.04	1.13	—

IV. The Basic Analysis Procedure

As noted in [1], even 2 Gbits of BVD-decoded data are insufficient to directly verify Reed–Solomon decoded SERs around 10^{-7} for the case of depth-8 interleaving. Instead, such performance must be inferred by extrapolating the simulated depth-8 curves along the accurately known family of curves for infinite interleaving. Each curve for depth-8 interleaving becomes nearly parallel to a member of the family of infinite interleaving curves, and 10^{-7} performance for depth-8 interleaving may be inferred by extrapolating along an "equivalent" infinite interleaving curve.

The selection and analysis of an appropriate set of codeword redundancies for four-stage decoding is illustrated in the following example. First, select a desired E_b/N_o operating point for the inner convolutional code using the software decoder. This choice is somewhat arbitrary, because the same analysis must be repeated for several values of E_b/N_o in order to determine the optimum operating point. For this example, a convolutional code signal-to-noise ratio E_b/N_o of 0.00 dB will be used. From Table 1, the average SER from the first stage of Viterbi decoding by the software decoder is the same as the average SER produced by the BVD at the BVD-equivalent operating point of 0.04 dB. The output SER from the first Reed–Solomon decoder stage is obtained from the BVD's performance curve in Fig. 5. If the target SER is around 2×10^{-7} (target BER around 1×10^{-7}), the highest redundancy codewords must yield an output SER on the order of 10^{-7} without any help from succeeding decoding stages. From Fig. 5, this can be accomplished at a BVD-equivalent signal-to-noise ratio E_b/N_o of 0.04 dB by using a codeword with correction capability E of approximately 47. From Table 1, the average SER from the second stage of Viterbi decoding with one known symbol every eight is the same as the average SER produced by the BVD with no known symbols at the BVD-equivalent operating points of 0.50, 0.36, 0.29, and 0.27 dB, for codewords with symbols at phases ±1, ±2, ±3, and ±4, respectively, from the known symbol. From Fig. 5, codewords with $E \approx 20$, 26, 29, and 30, respectively, can achieve SERs just under 10^{-7} for these four phases. Looking ahead to the next stage of Viterbi decoding, it can be shown that the biggest payoff comes from locating the second highest redundancy codeword at phase ±4. Then the third stage of Viterbi decoding is accomplished with one known symbol every four, and the BVD-equivalent operating points from Table 1 are 0.72 and 0.64 dB for phases ±1 and ±2, respectively. These require codewords with $E \approx 13$ and 15, respectively, and again it can be shown that the out-of-phase location ±2 makes the best utilization of the fourth and final Viterbi decoding stage. With two of these third highest redundancy codewords per block of eight placed at phases ±2, the final Viterbi decoding operation is accomplished with one known symbol every two, and from Table 1, the BVD-equivalent operating point for the unknown symbols at phase ±1 is 1.28 dB, requiring four lowest-redundancy codewords with $E \approx 5$. This selection process yields a redundancy profile $2E \approx (94, 10, 30, 10, 60, 10, 30, 10)$; this incurs a redundancy overhead cost of 0.58 dB, and the resulting concatenated code signal-to-noise ratio E_b/N_o is 0.58 dB. The overall average SER achieved by four-stage decoding using this redundancy set can be computed approximately by the formula given in [1], $SER = SER_a(1) + 7/8\ SER_b(2) + 3/4\ SER_c(3) + 1/2\ SER_d(4)$, where the indices a, b, c, and d refer to the strongest, next strongest, third strongest, and weakest codewords and (n) refers to decoding during stage n, $n = 1, 2, 3, 4$. Extrapolations from Fig. 5 for $SER_a(1)$, $SER_b(2)$, $SER_c(3)$, and $SER_d(4)$ yield an overall SER of approximately 2×10^{-7}.

Similar analyses starting with convolutional code operating points different from 0.00 dB yield different sets of optimal redundancies and different concatenated code E_b/N_o. It can be shown empirically that the optimum convolutional code operating point for four-stage decoding occurs within a range from approximately −0.10 to +0.05 dB, and that essentially identical performance (within one or two hundredths of a dB) is achievable by suitably selecting different redundancy sets within this range. Also, the best pattern of codeword redundancies always appears to be (a, d, c, d, b, d, c, d), where a is the highest redundancy, b the next highest, c the third highest, and d the lowest. This is the same pattern suggested by an earlier analysis of four-stage decoding in [2].

V. A More Refined Analysis Procedure

The analysis above may be refined by further studying the relationship between the performance curves for depth-8 interleaving and the "equivalent" infinite interleaving curves along which depth-8 SERs on the order of 10^{-7} are extrapolated. Table 2 and Fig. 6 attempt to quantify this equivalence.

Table 2. Equivalent error correction needed for infinite interleaving to yield the same SER.

BVD E_b/N_o, dB	Error correction for depth-8 interleaving												
	$E = 4$	$E = 5$	$E = 6$	$E = 10$	$E = 14$	$E = 16$	$E = 18$	$E = 30$	$E = 32$	$E = 34$	$E = 44$	$E = 48$	$E = 50$
−0.10	—	—	—	—	—	—	—	29.16	30.82	32.46	40.60	43.77	45.18
−0.05	—	—	—	—	—	—	—	28.70	30.34	31.98	40.07	43.44	—
0.00	—	—	—	—	—	—	—	28.26	29.89	31.50	39.90	—	—
0.05	—	—	—	—	—	—	—	27.86	29.45	31.08	—	—	—
0.10	—	—	—	—	—	—	—	27.51	29.10	—	—	—	—
0.15	—	—	—	—	13.78	15.52	17.24	27.25	29.15	—	—	—	—
0.20	—	—	—	—	13.57	15.30	17.02	—	—	—	—	—	—
0.30	—	—	—	9.73	13.23	14.94	16.63	—	—	—	—	—	—
0.40	—	—	—	9.52	12.99	14.74	16.34	—	—	—	—	—	—
0.50	3.96	4.89	5.80	9.36	12.86	14.49	—	—	—	—	—	—	—
0.60	3.88	4.82	5.72	9.31	—	—	—	—	—	—	—	—	—
0.70	3.82	4.73	5.64	9.17	—	—	—	—	—	—	—	—	—
0.80	3.81	4.73	5.63	—	—	—	—	—	—	—	—	—	—
0.90	3.81	4.70	5.64	—	—	—	—	—	—	—	—	—	—
1.00	3.84	4.67	5.59	—	—	—	—	—	—	—	—	—	—
1.10	3.79	4.76	—	—	—	—	—	—	—	—	—	—	—

Table 2 shows, for each Reed–Solomon code tested at depth-8 interleaving, the equivalent error correction capability needed to achieve the same SER if the interleaving were ideal. At each value of E_b/N_o, the equivalent error correction is obtained by linear interpolation on the log scale between the two adjacent infinite interleaving curves. It is quoted as a real number, not an integer, and thus does not represent a realizable code. For example, from Fig. 5 at 0.5 dB, the $E = 16$ curve for depth-8 achieves an SER about halfway between the infinite interleaving curves for $E = 14$ and $E = 15$. The corresponding equivalent error correction capability is listed in Table 2 as $E = 14.49$.

Figure 6 plots a normalized version of the numbers in Table 2. Each point in Table 2 is plotted with an x-coordinate equal to the depth-8 SER at the given value of E_b/N_o and a y-coordinate equal to the ratio of the actual depth-8 error correction capability to the equivalent infinite interleaving error-correction capability listed in Table 2. This ratio is referred to as the depth-8 error magnification factor. For purposes of computing Reed-Solomon code performance, the (nonindependent) symbol errors occurring in depth-8 interleaved codewords are effectively multiplied by the error magnification factor, as compared to an equal average number of independent symbol errors. The error magnification factor is a way of measuring the propensity for one long Viterbi decoder error burst to contribute more than one symbol error to a given Reed–Solomon codeword whenever the codewords are only finitely interleaved.

A more mechanized approach than visually extrapolating the depth-8 performance curves in Fig. 5 utilizes the error magnification factors presented in Fig. 6. The first step is to solve for the redundancies that

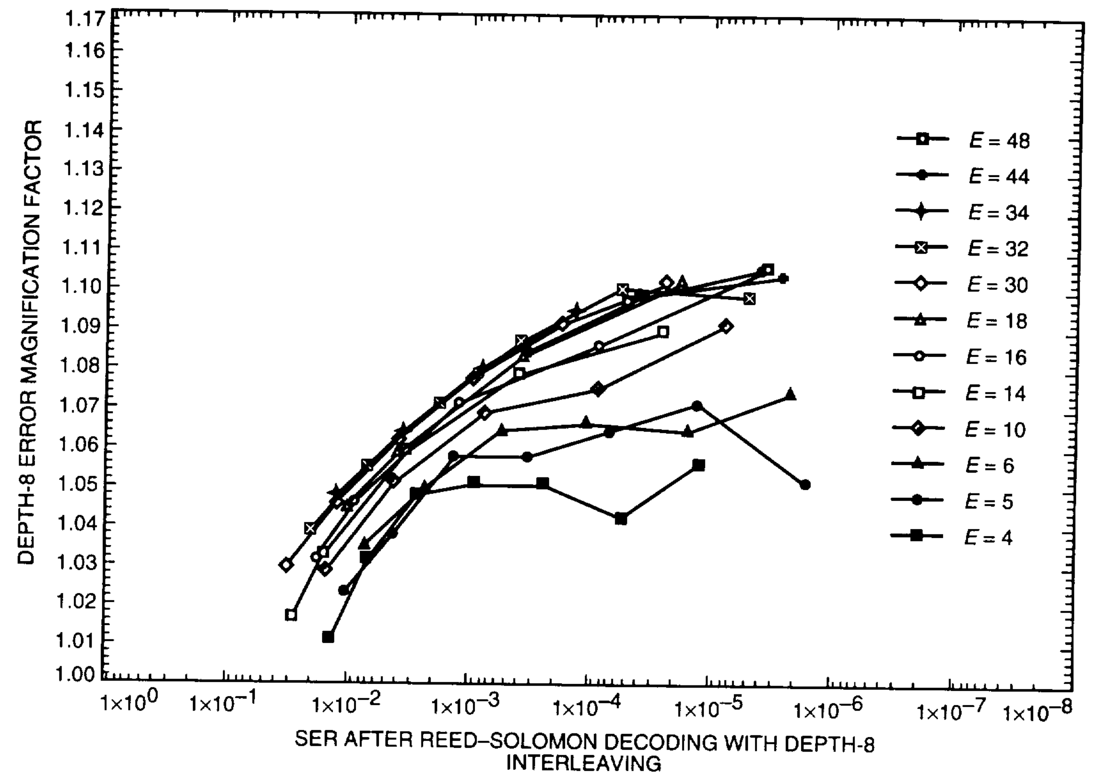


Fig. 6. Effective error magnification factors for Reed–Solomon decoding with depth-8 interleaving, as compared to Reed–Solomon decoding with infinite interleaving.

would be needed if the interleaving were ideal. For this solution, fractional redundancies and fractional error correction capabilities are permissible, and these can be obtained very accurately by interpolation between successive ideal interleaving curves. At each convolutional code operating point, the goal is to solve for a roughly "balanced" set of four redundancies, a, b, c, d, used in the pattern (a, d, c, d, b, d, c, d). A balanced set of redundancies is one for which each class of codewords contributes roughly equally to the overall SER. If the redundancies were not roughly balanced, essentially the same performance could be achieved at lower cost by reducing the redundancy of a codeword class that contributes only a tiny portion of the overall SER.

After a balanced set of fractional redundancies for ideal interleaving is obtained, the next step is to scale these upward by the error magnification factors for depth-8 interleaving and then round these numbers to the nearest or next higher even-integer $2E$. The integer roundoff causes some loss of balance and could cause worse performance if all the roundoffs were downward, hence the rationale for generally rounding upward. Finally, the slightly unbalanced performance of the rounded set of redundancies can be computed for depth-8 interleaving by again applying the magnification factors to obtain the equivalent ideal interleaving fractional redundancies and then interpolating between ideal interleaving curves at adjacent even-integer redundancies.

Figure 6 shows that for testable SERs between 10^{-2} and 10^{-5}, the depth-8 error magnification factor stays within a small range less than 1.11. The error magnification factor increases with decreasing SER but at a decreasing rate. In all cases plotted, it appears to be leveling off by the time it reaches an SER of 10^{-5}; it is not unreasonable to presume that this leveling off will continue through the untestable values of SER around 10^{-7}. The error magnification factors also increase with increasing codeword redundancy $2E$, but appear to increase very slowly for E above 10. Nominal depth-8 error magnification factors for the target SER around 10^{-7} have been inferred by extrapolating the family of curves in Fig. 6. The values used for this analysis are 1.13 for $E = 48 \pm 6$, 1.125 for $E = 32 \pm 4$, 1.12 for $E = 16 \pm 2$, 1.105 for $E = 10 \pm 1$, and 1.09 for $E = 5 \pm 1$.

Table 3 shows the results of this refined analysis procedure for four-stage decoding. At various candidate design point values of E_b/N_o for the software Viterbi decoder, a balanced set of fractional redundancies is obtained to yield an overall SER of 2×10^{-7} with ideal interleaving. The nominal error magnification factors for depth-8 interleaving are applied to the redundancies for ideal interleaving, and the resulting depth-8 redundancies are rounded to even integers. The corresponding SER is computed from the curves for infinite interleaving, again using the nominal error magnification factors. The overall signal-to-noise ratio for the concatenated code is then computed by adding the overhead imposed by the selected redundancies.

Table 3. Design values of redundancies for various possible operating points of the S/W Viterbi decoder, with redundancies *a*, *b*, *c*, *d* repeated according to pattern (*a*, *d*, *c*, *d*, *b*, *d*, *c*, *d*).

Design S/W decoder operating point*	Resulting concatenated code E_b/N_o, dB	Balanced redundancies for ideal interleaving				Assumed error magnification factors for depth-8 interleaving				Design redundancies for depth-8 interleaving				Resulting SER
		a	b	c	d	M_a	M_b	M_c	M_d	a	b	c	d	
−0.10	0.58	98.52	61.53	30.67	9.79	1.13	1.125	1.12	1.09	110	70	34	12	1.9×10^{-7}
−0.05	0.58	90.74	57.71	28.91	9.33	1.13	1.125	1.12	1.09	104	66	32	10	1.8×10^{-7}
0.00	0.58	83.14	53.96	27.02	8.88	1.13	1.125	1.12	1.09	94	60	30	10	2.1×10^{-7}
0.05	0.59	76.27	49.89	25.89	8.57	1.13	1.125	1.12	1.09	86	56	28	10	2.3×10^{-7}

* Convolutional code E_b/N_o, dB.

Note that essentially identical concatenated code design points just under 0.60 dB are obtained over a range of convolutional code design points from −0.10 to +0.05 dB, each using a custom-designed set of optimum redundancies. The set of redundancies listed in Table 3 for a convolutional code design point of 0.00 dB is the same as those discussed in the earlier example. There are many sets of "optimum" redundancies that achieve essentially the same performance. Table 4 lists 24 different redundancy sets that all produce an average SER of 2×10^{-7} at a concatenated code signal-to-noise ratio of 0.58 dB. As in [1], the recommendation in this article is to select the optimum redundancy set with the least spread in redundancies and the highest convolutional code operating point. This set is the one listed in Table 3 for a convolutional code design point of 0.00 dB, with redundancy pattern $(a, d, c, d, b, d, c, d) = (94, 10, 30, 10, 60, 10, 30, 10)$.

The foregoing procedure for selecting a set of redundancies has the advantage of allowing a major part of the analysis to take place without any assumptions about how to extrapolate the depth-8 SER performance data to the 10^{-7} range. This makes it possible to isolate and somewhat quantify the inaccuracies that might result from extrapolation. One might design a conservative set of redundancies for depth-8 interleaving by applying an extra-conservative set of magnification factors. This would require an easily calculable increase in the concatenated code signal-to-noise ratio. At concatenated code operating points just under 0.60 dB, an increase of all magnification factors by 0.05 above the nominal magnification factors costs just 0.03 dB in added overhead; an underestimate this large seems unlikely, as it would put three of the magnification factors above the top edge of the graph in Fig. 6. Designing for the adverse magnification factors would correspond to using $a, b, c, d = 98, 64, 32, 10$, instead of the nominal design, $a, b, c, d = 94, 60, 30, 10$, listed in Table 3 for a convolutional code design point of 0.00 dB. Conversely, once a set of depth-8 redundancies has been selected, the sensitivity of the predicted SER to the extrapolation assumptions could be tested by varying the assumed magnification factors for the final performance

Table 4. Various optimal redundancy sets *a*, *b*, *c*, *d*, repeated according to the pattern (*a*, *d*, *c*, *d*, *b*, *d*, *c*, *d*), that achieve *SER* = 2 × 10^{-7} at a concatenated code signal-to-noise ratio of 0.58 dB.*

Codeword redundancies				Signal-to-noise ratios, dB		*SER*
a	*b*	*c*	*d*	Concatenated	Convolutional	
94	60	30	10	0.58	0.00	2.0×10^{-7}
94	62	30	10	0.58	−0.00	2.0×10^{-7}
96	60	30	10	0.58	−0.00	2.0×10^{-7}
96	62	30	10	0.58	−0.00	2.0×10^{-7}
102	64	32	10	0.58	−0.03	2.0×10^{-7}
102	64	34	10	0.58	−0.04	2.0×10^{-7}
102	66	32	10	0.58	−0.04	2.0×10^{-7}
102	66	34	10	0.58	−0.05	2.0×10^{-7}
102	68	32	10	0.58	−0.05	2.0×10^{-7}
102	68	34	10	0.58	−0.05	2.0×10^{-7}
102	70	32	10	0.58	−0.05	2.0×10^{-7}
104	64	32	10	0.58	−0.04	2.0×10^{-7}
104	64	34	10	0.58	−0.05	2.0×10^{-7}
104	66	32	10	0.58	−0.05	2.0×10^{-7}
104	66	34	10	0.58	−0.05	2.0×10^{-7}
104	68	32	10	0.58	−0.05	2.0×10^{-7}
104	70	32	10	0.58	−0.05	2.0×10^{-7}
106	64	32	10	0.58	−0.04	2.0×10^{-7}
106	64	34	10	0.58	−0.05	2.0×10^{-7}
106	66	32	10	0.58	−0.05	2.0×10^{-7}
106	66	34	10	0.58	−0.06	2.0×10^{-7}
106	68	32	10	0.58	−0.05	2.0×10^{-7}
108	64	32	10	0.58	−0.05	2.0×10^{-7}
108	66	32	10	0.58	−0.05	2.0×10^{-7}

*The first listed redundancy set was chosen to support the Galileo LGA mission.

evaluation over a range of reasonable values. For example, it can be shown that the required operating point of the code nominally designed for 0.00 dB would increase to 0.04 dB if all the error magnification factors were increased by 0.05 above the nominal factors. Thus, the design mismatch only costs an additional 0.01 dB above the 0.03 dB that would accrue if the adverse magnification factors could be anticipated. Because of this relative insensitivity of the code's performance to the exact design parameters, the nominal design was recommended and is being implemented for the Galileo LGA mission.

VI. Four-Stage Redecoding Dynamics: An Example

Figure 7 depicts an example of how the four-stage redecoding process works. The block of eight Reed–Solomon codewords, with error correction capabilities (47, 5, 15, 5, 30, 5, 15, 5), is shown in five snapshots. The first snapshot depicts the bursts of errors emanating from the first-stage of Viterbi decoding before any Reed–Solomon decoding. The 8-bit symbol errors output from the Viterbi decoder are represented by the black left-to-right traces. Correctly decoded symbols occupy the gray regions of the code

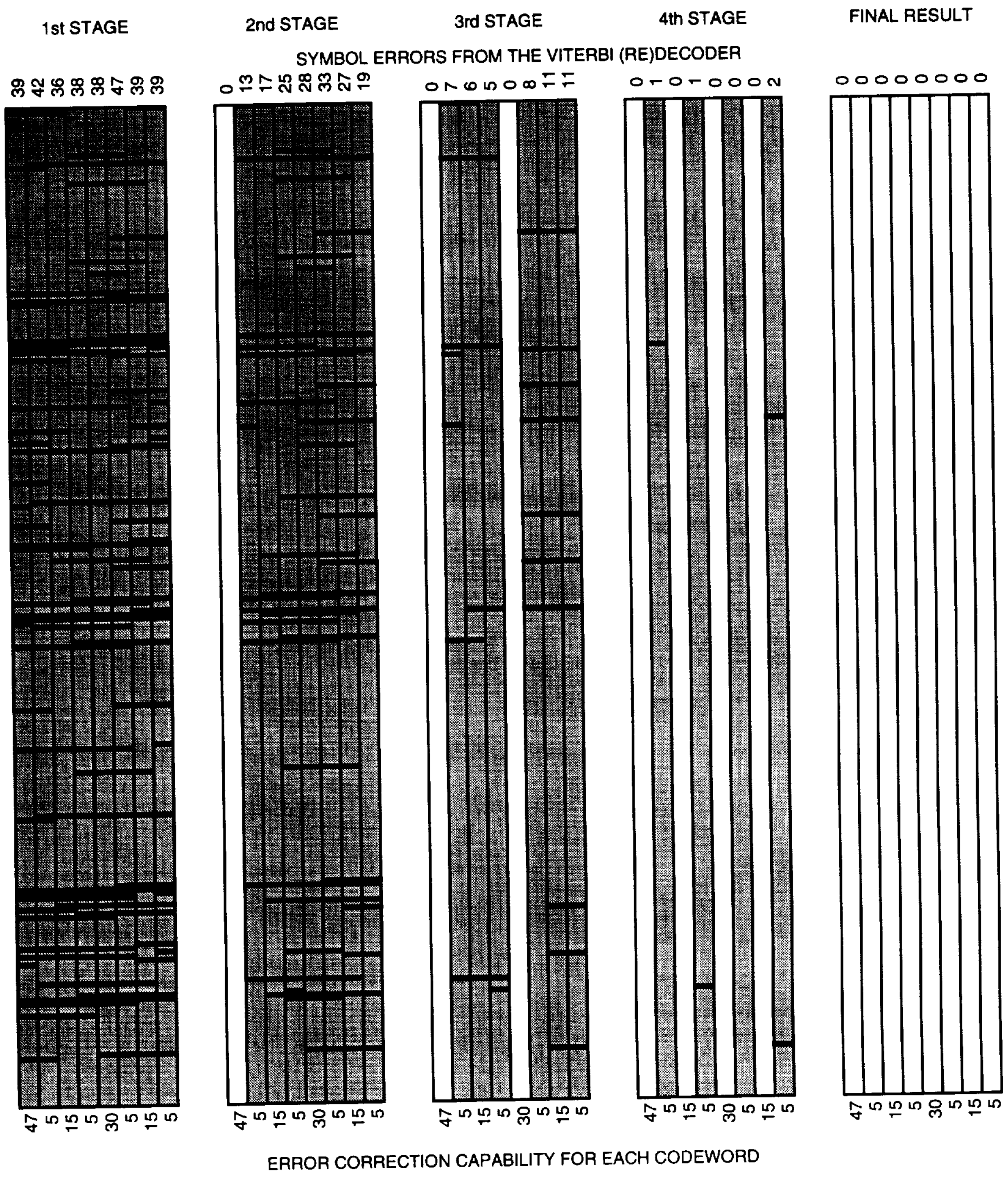


Fig. 7. Illustration of four-stage redecoding dynamics for a sample code block.

block. Shown at the top of the block are the symbol error counts in the individual codewords. These range from 36 to 47, making all the codewords undecodable except for the one with the highest redundancy. The second snapshot shows the code block after the first codeword is corrected by the first-stage of Reed–Solomon decoding. The corrected codeword, depicted in white, now has zero errors and is fed back to assist the second stage of Viterbi decoding. The output of the Viterbi redecoder is improved by the known symbols from the one known codeword, and the resulting error bursts are thinned out and shortened, as shown in the second snapshot. Now the codeword with correction capability 30 is barely decodable despite 28 errors, so this codeword has zero errors in the third snapshot. With only three unknown symbols between pairs of known symbols, the output from the third-stage Viterbi redecoder is improved to the point where both codewords with correction capability 15 can be decoded by the Reed–Solomon decoder. Finally, in the fourth snapshot, with four known symbols out of every eight, the fourth-stage Viterbi redecoder output sports only occasional isolated symbol errors, which are easily corrected by the final stage of Reed–Solomon decoding despite the low correction capability of the fourth-stage codewords.

This example was obtained from simulated data that were intentionally run at several tenths of a dB below the threshold E_b/N_o required to achieve a BER of 10^{-7}, because, at the design threshold, the Viterbi (re)decoder error bursts would have been sparse enough to make the illustration unenlightening. The choice of a below-threshold operating point also demonstrates another facet of the four-stage decoding process. As seen in the first two snapshots, at this low E_b/N_o, the first two codewords are very lucky to be decodable; in fact, some neighboring codewords have error counts equaling or exceeding the error correction capabilities of the first- and second-stage codewords. This emphasizes that the performance of the high-redundancy codes breaks down very rapidly as E_b/N_o is reduced below the design threshold, whereas the lower redundancy codes used in the third and fourth stages are relatively unaffected by a few tenths of a dB reduction.

VII. A Caveat: Undetected Errors

Throughout this analysis and that of [1], it has been assumed that Reed–Solomon codewords are always either correctable or undecodable. The possibility of undetected Reed–Solomon errors has not been considered. This has traditionally been a safe assumption for codes with large correction capabilities E, because from [3] the undetected error rate is bounded by $(1/E!)$ times the detected error rate. However, for the fourth-stage codewords with $E = 5$, the undetected error rate can be up to 10^{-2} times the detected error rate, and so the possibility of undetected errors cannot be ignored.

Undetected errors in the four weakest codewords do pose a real threat if any attempt is made to decode these words before the reliability of the Viterbi redecoder output is strengthened by having every other symbol known, as shown in the next-to-last snapshot in Fig. 7. Conversely, however, if the weakest codewords are always decoded subsequent to the final stage of Viterbi redecoding based on known symbols from all of the four stronger codewords, both detected and undetected errors are so rare that they do not breach the overall BER requirement of 10^{-7}. If E_b/N_o is reduced to the point where this assumption is no longer valid, the stronger codewords become undecodable first, and the fourth stage of decoding is never reached.

The following caveat suggests a very safe, conservative decoding algorithm that always utilizes exactly four stages as described in this article: "Decode no word before its time." Such a decoder takes four times as long to decode as a corresponding one-stage decoder. However, this extreme conservatism is unnecessary because the four codewords with correctabilities 47, 15, 30, and 15 do in fact detect their errors almost always. Therefore, it is safe to allow these codewords to be decoded as early as possible, regardless of whether the corresponding Viterbi (re)decoder output has been cleaned up by the successful decoding of stronger codewords in previous stages. The important caveat is that the four weakest codewords should never be decoded until the Viterbi redecoder utilizes information from all four of the stronger codewords. As long as this restriction is honored, there will be essentially no change in the overall output BER. Yet the modified algorithm can allow for a probabilistic speedup in decoding time, sometimes requiring four stages, three stages, or two stages, but never one stage.

VIII. Summary of Performance Results

Table 5 summarizes the performance results discussed above for four-stage decoding and compares them to previous results for one- and two-stage decoding. For a fair comparison, the one- and two-stage

SERs are recalculated here using the new software decoder calibration curves and the same assumed error magnification factors for depth-8 interleaving. The required signal-to-noise ratios for one- and two-stage decoding are lower than the values quoted in [1] by 0.03 and 0.01 dB, respectively.

Table 5. Performance comparisons for depth-8 interleaving at $SER = 2 \times 10^{-7}$, assuming no Reed–Solomon redecoding using erasure declarations.

Decoding stages	1	2	4
Codeword redundancies	(32,32,32,32,32,32,32,32)	(66,20,22,20,66,20,22,20)	(94,10,30,10,60,10,30,10)
Convolutional code E_b/N_o	0.56 dB	0.19 dB	0.00 dB
Concatenated code E_b/N_o	1.14 dB	0.77 dB	0.58 dB

The results in Table 5 do not include any effects from utilizing Reed–Solomon erasure declarations. As noted earlier, the performance improvement at an SER of 2×10^{-7} for a Reed–Solomon decoder that makes use of erasure declarations is roughly 0.19 dB for one-stage decoding, 0.02 dB for two-stage decoding, and 0.00 dB for four-stage decoding.

Two-stage decoding without erasure declarations is worth 0.37 dB relative to a baseline of one-stage decoding without erasure declarations. Adding erasure declarations gains another 0.02 dB for a total improvement of 0.39 dB. Four-stage decoding, with or without erasure declarations, gains 0.56 dB relative to the baseline and 0.17 or 0.19 dB relative to two-stage decoding with or without erasure declarations, respectively.

Uncertainties in the performance estimates stem mostly from the lack of enough data to directly verify decoded SERs around 10^{-7} with depth-8 interleaving. To first order, errors of this type are likely to affect performance predictions for one-, two-, and four-stage decoding in the same direction; hence, comparisons are not likely to change much. In absolute terms, the adverse uncertainty in four-stage decoding performance is likely to be less than 0.04 dB. The favorable uncertainty due to this effect is slightly smaller, as are the adverse uncertainties for one- and two-stage decoding. As mentioned earlier and in [1], there is an additional favorable uncertainty of a few hundredths of a dB for the multiple-stage decoding cases only, due to the technique of substituting "equivalent" BVD data with the same average SER but less benign burst characteristics, in analyzing the second, third, and fourth decoding stages. The magnitude of this effect has not been assessed, but it might provide an argument for adding a couple of extra redundant symbols to the strongest codeword only, in order to maintain a balanced codeword set if the weaker (redecoded) codewords achieve SERs slightly better than predicted.

As noted earlier, if E_b/N_o drops below the threshold designed to produce a BER of 10^{-7}, the performance of the highest redundancy Reed–Solomon codes falls apart, and the decoding of the interleaved code block never gets started. The overall BER increases dramatically according to the steep slope of the high-redundancy code performance curves. Conversely, if E_b/N_o is increased above the design threshold, further reduction in overall BER below 10^{-7} is hampered by the flatter slope of the performance curve for the four weakest codewords. Figure 8 shows the unusual performance curve that characterizes the four-stage Galileo LGA decoding system. Also shown for comparison are performance curves for the two-stage system analyzed in [1] and the standard one-stage concatenated system with a constant redundancy-32 Reed–Solomon code and no Viterbi redecoding. For four-stage decoding, the error rate falls off very steeply as E_b/N_o is increased toward the design threshold; in this region, performance is dominated by that of the highest-redundancy code(s). Upon reaching the design threshold, the performance curve flattens out; here the dominant error contribution comes from the weakest codewords. The lesson learned from considering the entire four-stage performance curve is that you get exactly what you ask for: a very steep descent reaching the required error rate at a minimum expenditure of E_b/N_o, but slow improvement

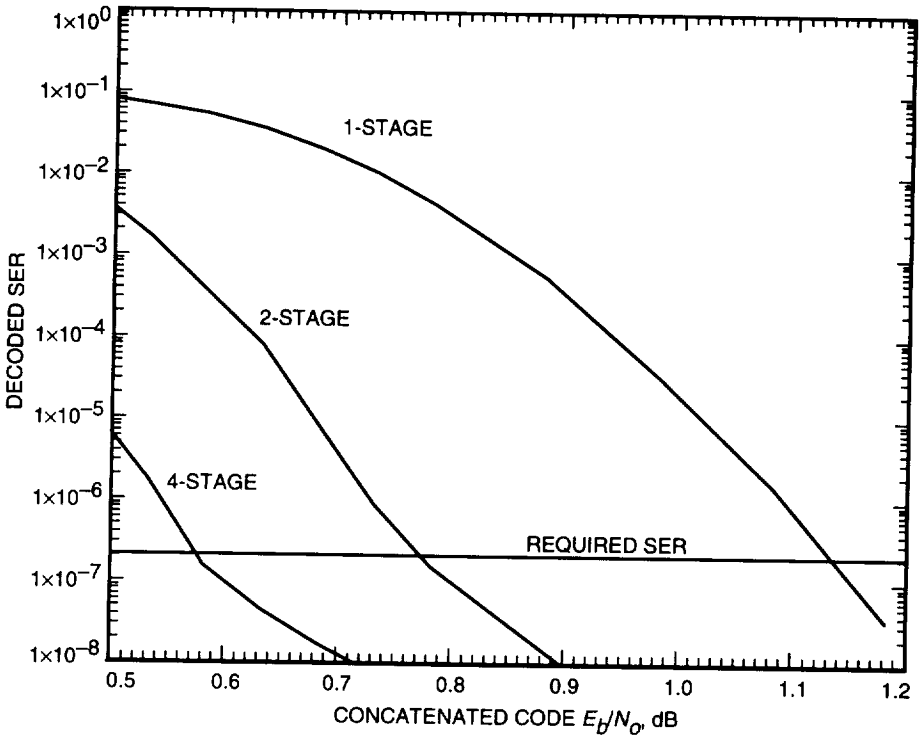


Fig. 8. Performance curves for one-, two-, and four-stage decoding with depth-8 interleaving and near-optimum redundancies.

beyond the requirement if further E_b/N_o is supplied. The same effect is evident but less noticeable for two-stage decoding. For one-stage decoding, the performance curve takes the traditional convex shape. The four-stage performance curve plunges most rapidly to the required SER level, reaching that point 0.56 dB more cheaply than one-stage decoding and 0.17 dB more cheaply than two-stage decoding. On this basis, the Galileo project selected four-stage decoding as the system for maximizing the possible data return.

References

[1] S. Dolinar and M. Belongie,"Enhanced Decoding for the Galileo S-Band Mission," *The Telecommunications and Data Acquisition Progress Report 42-114, April-June 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 96–111, August 15, 1993.

[2] O. Collins and M. Hizlan, "Determinate-State Convolutional Codes," *The Telecommunications and Data Acquisition Progress Report 42-107, July-September 1991*, Jet Propulsion Laboratory, Pasadena, California, pp. 36–56, November 15, 1991.

[3] R. J. McEliece and L. Swanson, "On the Decoder Error Probability for Reed–Solomon Codes," *The Telecommunications and Data Acquisition Progress Report 42-84, October-December 1985*, Jet Propulsion Laboratory, Pasadena, California, pp. 66–72, February 15, 1986.

TDA Progress Report 42-121

May 15, 1995

Testing the Performance of the Feedback Concatenated Decoder With a Nonideal Receiver

Y. Feria and S. Dolinar
Communications Systems Research Section

One of the inherent problems in testing the feedback concatenated decoder (FCD) at our operating symbol signal-to-noise ratio (SSNR) is that the bit-error rate is so low that we cannot measure it directly through simulations in a reasonable time period. This article proposes a test procedure that will give a reasonable estimate of the expected losses even though the number of frames tested is much smaller than needed for a direct measurement. This test procedure provides an organized robust methodology for extrapolating small amounts of test data to give reasonable estimates of FCD loss increments at unmeasurable minuscule error rates.

Using this test procedure, we have run some preliminary tests on the FCD to quantify the losses due to the fact that the input signal contains multiplicative non-white non-Gaussian noises resulting from the buffered telemetry demodulator (BTD). Besides the losses in the BTD, we have observed additional loss increments of 0.3 to 0.4 dB at the output of the FCD for several test cases with loop signal-to-noise ratios (SNRs) lower than 20 dB. In contrast, these loss increments were less than 0.1 dB for a test case with the subcarrier loop SNR at about 28 dB. This test procedure can be applied to more extensive test data to determine thresholds on the loop SNRs above which the FCD will not suffer substantial loss increments.

I. Introduction

Thus far, the feedback concatenated decoder (FCD) has only been tested with signals corrupted by pure additive white Gaussian noise (AWGN). In reality, the FCD takes input from the output of a receiver, such as the buffered telemetry demodulator (BTD), which contains multiplicative non-Gaussian noise. The FCD is composed of a Viterbi decoder (VD) and a Reed–Solomon (RS) decoder, as shown in Fig. 1. The RS decoder decodes four different types of codewords with different error correction capabilities: $E = 47, 30, 15, 5$. In each eight-codeword frame, the single codeword with the highest correctability, $E = 47$, is decoded first. This decoded word is passed back to the Viterbi decoder, which redecodes its data utilizing the new information. Then the RS decoder is able to decode the single codeword with the next highest correctability, $E = 30$, and it feeds this word back to the Viterbi decoder for another redecoding. At the next stage, the two codewords with correctability $E = 15$ are decoded and finally, after one more Viterbi redecoding, the RS decoder decodes the final four codewords with correctability $E = 5$.

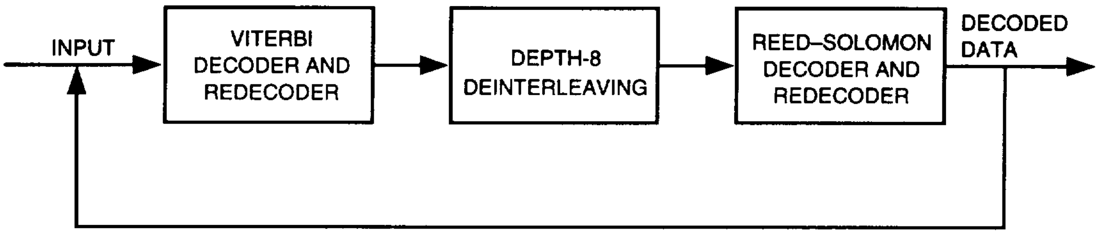

INPUT
VITERBI DECODER AND REDECODER
DEPTH-8 DEINTERLEAVING
REED–SOLOMON DECODER AND REDECODER
DECODED DATA

Fig. 1. The FCD structure.

Since the metrics in the Viterbi decoder are designed to be optimum for AWGN, they are not the optimum metrics for the actual BTD output; hence, there are additional losses in the Viterbi decoder. Similarly, the predicted performance of the four-stage RS decoder [1] is based on the error burst characteristics of a Viterbi decoder decoding symbols corrupted by pure AWGN and, hence, will be different for the actual BTD output. There are no analytical techniques for characterizing these losses; therefore, simulations are used to characterize the additional losses in the FCD due to the nonideal receiver (BTD).

The required error rate at the output of the FCD is extremely low. The required bit error rate (BER) is 10^{-7}, which corresponds to an 8-bit RS symbol error rate of 2×10^{-7} and an RS codeword error rate of 10^{-5} to 10^{-6}. To directly measure the FCD error rate, we would need to simulate several million codewords, which would take thousands of days using the current computing systems. In this article, we propose a test procedure that estimates FCD performance to the 10^{-7} level by applying sensitive extrapolation techniques to measurable hypothetical error rates for weaker RS codes (i.e., codes with lower correctability) within the same family of codes as the four actual RS codes used in the FCD.

II. Test Setup

We first generated an encoded data stream and modulated it with a suppressed carrier near baseband and four harmonics of a subcarrier (upper and lower sidebands) also at almost baseband. We then added white Gaussian noise to the modulated data and used the result as a test signal. Next we ran this test signal through the BTD, and at the BTD output, we measured the symbol error rate by comparing the hard symbols to the known test symbols. From this error rate, we computed the corresponding symbol signal-to-noise ratio (SSNR or E_s/N_o), assuming AWGN. We also made a second SSNR measurement from the split-symbol signal-to-noise ratio estimator built into the BTD. Finally, we fed the soft symbols obtained from the BTD to the FCD.

We decided to include the BTD in the test setup instead of modeling the BTD output with symbols containing multiplicative noises with a Tikhonov distribution. The reason is that the Tikhonov distribution is an appropriate assumption only for first-order loops, whereas the BTD actually uses second- or third-order tracking loops whose phase noise distribution is unknown.

We looked at the decoded output of the FCD and discarded any undecodable data before the receiver was in lock. From the in-lock decoded output, we counted how many 8-bit RS symbols the RS decoder corrected in each of its four stages of decoding. From the histogram of the numbers of corrected symbols, we estimated the performance of both the Viterbi decoder and the Reed–Solomon decoder in each decoding stage, and we used these measurements to estimate additional losses that show up at the output of the FCD but are not apparent at the output of the BTD. The analysis method for obtaining these estimates is described in the next section.

The test setup is shown in Fig. 2. This setup consists of a random information-bit generator, a carrier-subcarrier modulator, an AWGN generator, a receiver (BTD), and a decoder (FCD). The test signal does not have filtering effects on it; hence, it can be generated at a high speed. The speed is crucial in this case, since hundreds or thousands of frames need to be generated in a reasonable amount of time.

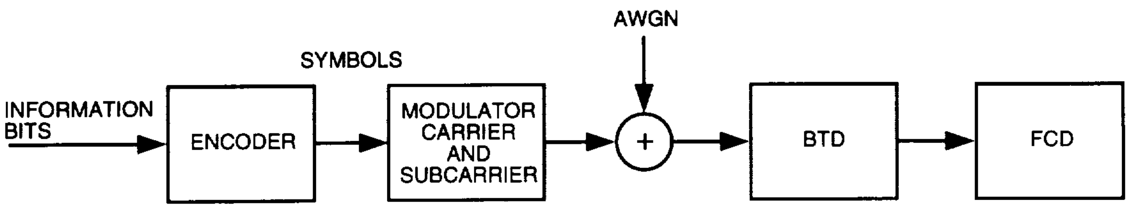


Fig. 2. Test setup.

The input to the receiver (BTD) is an encoded symbol stream on a suppressed carrier and the first four harmonics of a square-wave subcarrier at almost baseband. The size of the losses depends on the SSNR and the parameters of the carrier, subcarrier, and symbol synchronization loops, such as loop bandwidths and window sizes.

For our tests, we arbitrarily picked six sets of typical receiver parameters as examples, designated as cases A through F. In the first two cases, the SSNR is chosen to be −5 dB, which is a typical value in operation slightly above the design threshold where decoder errors are very rare. The loop SNRs are chosen to be about 20 and 18 dB for cases A and B, respectively. In cases C through F, the input SSNR is set at −5.5 dB. This is to push the effective bit SNR slightly below the design threshold, where the decoder may fail to decode. The loop SNRs are varied from about 20 to 16 dB, where below 16 dB the loops may have cycle slips.

Table 1 summarizes the receiver parameters associated with cases A through F, along with the corresponding estimates of losses at the output of the BTD before any decoding by the FCD. It is seen that BTD loss increments on the order of 0.3 dB are typical for all test cases except case D, which has a high subcarrier loop SNR of 28.5 dB and a resulting BTD loss increment under 0.1 dB.

Table 1. FCD test conditions.

Case	Loop BW, Hz		Window size	Loop SNR, dB	Carrier Doppler rate, mHz/s	E_s/N_o at BTD input, dB	E_s/N_o at BTD output, dB	BTD loss increment, dB
A	Carrier	0.10		20.5	0.1	−5.22	−5.49	0.27
	Subcarrier	0.05	1.0	19.3				
	Symbol	0.02	0.5	16.1				
B	Carrier	0.17		18.2	0.1	−5.22	−5.54	0.32
	Subcarrier	0.06	1.0	18.6				
	Symbol	0.01	0.5	18.0				
C	Carrier	0.10		19.7	0.0	−5.72	−6.01	0.29
	Subcarrier	0.05	1.0	18.5				
	Symbol	0.02	0.5	15.2				
D	Carrier	0.08		20.7	0.0	−5.72	−5.79	0.07
	Subcarrier	0.04	0.25	27.9				
	Symbol	0.01	0.25	21.6				
E	Carrier	0.10		19.7	0.0	−5.72	−5.98	0.26
	Subcarrier	0.05	1.0	18.5				
	Symbol	0.01	0.5	18.2				
F	Carrier	0.23		16.1	0.0	−5.72	−6.06	0.34
	Subcarrier	0.05	1.0	18.5				
	Symbol	0.02	0.5	15.2				

III. Classifying and Measuring the Losses

We classified the losses due to the nonideal receiver into several categories. The first category is the loss measured at the output of the receiver without any decoding; this is the BTD loss increment reported in Table 1. Any extra loss beyond the BTD loss increment that is measurable at the output of the full FCD is referred to as the FCD loss increment. The FCD loss increment is further subclassified into two types of stage-by-stage losses. The VD loss increment for a given decoding stage is the loss measured at the output of the Viterbi decoder assuming correct RS decoding in previous stages but without any Reed–Solomon decoding in succeeding stages; this loss is measured relative to the performance of a stand-alone Viterbi decoder operating with differing amounts of known information from stage to stage. The RS loss increment for a given stage is the loss measured at the output of the RS decoder assuming the observed average error rate from the Viterbi decoder for that stage; this loss is measured relative to the performance of a Reed–Solomon decoder operating with depth-8 interleaved symbols corrupted by pure AWGN.

The RS loss increment is referred to the FCD's performance with codewords interleaved to depth 8, not infinitely interleaved. As reported in [1], there is a 0.06- to 0.07-dB degradation due to finite depth-8 interleaving, but that loss is already accounted for in the FCD's performance baseline with an ideal BTD.

It should be emphasized that all the loss components evaluated in our tests arise from the nonideal noise originating in the receiver, and the various categories of loss increments estimate the successive degradations caused by the corrupted symbols as the processing moves further downstream from the receiver. Ideally, we would like to know the losses in each of the components, so that in the event of a fault, we can pinpoint where the fault may be. We also want to quantify the losses in smaller components so that we know where the losses are more significant and may need to be improved in the future.

A. BTD Loss Increment

The symbol SNR (SSNR or E_s/N_o) at the input to the BTD was −5.22 dB for cases A and B, and −5.72 dB for cases C, D, E, and F. These input SSNRs were achieved by keeping four harmonics from full-spectrum signals with SSNRs of −5.00 and −5.50 dB, respectively.

The SSNR at the output of the BTD was measured using the split symbol estimator. This estimate was also corroborated by measuring the hard-limited symbol error rate and looking up the corresponding SSNR on the standard performance curve for an uncoded AWGN channel. In all six test cases, the two SSNR estimation techniques gave almost identical estimates. The difference between the estimated output SSNR and the tested input SSNR is what we call the loss in the receiver or the BTD loss increment. Note that this definition of the BTD loss increment does not include the 0.22 dB lost before the BTD input due to using only four harmonics.

B. Stage-by-Stage VD Loss Increments

The effective bit SNR (BSNR or E_b/N_o) at the output of the Viterbi decoder for each decoding stage was estimated by counting the number of 8-bit Reed–Solomon code symbols corrected by the FCD in that stage. If it can be assumed that the FCD always decodes the truth data, then the observed symbol correction rate from the FCD equals the Viterbi decoder's output symbol error rate (SER) for 8-bit Reed–Solomon symbols. This is the output SER for a Viterbi decoder operating in a stand-alone mode but with different patterns of known symbols from previous RS decoding stages. The measured SER is mapped to a corresponding effective BSNR using the performance curve for a stand-alone Viterbi decoder for Galileo's (14,1/4) convolutional code, given a particular pattern of known 8-bit symbols from previous RS decoding stages (assumed successful); these Viterbi decoder reference curves were obtained in [1] and are reproduced here as Fig. 3. The VD loss increment for the given decoding stage is the difference between this measured effective BSNR and the BSNR computed from the estimated SSNR at

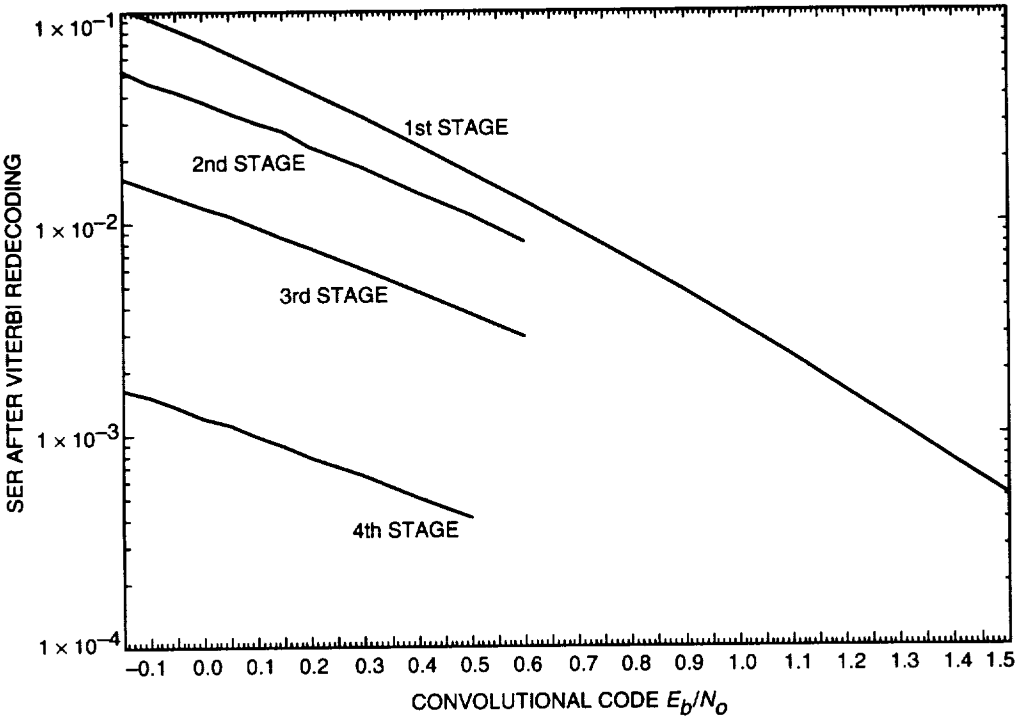


Fig. 3. Stage-by-stage Viterbi decoder performance for decoding Galileo's (14,1/4) convolutional code.

the output of the BTD. The values of BSNR used in these calculations are per bit at the output of the Viterbi decoder, and they do not include the 0.58-dB overhead to account for the average rate of the RS codewords.

C. Stage-by-Stage RS Loss Increments

The stage-by-stage RS loss increments cannot be measured directly using reasonable amounts of test data. They are estimated by a complicated method similar to that used in [1] for estimating the losses due to using depth-8 interleaving instead of infinite interleaving.

The four stages of the FCD are designed to be in "balance" with each other [1]. At the design operating point where the required error rate of 10^{-7} is just barely achieved, all four stages contribute comparable portions to the overall error rate. If the operating point is at a lower E_b/N_o than the design point, the performance of the RS code with the highest correctability, $E = 47$, deteriorates much more rapidly than the others, and so the error rate is dominated by errors from the first stage. If the operating point is at a higher E_b/N_o than the design point, the code with the lowest correctability, $E = 5$, improves very slowly relative to the others, and the error rate is dominated by errors from the fourth stage.

The effects on FCD performance of the non-AWGN introduced by the BTD must be evaluated stage by stage. If the design balance point is disturbed, the performance degradation will be dominated by that of the most affected stage.

1. Method for Estimating Losses Due to Depth-8 Interleaving. The analysis in [1] introduced a technique for estimating stage by stage the performance difference between a hypothetical FCD processing infinitely interleaved Reed–Solomon symbols and the actual FCD that must work with symbols interleaved only to depth 8. Depth-8 performance could be directly simulated only to an overall error rate of about 10^{-5} or 10^{-6}. Estimates of the design operating point required to produce a 10^{-7} error rate were obtained by extrapolation. The extrapolation method was to compare the simulated depth-8 data with an entire family of Reed–Solomon performance curves based on infinite interleaving, for all possible values of the error correction capability E of the code. The infinite-interleaving performance curves could be accurately calculated to error rates below 10^{-7}, and the depth-8 performance data were extrapolated to the 10^{-7} level by reference to the family of infinite-interleaving curves. This extrapolation was accomplished by first calculating "error magnification factors," relating (in the region where depth-8 data existed) the actual Reed–Solomon error correction capability to that of a code that would achieve the identical output error rate if its input symbols had the same input error rate but were infinitely interleaved. The error magnification factors were found to vary slowly and smoothly over the range of depth-8 data, and they could be extrapolated from the 10^{-5} level to the 10^{-7} level with a high degree of confidence.

2. Test Method for Estimating Losses Due to the BTD. In the present tests, we are trying to estimate 10^{-7} performance with much less data than was available in [1] for determining the effects of depth-8 interleaving. However, the basic extrapolation principle is the same. We first measure stage-by-stage FCD error rates, under the actual conditions introduced by the nonideal BTD, to the lowest error level that can be feasibly tested (in this case about 10^{-3} or 10^{-4}). Then all of the measured data are converted to equivalent error magnification factors by reference to the entire family of RS performance curves based on infinite interleaving; these reference curves are shown in Fig. 4. The magnification factors are extrapolated to the required 10^{-7} error level to give an estimate of the total degradation relative to infinite interleaving. Finally, the degradation due to depth-8 interleaving, already estimated in [1], is subtracted to give the net degradation due to the nonideal BTD.

The degradation measured in terms of error magnification factors is translated into an equivalent SNR loss by means of the calibration curves shown in Fig. 5. This figure plots the error magnification factor at an RS output SER of 10^{-7} versus the Viterbi decoder bit SNR that would achieve the same SER according to the stand-alone first-stage Viterbi decoder reference performance curve in Fig. 3, and assuming infinite interleaving. It is seen from Fig. 5 that the translation from magnification factors into SNR losses follows a nearly universal straight-line rule, regardless of the error correction capability E of the outer Reed–Solomon code. The calibration rule for all values of E greater than or equal to approximately 15 is that 8 dB of error magnification factor equals 1 dB of equivalent SNR loss. For E less than 15, this ratio drops drops off very gradually, staying above 6 to 1 for all values of E greater than or equal to 2.

A difference between these tests and the simulations in [1] is that for these tests the nonideal error rates were not directly measured, but instead were estimated without reference to known "truth" data. These estimates were obtained using histograms of Reed–Solomon symbol corrections reported by the FCD. A similar method[1] utilized only average symbol correction rates rather than entire histograms; this method allows accurate stage-by-stage measurement of the VD loss increment, but does not produce an estimate of the RS loss increment.

Suppose that a code with error correction capability E actually reports $e \leq E$ corrections for a given codeword. Then, assuming that this corrected codeword is not erroneous, any Reed–Solomon code with the same block length and correction capability $E' \geq e$ would have corrected a corresponding codeword with symbol errors in the same e places, whereas codes with correction capability $E' < e$ would have failed to decode (or possibly decoded incorrectly). By collecting a histogram of observed values of e for different decoded codewords, we can simultaneously estimate the RS decoded error rates for a whole family of codes with error correction capabilities $E' \leq E$. After noting RS output SER as a function of E', we look up the corresponding ideal error correction capabilities E^* that would achieve the same SER values under an infinite interleaving assumption. This yields the error magnification factors E'/E^*

[1] S. Shambayati, "DGT Bit Error Rate Inference From Reed–Solomon Correction Rate Per Correctable Reed–Solomon Symbol," JPL Interoffice Memorandum 3393-94-SS02, Rev. A (internal document), Jet Propulsion Laboratory, Pasadena, California, May 15, 1995.

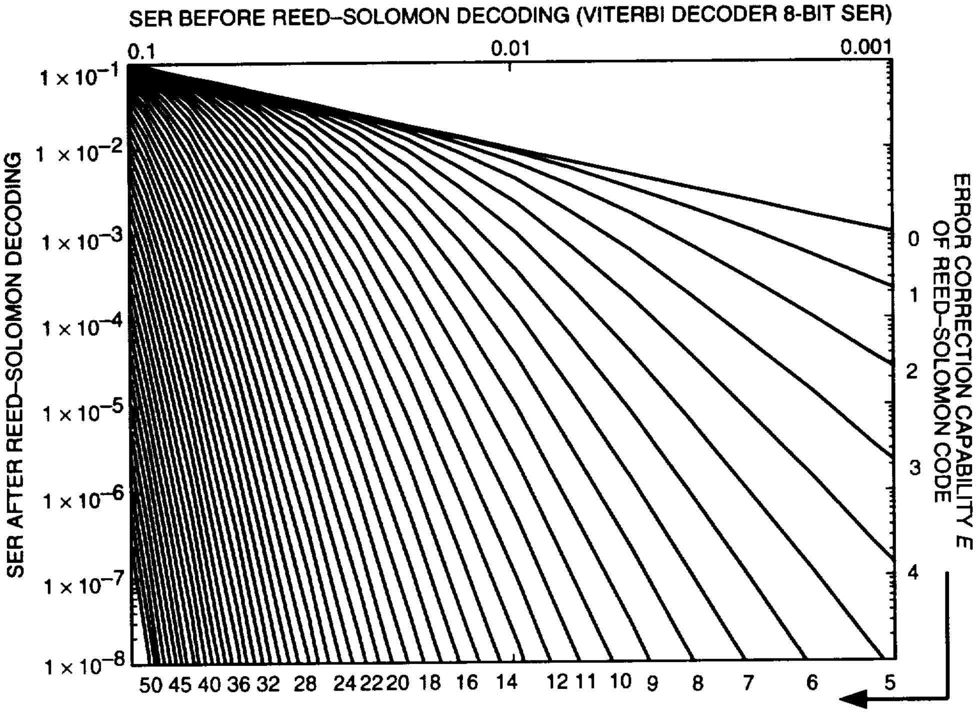


Fig. 4. Ideal Reed–Solomon performance curves for independent symbol errors (infinite interleaving).

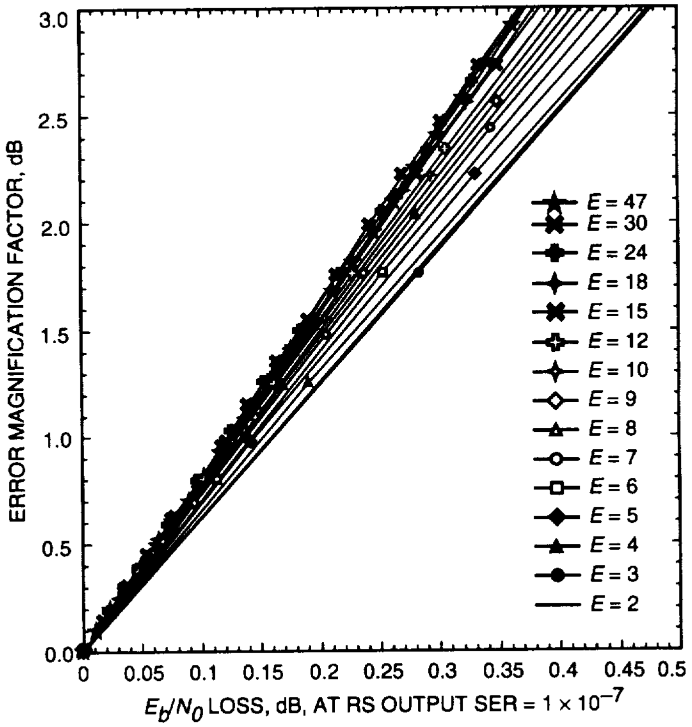


Fig. 5. Error magnification factor calibration curves for Galileo's (14,1/4) convolutional code concatenated with a blocklength-255 Reed–Solomon code.

as a function of SER. For the purposes of this computation, the equivalent ideal correctabilities E^* are determined as nonintegral values by interpolating between the discrete integer-valued curves in Fig. 4.

The process of measuring error magnification factors as a function of RS output SER must be repeated for each of the four stages separately. At each stage, error magnification factors are computed for hypothetical values of correctability E' less than or equal to the actual correctability of the RS code used in that stage. For this calculation, data are only collected from the specific codeword(s) designed to be corrected during that decoding stage. Very small values of E' are discarded if they are less than the code's block length (255) times the RS input average SER (i.e., the average output 8-bit SER from the Viterbi decoder), because they would not correspond to useful codes (even hypothetically) at the given input SER. Values of E' greater than or equal to the maximum number of corrected symbol errors e are also discarded because, for these values of correctability, there are insufficient data to detect an error rate greater than zero.

3. Test Results for Estimating the Nonideal BTD Effects. Figures 6 through 11 show, for cases A through F, the measured RS output SER for hypothetical correctabilities E' in each of the four decoding stages. The measured SERs for different values of E' are plotted as small circles at the same value of Viterbi decoder bit SNR. In the first stage, this is the effective VD BSNR after accounting for the BTD and VD loss increments. In stages 2 through 4, the horizontal coordinate plotted in Figs. 6 through 11 is an equivalent first-stage VD BSNR computed by looking up the output SER of the Viterbi *redecoder* on the first-stage VD performance curve in Fig. 3. Also shown in Figs. 6 through 11 is a family of reference performance curves assuming infinite interleaving and different values of correctability. The horizontal coordinate of the reference curves is similarly normalized to an equivalent first-stage BSNR. The figures show one small circle and one reference curve for each value of E' between the minimum and maximum values described above (and labeled explicitly in the figures).

Notice that the FCD test points represented by the small circles are generally displaced slightly to the right of the corresponding reference curves assuming infinite interleaving and AWGN. This same conclusion holds relative to the slightly degraded set of reference curves reported in [1] for depth-8 interleaving but still assuming ideal AWGN. The RS loss increment in the first stage is the horizontal displacement of the small circles from the depth-8 reference curves. For stages 2 through 4, this horizontal displacement represents the sum of the RS and VD loss increments for the given stage minus the VD loss increment for the first stage.

The RS loss increments that can be observed directly as horizontal displacements in Figs. 6 through 11 are for SERs several orders of magnitude higher than 10^{-7} and hypothetical values of correctability much lower than those of the actual four RS codes used in the FCD. The RS loss increment is extrapolated to the 10^{-7} level by first taking the SERs plotted as small circles and reinterpreting them as equivalent error magnification factors. The results are shown in Figs. 12 through 15. It is seen that the magnification factors for the first three stages approach or exceed 1 dB for output SERs in the 10^{-3} to 10^{-4} range. At the measured rate of increase of magnification factors between 10^{-2} and 10^{-4}, it is likely that the error magnification factors will be around 2 dB, and possibly as high as 3 dB, when the error rate is reduced to the order of 10^{-7}. In the fourth stage, the data are more difficult to extrapolate, but the magnification factors are somewhat lower than in the other three stages.

Since the error magnification factors are computed relative to an equivalent performance curve under an infinite interleaving assumption, the computation of the RS loss increment requires an adjustment to account for the portion of the error magnification that is due to depth-8 interleaving; this was already predicted and accounted for by the analysis in [1]. Figure 16 shows that the error magnification factors for depth-8 interleaving (assuming AWGN) are consistently below 0.5 dB and seem to approach 0.5 dB very reliably at 10^{-7} SER for all except the very lowest values of correctability. For small values of correctability, the extrapolated value of the magnification factor may be 0.1 to 0.2 dB smaller.

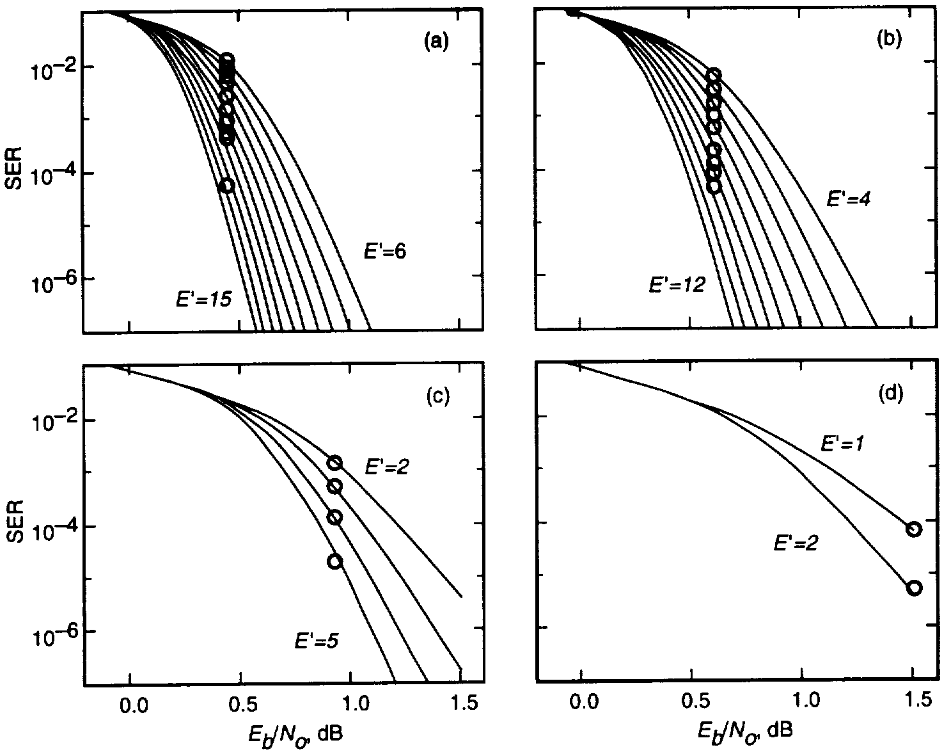


Fig. 6. Measured SER compared to ideal interleaving SER curves (case A): SER in (a) stage 1, (b) stage 2, (c) stage 3, and (d) stage 4.

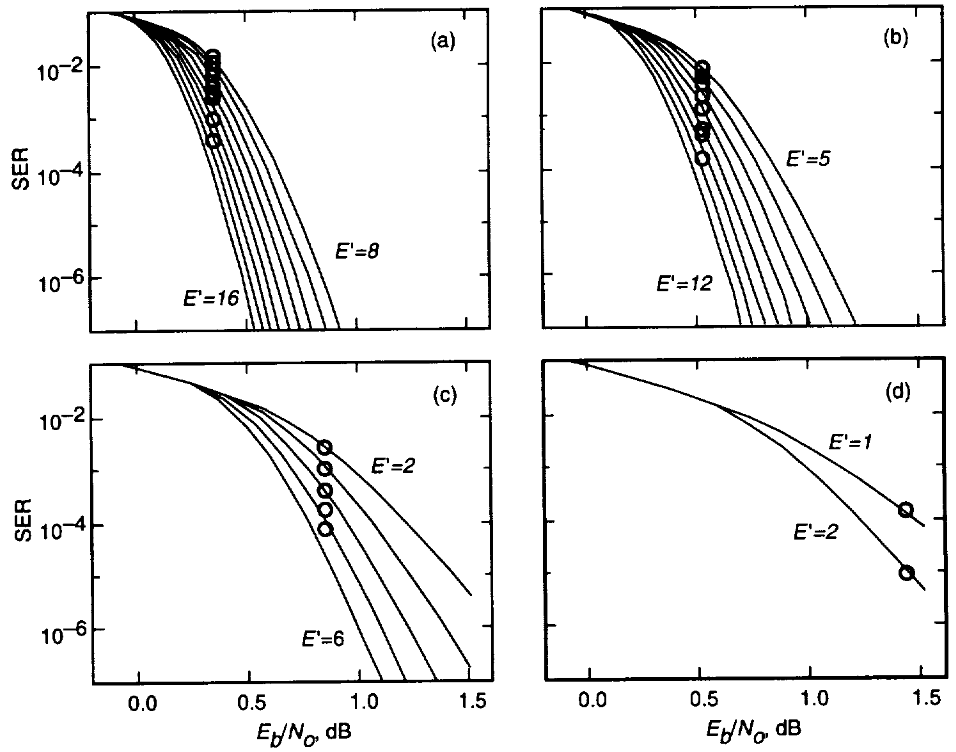


Fig. 7. Measured SER compared to ideal interleaving SER curves (case B): SER in (a) stage 1, (b) stage 2, (c) stage 3, and (d) stage 4.

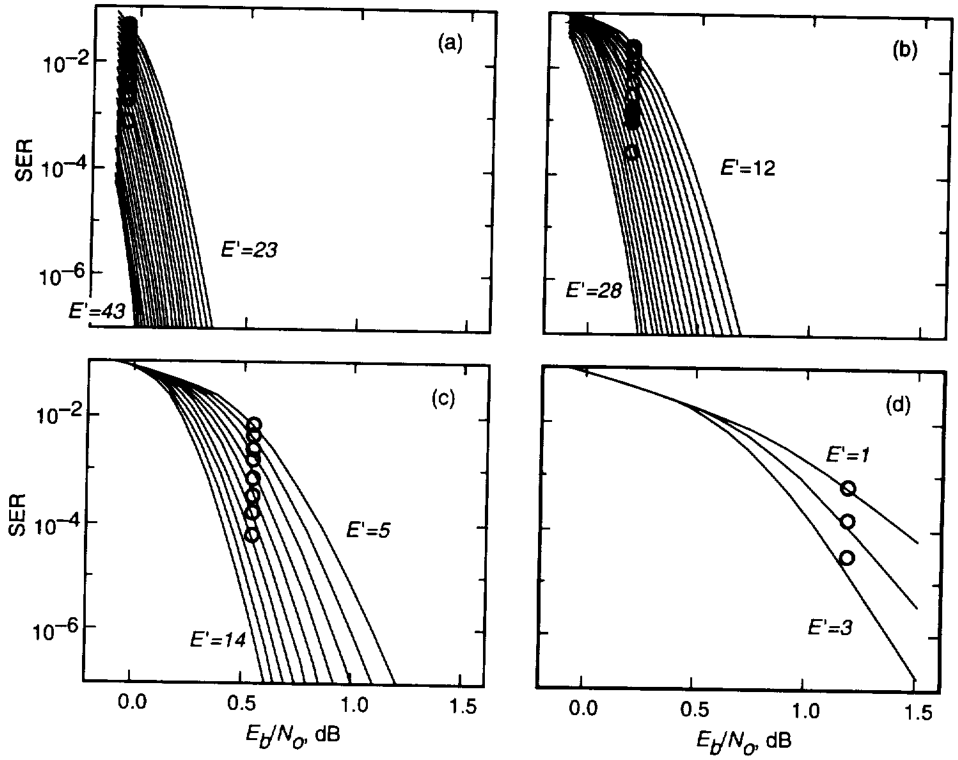


Fig. 8. Measured SER compared to ideal interleaving SER curves (case C): SER in (a) stage 1, (b) stage 2, (c) stage 3, and (d) stage 4.

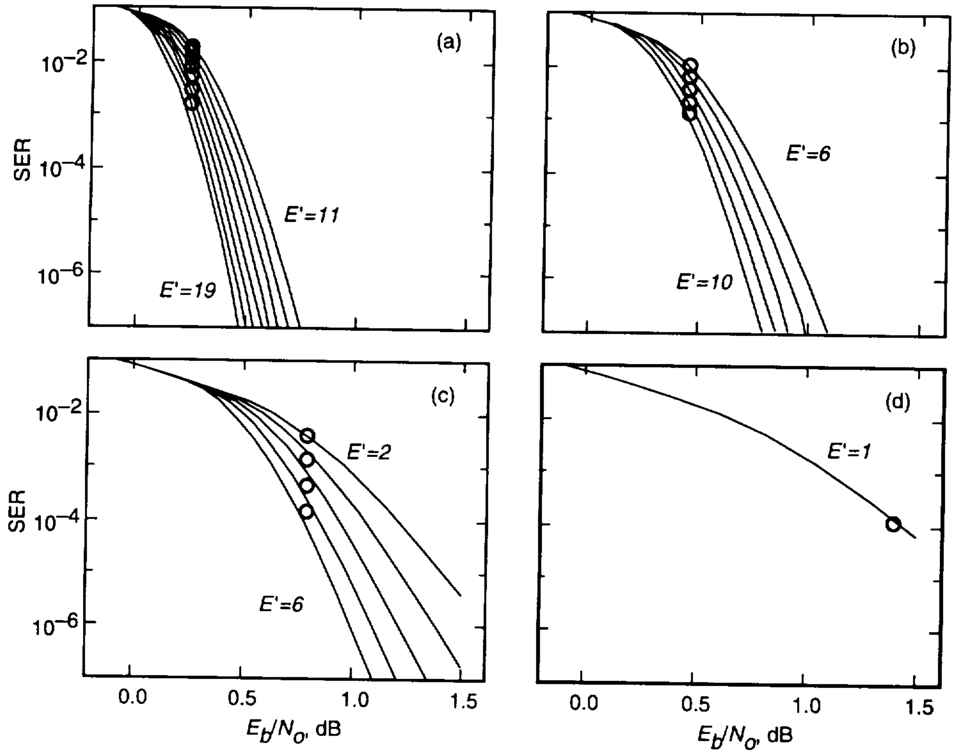


Fig. 9. Measured SER compared to ideal interleaving SER curves (case D): SER in (a) stage 1, (b) stage 2, (c) stage 3, and (d) stage 4.

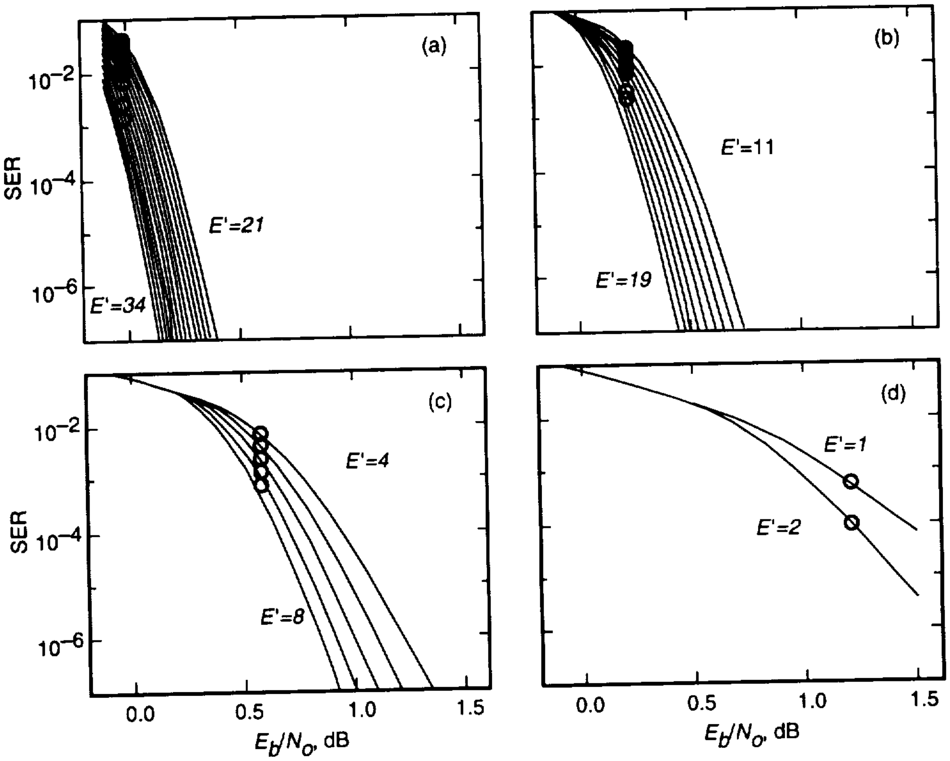


Fig. 10. Measured SER compared to ideal interleaving SER curves (case E): SER in (a) stage 1, (b) stage 2, (c) stage 3, and (d) stage 4.

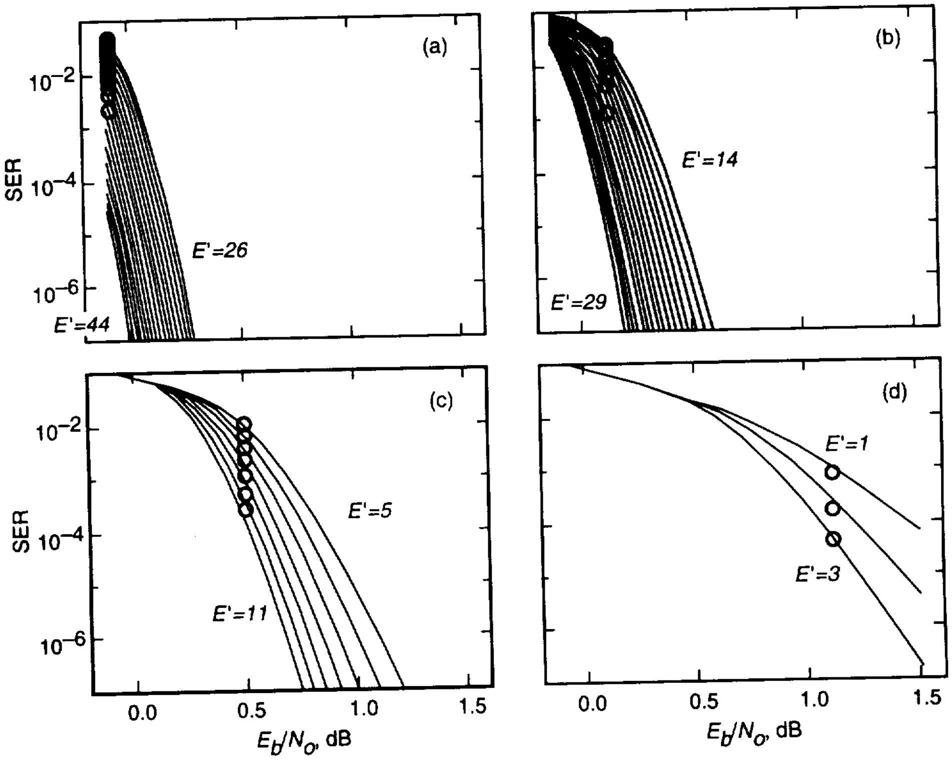


Fig. 11. Measured SER compared to ideal interleaving SER curves (case F): SER in (a) stage 1, (b) stage 2, (c) stage 3, and (d) stage 4.

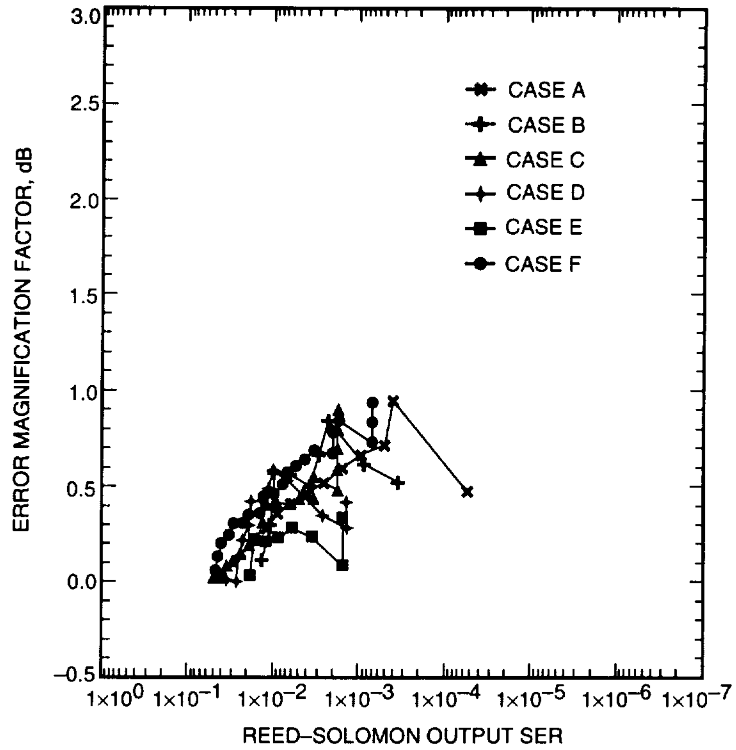


Fig. 12. Measured first-stage error magnification factors.

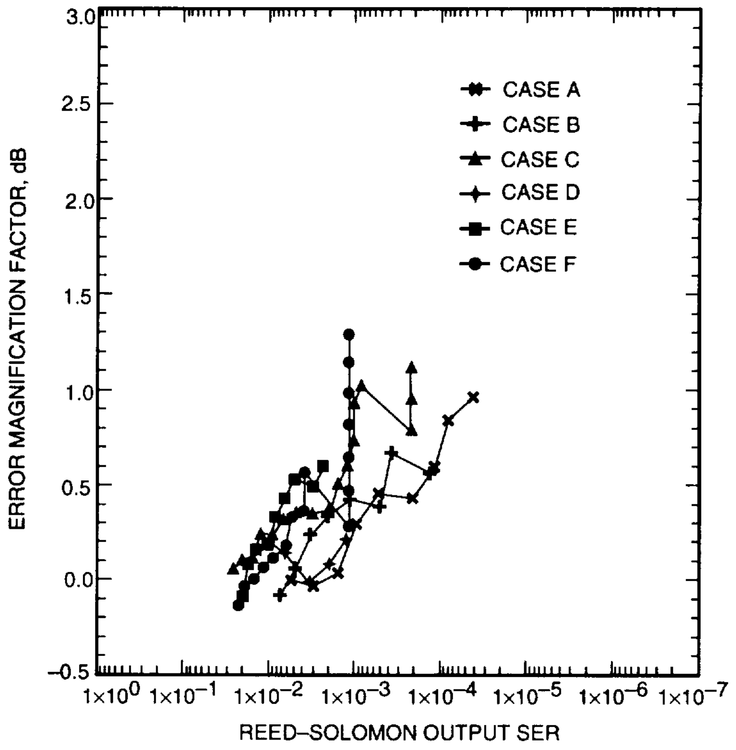


Fig. 13. Measured second-stage error magnification factors.

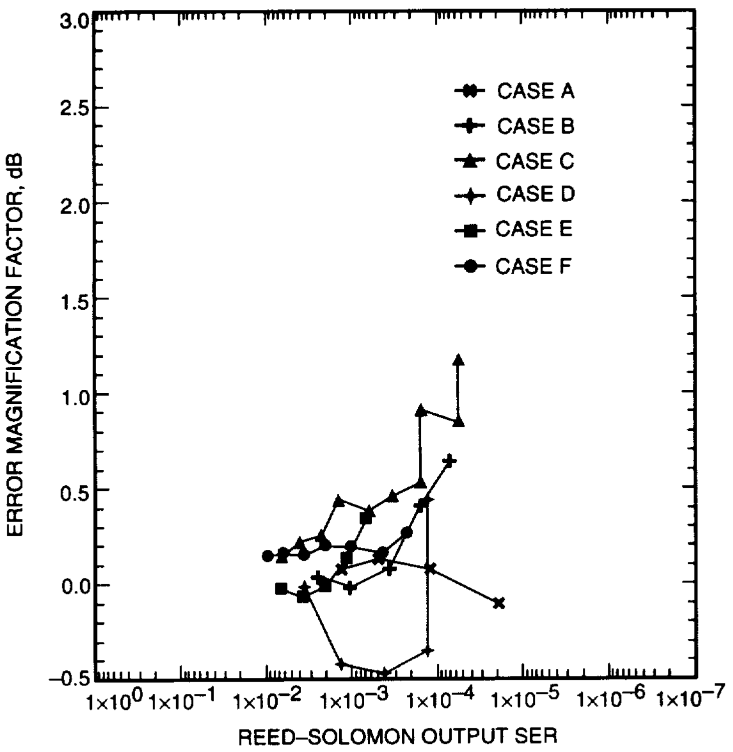


Fig. 14. Measured third-stage error magnification factors.

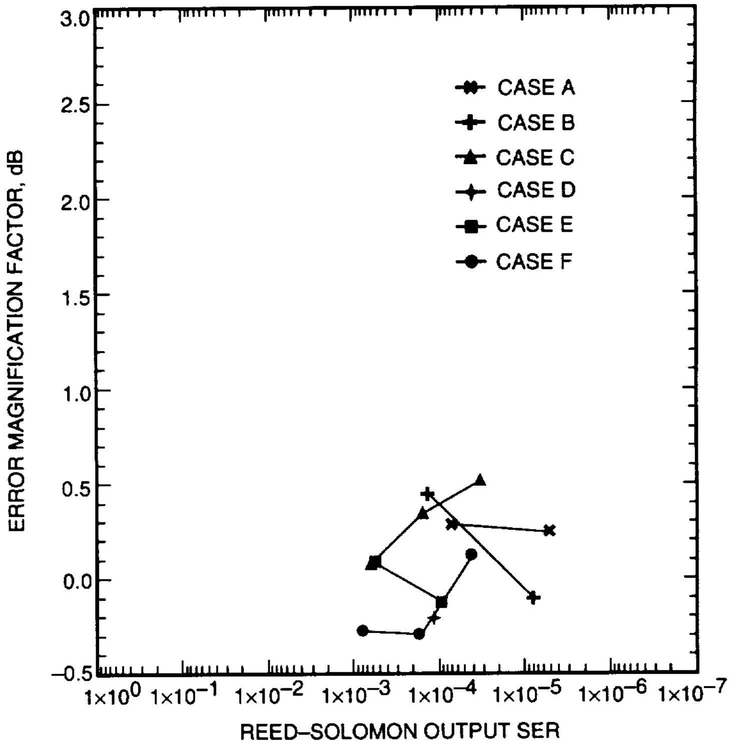


Fig. 15. Measured fourth-stage error magnification factors.

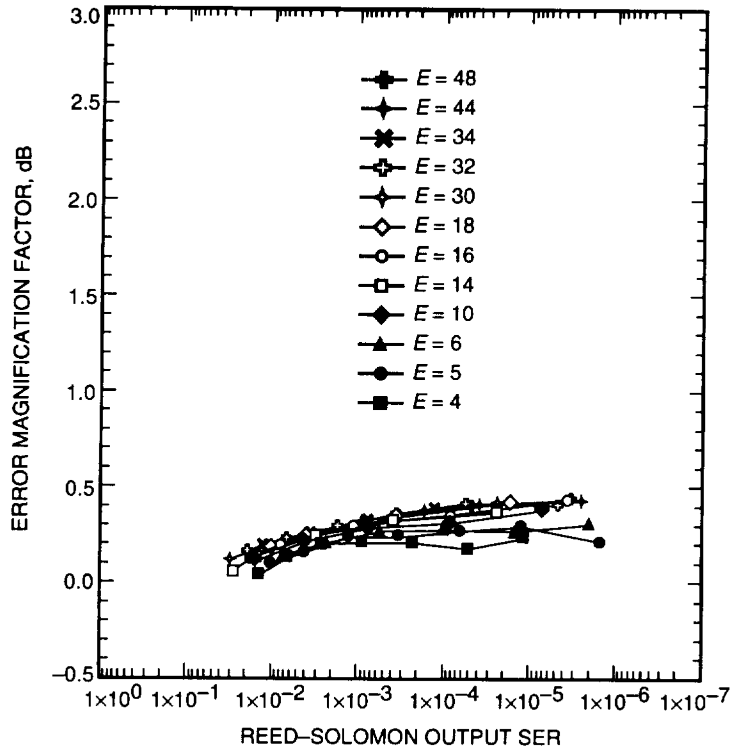


Fig. 16. Reference error magnification factors for depth-8 interleaving.

It should be noted that the reference curves show error magnification factors computed by varying the channel SNR but keeping the correctability fixed, while the test data curves show magnification factors computed by varying the hypothetical correctability at a fixed channel SNR for each of cases A through F. Thus, it is not legitimate to subtract the curves point by point. However, our extrapolation procedure still provides a good estimate, because the reference curves all cluster together and approach a very robust extrapolated value almost independent of correctability.

The final adjustment required to obtain the RS loss increment is to convert the error magnification factors into equivalent SNR losses according to the calibration curves in Fig. 5. This means dividing the net magnification factor (relative to the depth-8 AWGN reference) by 8 for stages with hypothetical correctabilities $E' \geq 15$, and by approximately 6 or 7 for stages with lower correctabilities. This results in estimated RS loss increments up to approximately 0.2 dB for the first three stages except for case D, and no more than approximately 0.1 dB for the fourth stage of all cases and for all stages of case D. However, it must be emphasized that these estimates are based on extrapolating some very ragged error magnification factor test data in Figs. 12 through 15 over three or four orders of magnitude in RS output SER, and the estimates might easily be off by 1 dB or so in magnification factor units, which is equivalent to a little more than 0.1 dB in SNR loss.

4. Discussion of the Extrapolation Method. If the error magnification factor extrapolations in Figs. 12 through 15 seem somewhat mysterious, here is a brief explanation in terms of the more understandable error rate measurements shown in Figs. 6 through 11. In the latter figures, the small circles represent hypothetical RS output SERs for codes with smaller correctabilities E' than the actual code's correctability. The desired but unmeasurable test datum is the small circle that would correspond to $E' = E_i$, where E_i is the actual codeword correctability in the ith decoding stage. We must try to estimate where this unmeasurable small circle might lie. The simplest method is to assume that it falls on the corresponding reference curve for the same value of correctability. The trail of small circles would be extended downward in a straight vertical line at the constant value of effective bit SNR shown in the figures, until the assumed reference curve is intersected. The reference curve may be the infinitely interleaved performance curve shown in the figures or the corresponding depth-8 performance curve (not shown). This method of extrapolating to an assumed reference curve precludes the detection of any deviation or loss relative to the reference.

We notice from Figs. 6 through 11 that the small circles begin to deviate more and more from their corresponding reference curves as the hypothetical correctability E' increases and the RS output SER gets smaller. The error magnification factors in Figs. 12 through 15 quantify this increasing deviation from the reference. By extrapolating along the trend of increasing magnification factors measurable for small values of E', we can obtain an estimate of how far above the reference curve the small circle would be either at the true value of correctability or at the value that yields an error rate around 10^{-7}. This produces an estimate of the loss relative to the assumed reference.

IV. Summary of Test Results

The measured VD and RS loss increments for cases A through F are reported in Table 2. The VD loss increments are mostly between 0.1 and 0.2 dB, except for case D, which has negligibly small VD loss increments. The RS loss increments estimated in the previous section range up to approximately 0.2 dB, not allowing for at least 0.1 dB possible error in extrapolating the data to the 10^{-7} SER level. The composite FCD loss increment, obtained roughly as the sum of the VD and RS loss increments for the most affected stage, is estimated to be approximately 0.3 to 0.4 dB for all cases except case D, for which the FCD loss increment is less than 0.1 dB. Again, the estimates of the composite FCD loss increment do not include the numerical uncertainties (positive or negative) in extrapolating the RS loss increments to the 10^{-7} SER level.

Cases C and F produced an effective operating point a few tenths of a dB lower than the design threshold required for a 10^{-7} error rate. As a result, the FCD failed to decode a few frames. In the previous section, we described our test procedure as if these undecodable frames never existed, and the numerical results in Table 2 are based on ignoring these frames. In the next section, adjustments are made to approximately account for the bias introduced by ignoring the undecodable frames. These adjustments add less than 0.1 dB to the composite FCD loss increment for cases C and F only.

V. Discussion of Test Results

A. Statistical Confidence in the Numerical Results

One of our concerns about the test results is that the measured error magnification factors for these tests jump around wildly, and, thus, it is much harder to confidently extrapolate the RS loss increments due to the nonideal BTD than the corresponding losses reported in [1] due to depth-8 interleaving. Some of this erratic behavior is purely statistical, as a result of the small number of frames tested. If error bars were shown in Figs. 12 through 15, they would lengthen dramatically proceeding from left to right as the SER decreases to the point of undetectability for the small number of frames tested.

Figure 17 illustrates how the statistical fluctuations can be smoothed out for the first stage by having eight times as much data. In Figs. 12 through 15, the only data used for the calculation of the magnification factors in a given stage came from the specific codeword(s) decoded during that stage. For example, the data for the first stage are from the observed RS symbol corrections in the single codeword with the highest correctability, $E = 47$. It would be equally valid to perform hypothetical first-stage decodings with correctabilities $E' \leq 47$ on all eight codewords, if it can be assumed that the correct symbols are eventually known in all eight codewords by the end of the fourth decoding stage. The data obtained from all eight codewords are plotted in Fig. 17. Notice the improved smoothness of the curves relative to those

Table 2. FCD test results.

Case	Number of decoded frames	BSNR at BTD output, dB	Stage	BSNR at VD output, dB	VD loss increment, dB	RS loss increment, dB	Composite FCD loss increment, dB
A	1184	0.53	1	0.43	0.10	0.10	0.3
			2	0.42	0.11	0.15	
			3	0.43	0.10	<0.1	
			4	0.40	0.13	<0.1	
B	372	0.48	1	0.34	0.14	0.15	0.3
			2	0.32	0.16	0.15	
			3	0.32	0.16	0.10	
			4	0.27	0.21	<0.1	
C	491 (3 frames failed)	0.01	1	−0.08	0.09	0.20	0.4
			2	−0.10	0.11	0.20	
			3	−0.14	0.15	0.20	
			4	−0.17	0.18	0.10	
D	100	0.23	1	0.21	0.02	<0.1	<0.1
			2	0.24	−0.01	<0.1	
			3	0.24	−0.01	<0.1	
			4	0.22	0.01	<0.1	
E	100	0.04	1	−0.05	0.09	0.10	0.3
			2	−0.06	0.10	0.20	
			3	−0.08	0.12	0.20	
			4	−0.13	0.17	<0.1	
F	99 (1 frame failed)	−0.04	1	−0.13	0.09	0.20	0.4
			2	−0.16	0.12	0.25	
			3	−0.20	0.16	0.10	
			4	−0.29	0.25	0.10	

in Fig. 12. This procedure cannot be repeated for stages two through four, because the output error characteristics from the Viterbi *redecoder* affect each codeword differently, depending on the placement of the codeword relative to codewords decoded in previous stages.

In Fig. 15, we observed a dearth of data for making extrapolations of fourth-stage error magnification factors. This is not a problem that can be cured by testing just a few more frames. When the decoder is operating at the design threshold and above, each fourth-stage codeword will report very few symbol corrections e. Values of e close to the code's correction capability, such as $e = 5$ or $e = 4$, will be highly unlikely. Thus, with reasonable amounts of test data, there may only be two or three distinct values of the hypothetical correctability E' for which any test results exist. It is difficult to justify extrapolations of the error magnification factor curves based on only two or three points. Fortunately, as pointed out earlier, the fourth-stage magnification factors seem to be somewhat more benign than those of the earlier stages, and an accurate extrapolation is not necessary if the overall FCD loss increment is dominated by the error magnifications in earlier stages.

Better estimates of the fourth-stage error magnification factor might be obtained by modifying the test procedure to more closely resemble the analysis in [1], fixing a particular value of correctability (e.g., $E' = 2$ or $E' = 3$) and running a series of tests with the same loop parameters but different SSNR values. In fact, because of the empirically observed near universality of the error magnification factor curves for similar values of E', testing different SSNR values is an appropriate way to merge more data into the magnification factor estimates for any stage.

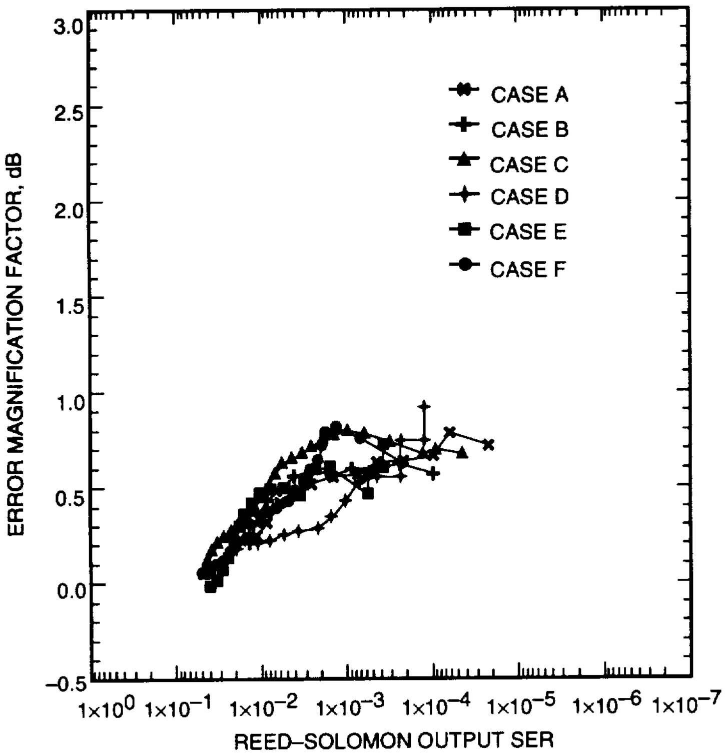


Fig. 17. First-stage error magnification factors using eight times as much data.

B. Tests Conducted Below Design Threshold

It was pointed out earlier that a few frames failed to decode for test cases C and F. As expected, these failures always happened on the first-stage codeword, because the effective operating point (accounting for all loss increments) was below the design threshold. The effective operating point for case E was also below threshold, but by luck no decoding failures occurred over the small sample size of 100 frames. In all of the analyses up to now, the data from undecodable frames have been completely ignored. There was no report from the FCD on what the correct symbols were, and exact SERs and error magnification factors cannot be computed. However, ignoring these frames biases the results optimistically. Figure 18 shows first-stage magnification factors computed for cases C and F by assuming that there were exactly 48 errors in the undecodable codewords. The magnification factors are increased relative to those reported in Fig. 12, reaching well above 1 dB at SERs near 10^{-3}. Extrapolated magnification factors of 3 dB or higher at a 10^{-7} SER are certainly imaginable based on these adjusted data.

We have consistently made the assumption that the codewords decoded by the FCD represent the truth data. This is a valid assumption as long as the test procedure is being applied at design threshold and above. Below the design threshold, there is the possibility of encountering undecodable frames, as in test cases C and F. One might also worry about incorrectly decoded codewords in the fourth stage, where the small correction capability, $E = 5$, implies that there is a non-negligible probability of making a decoding error. However, this should not happen unless the loss mechanism somehow concentrates its deleterious effects on the fourth decoding stage and, thus, dramatically disturbs the design balance point. In the usual circumstances, losses that drop the effective operating point below threshold will show up as detected decoding failures on the first stage, because first-stage decoding performance declines most sharply as the SNR drops below threshold.

It should be noted that the complicated test procedure described in this article is primarily intended for analyzing FCD performance when codeword errors are rare, i.e., at threshold or above. Below thresh-

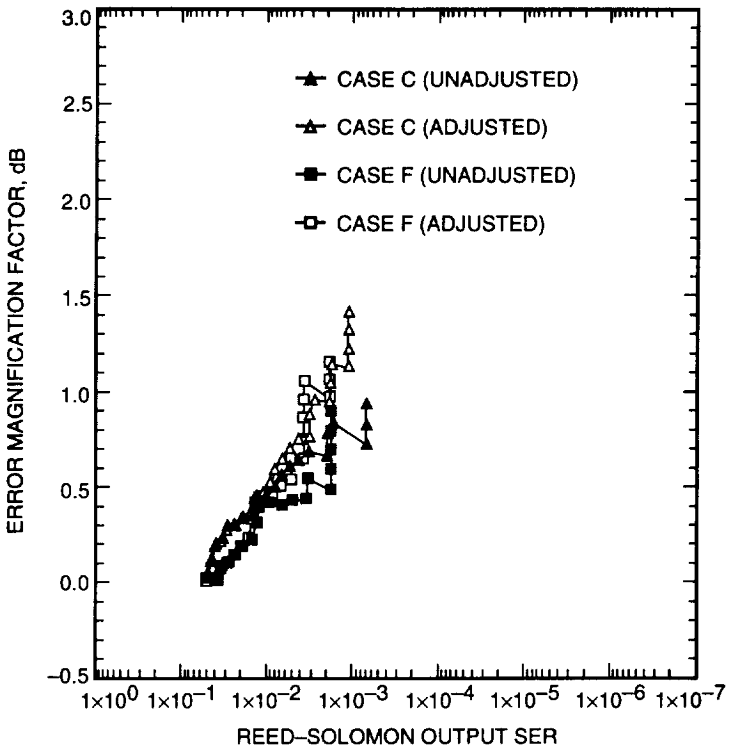


Fig. 18. First-stage error magnification factors adjusted for undecoded frames.

old, the FCD's performance deteriorates very rapidly, and there are sufficient codeword errors to make simple error counting tests reliable. Thus, the extra complications needed to account for undecodable or incorrectly decoded frames should not pose a problem in practice: at operating points where decoding failures are likely, a simpler test procedure should be substituted for the one described here.

Here is an illustration of how different the conclusions are for a test conducted below threshold. For case C, the observed first-stage codeword failure rate was 0.006, based on 3 failures out of 491 first-stage codewords. When a first-stage codeword fails, about 20 percent of the symbols are erroneous, so the RS output SER is around 10^{-3}. Due to the small number of observed undecodable words, this estimate is not highly accurate, but it still gives a ballpark number. From Table 2, the effective BSNR at the Viterbi decoder output is −0.08 dB, which is under the design threshold of 0.00 dB quoted in [1] for achieving a 10^{-7} SER. From Figs. 3 and 4, it is seen that four orders of magnitude in RS output SER are equivalent to about 0.17 dB of BSNR at the high slope of the first-stage code's performance curve. Therefore, the first-stage RS loss increment for case C is slightly less than 0.1 dB rather than the 0.2 dB quoted in Table 2. This apparent contradiction is resolved as follows. The calculations in this paragraph measure the RS loss increment at the actual test conditions for case C, i.e., at an operating point producing an SER around 10^{-3}. The calculations reported in Table 2 estimate how big the losses would be if the operating point had been adjusted to produce an SER around 10^{-7}. The calculations for 10^{-3} SER can be directly verified by reference to the error magnification curves in Figs. 12 and 16 without any need for extrapolation. The observed and reference error magnification factors at 10^{-3} SER are about 0.85 dB and 0.3 dB, respectively, translating into a net SNR loss of about 0.07 dB. This correlates well with the calculation based on just three codeword failures. The additional 0.1 dB of RS loss increment predicted in Table 2 for 10^{-7} SER results from extrapolating the magnification factors in Fig. 12, along their observed rate of increase, all the way to the 10^{-7} SER level. The increasing error magnification factors correspond to a slight flattening of the high-slope first-stage performance curve as compared to the reference ideal curves in Fig. 4. This flattening causes the RS loss increment to increase as the RS output SER is made smaller.

C. Combining the Results From All Four Stages

We have described a procedure for evaluating stage-by-stage FCD loss increments as the sum of stage-by-stage VD and RS loss increments, but we have not emphasized how to obtain the composite FCD loss increment taking into account all four stages. If the four stage-by-stage FCD loss increments are identical, then the composite loss increment is the same. The composite loss increment is no worse than the worst of the stage-by-stage loss increments, and it approaches this limit when one stage dominates the FCD's performance. Between these two extremes, it would be proper to calculate an average of the stage-by-stage loss increments by explicitly considering the effect of each stage on the overall SER or BER of the FCD. However, this complicated analysis would only improve the estimate over a narrow band of loss combinations, because the performance of the FCD passes very quickly into dominance by the performance of its weakest stage whenever its design balance point is disturbed.

We have also glossed over the precise error rate at which the FCD loss increment is evaluated. While Galileo has a very specific overall BER requirement of 1×10^{-7}, we have spoken more vaguely of reaching on each stage a target error rate on the order of 10^{-7}, and the error rates we have aimed at this target are 8-bit RS SERs rather than BERs. Given the several orders of magnitude range over which the error magnification factors must be extrapolated, there is no need to be more precise in specifying the exact target, since the overall BER is about half the overall 8-bit SER for a long-constraint convolutional code, and about one to two times the stage-by-stage 8-bit SERs.

Since our test procedure focuses on estimating individual stage-by-stage losses, it is also applicable to testing a simple one-stage concatenated decoder without feedback. The RS loss increments seen in our tests are qualitatively, if not quantitatively, similar to the RS loss increments that would be measured if the (14, 1/4) convolutional code were concatenated with the standard 16-error-correcting RS outer code and asked to perform at a 10^{-7} SER level with nonideal input from the BTD.

VI. Conclusion

This article presents a test procedure that tests the performance of the FCD when the resulting BER is very low (10^{-7}) and cannot be measured directly through simulations in a reasonable amount of time. Using this test procedure, we have tested the FCD taking the input from the BTD which contains multiplicative colored non-Gaussian noises. The preliminary test results show that there are about 0.3- to 0.4-dB loss increments in the FCD when the loop SNRs are lower than 20 dB as compared to analytical results assuming AWGN. In one test case, where we had the subcarrier-loop SNR around 28 dB, the loss increment in the FCD was less than 0.1 dB.

The numerical test results reported in this article are rough estimates due to the small amount of test data and test cases that were run. However, the test procedure described herein should be used as a template for conducting more extensive performance tests on the FCD in the future. This template provides an organized robust methodology for extrapolating small amounts of test data to give reasonable estimates of FCD loss increments at unmeasurable minuscule error rates.

Reference

[1] S. Dolinar and M. Belongie,"Enhanced Decoding for the Galileo Low-Gain Antenna Mission: Viterbi Redecoding With Four Decoding Stages," *The Telecommunications and Data Acquisition Progress Report 42-121, January-March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 96–109, May 15, 1995.

Appendix

Step-by-Step Test Procedure

Follow this procedure to test the FCD at operating points that produce an output BER of around 10^{-7}

(1) Choose a set of loop parameters for testing. Based on the preliminary results in Tables 1 and 2 or on more extensive similar test results, guess a value of SSNR that will produce an output BER around 10^{-7}. Generate a number of frames of encoded data, modulate a carrier and a subcarrier with the data, add channel noise, and feed the resulting test signal through the BTD. Pass the output of the BTD through the FCD and note the results of the decoding.

(2) Estimate the SSNR at the output of the BTD using the split symbol estimator $\widehat{SSNR}$. Compute the BTD loss increment, in dB, as $\Delta L_{BTD} = 10 \log_{10} SSNR/\widehat{SSNR}$.

(*) Repeat steps 3 through 10 for the output from each individual decoding stage, $i = 1, 2, 3, 4$. For these steps, an ith stage codeword is defined as a codeword with correctability E_i, where $E_1 = 47$, $E_2 = 30$, $E_3 = 15$, and $E_4 = 5$.

(3) Observe the number of corrected symbols e_i in each ith-stage codeword in each frame. If any ith-stage codeword is undecodable, record this event as $e_i = E_i + 1$, but be aware that, if this event occurs frequently, the test procedure is being used outside its intended range.

(4) Compute the VD output SER, $SER_{VD}(i)$, for ith-stage codewords as the sum of all the observed values of e_i divided by 255 times the total number of ith-stage codewords.

(5) Look up the measured value of $SER_{VD}(i)$ on the Viterbi decoder performance curve for the ith stage for Galileo's (14,1/4) code (Fig. 3) and interpolate to find the corresponding value of BSNR. Compute the VD loss increment, in dB, as $\Delta L_{VD}(i) = 10 \log_{10}(4 \times \widehat{SSNR}/BSNR)$.

(6) Compute output SERs, $SER_{RS}(i, E')$, for RS codes with hypothetical correctabilities E' greater than $255 \times SER_{VD}(i)$ and strictly less than the maximum value of e_i observed in step 3: $SER_{RS}(i, E')$ is computed as the sum of the observed values of e_i for only those ith-stage codewords with $e_i > E'$, divided by 255 times the total number of ith-stage codewords.

(7) Compute a lookup table of ideal RS output SERs, $SER^*_{RS}(i, E)$, for RS codes with varying correctabilities E facing independent symbol errors occurring with rate $SER_{VD}(i)$. This table generates the ideal RS performance curves shown in Fig. 4. Be sure that the lookup table encompasses sufficient values of E for the interpolation in the next step.

(8) For each value of hypothetical correctability E' determined in step 6, interpolate using the lookup table in step 7 to find an equivalent ideal correctability E^* such that $SER^*_{RS}(i, E^*) = SER_{RS}(i, E')$. Be sure to perform this interpolation based on logarithms of error rates, e.g., for "linear" interpolation,

$$E^* = E_0^* + \frac{\log[SER^*_{RS}(i, E_0^*)/SER_{RS}(i, E')]}{\log[SER^*_{RS}(i, E_0^*)/SER^*_{RS}(i, E_0^* + 1)]}$$

where E_0^* is the largest value of E for which $SER_{RS}^*(i,E) \geq SER_{RS}(i,E')$. For each value of E', compute a corresponding ith-stage error magnification factor, measured in dB, as $MF(i,E') = 10\log_{10} E'/E^*$.

(9) Plot $MF(i,E')$ versus $SER_{RS}(i,E')$, varying the parameter E', to obtain a curve like those in Figs. 12 through 15. Use good engineering judgment to extrapolate these curves to the desired RS output SER level around 10^{-7}.

(10) Subtract 0.5 dB, or a little less, from the extrapolated error magnification factor obtained in step 9. Then divide by 8, or a little less, to get a corresponding SNR loss. The result is the RS loss increment, $\Delta L_{RS}(i)$, for stage i, compared to the reference performance derived in [1] for depth-8 interleaving and AWGN. The values to subtract or divide by depend on the values E' contributing to the error magnification factor curve: Subtract 0.5 dB and divide by 8 when E' is approximately 15 or greater, and reduce these calibration values slightly to 0.4 or 0.3 dB and 7 or 6 when E' is smaller. Since different values of E' contribute to the same error magnification factor curve, the exact calibration requires an exercise of good judgment.

(11) The FCD loss increment for the ith decoding stage is the sum of the ith-stage VD and RS loss increments (measured in dB). The composite FCD loss increment for all decoding stages is approximately the largest of the stage-by-stage FCD loss increments.

TDA Progress Report 42-121

May 15, 1995

Performance of Residual Carrier Array-Feed Combining in Correlated Noise

H. H. Tan
Telecommunications Systems Section
and
University of California, Irvine

An array feed combining system for the recovery of signal-to-noise ratio (SNR) loss due to antenna reflector deformation has been implemented and is currently being evaluated on the Jet Propulsion Laboratory's 34-meter DSS-13 antenna. In this system, the defocused signal field captured by a focal plane array feed is recovered using real-time signal-processing and signal-combining techniques. The current signal-processing and signal-combining algorithms are optimum under the assumption that the white Gaussian noise processes in the received signals from different array elements are mutually uncorrelated. Experimental data at DSS 13 indicate that these noise processes are indeed mutually correlated. The main result of this article is an analytical derivation of the actual SNR performance of the current suboptimal signal-combining algorithm in this correlated-noise environment. The analysis here shows that the combined signal SNR can either be improved or degraded depending on the relation between the array signal and noise correlation coefficient phases. Further performance improvement will require the development of signal-combining methods that take into account the correlated noises.

I. Introduction

Operation of deep-space communication networks at higher carrier frequencies has the advantage of greater antenna gains as well as increased bandwidths for enhancing telemetry capabilities. However, the use of higher frequencies also has certain disadvantages. These include more stringent antenna pointing requirements and larger receiving antenna signal-to-noise ratio (SNR) losses due to mechanical deformations of large reflector surfaces. These SNR losses become more significant at higher frequencies when carrier wavelengths become smaller than the mechanical imperfections of the reflector. This is the case in the Jet Propulsion Laboratory's Deep Space Network plan to employ Ka-band (32-GHz) communications using 34- and 70-meter receiving antennas.

An array feed combining system for the recovery of the SNR loss due to antenna reflector deformation has been proposed and analyzed in [1]. In this system, a focal plane feed array is used to collect the defocused signal fields. All the signal power captured by the feed array is then recovered using real-time signal-processing and signal-combining techniques. In phase and quadrature, baseband signal samples are obtained from the downconverted received signal of each of the array feed elements and then are recombined after application of combiner weights. The optimum combiner weights that maximize the combined signal SNR were derived in [1] under the assumption that the white Gaussian noise processes in the received signals from different array elements are mutually uncorrelated. These optimum weights depend on unknown signal and noise parameters that need to be estimated. The work in [1] proposed to estimate the optimum weights from the observed residual carrier received signal samples using a maximum likelihood (ML) estimation of these unknown parameters. The actual combined signal SNR in this uncorrelated-noise environment was also derived in [1] when the estimated weights were used in place of the optimum weight coefficients.

The array feed combining system is currently being evaluated at the JPL DSS-13 34-meter antenna. Although the work in [1] assumed mutually uncorrelated-noise processes, experimental data [2] indicate that the white noise processes in the received signals from different feed elements are indeed correlated, with correlation coefficients of the order of 0.01 under clear-sky conditions. Since the noise in each of the array feed element signals consists of receiver white noise plus noise due to background radiation, this small correlation is conjectured to be caused by near-field atmospheric background noise. Although the observed correlation in [2] is quite small in the current array feed combining system, future planned improvements in the the receiver noise temperature could magnify the effect of atmospheric background noise and result in considerably higher amounts of correlation. Thus, it is important to determine the performance of the signal-combining system proposed in [1] when the white Gaussian noise processes in the signals from different array elements are mutually correlated. That is the objective of this article, which provides an exact analysis of the combined signal SNR performance in this correlated-noise environment.

The performance analysis here considers only the signal combining algorithm proposed in [1], which was designed to operate in the environment where the white Gaussian noise processes in the signals from different array elements are mutually uncorrelated. The effect of the correlation is twofold. First, the optimum combining weights developed in [1] are no longer optimal in this correlated-noise environment. The other effect of this correlation is on the resulting combined-signal SNR performance. The analysis here shows that the combined-signal SNR can be either improved or degraded depending on the relation between the array signal and noise correlation coefficient phases. Further performance improvement will require effective combining systems that take into account the correlations between the array feed element noise processes. Our work on this problem is still in progress.

II. Array Feed Signals and Combining Algorithm

Consider a K-element array and the NASA Deep Space Network standard residual carrier modulation with a binary phase shift key (BPSK)-modulated square-wave subcarrier [3]. The received signal from each array element is downconverted to baseband and sampled. The combining system proposed in [1] uses only the residual carrier portion of the received signal spectrum to estimate the unknown parameters in the combiner weights. The full spectrum modulated signals from the array elements, which contain both the modulated sidebands as well as the residual carrier spectrum, are subsequently combined. In this system [1], the higher-bandwidth primitive baseband signal samples are low-pass filtered by averaging successive blocks of M_B samples to yield a full-spectrum signal stream B for each array element. Additive white Gaussian noise is assumed to be present in the primitive baseband signal sequences from each of the array elements. Let

$$y_k(i_B) = V_k[\cos\delta + j\, s(i_B)\sin\delta] + n_k(i_B), \quad i_B = 1, 2, \cdots \tag{1}$$

denote the stream B signal samples from the kth array element. The complex signal parameters

$$V_k = |V_k|\, e^{j\,\theta_k} \tag{2}$$

represent the unknown signal amplitude and phase parameters induced by the antenna reflector deformation. Moreover, δ is the modulation index, $s(i_B) = \pm 1$ is the transmitted data, and $\{n_k(i_B)\}$ is the zero-mean white Gaussian noise corruption in the stream B signal samples from the kth array element. The primitive baseband signal samples are also more narrowly low-pass filtered by averaging successive blocks of M_A samples to yield a residual carrier signal stream A for each array element. Clearly $M_A > M_B$, and $\eta = M_A/M_B$ is the ratio of the bandwidth of stream B to stream A. Let

$$u_k(i_A) = V_k \cos\delta + m_k(i_A), \quad i_A = 1, 2, \cdots \tag{3}$$

denote the stream A signal samples from the kth array element. Here $\{m_k(i_A)\}$ is the zero-mean white Gaussian noise corruption in the stream A signal samples from the kth array element.

Let $\underline{A}^T$ and $\underline{A}^\dagger$ denote the transpose and complex conjugate transpose of the matrix $\underline{A}$, respectively. The white noise sequences corresponding to different array elements are assumed to be correlated. To specify these correlations, consider

$$\begin{aligned} \underline{n}(i_B) &= (n_1(i_B), \cdots, n_K(i_B))^T \\ \underline{m}(i_A) &= (m_1(i_A), \cdots, m_K(i_A))^T \end{aligned}$$

Then $\{\underline{n}(i_B)\}$ and $\{\underline{m}(i_A)\}$ are each sequences of independent identically distributed (i.i.d.) zero-mean complex Gaussian random vectors of dimension K. The respective covariance matrices

$$\begin{aligned} \underline{R}_B &= \{r_{Bkj}\} &= \mathrm{E}\left[\underline{n}(i_B)\underline{n}(i_B)^\dagger\right] \\ \underline{R}_A &= \{r_{Akj}\} &= \mathrm{E}\left[\underline{m}(i_A)\underline{m}(i_A)^\dagger\right] \end{aligned}$$

of $\underline{n}(i_B)$ and $\underline{m}(i_A)$ then specify the mutual correlations between the white noises in the signal streams from different array elements. For example, r_{Bkj} is the correlation between the noise variables $n_k(i_B)$ and $n_j(i_B)$ in the stream B signals from the kth and jth array elements, respectively. Moreover, define

$$\rho_{Bkj} = \frac{r_{Bkj}}{\sqrt{r_{Bkk} r_{Bjj}}} = |\rho_{Bkj}|\, \mathrm{e}^{j\varphi_{Bkj}} \tag{4}$$

to be the correlation coefficient between the noise samples $n_k(i_B)$ and $n_j(i_B)$. We shall assume as in [1] that the complex Gaussian noise samples $n_k(i_B)$ and $m_k(i_A)$ each has statistically independent real and imaginary parts of equal variance. This assumption is not required for the following analysis, but is made to maintain consistency with the results reported in [1]. So, $2\sigma_{Bk}^2 = r_{Bkk}$ and $2\sigma_{Ak}^2 = r_{Akk}$ are the respective variances of $n_k(i_B)$ and $m_k(i_A)$, where σ_{Bk}^2 and σ_{Ak}^2 are the respective variances of the real or imaginary parts. Because of the different averaging rates in streams A and B on the primitive baseband signals, it follows that $\underline{R}_B = \eta \underline{R}_A$. Finally, these different averaging rates also imply that $\underline{m}(i_A)$ is independent of $\underline{n}(i_B)$ provided that $i_A < i_B$ and the samples averaged to yield $\underline{m}(i_A)$ occurred prior to the samples averaged to yield $\underline{n}(i_B)$.

The complex combining weight coefficients w_k, $1 \le k \le K$, given by

$$w_k = \frac{V_k^*}{2\eta\sigma_{Ak}^2} = \frac{V_k^*}{2\sigma_{Bk}^2} \tag{5}$$

were shown in [1] to maximize the SNR of the combiner output in the uncorrelated-noise case, resulting in a maximum possible SNR equal to

$$\gamma = \sum_{k=1}^{K} \frac{|V_k|^2}{2\eta\sigma_{Ak}^2} = \sum_{k=1}^{K} \frac{|V_k|^2}{2\sigma_{Bk}^2} \tag{6}$$

That is, the optimum attainable SNR in the uncorrelated-noise case is equal to the sum of the SNRs of each of the feed array element outputs. The signal parameters V_k and the noise variances σ_{Ak}^2 are unknown parameters that need to be estimated to obtain an estimate of the optimum weight coefficients.

Assume that these unknown parameters are not random. The estimates for V_k and σ_{Ak}^2 developed in [1] are univariate sampling estimates based on the stream A residual carrier signal samples $\{u_k(i_A)\}$. In the uncorrelated-noise case, the stream A signal samples from different array elements are statistically independent. Hence, estimates of the weight coefficients w_k based on these estimates of V_k and σ_{Ak}^2 are also mutually independent. However, in the correlated-noise environment, these signal streams are no longer mutually independent and, hence, the resulting estimates for w_k are also no longer independent. In order to put this dependence in the proper perspective for the SNR performance analysis below, we will describe the estimation techniques developed in [1] in terms of multivariate sampling estimates based on the vector of stream A signal samples $\{\underline{u}(i_A)\}$ where

$$\underline{u}(i_A) = (u_1(i_A), \cdots, u_K(i_A))^T$$

Instead of estimating V_k directly, consider estimating $X_k = V_k \cos\delta$. Define

$$\underline{X} = (X_1, \cdots, X_K)^T$$

Then it follows from Eq. (3) that $\{\underline{u}(i_A)\}$ is an i.i.d. sequence of complex Gaussian random vectors with mean $\underline{X}$ and covariance matrix $\underline{R}_A$. It follows from multivariate statistical analysis [4,5] that, based on observations $\{\underline{u}(i_A - 1), \cdots, \underline{u}(i_A - L)\}$,

$$\underline{\hat{X}}(i_A) = \left(\hat{X}_1(i_A), \cdots, \hat{X}_K(i_A)\right)^T = \frac{1}{L} \sum_{l=i_A-L}^{i_A-1} \underline{u}(l) \tag{7}$$

is the ML sample mean estimate of $\underline{X}$ and

$$\underline{\hat{R}}_A(i_A) = \frac{1}{L-2} \sum_{l=i_A-L}^{i_A-1} \left[\underline{u}(l) - \underline{\hat{X}}(i_A)\right] \left[\underline{u}(l) - \underline{\hat{X}}(i_A)\right]^{\dagger} \tag{8}$$

is equal to $(L - 1)/(L-2)$ times the corresponding sample covariance estimate of $\underline{R}_A$. The approach in [1] uses $\hat{X}_k(i_A)$ as the estimate of X_k and consequently $\hat{V}_k(i_A) = \hat{X}_k(i_A)/\cos\delta$ as the estimate of V_k. Moreover, the kth diagonal element $2\hat{\sigma}_{Ak}^2(i_A)$ of $\underline{\hat{R}}_A(i_A)$ is used in [1] as the estimate of $2\sigma_{Ak}^2$, which is the kth diagonal element of $\underline{R}_A$. Finally, the estimate given by

$$\hat{w}_k(i_A) = \frac{\hat{V}_k^*(i_A)}{2\eta\hat{\sigma}_{Ak}^2(i_A)} = \frac{\hat{X}_k^*(i_A)}{2\eta \cos\delta\, \hat{\sigma}_{Ak}^2(i_A)} \tag{9}$$

was shown in [1] to be an unbiased estimate of the optimum combining weight coefficient w_k given by Eq. (5) in the uncorrelated-noise case. These weight coefficient estimates are used in a sliding window structure to produce the following combiner output sequence:

$$z(i_B) = \sum_{k=1}^{K} \hat{w}_k \left(\tilde{i}_A\right) \, y_k(i_B) \tag{10}$$

where $\tilde{i}_A$ is the largest integer less than i_B, so that the residual carrier signal samples $\{u_k(\tilde{i}_A - 1), \ldots, u_k(\tilde{i}_A - L)\}$ used for estimating $\hat{w}_k(\tilde{i}_A)$ occur before the full-spectrum signal sample $y_k(i_B)$.

III. SNR Performance Analysis

The objective is to determine the actual SNR of the combiner output in the correlated-noise environment. From Eqs. (1) and (10), the combiner output can be written as

$$z(i_B) = s_c(i_B) + n_c(i_B) \tag{11}$$

where

$$s_c(i_B) = \sum_{k=1}^{K} \hat{w}_k \left(\tilde{i}_A\right) \, V_k \, e^{j\, s(i_B)\, \delta} \tag{12}$$

and

$$n_c(i_B) = \sum_{k=1}^{K} \hat{w}_k \left(\tilde{i}_A\right) \, n_k(i_B) \tag{13}$$

are the signal and noise components, respectively. Since the residual carrier signal samples used for the estimates $\hat{w}_k(\tilde{i}_A)$ occur prior to the full spectrum signal samples $y_k(i_B)$, and since $\{m_k(i_A)\}$ and $\{n_j(i_B)\}$ are i.i.d. sequences, it follows that $\hat{w}_k(\tilde{i}_A)$ and $n_j(i_B)$ are uncorrelated random variables for every k and j. Each $n_j(i_B)$ has zero mean. It then follows from Eqs. (13) and (12) that $n_c(i_B)$ also has zero mean and is, moreover, uncorrelated with $s_c(i_B)$. Let $\mathrm{Var}[Z] = \mathrm{E}\left[|Z - \mathrm{E}[Z]|^2\right]$ denote the variance of a complex random variable Z. Thus it follows from Eq. (11) that the actual SNR of the combiner signal output $z(i_B)$ given by Eq. (10) can be written as

$$\gamma_{ML} = \frac{|\mathrm{E}[z(i_B)]|^2}{\mathrm{Var}[z(i_B)]} = \frac{|\mathrm{E}[s_c(i_B)]|^2}{\mathrm{Var}[s_c(i_B)] + \mathrm{Var}[n_c(i_B)]} \tag{14}$$

It is well known [4,5], that $\underline{\hat{X}}(i_A)$ and $\underline{\hat{R}}_A(i_A)$ are statistically independent and that $2(L-2)\hat{\sigma}^2_{Ak}(i_A)/\sigma^2_{Ak}$ has a chi-square distribution with $2(L-1)$ degrees of freedom. As a result of these properties, it follows from Eq. (9) in a derivation similar to that in [1] that, for $1 \le k \le K$,

$$\mathrm{E}\left[\hat{w}_k(\tilde{i}_A)\right] = w_k \tag{15}$$

where $\{w_k\}$ are the optimal combining weights given by Eq. (5). That is, the estimated weight coefficients are unbiased as in the uncorrelated-noise case [1]. It then follows from Eqs. (12), (15), (5), and (6) that for both the correlated- and uncorrelated-noise cases,

$$|\mathrm{E}[s_c(i_B)]| = |\mathrm{e}^{j\, s(i_B)\, \delta} \sum_{k=1}^{K} \mathrm{E}[\hat{w}_k(\tilde{i}_A)] V_k| = \gamma \tag{16}$$

Consider next the variances of $s_c(i_B)$ and $n_c(i_B)$ in Eq. (14). Using Eqs. (12) and (15), we have

$$\mathrm{Var}[s_c(i_B)] = \sum_{k=1}^{K} \sum_{j=1}^{K} \mathrm{E}\left[(\hat{w}_k(\tilde{i}_A) - w_k)(\hat{w}_j(\tilde{i}_A) - w_j)^* V_k V_j^*\right] \tag{17}$$

where w_k is given by Eq. (5). Consider first the case when the Gaussian noise processes in the signals from different array elements are mutually uncorrelated. Since $\hat{w}_k(\tilde{i}_A)$ and $\hat{w}_j(\tilde{i}_A)$ are pairwise independent for $k \neq j$ in this case, the variance of $s_c(i_B)$ can be written as

$$\mathrm{Var}_U[s_c(i_B)] = \sum_{k=1}^{K} \mathrm{Var}\left[\hat{w}_k\left(\tilde{i}_A\right)\right] |V_k|^2 \tag{18}$$

Let

$$\beta_1 = 2\mathcal{R}e \left\{ \sum_{k=1}^{K} \sum_{j=k+1}^{K} V_k V_j^* \left\{ \mathrm{E}\left[\hat{w}_k\left(\tilde{i}_A\right) \hat{w}_j^*\left(\tilde{i}_A\right)\right] - w_k w_j^* \right\} \right\} \tag{19}$$

Combining Eqs. (17), (18), and (19) then yields

$$\mathrm{Var}[s_c(i_B)] = \mathrm{Var}_U[s_c(i_B)] + \beta_1 \tag{20}$$

Recall that $n_c(i_B)$ has zero mean and $\hat{w}_k(\tilde{i}_A)$ is statistically independent of $n_j(i_B)$ for all k and j. Then, similar to the derivation leading to Eq. (20), we can write

$$\mathrm{Var}[n_c(i_B)] = \sum_{k=1}^{K} \sum_{j=1}^{K} \mathrm{E}\left[\hat{w}_k\left(\tilde{i}_A\right) \hat{w}_j^*\left(\tilde{i}_A\right)\right] \mathrm{E}\left[n_k(i_B) n_j^*(i_B)\right] = \mathrm{Var}_U[n_c(i_B)] + \beta_2 \tag{21}$$

where

$$\beta_2 = 2\mathcal{R}e \left\{ \sum_{k=1}^{K} \sum_{j=k+1}^{K} \mathrm{E}\left[\hat{w}_k\left(\tilde{i}_A\right) \hat{w}_j^*\left(\tilde{i}_A\right)\right] \mathrm{E}\left[n_k\left(i_B\right) n_j^*\left(i_B\right)\right] \right\} \tag{22}$$

and where

$$\mathrm{Var}_U[n_c(i_B)] = \sum_{k=1}^{K} \mathrm{E}\left[|\hat{w}_k\left(\tilde{i}_A\right)|^2\right] \ \mathrm{Var}\left[|n_k(i_B)|^2\right]$$

is the variance of $n_c(i_B)$ in the uncorrelated-noise case. It then follows from Eqs. (14) and (16) that the actual SNR of the combiner output in the uncorrelated-noise case is given by

$$\gamma_{ML}^{U} = \frac{\gamma^2}{\mathrm{Var}_U[s_c(i_B)] + \mathrm{Var}_U[n_c(i_B)]} \tag{23}$$

So it follows from Eqs. (14), (16), (20), (21), and (23) that

$$\gamma_{ML} = \gamma_{ML}^{U}\left(\frac{1}{1+d}\right) \tag{24}$$

where

$$d = \frac{\beta_1 + \beta_2}{\mathrm{Var}_U[s_c(i_B)] + \mathrm{Var}_U[n_c(i_B)]} \tag{25}$$

The factor $1/(1+d)$ in Eq. (24) represents the improvement in SNR caused by the correlation between the noises in the signals received from different array elements. Note in particular that β_1 and β_2 can be either positive or negative in value. Hence, an SNR improvement is obtained when d is negative and a degradation is obtained otherwise.

Expressions for $\mathrm{Var}_U[s_c(i_B)]$ and $\mathrm{Var}_U[n_c(i_B)]$ are given in [1]. Thus, we need only determine β_1 and β_2 to obtain d and thereby obtain an expression for γ_{ML} from Eq. (24). In order to do this, we need only obtain an expression for $\mathrm{E}[\hat{w}_k(\tilde{i}_A)\hat{w}_j^*(\tilde{i}_A)]$ when $k \neq j$. Using the property that $\underline{\hat{X}}(i_A)$ is statistically independent of $\underline{\hat{R}}_A(i_A)$, it then follows from Eqs. (9), (7), and (8) that, for $k \neq j$,

$$\mathrm{E}\left[\hat{w}_k\left(\tilde{i}_A\right)\hat{w}_j^*\left(\tilde{i}_A\right)\right] = \frac{1}{4\eta^2\cos^2\delta}\mathrm{E}\left[\hat{X}_k^*\left(\tilde{i}_A\right)\hat{X}_j\left(\tilde{i}_A\right)\right]\mathrm{E}\left[\frac{1}{\hat{\sigma}_{Ak}^2(\tilde{i}_A)\hat{\sigma}_{Aj}^2(\tilde{i}_A)}\right] \tag{26}$$

Since $\underline{\hat{X}}(i_A)$ has mean $\underline{X}$ and covariance matrix $\frac{1}{L}\underline{R}_A = \frac{1}{\eta L}\underline{R}_B$ [5], it follows that

$$\mathrm{E}\left[\hat{X}_k^*\left(\tilde{i}_A\right)\hat{X}_j\left(\tilde{i}_A\right)\right] = \frac{1}{\eta L}r_{Bkj}^* + X_k^*X_j \tag{27}$$

Recall that $2(L-2)\hat{\sigma}_{Ak}^2(\tilde{i}_A)$ is the kth diagonal element of the matrix $(L-2)\underline{\hat{R}}_A(\tilde{i}_A)$. Let

$$\underline{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{12}^* & A_{22} \end{bmatrix}$$

be a 2 × 2 matrix where A_{11} and A_{22} are the kth and jth diagonal elements, respectively, and A_{12} is the element in the kth row and jth column of $(L-2)\underline{\hat{R}}_A(\tilde{i}_A)$. So we have

$$\mathrm{E}\left[\frac{1}{\hat{\sigma}^2_{Ak}(\bar{i}_A)\hat{\sigma}^2_{Aj}(\bar{i}_A)}\right] = 4(L-2)^2\mathrm{E}\left[\frac{1}{A_{11}A_{22}}\right] \tag{28}$$

Complex multivariate statistical sampling theory [4,5] has shown that $\underline{A}$ has the same distribution as that of $\sum_{i=1}^{L-1} \underline{Z}_i \underline{Z}_i^\dagger$ where $\{\underline{Z}_i\}$ is a sequence of i.i.d. zero-mean complex Gaussian random vectors with covariance matrix $\underline{\Sigma}$ given by

$$\underline{\Sigma} = \begin{bmatrix} r_{Akk} & r_{Akj} \\ r^*_{Akj} & r_{Ajj} \end{bmatrix} = \frac{1}{\eta} \begin{bmatrix} r_{Bkk} & r_{Bkj} \\ r^*_{Bkj} & r_{Bjj} \end{bmatrix} \tag{29}$$

This type of distribution is called a complex Wishart distribution [4,5], with parameters $\underline{\Sigma}$ and $(L-1)$. Denote the determinant and trace of a matrix $\underline{A}$ by $|\underline{A}|$ and $\mathrm{tr}(\underline{A})$, respectively. Then if $L \geq 4$, the joint Wishart probability density of (A_{11}, A_{22}, A_{12}) is given by [4]

$$p(A_{11}, A_{22}, A_{12}) = \frac{\left(A_{11}A_{22} - |A_{12}|^2\right)^{L-3}}{\pi\Gamma(L-1)\Gamma(L-2)|\underline{\Sigma}|^{L-1}} \exp\left[-\mathrm{tr}(\underline{\Sigma}^{-1}\underline{A})\right] \tag{30}$$

for $A_{11}, A_{22} \geq 0$ and $|A_{12}|^2 \leq A_{11}A_{22}$, where $\Gamma(x)$ is the gamma function. The derivation in Appendix A obtains the expression given by Eq. (A-5) for $\mathrm{E}[1/A_{11}A_{22}]$ starting from Eq. (30). Define for $L \geq 4$ and $0 \leq x < 1$,

$$f_L(x) = (L-2)(1-x)^{L-3} \sum_{k=0}^{\infty} \binom{k+L-3}{k} \frac{x^k}{k+L-2} \tag{31}$$

Assume that the correlation coefficients between noise components of the kth and jth array element outputs ρ_{Bkj} given by Eq. (4) are always less than one in magnitude. Then, by using Eqs. (28), (49), (31), and (27), Eq. (26) can be written as

$$\mathrm{E}\left[\hat{w}_k\left(\bar{i}_A\right)\hat{w}^*_j\left(\bar{i}_A\right)\right] = f_L\left(|\rho_{Bkj}|^2\right)\left[\frac{1}{\eta L\cos^2\delta}\,\frac{\rho^*_{Bkj}}{2\sigma_{Bk}\sigma_{Bj}} + \frac{V^*_k V_j}{4\sigma^2_{Bk}\sigma^2_{Bj}}\right] \tag{32}$$

When $|\rho_{Bkj}| < 1$ and $L \geq 4$, we obtain, by using Eqs. (2), (4), (5), and (32) in Eqs. (19) and (22),

$$\begin{aligned}\beta_1 + \beta_2 &= 2\sum_{k=1}^{K}\sum_{j=k+1}^{K}\left\{f_L\left(|\rho_{Bkj}|^2\right)\left[\frac{|\rho_{Bkj}|^2}{\eta L\cos^2\delta} + \frac{|V_k|^2|V_j|^2}{4\sigma^2_{Bk}\sigma^2_{Bj}}\right.\right.\\ &\quad \left.\left. + \left(1 + \frac{1}{\eta L\cos^2\delta}\right)\frac{|V_k||V_j|}{2\sigma_{Bk}\sigma_{Bj}}|\rho_{Bkj}|\cos\left(\vartheta_{kj} - \varphi_{Bkj}\right)\right] - \frac{|V_k|^2|V_j|^2}{4\sigma^2_{Bk}\sigma^2_{Bj}}\right\}\end{aligned} \tag{33}$$

where φ_{Bkj} is the phase of the correlation coefficient ρ_{Bkj} between $n_k(i_B)$ and $n_k(i_B)$ and where $\vartheta_{kj} = \theta_k - \theta_j$ is the phase difference between the signal components of the kth and jth array elements. Finally, by using Eqs. (44) and (48) of [1] for $\mathrm{Var}_U[s_c(i_B)]$ and $\mathrm{Var}_U[n_c(i_B)]$, respectively, Eq. (25) can be written as

$$d = \frac{\beta_1 + \beta_2}{\left(\frac{L-2}{L-3}\right)[\gamma + (\gamma + K)/\eta L \cos^2 \delta] + \left(\frac{1}{L-3}\right)\sum_{k=1}^{K}\left(\frac{|V_k|^2}{2\sigma_{Bk}^2}\right)^2} \tag{34}$$

where $\beta_1 + \beta_2$ is given by Eq. (33) and γ is given by Eq. (6). In order to arrive at an explicit expression for γ_{ML}, we note that Eqs. (44) and (48) of [1] in Eq. (23) give

$$\gamma_{ML}^{U} = \frac{\gamma^2}{\left(\frac{L-2}{L-3}\right)[\gamma + (\gamma + K)/\eta L \cos^2 \delta] + \left(\frac{1}{L-3}\right)\sum_{k=1}^{K}\left(\frac{|V_k|^2}{2\sigma_{Bk}^2}\right)^2} \tag{35}$$

So the actual SNR of the combiner output in the correlated-noise case can be determined from Eqs. (24), (34), and (35) when $L \geq 4$ and $|\rho_{Bkj}| < 1$. The two measures of particular interest in understanding the SNR performance are $1/(1+d)$ and γ_{ML}/γ. The measure $1/(1+d)$ represents the gain in SNR caused by the correlation between the array element noises and will be referred to as the *correlation gain*. In the uncorrelated-noise case, γ/γ_{ML} represents the loss in SNR due to the combining algorithm since γ is the maximum possible achievable SNR. We shall adopt the same measure here and define γ_{ML}/γ as the *combining gain* for ease of comparison with the uncorrelated-noise case. The combining gain also represents the gain in SNR over the sum of SNRs of the individual array element outputs.

Let us examine the characteristics of the SNR performance. In the uncorrelated-noise case, the actual SNR performance γ_{ML}^{U} converges to the maximum possible SNR achievable γ as the number of samples L approaches infinity. It is interesting to also examine the combining gain in the correlated-noise case as the number of samples approaches infinity. It is shown in Appendix B that $f_L(x) \to 1$ as $L \to \infty$ for $0 \leq x < 1$. Assume that the pairwise noise correlation coefficients ρ_{Bkj} are all less than one in magnitude. Then, taking the limit as $L \to \infty$ in Eqs. (34) and (33) yields

$$\lim_{L \to \infty} d = \frac{2}{\gamma} \sum_{k=1}^{K} \sum_{j=k+1}^{K} \frac{|V_k||V_j|}{2\sigma_{Bk}\sigma_{Bj}} |\rho_{Bkj}| \cos(\vartheta_{kj} - \varphi_{Bkj}) \tag{36}$$

So the limiting value of d can also be of either sign, positive or negative. In fact, the limiting value is always negative if $\vartheta_{kj} - \varphi_{Bkj} = \pi$ for all $k \neq j$, and always positive if $\vartheta_{kj} - \varphi_{Bkj} = 0$ for all $k \neq j$. It then follows from Eq. (24) that as $L \to \infty$, the limiting value of the actual SNR performance γ_{ML} in the correlated-noise case can be either greater or smaller than the maximum possible SNR γ in the uncorrelated-noise case, depending on the relation between the signal and noise correlation phases. This is not really that surprising, since the maximum possible SNR performance in the correlated-noise case is generally not equal to γ.

Bounds on the actual SNR performance γ_{ML} that depend on a fewer number of parameters than the exact expression are also useful. We shall derive upper and lower bounds that depend only on the maximum magnitude of the noise correlation coefficients and on γ, the sum of the SNRs of the individual array element outputs. We first note the following inequalities derived in [1] for this purpose:

$$\frac{\gamma^2}{K} = \frac{1}{K}\left(\sum_{k=1}^{K}\frac{|V_k|^2}{2\sigma_{Bk}^2}\right)^2 \leq \sum_{k=1}^{K}\left(\frac{|V_k|^2}{2\sigma_{Bk}^2}\right)^2 \leq \gamma^2 \tag{37}$$

Similar to the left-hand inequality of Eq. (37), we have

$$\left(\sum_{k=1}^{K} \frac{|V_k|}{\sqrt{2\sigma_{Bk}^2}}\right)^2 \leq K \left(\sum_{k=1}^{K} \frac{|V_k|^2}{2\sigma_{Bk}^2}\right) = K\gamma \tag{38}$$

Applying the left-hand inequality of Eq. (37), the inequality of Eq. (38) gets the following upper bounds:

$$2\sum_{k=1}^{K}\sum_{j=k+1}^{K} \frac{|V_k|^2|V_j|^2}{4\sigma_{Bk}^2\sigma_{Bj}^2} \leq \gamma^2\left(1 - \frac{1}{K}\right) \tag{39}$$

and

$$2\sum_{k=1}^{K}\sum_{j=k+1}^{K} \frac{|V_k||V_j|}{2\sigma_{Bk}\sigma_{Bj}} \leq (K-1)\gamma \tag{40}$$

Let

$$\rho_{max} = \max_{k \neq j} |\rho_{Bkj}|$$

be the maximum magnitude of the correlation coefficients between array element noise components. Note from Eq. (33) that the worst-case phase resulting in the largest possible d occurs when $\vartheta_{kj} - \varphi_{Bkj} = 0$ for all $k \neq j$. Hence, application of the left-hand inequality in Eq. (37), the inequalities of Eqs. (39) and (40), and the bounds of Eq. (B-7) on $f_L(x)$ given in Appendix B yields the following upper bound on the worst-case d:

$$d \leq \frac{(L-2)(K-1)\rho_{max}\left[\gamma + (K\rho_{max} + \gamma)/\eta L \cos^2\delta\right] + \gamma^2(1 - 1/K)}{(L-2)\left[\gamma + (\gamma + K)/\eta L \cos^2\delta\right] + \gamma^2/K} \tag{41}$$

Similarly, since the best-case phase resulting in the most negative possible d occurs when $\vartheta_{kj} - \varphi_{Bkj} = \pi$ for all $k \neq j$, the following lower bound on the best-case d can be obtained:

$$d \geq -\frac{(L-2)(K-1)\rho_{max}\gamma\left[1 + 1/\eta L \cos^2\delta\right]}{(L-2)\left[\gamma + (\gamma + K)/\eta L \cos^2\delta\right] + \gamma^2/K} \tag{42}$$

Finally, using the inequalities of Eq. (37) in Eq. (35) yields the following bounds on the actual SNR performance γ_{ML}^U in the uncorrelated-noise case:

$$\gamma_{ML}^U \leq \frac{(L-3)\gamma^2}{(L-2)\left[\gamma + (\gamma + K)/\eta L \cos^2\delta\right] + \gamma^2/K} \tag{43}$$

and

$$\gamma_{ML}^U \geq \frac{(L-3)\gamma^2}{(L-2)\left[\gamma + (\gamma + K)/\eta L \cos^2\delta\right] + \gamma^2} \tag{44}$$

An upper bound on the actual SNR performance γ_{ML} is obtained by using the lower bound of Eq. (42) on d and the upper bound of Eq. (43) on γ_{ML}^U in Eq. (24). Similarly, a lower bound on γ_{ML} is obtained by using, instead, the upper bound of Eq. (41) on d and the lower bound of Eq. (44) on γ_{ML}^U.

IV. Numerical Example

We consider here the numerical example in [1] of using a $K = 7$ element array feed in the JPL Deep Space Network. In this example, a modulation index $\delta = 80$ deg and a primitive sample period $T_0 = 2.5 \times 10^{-8}$ s are assumed. The full-spectrum modulation signal is assumed to be of bandwidth 2×10^6 Hz, which yields $M_B = 20$. Moreover, the ratio of the full-spectrum bandwidth to the residual carrier bandwidth $\eta = M_A/M_B = 200$. Nominal P_T/N_0 of 55 and 65 dB-Hz are considered with corresponding $\gamma = (P_T/N_0)M_B T_0$. Upper and lower bounds on the combining gain γ_{ML}/γ are shown in Fig. 1 as a function of the number of samples L averaged to obtain the weight estimates. Here $P_T/N_0 = 55$ dB-Hz, and maximum correlation coefficient magnitudes ρ_{max} of 0.01 and 0.02 are considered. Convergence of these bounds to within 0.01 dB of their limiting values occurs at about $L = 3000$ samples. This corresponds to an averaging time of $M_A T_0 L = 0.3$ s and supports real-time operations for antenna deformation compensation. The limiting upper bounds on the combining gain are about 0.26 and 0.56 dB for ρ_{max} equal to 0.01 and 0.02, respectively. The corresponding lower bounds on the combining gain are -0.26 and -0.50 dB, respectively. The actual limiting value for the combining gain, which is given by Eq. (36), will fall between these bounds. Similar results are shown in Fig. 2 for $P_T/N_0 = 65$ dB-Hz, where convergence of the bounds occurs at smaller values of L to virtually the same limiting values as the $P_T/N_0 = 55$ dB-Hz case.

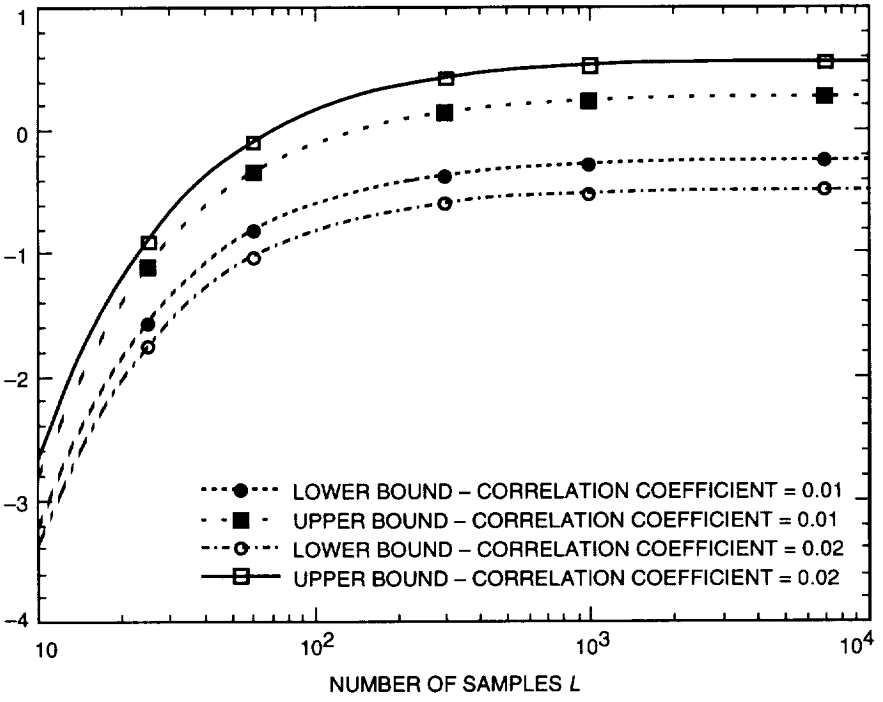


Fig. 1. Combining gain versus L for P_T/N_0 = 55 dB-Hz.

Figures 3 and 4 plot upper and lower bounds on the correlation gain $1/(1 + d)$ for ρ_{max} equal to 0.01 and 0.02. Figure 3 considers $P_T/N_0 = 55$ dB-Hz and Fig. 4 considers $P_T/N_0 = 65$ dB-Hz. The limiting values of these bounds are identical to the limiting values of the corresponding bounds on the combining gain. The differences between the behavior of the lower bounds at $P_T/N_0 = 55$ dB-Hz and those at $P_T/N_0 = 65$ dB-Hz are due to the looseness of these lower bounds at small values of L. For a large number L of samples, the upper and lower bounds on the combining gain diverge as the maximum correlation coefficient magnitude increases. This can be seen from Fig. 5, which shows the upper and lower bounds on combining gain for $P_T/N_0 = 55$ dB-Hz at $L = 5000$ samples as ρ_{max} increases from 0.01 to 0.1. The upper bound increases from 0.26 to 3.96 dB and the lower bound decreases from -0.26 to -2.05 dB in this range of ρ_{max}. The observed correlation coefficients of 0.01 magnitude in [2] were

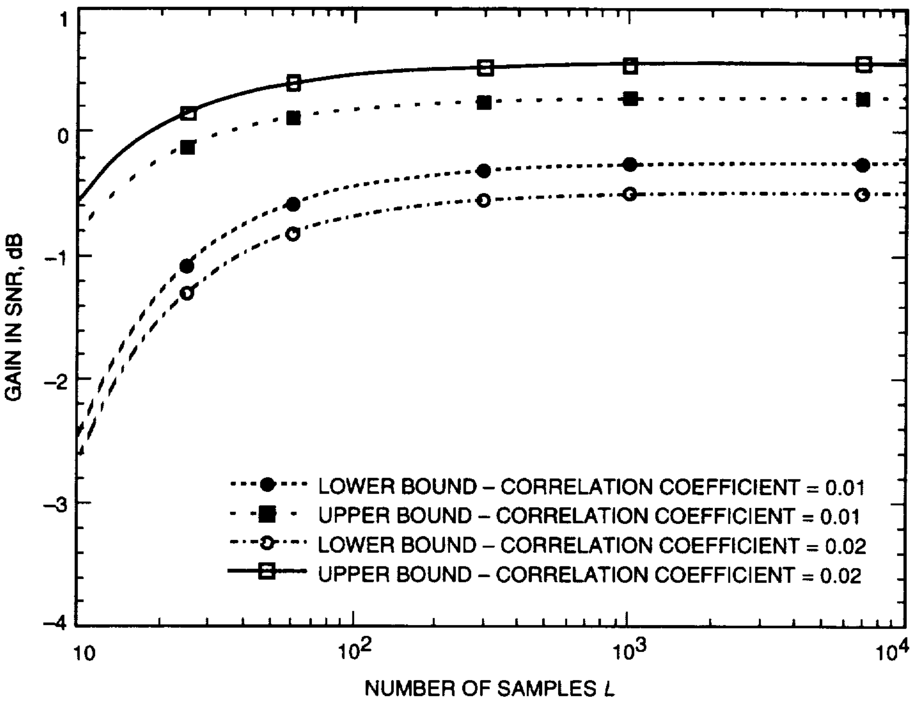


Fig. 2. Combining gain versus L for P_T/N_0 = 65 dB-Hz.

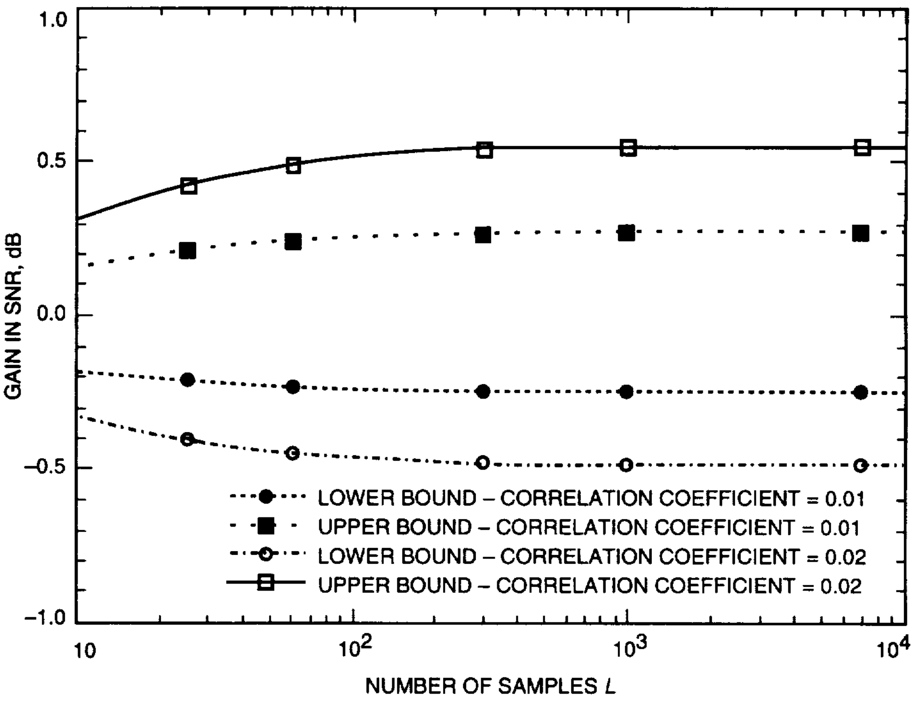


Fig. 3. Correlation gain versus L for P_T/N_0 = 55 dB-Hz.

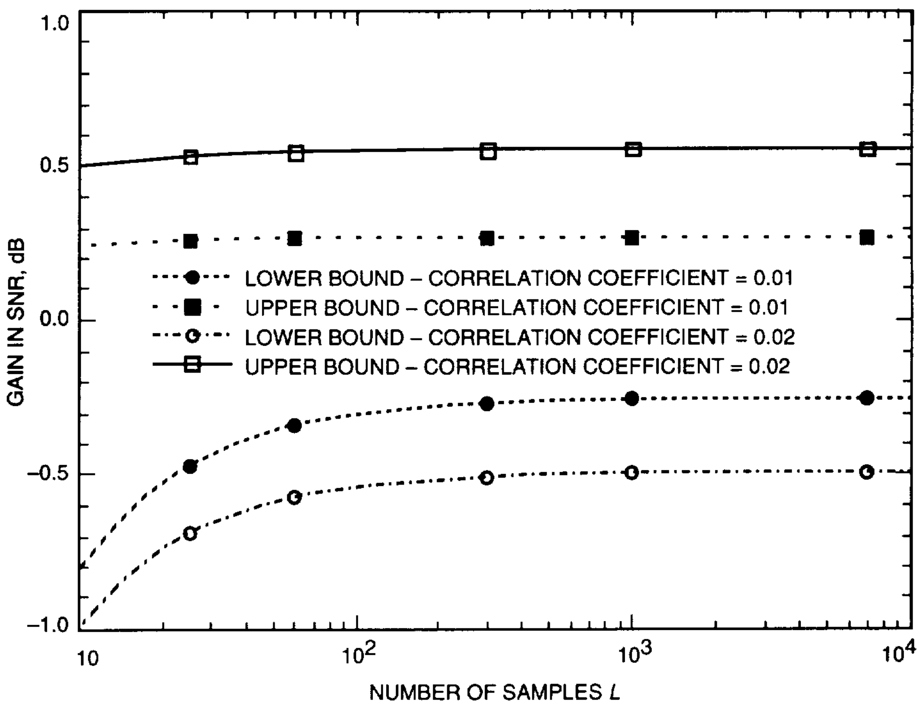


Fig. 4. Correlation gain versus *L* for P_T/N_0 = 65 dB-Hz.

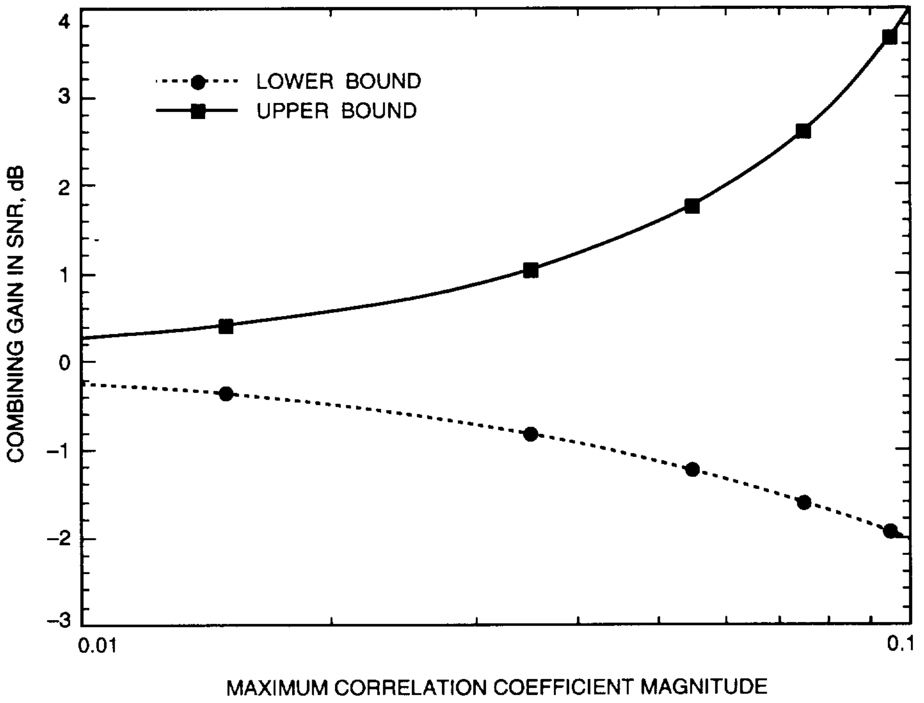


Fig. 5. Combining gain versus ρ_{max} for P_T/N_0 = 55 dB-Hz and *L* = 5000.

obtained in clear-sky conditions with a receiver noise temperature of 90 K and a system noise temperature of 120 K. Improvement of the receiver noise temperature to 25 K will increase the correlation coefficient magnitude to about 0.02. As noted above, a maximum possible improvement of 0.56 dB and a maximum possible degradation of -0.50 dB results. Preliminary measurements at DSS 13 indicate that even larger amounts of correlation occur under adverse weather conditions. This will result in even larger potential improvement or degradation of SNR performance relative to the uncorrelated-noise case.

V. Conclusion

An array feed combiner system for the recovery of SNR loss due to antenna reflector deformation has been implemented and is currently being evaluated on the Jet Propulsion Laboratory 34-meter DSS-13 antenna. The current signal-combining algorithms are optimum under the assumption that the white Gaussian noise processes in the received signals from different array elements are uncorrelated. Experimental data at DSS 13 indicate that these noise processes are indeed mutually correlated. The main result of this article is an analytical derivation of the actual SNR performance of the current suboptimal signal-combining algorithm in this correlated-noise environment. The analysis here shows that the combined-signal SNR can be either improved or degraded depending on the relation between the array signal and noise correlation coefficient phases. Further performance improvement will require the development of effective combining systems that take into account the correlations between the array feed element noise processes.

References

[1] V. A. Vilnrotter, E. R. Rodemich, and S. J. Dolinar, Jr., "Real-Time Combining of Residual Carrier Array Signals Using ML Weight Estimates," *IEEE Transactions Communications*, vol. COM-40, no. 3, pp. 604–615, March 1992.

[2] B. A. Iijima, V. A. Vilnrotter, and D. Fort, "Correlator Data Analysis for the Array Feed Compensation System," *The Telecommunications and Data Acquisition Progress Report 42-117, January—March 1994*, Jet Propulsion Laboratory, Pasadena, California, pp. 110–118, May 15, 1994.

[3] J. H. Yuen, *Deep Space Telecommunications Systems Engineering*, Chapter 5, New York: Plenum, 1983.

[4] N. R. Goodman, "Statistical Analysis Based on a Certain Multivariate Complex Gaussian Distribution (An Introduction)," *Annals of Mathematical Statistics*, vol. 34, pp. 152–177, 1963.

[5] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 2nd ed., New York: Wiley, 1984.

[6] I. S. Gradshteyn and I. M. Ryzhik, *Tables of Integrals, Series and Products*, Corrected and Enlarged 4th ed., London: Academic Press, 1980.

[7] J. Riordan, *An Introduction to Combinatorial Analysis*, New York: Wiley, 1958.

Appendix A

Derivation of $E\left[\frac{1}{A_{11}A_{22}}\right]$

We first obtain the joint probability density function $p(A_{11}, A_{22})$ of (A_{11}, A_{22}) by integrating Eq. (30) over the complex region $\mathcal{S} = \{A_{12} : |A_{12}| < \sqrt{A_{11}A_{22}}\}$ of values taken on by A_{12}. Let $\underline{G} = \{G_{ij}\} = \underline{\Sigma}^{-1}$ and convert the variables G_{12} and A_{12} into polar coordinates: $G_{12} = |G_{12}|\mathrm{e}^{j\psi}$ and $A_{12} = r\mathrm{e}^{j\phi}$. Then it follows from Eq. (30) that

$$
\begin{aligned}
p(A_{11}, A_{22}) &= \frac{\mathrm{e}^{-(G_{11}A_{11}+G_{22}A_{22})}}{\pi\Gamma(L-1)\Gamma(L-2)|\underline{\Sigma}|^{L-1}} \int_{\mathcal{S}} \left(A_{11}A_{22} - |A_{12}|^2\right)^{L-3} \mathrm{e}^{-2\mathcal{R}e(G_{12}^{*}A_{12})}\, dA_{12} \\
&= \frac{\mathrm{e}^{-(G_{11}A_{11}+G_{22}A_{22})}}{\pi\Gamma(L-1)\Gamma(L-2)|\underline{\Sigma}|^{L-1}} \int_0^{\sqrt{A_{11}A_{22}}} r\left(A_{11}A_{22} - r^2\right)^{L-3} \left[\int_0^{2\pi} \mathrm{e}^{-2r|G_{12}|\cos(\phi-\psi)}\, d\phi\right] dr \\
&= \frac{2\mathrm{e}^{-(G_{11}A_{11}+G_{22}A_{22})}}{\Gamma(L-1)\Gamma(L-2)|\underline{\Sigma}|^{L-1}} \int_0^{\sqrt{A_{11}A_{22}}} r\left(A_{11}A_{22} - r^2\right)^{L-3} I_0(2r|G_{12}|)\, dr \qquad \text{(A-1)}
\end{aligned}
$$

where $I_0(x)$ is the zero-order modified Bessel function of the first kind, which has series representation

$$
I_0(x) = \sum_{k=0}^{\infty} \frac{x^{2k}}{2^{2k}(k!)^2} \qquad \text{(A-2)}
$$

By making a change of the variable of integration using the series of Eq. (A-2) and the integral relation (3.251) of [6], the integral in Eq. (A-1) can be written as

$$
\begin{aligned}
\int_0^{\sqrt{A_{11}A_{22}}} r\left(A_{11}A_{22} - r^2\right)^{L-3} I_0(2r|G_{12}|)\, dr &= (A_{11}A_{22})^{L-2} \sum_{k=0}^{\infty} \frac{\left(A_{11}A_{22}|G_{12}|^2\right)^k}{k!} \int_0^1 \left(1 - s^2\right)^{L-3} s^{2k+1}\, ds \\
&= (A_{11}A_{22})^{L-2} \sum_{k=0}^{\infty} \frac{\left(A_{11}A_{22}|G_{12}|^2\right)^k}{k!} \left[\frac{\Gamma(k+1)\Gamma(L-2)}{2\Gamma(k+L-1)}\right] \\
&\qquad \text{(A-3)}
\end{aligned}
$$

Substituting Eq. (A-3) into Eq. (A-1) and using the fact that $\Gamma(n) = (n-1)!$ for integer n, we obtain

$$
p(A_{11}, A_{22}) = \frac{\left(A_{11}A_{22}\right)^{L-2} \mathrm{e}^{-(G_{11}A_{11}+G_{22}A_{22})}}{(L-2)!\,|\underline{\Sigma}|^{L-1}} \sum_{k=0}^{\infty} \frac{(A_{11}A_{22}|G_{12}|^2)^k}{k!(k+L-2)!} \qquad \text{(A-4)}
$$

Using Eq. (A-4), integrating term by term in the series, and obtaining $\underline{G} = \underline{\Sigma}^{-1}$ and $|\underline{\Sigma}|$ directly from Eq. (29) in terms of ρ_{Bkj}, r_{Bkk}, and r_{Bjj}, we have

$$\mathrm{E}\left[\frac{1}{A_{11}A_{22}}\right] = \frac{1}{(L-2)|\underline{\Sigma}|^{L-1}(G_{11}G_{22})^{L-2}} \sum_{k=0}^{\infty} \binom{k+L-3}{k} \frac{\left(\frac{|G_{12}|^2}{G_{11}G_{22}}\right)^k}{k+L-2}$$

$$= \frac{\eta^2\left(1-|\rho_{Bkj}|^2\right)^{L-3}}{(L-2)r_{Bkk}r_{Bjj}} \sum_{k=0}^{\infty} \binom{k+L-3}{k} \frac{|\rho_{Bkj}|^{2k}}{k+L-2} \tag{A-5}$$

The series in Eq. (A-5) can be shown to converge by using the ratio convergence test whenever $|\rho_{Bkj}| < 1$.

Appendix B
Bounds on $f_L(x)$

Let $L \geq 4$ and $0 \leq x < 1$. We will obtain upper and lower bounds on $f_L(x)$ that are asymptotically tight in the limit as $L \to \infty$. First note that

$$\frac{L-2}{k+L-2} = \left(\frac{L-3}{k+L-3}\right)\left(\frac{L-2}{L-3}\right)\left(\frac{k+L-3}{k+L-2}\right) \tag{B-1}$$

and that for $k \geq 0$,

$$\frac{L-3}{L-2} \leq \frac{k+L-3}{k+L-2} \leq 1 \tag{B-2}$$

Using these bounds of Eq. (B-2) in Eq. (B-1), we have

$$\frac{L-3}{k+L-3} \leq \frac{L-2}{k+L-2} \leq \left(\frac{L-2}{L-3}\right)\frac{L-3}{k+L-3} \tag{B-3}$$

Next, by using the bounds of Eq. (B-3) in Eq. (31), we get the following:

$$(1-x)^{L-3} \sum_{k=0}^{\infty} \binom{k+L-4}{k} x^k \leq f_L(x) \tag{B-4}$$

$$f_L(x) \leq \left(\frac{L-2}{L-3}\right)(1-x)^{L-3} \sum_{k=0}^{\infty} \binom{k+L-4}{k} x^k \tag{B-5}$$

It can be shown in [7] that for $0 \leq x < 1$,

$$\sum_{k=0}^{\infty} \binom{k+L-4}{k} x^k = \left(\frac{1}{1-x}\right)^{L-3} \tag{B-6}$$

Using Eq. (B-6) in Eqs. (B-4) and (B-5), we then obtain, for $0 \leq x < 1$,

$$1 \leq f_L(x) \leq \frac{L-2}{L-3} \tag{B-7}$$

The upper and lower bounds given in Eq. (B-7) are both asymptotically tight in the limit as $L \rightarrow \infty$. So we can conclude that for $0 \leq x < 1$, $f_L(x) \rightarrow 1$ as $L \rightarrow \infty$, where the convergence is uniform in x.

TDA Progress Report 42-121

May 15, 1995

Minimal Trellises for Linear Block Codes and Their Duals

A. B. Kiely, S. Dolinar, and L. Ekroot
Communications Systems Research Section

R. J. McEliece
California Institute of Technology
and
Communications Systems Research Section

W. Lin
California Institute of Technology

We consider the problem of finding a trellis for a linear block code that minimizes one or more measures of trellis complexity for a fixed permutation of the code. We examine constraints on trellises, including relationships between the minimal trellis of a code and that of the dual code. We identify the primitive structures that can appear in a minimal trellis and relate this to those for the minimal trellis of the dual code.

I. Introduction

Every linear block code can be represented by a minimal trellis, originally introduced by Bahl et al. [1], which is a labeled graph that can be used as a template for encoding or decoding. As shown by McEliece,[1] the minimal trellis simultaneously minimizes the maximum number of states, the total numbers of vertices and edges in the trellis, and the total numbers of additions and path comparisons required for decoding with the Viterbi algorithm.

In this article, we examine properties of the minimal trellis representation of a code and its dual for a fixed permutation. A companion article [2] uses these results to examine the problem of finding a permutation that minimizes one or more trellis complexity measures.

Section II reviews the subject of minimal trellises for a fixed permutation of a code. We examine the building blocks of such trellises and identify several different measures of trellis size or complexity. In Section III, we illustrate the connection between the minimal trellis of a code and that of the dual code. The section includes results that describe the structure and complexity of trellises for self-dual and other special codes.

[1] R. J. McEliece, "On The BCJR Trellis for Linear Block Codes," submitted to *IEEE Trans. Inform. Theory*.

II. Minimal Trellis Representation of a Code

A. The Minimal Span Generator Matrix

For any linear (n, k) block code $\mathcal{C}$ over $GF(q)$ there exists a minimal span generator matrix (MSGM) representing $\mathcal{C}$. A minimal trellis $\mathcal{T}$ for the code can be constructed from the MSGM. The trellis has $n + 1$ levels of vertices and n levels of edges. The vertex levels, called depths, are numbered from 0 to n; the edge levels, called stages, are numbered from 1 to n. Each stage of edges corresponds to one stage of encoding or decoding using the trellis. Each vertex at depth i represents a possible encoder state after the ith stage of encoding. The ith stage corresponds to the ith column of the generator matrix, whereas the ith depth corresponds to the "space between" columns i and $i + 1$.

The edge span of any row of the generator matrix is the smallest set of consecutive integers (stages) containing its nonzero positions. The vertex span of the row is the set of depths i such that at least one nonzero symbol occurs before and after depth i. Using the generator matrix to encode k information symbols in n stages of encoding, the edge span of the jth row represents the interval of stages during which the jth information symbol can affect the encoder output. The vertex span of the jth row is the set of depths at which the jth information symbol can affect the encoder state.

For example, the (6,3) shortened Hamming code has the minimal span generator matrix

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{1}$$

The edge spans are $\{1, 2, 3\}$, $\{2, 3, 4, 5, 6\}$, and $\{3, 4, 5\}$. The vertex spans are $\{1, 2\}$, $\{2, 3, 4, 5\}$, and $\{3, 4\}$. We use the term span length to refer to the cardinality of a span.

A remarkable result is that the MSGM simultaneously makes all of the spans as short as possible: The edge spans (vertex spans) for any other generator matrix representing $\mathcal{C}$ always contain the corresponding spans of some row-permuted MSGM.[2] Any generator matrix can be put into minimal span form using the following greedy algorithm: At each step, perform any row operation that reduces the edge span of any row of the matrix. The rows of the MSGM are then "atomic codewords," according to the terminology of Kschischang and Sorokine [5].

Each vertex or state at a given depth can be uniquely labeled using k or fewer symbols from $GF(q)$. But any given state-label symbol can be reused to represent several information symbols, as long as the vertex spans of the corresponding rows of the generator matrix do not overlap. This reassignment of state-label symbols to multiple rows of the generator matrix is the key to efficient trellis representations of the code.

For example, the minimal trellis $\mathcal{T}$ produced for the (6,3) shortened Hamming code with MSGM given in Eq. (1) is shown in Fig. 1. For this trellis, we can define the binary state label to be $s_2 s_1$, where $s_2 = 1$ at depth i if the second information bit is 1 and i is within the vertex span of the second row, and $s_1 = 1$ if either (1) the first information bit is 1 and i is within the vertex span of the first row or (2) the third information bit is 1 and i is within the vertex span of the third row. This time-sharing arrangement for state bit s_1 is possible because the vertex spans of the first and third rows do not overlap.

In the sequel, we will be interested primarily in nondegenerate codes, which we define as codes whose minimum distance d and dual code minimum distance $d^{\perp}$ are both at least 2. Degenerate codes have a simple interpretation: If $d < 2$, the vertex span of some row of the MSGM must be empty; if $d^{\perp} < 2$,

[2] Ibid.

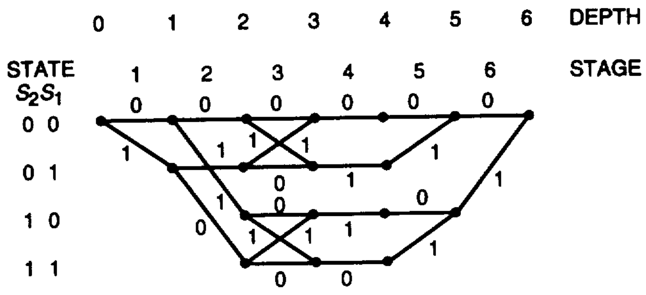


Fig. 1. A minimal trellis for the (6,3) shortened Hamming code.

some column of the generator matrix must be identically zero. For a degenerate code, we can simply ignore the extraneous symbol positions (if $d^{\perp} < 2$) and/or separately decode the unprotected information symbols (if $d < 2$). The code consisting of the remaining code symbols is then nondegenerate.

B. Past and Future Subcodes

Following Forney [3], let us define the ith past and future subcodes, denoted $\mathcal{P}_i$ and $\mathcal{F}_i$, to be the sets of all codewords whose vertex spans are contained in $[0, i-1]$ and $[i+1, n]$, respectively. The dimensions of these codes can be easily determined from the MSGM: $f_i \triangleq \dim(\mathcal{F}_i)$ is the number of rows for which the leftmost nonzero entry lies in column $i+1$ or later, and $p_i \triangleq \dim(\mathcal{P}_i)$ is the number of rows for which the rightmost nonzero entry lies in column i or earlier.[3] This implies that p_i and f_i are monotonic,

$$0 = p_0 \leq p_1 \leq \cdots \leq p_n = k = f_0 \geq f_1 \geq \cdots \geq f_n = 0 \tag{2}$$

and never change by more than 1 from one index to the next.

For each $1 \leq i \leq n$, we define the left- and right-basis indicators, $l_i, r_i \in \{0, 1\}$, to identify the positions where the future and past dimensions change:

$$l_i \triangleq f_{i-1} - f_i \qquad r_i \triangleq p_i - p_{i-1}$$

For any i, $l_i = 1$ if and only if the edge span of some row of the MSGM G *begins* in column i, or equivalently, the ith column of G is linearly independent of the $i-1$ columns to the *left*. Similarly, $r_i = 1$ if and only if the edge span of some row of G *ends* in column i, i.e., the ith column of G is linearly independent of the $n-i$ columns to the *right*. The columns where $l_i = 1$ and the columns where $r_i = 1$ each forms a basis for the column space of G, and these sets are called the left basis and the right basis, respectively. The positions of the left and right basis columns can be regarded as information positions when the generator matrix is used to encode the information left to right or right to left, respectively.

C. Primitive Structures of a Minimal Trellis

There are four basic building blocks that can be used to construct the minimal trellis for any nondegenerate code. At any given stage i, all primitive structures are of the same type, which is determined by the values of l_i and r_i. The primitive structures are

[3] Ibid.

(1) Simple extension ($-$): This primitive structure appears at stage i when $l_i = 0, r_i = 0$, e.g., stage 4 in Fig. 1. Simple extensions at stage i imply a single edge out of each vertex at depth $i-1$ and a single edge into each vertex at depth i; hence, the number of vertices remains constant.

(2) Simple expansion ($<$): This corresponds to $l_i = 1, r_i = 0$, e.g., stages 1 and 2 in Fig. 1. There are q edges out of each vertex at depth $i-1$, and a single edge into each vertex at depth i, hence, multiplying by q the number of states from one vertex depth to the next.

(3) Simple merger ($>$): This corresponds to $l_i = 0, r_i = 1$, e.g., stages 5 and 6 in Fig. 1. A simple merger is a time-reversed simple expansion, reducing the number of states by a factor of q.

(4) Butterfly (⧖): This corresponds to $l_i = 1, r_i = 1$, e.g., stage 3 in Fig. 1. There are q edges out of each vertex at depth $i-1$ and q edges into each vertex at depth i; hence, the number of states is constant.

The total numbers of such primitive structures in the trellis are denoted by N_-, $N_<$, $N_>$, and $N_{⧖}$, respectively. For example, the trellis in Fig. 1 has $N_< = 3 = N_>$, $N_{⧖} = 2$, $N_- = 4$. Because the graph has exactly one initial node and one terminal node, the total number of simple expansions must equal the total number of simple mergers:

$$N_< = N_>$$

The total number of edges in the trellis, E, can be found by counting the number of edges associated with each primitive trellis structure:

$$E = N_- + qN_< + qN_> + q^2 N_{⧖} \tag{3}$$

Similarly, the total number of mergers M is the sum of the number of simple mergers and the mergers included in butterflies:

$$M = N_> + qN_{⧖} \tag{4}$$

If we count the total number of vertices associated with each primitive structure, then each vertex in the trellis (excluding initial and terminal nodes) will be counted twice, so the total number of vertices V satisfies

$$2V - 2 = 2N_- + (q+1)N_< + (q+1)N_> + 2qN_{⧖}$$

which gives

$$V = 1 + N_- + (q+1)N_< + qN_{⧖} \tag{5}$$

Combining Eqs. (3), (4), and (5), we find

$$M = \frac{E - V + 1}{q - 1}$$

This is the generalization of the binary version of this result found by McEliece.[4]

[4] Ibid.

D. Measures of Trellis Complexity for Viterbi Decoding

The vertex space dimension at depth i is

$$v_i = k - f_i - p_i, \qquad i = 0, \cdots, n \tag{6}$$

and the edge space dimension at stage i is

$$e_i = k - f_i - p_{i-1}, \qquad i = 1, \cdots, n \tag{7}$$

The total number of vertices at depth i is q^{v_i}, and the total number of edges at stage i is q^{e_i}. Of course, $v_i \geq 0$ for all i since at least one vertex must exist at each depth. Also, for nondegenerate codes, $e_i \geq 1$ for all i, i.e., no stage consists of a single edge.

The most commonly used measure of Viterbi decoding complexity for a minimal trellis is the maximum dimension of its state space,

$$s_{\max} \triangleq \max_i v_i \tag{8}$$

This complexity metric has been cited as one of the essential characteristics of any code [6]. Similarly, the maximum dimension of the edge space is

$$e_{\max} \triangleq \max_i e_i \tag{9}$$

Forney argues that this is a more relevant complexity measure because, unlike $s_{\max}$, this quantity cannot be reduced by combining adjacent stages of a trellis [4].

A different metric, used in McEliece's derivation of the MSGM,[5] is the total length of all the edge spans of the rows of the MSGM:

$$\varepsilon \triangleq \sum_{j=1}^{k} \varepsilon_j \tag{10}$$

where ε_j denotes the length of the edge span of the jth row of the MSGM. A similar span length metric is the total length of all the vertex spans:

$$\nu \triangleq \sum_{j=1}^{k} \nu_j$$

where $\nu_j = \varepsilon_j - 1$ is the length of the vertex span of the jth row of the MSGM. These two metrics are equivalent to the sums of all the edge dimensions or vertex dimensions (summed over stages or depths, respectively):

[5] Ibid.

$$\nu + k = \varepsilon = \sum_{i=1}^{n} e_i = k + \sum_{i=0}^{n} v_i$$

McEliece argues that more meaningful measures of Viterbi decoding complexity are the total numbers of edges E, vertices V, and mergers M, rather than simply the vertex or edge dimensionality:[6]

$$E = \sum_{i=1}^{n} q^{e_i} \tag{11}$$

$$V = \sum_{i=0}^{n} q^{v_i} \tag{12}$$

$$M = \sum_{i=1}^{n} r_i q^{v_i} = \sum_{i=1}^{n} l_i q^{v_{i-1}} = \frac{1}{q}\sum_{i=1}^{n} l_i q^{e_i} = \frac{1}{q}\sum_{i=1}^{n} r_i q^{e_i} \tag{13}$$

E is equal to the number of binary additions required to compute path metrics, and M is the number of q-ary comparisons required to merge trellis paths. The computational complexity of Viterbi decoding is proportional to E.[7]

III. Minimal Trellis Representation of the Dual Code

In this section, we explore the relationship between the minimal trellises for a code C and its dual $C^\perp$.

A. Past and Future Subcode Relationships

As discussed in Section II.B, $l_i = 0$ if and only if the ith column of the MSGM can be written as some linear combination of the $i-1$ columns to its left. In other words, there exists a dual codeword y of the form

$$y = \underbrace{XXX\cdots X}_{i-1} 1 \underbrace{000\cdots 0}_{n-i}$$

Where $XXX\cdots X$ denotes some sequence of symbols from $GF(q)$. Defining $y_1, y_2, \cdots, y_{n-k}$ in this manner for each of the left-dependent columns in the MSGM produces $n-k$ dual codewords of the form

$$\begin{array}{lcll} y_1 & = & XXX\cdots X1000\cdots & \qquad 0 \\ y_2 & = & XXX\cdots & X1000\cdots 0 \\ \vdots & & & \\ y_{n-k} & = & XXX\cdots & \qquad X1 \end{array}$$

These dual codewords are clearly linearly independent and, thus, can be used as the rows of the generator matrix for $C^\perp$. We see that the positions where $r_i^\perp = 1$ are precisely the positions where $l_i = 0$; the same argument applied to the right-dependent columns shows that the positions where $l_i^\perp = 1$ are precisely the

[6] Ibid.

[7] Ibid.

positions where $r_i = 0$. Here $l_i^{\perp}$ and $r_i^{\perp}$ are the left- and right-basis indicators for $C^{\perp}$. These observations lead to the following theorem.

Theorem 1. For each $0 \leq i \leq n$, the left- and right-basis indicators for a code and its dual are related by

$$l_i + r_i^{\perp} = l_i^{\perp} + r_i = 1$$

and the dimensions p_i, f_i of the past and future subcodes of a code are given in terms of those of the dual code $p_i^{\perp}$, $f_i^{\perp}$ as follows:

$$\begin{aligned} p_i &= k - n + i + f_i^{\perp} \\ f_i &= k - i + p_i^{\perp} \end{aligned}$$

We believe that this result, which relates minimal trellises of a code and dual for any *fixed* permutation, is more fundamental than similar dual relationships for permutations of codes. This result is also contained in [4], but is derived by first considering permutations of codes.

B. Primitive Trellis Structures for the Dual Code

Much information about the trellis for the dual code can be inferred from the trellis structure of the code. For example, if the code has a simple expansion at the ith stage, then $l_i = 1, r_i = 0$, which implies, using Theorem 1, that the dual code has $l_i^{\perp} = 1$, $r_i^{\perp} = 0$; hence, the trellis of the dual code also has a simple expansion structure at this stage. Repeating this procedure, we find the "dual" of each primitive structure, shown in Table 1.

Given an unlabeled trellis, Table 1 can be used to determine the number and type of primitive structures present at every depth of the trellis for the dual code. However, we cannot in general determine the interconnections without additional information about the code.

The dual relationship for primitive structures shown in Table 1 implies that

$$N_{<}^{\perp} = N_{<} = N_{>} = N_{>}^{\perp}$$

and

$$N_{-} = qN_{\text{⧖}}^{\perp}$$

C. Dual-Code Complexity Measures

The following well-known result, first noted by Forney [3], is a consequence of Eq. (6) and Theorem 1.

Lemma. A code and its dual code have equivalent vertex spaces, namely, for each i,

$$v_i = v_i^{\perp}$$

Table 1. Dual primitive structures.

Code structure	Dual structure
Simple extension (−) $l_i = 0, r_i = 0$	Butterfly (⧖) $l_i^{\perp} = 1, r_i^{\perp} = 1$
Simple expansion (<) $l_i = 1, r_i = 0$	Simple expansion (<) $l_i^{\perp} = 1, r_i^{\perp} = 0$
Simple merger (>) $l_i = 0, r_i = 1$	Simple merger (>) $l_i^{\perp} = 0, r_i^{\perp} = 1$
Butterfly (⧖) $l_i = 1, r_i = 1$	Simple extension (−) $l_i^{\perp} = 0, r_i^{\perp} = 0$

Consequently, many of the trellis complexity measures for a code can be determined by evaluating the same measure on the dual code:

$$V = V^{\perp}$$

$$s_{\max} = s_{\max}^{\perp}$$

$$\varepsilon - k = \nu = \nu^{\perp} = \varepsilon^{\perp} - (n - k)$$

Note that this implies $\varepsilon = \varepsilon^{\perp}$ for any rate 1/2 code.

The number of edges in the minimal trellis of a code and its dual is not as conveniently related. From Eq. (7) and Theorem 1,

$$e_i = e_i^{\perp} + (1 - r_i^{\perp} - l_i^{\perp})$$

for each $1 \leq i \leq n$. Consequently, since $|1 - r_i^{\perp} - l_i^{\perp}| \leq 1$, and from the definition of E,

$$\frac{1}{q}E \leq E^{\perp} \leq qE$$

Equality is possible only for the degenerate $(n, n, 1)$ code or its dual.

D. Minimal Trellises for Self-Dual and Other Special Codes

For self-dual codes, the theory of the previous two sections collapses neatly to yield stronger results because, for any such code, $l_i = l_i^{\perp}$ and $r_i = r_i^{\perp}$ for all i. Consequently, from Theorem 1,

Theorem 2. For any self-dual code C, for each $i = 1, 2, \cdots n$, either

(1) $l_i = 1$ and $r_i = 0$, or

(2) $l_i = 0$ and $r_i = 1$

i.e., every stage corresponds to an information symbol when encoding from one direction and a parity symbol when encoding from the other direction. The only primitive trellis structures in $\mathcal{T}(C)$ are simple expansions and simple mergers.

The converse of Theorem 2 does not hold: A code whose minimal trellis contains only simple expansions and mergers need not be self-dual. However, such a trellis can always be relabeled to represent a self-dual code.

The following theorem, which is a consequence of Theorem 2 and Eqs. (3), (4), and (5), shows that, for self-dual codes, the complexity measures E, V, and M are linearly related, and the maximum edge and vertex dimensions are equal.

Theorem 3. For any self-dual code,

$$V = \frac{q+1}{2q}E + 1$$

$$M = \frac{1}{2q}E$$

$$s_{\max} = e_{\max}$$

There is another case where we can restrict the type of structures that can appear in the trellis for a code:

Theorem 4. If C is a code with all codeword weights divisible by some integer $m > 2$, then,

(1) There does not exist a position i such that $l_i = r_i = 1$, i.e., $\mathcal{T}(C)$ contains no butterfly structures.

(2) C cannot have rate greater than $\frac{1}{2}$.

(3) $e_{\max} = s_{\max}$

(4) $V \geq \left(\frac{q+1}{q}\right) E + 1$

(5) $M \leq \frac{1}{2q} E$

(6) $V^{\perp} \leq \left(\frac{q+1}{q}\right) E^{\perp} + 1$

(7) $M^{\perp} \geq \frac{1}{2q} E^{\perp}$

Proof: If $l_i = r_i = 1$, then the ith column begins and ends spans in the MSGM. This implies the existence of codewords of the form $x = XXX \cdots X10^{n-i}$ and $y = 0^{i-1}(-1)XXX \cdots X$, where (-1) denotes the additive inverse of 1 in $GF(q)$ and $XXX \cdots X$ denotes some string of symbols in $GF(q)$. Then $x + y$ is a codeword of weight $|x| + |y| - 2$, which cannot be divisible by m. This proves (1). From (1), we have $l_i + r_i \leq 1$ for all i, so $2k = \sum_{i=1}^{n}(l_i + r_i) \leq \sum_{i=1}^{n} 1 = n$, which proves (2). The fact that $\mathcal{T}(C)$ can have no butterfly structures proves (3). From Eq. (13), $2qM = \sum_{i=1}^{n}(l_i + r_i)q^{e_i} \leq \sum_{i=1}^{n} q^{e_i} = E$, proving (5), and (4) follows directly. Since $l_i + r_i \leq 1$, Theorem 1 implies $l_i^{\perp} + r_i^{\perp} \geq 1$, which gives (6) and (7). □

Codes for which all codeword weights are divisible by some integer other than one are called divisible codes [7]. Examples of divisible codes include the (31,10,12) binary cyclic codes and doubly even self-dual codes such as the extended Golay code.

The converse of Theorem 4 does not hold—a code is not necessarily divisible when $l_i + r_i \leq 1$ for all i. If a code and its dual satisfy the conditions of Theorem 4, then the code strongly resembles a self-dual code: The code must have rate 1/2 and its trellis contain only simple expansions and simple mergers.

IV. Conclusion

In this article, we have examined the trellis complexity problem by first considering the minimal span generator matrix for a fixed permutation of a code. McEliece showed that the so-called minimal trellis indeed minimizes not only the maximum state dimension of the trellis but also a whole gamut of complexity measures.[8] Here we have augmented the list of reasonable complexity measures and interrelated them. We have also illustrated the connection between the complexity measures and the four primitive structures of a minimal trellis for a nondegenerate code.

We developed some useful relationships between the minimal trellis of a code and that of its dual. The duality relationships lead to interesting connections among several of the complexity measures for the special case of self-dual codes.

Acknowledgment

A portion of W. Lin's contribution was sponsored by AFOSR grant no. F4960-94-1-005. Part of this work was presented at the 32nd Annual Allerton Conference on Communication, Control, and Computing in Allerton, Illinois, in October 1994.

References

[1] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 2, pp. 284–287, March 1974.

[2] A. B. Kiely, S. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Trellis Complexity Bounds for Decoding Linear Block Codes," *The Telecommunications and Data Acquisition Progress Report 42-121, January–March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 159–172, May 15, 1995.

[3] G. D. Forney, "Coset Codes—Part II: Binary Lattices and Related Codes (Appendix A)," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1152–1187, 1988.

[4] G. D. Forney, "Dimension/Length Profiles and Trellis Complexity of Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6., pp. 1741–1752, November 1994.

[8] Ibid.

[5] F. R. Kschischang and V. Sorokine, "On the Trellis Structure of Block Codes," *Proceedings of the 1994 IEEE International Symposium on Information Theory*, Trondheim, Norway, June 27–July 1, 1994, p. 337, 1994.

[6] D. J. Muder, "Minimal Trellises for Block Codes," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1049–1053, 1988.

[7] H. N. Ward, "Divisible Codes," *Arch. Math.*, vol. 36, pp. 485–494, 1991.

N95-32234

TDA Progress Report 42-121 May 15, 1995

Trellis Complexity Bounds for Decoding Linear Block Codes

A. B. Kiely, S. Dolinar, and L. Ekroot
Communications Systems Research Section

R. J. McEliece
California Institute of Technology
and
Communications Systems Research Section

W. Lin
California Institute of Technology

We consider the problem of finding a trellis for a linear block code that minimizes one or more measures of trellis complexity. The domain of optimization may be different permutations of the same code or different codes with the same parameters. Constraints on trellises, including relationships between the minimal trellis of a code and that of the dual code, are used to derive bounds on complexity. We define a partial ordering on trellises: If a trellis is optimum with respect to this partial ordering, it has the desirable property that it simultaneously minimizes all of the complexity measures examined. We examine properties of such optimal trellises and give examples of optimal permutations of codes, most notably the (48,24,12) quadratic residue code.

I. Introduction

A minimal trellis is a labeled graph that can be used as a template for encoding or decoding. In [6], we examined properties of trellises for fixed permutations of a code. A code's minimal trellis is unique as long as the ordering of the code's symbols is fixed. However, different permutations of the symbols yield different minimal trellises. An optimum minimal trellis for the code is one that minimizes a suitable measure of trellis complexity over all possible permutations of the code. There are no known efficient algorithms for constructing optimum minimal trellises.

We expand the results of [6] to examine the problem of finding a permutation that minimizes one or more trellis complexity measures. We extend these results to the problem of finding a minimal complexity trellis over all codes with the same parameters. We identify certain sufficient conditions for a code or a permutation to simultaneously minimize all of the complexity measures.

In Section II, we discuss dimension/length profiles of a code [3,11], which are equivalent to Wei's generalized Hamming weights [12]. The dimension/length profiles are used to derive some straightforward complexity bounds. We summarize some properties of these profiles, including duality relationships.

We define a partial ordering on minimal trellises in Section III. If the minimal trellises for two codes are comparable in terms of this partial ordering, then each of the complexity measures for one trellis is bounded by the same measure evaluated for the other trellis. This partial ordering can sometimes be used to identify the permutation of a code with the least (or most) complex minimal trellis, or the code with the lowest (or highest) complexity trellis of all codes with the same parameters. The extremal codes determined by this partial ordering turn out to meet the complexity bounds described in Section II. We illustrate certain properties and give examples of such permutations and codes.

II. Trellis Complexity Bounds

The minimal trellis results of [6] assume a fixed coordinate ordering for the code. However, the trellis structure and, hence, trellis complexity are different for different permutations of the code coordinates. Massey refers to the procedure of reordering the code symbols to reduce the trellis complexity as "the *art* of trellis decoding" [9, p. 9].

In this section, we identify code parameters that affect the possible trellis complexity, describe upper and lower bounds based on these parameters, and illustrate properties of certain codes that have low complexity trellises. Our results apply to a gamut of possible complexity measures introduced in [6]: the maximum vertex (state) and edge dimensions $(s_{\max}, e_{\max})$, the total vertex and edge spans (ν, ε), and the total numbers of vertices, edges, and mergers (V, E, M). In this article, all theorems are presented without proof; proofs are supplied in a separate article.[1]

First, some notation: Let $\mathcal{S}_n$ denote the set of all permutations of $\{1, 2, \cdots, n\}$, and for any $\pi \in \mathcal{S}_n$, let $\mathcal{C}\pi$ denote the code $\mathcal{C}$ with coordinates reordered according to π. Because the code and dual code provide symmetric constraints on the code's minimal trellis, the complexity bounds are developed by considering the characteristics of both the code and its dual. We refer to an (n, k, d) code over $GF(q)$ with dual distance $d^{\perp}$ as an $(n, k, d, d^{\perp})$ code.

A. Bounds Relating One Complexity Measure to Another

The following lemma arises from the definitions of $s_{\max}$ and $e_{\max}$ and from the fact that the vertex and edge dimensions, v_i and e_i, change by no more than one unit from one index to the next.

Lemma 1. The vertex dimensions and edge dimensions are upper bounded by

$$v_i \le \min\{i, n-i, s_{\max}\}, \qquad 0 \le i \le n$$

$$e_i \le \min\{i, n+1-i, e_{\max}\}, \qquad 1 \le i \le n$$

Summing the inequalities in Lemma 1 leads to the following bounding relationships among the complexity measures.

Theorem 1. The total complexity measures ν, ε, V, E are upper bounded in terms of the maximum complexity measures $s_{\max}$, $e_{\max}$ by

$$\nu \le s_{\max}(n - s_{\max}) \tag{1}$$

[1] A. B. Kiely, S. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Trellis Decoding Complexity of Linear Block Codes," submitted to *IEEE Trans. Inform. Theory*.

$$\varepsilon \leq e_{\max}(n + 1 - e_{\max}) \tag{2}$$

$$V \leq \left[n + \frac{q+1}{q-1} - 2s_{\max} \right] q^{s_{\max}} - \frac{2}{q-1} \tag{3}$$

$$E \leq \left[n + \frac{2q}{q-1} - 2e_{\max} \right] q^{e_{\max}} - \frac{2q}{q-1} \tag{4}$$

Since the average edge dimension over all stages is ε/n and the average vertex dimension over the last n depths is ν/n, loose lower bounds on V and E can be obtained from Jensen's inequality.

Theorem 2. The total complexity measures V, E are lower bounded in terms of the total span length complexity measures ν, ε by

$$V \geq 1 + nq^{\nu/n}$$

$$E \geq nq^{\varepsilon/n}$$

There are also tighter lower bounds on V and E in terms of ν and ε.

Theorem 3. Given a total span length ν, or equivalently ε, let $\Delta\varepsilon = \varepsilon - e^-(n + 1 - e^-)$ and $\Delta\nu = \nu - s^-(n - s^-)$, where $e^- \leq (n+1)/2$ and $s^- \leq n/2$ are the largest integers such that $\Delta\varepsilon \geq 0$ and $\Delta\nu \geq 0$. Then

$$V \geq \left[n + \frac{q+1}{q-1} - 2s^- \right] q^{s^-} - \frac{2}{q-1} + (q-1)q^{s^-}\Delta\nu$$

$$E \geq \left[n + \frac{2q}{q-1} - 2e^- \right] q^{e^-} - \frac{2q}{q-1} + (q-1)q^{e^-}\Delta\varepsilon$$

This theorem follows from the observation that, for a given ν or ε, a vertex or edge dimension profile such as the one in Fig. 1 minimizes V or E. Notice the similarity of these lower bounds in terms of s^- and e^- with the corresponding upper bounds, Eqs. (3) and (4), in terms of $s_{\max}$ and $e_{\max}$.

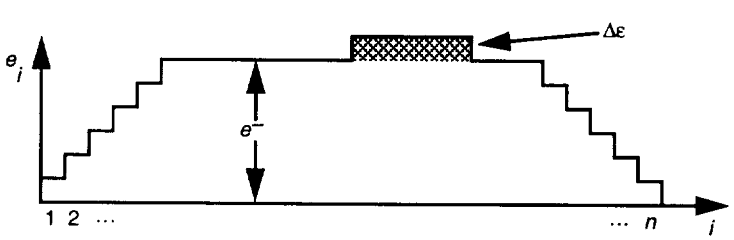


Fig. 1. An edge dimension profile that minimizes E subject to a constraint on total edge span ε.

B. Complexity Lower Bounds Based on MSGM Span Length

Every row of a generator matrix for an $(n, k, d, d^\perp)$ code must have edge-span length $\varepsilon_i \geq d$ and vertex-span length $\nu_i \geq d - 1$. Applying this simple bound to both the code and the dual code and using the fact that $\nu^{\perp} = \nu = \varepsilon - k$ leads to the following lower bounds on the span length complexity measures ν and ε.

Theorem 4. The total lengths ν and ε of the vertex spans and edge spans for any $(n, k, d, d^{\perp})$ code are lower bounded by

$$\nu \geq \max\left\{k(d-1), (n-k)(d^{\perp}-1)\right\}$$

$$\varepsilon \geq k + \max\left\{k(d-1), (n-k)(d^{\perp}-1)\right\}$$

Applying the Singleton bound to the inequalities in this theorem gives the weaker bounds $\nu \geq (d-1)(d^{\perp}-1)$ and $\varepsilon \geq k + (d-1)(d^{\perp}-1)$.

We say that a code meeting the bounds in Theorem 4 with equality is a minimal span code. An example is the $(n, 1, n, 2)$ repetition code. To construct a nondegenerate $(n, k, d, 2)$ binary minimal span code for any $d > 2$ and $n \geq d + (k-1)\lceil d/2 \rceil$, let the first row of the minimal span generator matrix (MSGM) be

$$\underbrace{111\cdots1}_{d}\underbrace{000\cdots0}_{n-d}$$

and form each successive row by cyclically shifting the previous row at least $\lceil d/2 \rceil$ positions but not more than d positions to the right, such that the total of all the shifts is $n - d$. The dual of a minimal span code is also a minimal span code. These codes are not usually good in terms of distance, though they have very low complexity trellises.

The span length bounds in Theorem 4, combined with the bounds of Eqs. (1) and (2), lead to lower bounds on the complexity measures $s_{\max}$, $e_{\max}$ for any $(n, k, d, d^{\perp})$ code:

$$s_{\max}(n - s_{\max}) \geq \max\left\{k(d-1), (n-k)(d^{\perp}-1)\right\}$$

$$e_{\max}(n + 1 - e_{\max}) \geq k + \max\left\{k(d-1), (n-k)(d^{\perp}-1)\right\}$$

A slightly weaker version of this bound on $s_{\max}$ has been proved for both linear and nonlinear codes [8]. This bound implies, for instance, that the average edge dimension $e_{\max}$ can never be lower than the asymptotic coding gain kd/n. We can also obtain bounds on V and E for any $(n, k, d, d^{\perp})$ code by substituting the right-hand sides of the bounds in Theorem 4 for ν and ε in Theorems 2 and 3.

C. Dimension/Length Profiles

We can see from the definitions of the complexity measures in [6] that a permutation of C that makes f_i and p_i large (small) wherever possible will produce a low (high) complexity trellis. It is useful, therefore, to find bounds on these quantities.

The support of a vector x is the set of nonzero positions in x. The support of a set of vectors is the union of the individual supports.

Definition 1. For a given code C and any $0 \leq i \leq n$, let $K_i(C)$ be the maximum dimension of a linear subcode of C having support whose size is no greater than i. The set $\{K_i(C), i = 0, \cdots, n\}$ is called the dimension/length profile (DLP) [3,11].

The DLP and similar concepts have been used recently by many other researchers to bound trellis complexity. Extensive bibliographies are given in [3] and Kiely et al.[2] Since the past and future subcodes $\mathcal{P}_i$ and $\mathcal{F}_i$ are subcodes of $\mathcal{C}$ with support size no larger than i and $n-i$, respectively, the past and future subcode dimensions are bounded by the DLP

$$p_i \leq \max_{\pi \in \mathcal{S}_n} p_i(\mathcal{C}\pi) = K_i(\mathcal{C}) \tag{5}$$

$$f_i \leq K_{n-i}(\mathcal{C}) \tag{6}$$

These bounds, which also appeared in [4, Eq. (1.4)], are tight in the following sense: For any i, there exists a permuted version of $\mathcal{C}$ that meets the bound of Eq. (5) and one that meets Eq. (6), though it may not be possible to meet both simultaneously. The DLP of a code can be used to lower bound the trellis complexity for any permutation of that code, as we shall see in Section II.E.

Since each $K_i(\mathcal{C})$ is associated with a linear subcode of $\mathcal{C}$, we can use bounds on the best possible linear codes (i.e., codes with the largest possible minimum distance) to upper bound the DLP:

Theorem 5. For an $(n, k, d, d^{\perp})$ code $\mathcal{C}$ and any $0 \leq i \leq n$,

$$p_i \leq K_i(\mathcal{C}) \leq \overline{K}_i(n, k, d, d^{\perp})$$

$$f_i \leq K_{n-i}(\mathcal{C}) \leq \overline{K}_{n-i}(n, k, d, d^{\perp})$$

where

$$\overline{K}_i(n, k, d, d^{\perp}) \triangleq \min[k_{\max}(i, d), k - n + i + k_{\max}(n - i, d^{\perp})]$$

and $k_{\max}(m, d)$ is the largest possible dimension for any q-ary linear block code of length m and minimum distance d. The set $\{\overline{K}_i(n, k, d, d^{\perp}), i = 0, \ldots, n\}$ is called the upper dimension/length profile (UDLP) for the code parameters $(n, k, d, d^{\perp})$.

Bounds based on the UDLP may be loose, as it may not be possible for a single $(n, k, d, d^{\perp})$ code and its dual to both have a series of subcodes, all with the maximum code dimensions. However, these bounds are important practically, because much data about the best possible codes have been tabulated [1] and, in many cases, the UDLP bounds can be achieved with equality.

Since for any (n, k) code $\mathcal{C}$, p_i and f_i both reach maximum values of k ($f_0 = k$ and $p_n = k$) and can fall from these values at a maximum rate of one unit per trellis stage, p_i and f_i are lower bounded as follows:

$$K_i(\mathcal{C}) \geq p_i \geq \underline{K}_i(n, k) \triangleq \max(0, k - n + i) \tag{7}$$

$$K_{n-i}(\mathcal{C}) \geq f_i \geq \underline{K}_{n-i}(n, k) = \max(0, k - i) \tag{8}$$

The set $\{\underline{K}_i(\mathcal{C}), i = 0, 1, \cdots, n\}$ is called the lower dimension/length profile (LDLP) for the code parameters (n, k). The LDLP stays at 0 until the last possible depth before it can rise linearly at the rate of one dimension per depth to reach its final value of k at depth n. The LDLP can be used to upper bound the complexity of a minimal trellis for an arbitrary (n, k) code.

[2] Ibid.

D. Properties of Dimension/Length Profiles

The DLPs possess many of the same properties as the past and future subcode dimensions that they bound. For example, the monotonicity and unit increment properties of $\{p_i\}$ also hold for $K_i(C)$, $\overline{K}_i(C)$, and $\underline{K}_i(C)$: The increments $\overline{K}_{i+1}(n,k,d,d^\perp)-\overline{K}_i(n,k,d,d^\perp)$, $K_{i+1}(C)-K_i(C)$, and $\underline{K}_{i+1}(n,k)-\underline{K}_i(n,k)$ must equal 0 or 1 for all i. Similarly, duality properties can be easily extended.

There is a convenient relationship between the DLP of a code and that of its dual, stated in [4, Eq. (1.12)] and [3, Theorem 3], which is equivalent to the duality relationship for generalized Hamming weights [12, Theorem 3]. Similar relationships hold for the upper and lower dimension/length profiles:

Lemma 2. For all $0 \le i \le n$, the DLP, UDLP, and LDLP satisfy the following duality relationships:

$$K_i(C^\perp) = i - k + K_{n-i}(C)$$

$$\overline{K}_i(n, n-k, d^\perp, d) = i - k + \overline{K}_{n-i}(n,k,d,d^\perp)$$

$$\underline{K}_i(n, n-k) = i - k + \underline{K}_{n-i}(n,k)$$

E. Complexity Bounds From Dimension/Length Profiles

The DLP bounds, Eqs. (5) and (6), combined with the complexity definitions lead to simple bounds on trellis complexity that are useful when the DLP of a given code is known. These bounds can be tightened slightly by using the additional fact that the vertex and edge dimensions must be nonnegative everywhere.

Theorem 6. The complexity measures for the minimal trellis $\mathcal{T}(C\pi)$ corresponding to any permutation π of a given (n,k) code C are lower bounded by

$$s_{\max}(C\pi) \ge \max_{i\in[0,n]} (k - K_i(C) - K_{n-i}(C)) \tag{9}$$

$$e_{\max}(C\pi) \ge \max_{i\in[1,n]} (k - K_{i-1}(C) - K_{n-i}(C)) \tag{10}$$

$$\varepsilon(C\pi) \ge \sum_{i=0}^{n} \max\{0, k - K_i(C) - K_{n-i}(C)\} \tag{11}$$

$$V(C\pi) \ge \sum_{i=0}^{n} q^{\max\{0,k-K_i(C)-K_{n-i}(C)\}} \tag{12}$$

$$E(C\pi) \ge \sum_{i=1}^{n} q^{\max\{0,k-K_{i-1}(C)-K_{n-i}(C)\}} \tag{13}$$

$$M(C\pi) \ge \frac{1}{q}\sum_{i=1}^{n} [K_i(C) - K_{i-1}(C)] q^{\max\{0,k-K_{i-1}(C)-K_{n-i}(C)\}} \tag{14}$$

The DLP bound, Eq. (9), on state complexity has been derived in [3,11].[3] Some of the bounds in Theorem 6 can be improved slightly when $\mathcal{C}$ is nondegenerate, because this condition implies that $e_i \geq 1$.

The UDLP bound (Theorem 5) leads to similar lower bounds on trellis complexity that apply to all codes with given code parameters.

Theorem 7. The complexity measures for the minimal trellis $\mathcal{T}(\mathcal{C})$ representing any $(n, k, d, d^{\perp})$ code $\mathcal{C}$ are lower bounded by

$$s_{\max}(\mathcal{C}) \geq \max_{i \in [0,n]} \left[k - \overline{K}_i(n,k,d,d^{\perp}) - \overline{K}_{n-i}(n,k,d,d^{\perp})\right] \tag{15}$$

$$e_{\max}(\mathcal{C}) \geq \max_{i \in [1,n]} \left[k - \overline{K}_{i-1}(n,k,d,d^{\perp}) - \overline{K}_{n-i}(n,k,d,d^{\perp})\right] \tag{16}$$

$$\varepsilon(\mathcal{C}) \geq \sum_{i=1}^{n} \max\left\{0, k - \overline{K}_{i-1}(n,k,d,d^{\perp}) - \overline{K}_{n-i}(n,k,d,d^{\perp})\right\} \tag{17}$$

$$V(\mathcal{C}) \geq \sum_{i=0}^{n} q^{\max\left\{0, k-\overline{K}_i(n,k,d,d^{\perp}) - \overline{K}_{n-i}(n,k,d,d^{\perp})\right\}} \tag{18}$$

$$E(\mathcal{C}) \geq \sum_{i=1}^{n} q^{\max\left\{0, k-\overline{K}_{i-1}(n,k,d,d^{\perp}) - \overline{K}_{n-i}(n,k,d,d^{\perp})\right\}} \tag{19}$$

$$M(\mathcal{C}) \geq \frac{1}{q} \sum_{i=1}^{n} \left[\overline{K}_i(n,k,d,d^{\perp}) - \overline{K}_{i-1}(n,k,d,d^{\perp})\right] q^{\max\left\{0, k-\overline{K}_{i-1}(n,k,d,d^{\perp}) - \overline{K}_{n-i}(n,k,d,d^{\perp})\right\}} \tag{20}$$

Finally, the LDLP bounds, Eqs. (7) and (8), lead immediately to simple explicit upper bounds on the various complexity measures that apply to all codes with a given length and dimension.

Theorem 8. The complexity measures for the minimal trellis $\mathcal{T}(\mathcal{C})$ corresponding to any (n, k) code $\mathcal{C}$ are upper bounded by

$$s_{max}(\mathcal{C}) \leq \min(k, n-k) \tag{21}$$

$$e_{max}(\mathcal{C}) \leq \min(k, n-k+1) \tag{22}$$

$$\varepsilon(\mathcal{C}) \leq k(n-k+1) \tag{23}$$

$$V(\mathcal{C}) \leq \left[n + \frac{q+1}{q-1} - 2\min(k, n-k)\right] q^{\min(k,n-k)} - \frac{2}{q-1} \tag{24}$$

[3] A. Lafourcade and A. Vardy, "Lower Bounds on Trellis Complexity of Block Codes," submitted to *IEEE Trans. Inform. Theory*.

$$E(\mathcal{C}) \leq \left[n + \frac{2q}{q-1} - 2\min(k, n-k+1)\right] q^{\min(k,n-k+1)} - \frac{2q}{q-1} \tag{25}$$

$$M(\mathcal{C}) \leq \left[\frac{1}{q-1} + \max(0, 2k-n)\right] q^{\min(k,n-k)} - \frac{1}{q-1} \tag{26}$$

The inequality, Eq. (21), is the well-known Wolf bound [13]. Note that Eqs. (2) through (4) are tighter than Eqs. (23) through (25), except when Eqs. (21) and (22) are met with equality, in which case the bounds are the same.

III. Best and Worst Trellises

A. Uniform Comparability

In general, to determine which of two minimal trellises is less complex, we must first choose the relevant complexity measure. However, in some cases, one trellis may be simpler than another at every stage and depth with respect to all of the complexity measures simultaneously.

Definition 2. For two (n,k) codes $\mathcal{C}_1, \mathcal{C}_2$ having minimal trellises $\mathcal{T}(\mathcal{C}_1)$ and $\mathcal{T}(\mathcal{C}_2)$, we say that $\mathcal{T}(\mathcal{C}_1) \preceq \mathcal{T}(\mathcal{C}_2)$ if $p_i(\mathcal{C}_1) \geq p_i(\mathcal{C}_2)$ and $f_i(\mathcal{C}_1) \geq f_i(\mathcal{C}_2)$ for all i. If either $\mathcal{T}(\mathcal{C}_1) \preceq \mathcal{T}(\mathcal{C}_2)$ or $\mathcal{T}(\mathcal{C}_2) \preceq \mathcal{T}(\mathcal{C}_1)$, then the two trellises are uniformly comparable.

The binary relation $\preceq$ defines a partial ordering on any set of codes with the same length and dimension. If $\mathcal{T}(\mathcal{C}_1) \preceq \mathcal{T}(\mathcal{C}_2)$ and $\mathcal{T}(\mathcal{C}_2) \preceq \mathcal{T}(\mathcal{C}_1)$, then the two minimal trellises have equivalent complexity, though they may not have the same structure.

Note that if $\mathcal{T}(\mathcal{C}_1) \preceq \mathcal{T}(\mathcal{C}_2)$, then at every depth and stage, $\mathcal{T}(\mathcal{C}_1)$ has no more vertices or edges than $\mathcal{T}(\mathcal{C}_2)$, but the converse is not necessarily true. We define comparability in terms of past and future dimensions rather than edge and vertex dimensions because this gives a closer connection to the dimension/length profiles.

Theorem 9. If $\mathcal{T}(\mathcal{C}_1) \preceq \mathcal{T}(\mathcal{C}_2)$, then all of the following trellis complexity measures for $\mathcal{C}_1$ are upper bounded by those for $\mathcal{C}_2$:

(1) Maximum state complexity: $s_{\max}(\mathcal{C}_1) \leq s_{\max}(\mathcal{C}_2)$

(2) Total span lengths: $\varepsilon(\mathcal{C}_1) \leq \varepsilon(\mathcal{C}_2)$, $\nu(\mathcal{C}_1) \leq \nu(\mathcal{C}_2)$

(3) Total vertices: $V(\mathcal{C}_1) \leq V(\mathcal{C}_2)$

(4) Total edges: $E(\mathcal{C}_1) \leq E(\mathcal{C}_2)$

(5) Total number of path mergers: $M(\mathcal{C}_1) \leq M(\mathcal{C}_2)$

If two minimal trellises are not uniformly comparable, then the choice of the less complex trellis may depend on which of the complexity measures is used as the criterion.

Uniform comparability is a very strong property that is not guaranteed to exist between any two trellises. Our motivation for defining it and studying its consequences lies in the correspondingly strong results obtained for the problem of finding a minimal trellis in the first place, i.e., finding the least complex trellis that represents a fixed permutation of a fixed code. As shown by McEliece,[4] the minimal trellis is uniformly less complex at every stage and depth than any other trellis that represents the code.

[4] R. J. McEliece, "On The BCJR Trellis for Linear Block Codes," submitted to *IEEE Trans. Inform. Theory.*

We define four categories of best and worst minimal trellises based on uniform comparability:

Definition 3. For a fixed code $\mathcal{C}$, a permutation π^* and the corresponding minimal trellis $\mathcal{T}(\mathcal{C}\pi^*)$ are

(1) Uniformly efficient if $\mathcal{T}(\mathcal{C}\pi^*) \preceq \mathcal{T}(\mathcal{C}\pi)$ for all $\pi \in \mathcal{S}_n$

(2) Uniformly inefficient if $\mathcal{T}(\mathcal{C}\pi) \preceq \mathcal{T}(\mathcal{C}\pi^*)$ for all $\pi \in \mathcal{S}_n$

Definition 4. An $(n, k, d, d^\perp)$ code $\mathcal{C}^*$ and its corresponding minimal trellis $\mathcal{T}(\mathcal{C}^*)$ is

(2) Uniformly concise if $\mathcal{T}(\mathcal{C}^*) \preceq \mathcal{T}(\mathcal{C})$ for all $(n, k, d, d^\perp)$ codes $\mathcal{C}$

(2) Uniformly full if $\mathcal{T}(\mathcal{C}) \preceq \mathcal{T}(\mathcal{C}^*)$ for all (n, k) codes $\mathcal{C}$

If a minimal trellis is uniformly efficient or uniformly concise, we can drop the qualifier "minimal" and refer to it simply as a uniformly efficient trellis or a uniformly concise trellis, respectively. As shown later in Theorem 17, the two worst-case categories, uniformly inefficient and uniformly full, turn out to be equivalent.

The inclusion of $d^\perp$ in the above definition elucidates symmetries that are hidden by consideration of only n, k, and d. First, it preserves duality relationships, as we shall see below in Theorem 10. Second, from a practical point of view, d and $d^\perp$ have symmetric impact on the potential trellis complexity. There also appears to be a deep connection between d and $d^\perp$ for good codes: Often when d is large, $d^\perp$ must also be large, e.g., the extended Hamming codes and maximum distance separable (MDS) codes.

A direct consequence of [6, Theorem 1] is that uniform comparability of codes and their duals **are** equivalent:

Theorem 10. $\mathcal{T}(\mathcal{C}_1) \preceq \mathcal{T}(\mathcal{C}_2)$ if and only if $\mathcal{T}(\mathcal{C}_1^\perp) \preceq \mathcal{T}(\mathcal{C}_2^\perp)$. Consequently,

(1) A permutation π^* is uniformly efficient for $\mathcal{C}$ if and only if π^* is uniformly efficient for $\mathcal{C}^\perp$.

(2) A permutation π^* is uniformly inefficient for $\mathcal{C}$ if and only if π^* is uniformly inefficient for $\mathcal{C}^\perp$ [4, Theorem 1].

(3) $\mathcal{C}^*$ is uniformly concise if and only if $\mathcal{C}^{*\perp}$ is uniformly concise.

(4) $\mathcal{C}^*$ is uniformly full if and only if $\mathcal{C}^{*\perp}$ is uniformly full.

In the next sections, we show that the trellis complexity bounds derived in Section II.E are met exactly for the four categories of extremal minimal trellises.

B. Best Permutations

The following theorem shows that uniformly efficient trellises are those that achieve the DLP bounds in Eqs. (5), (6), and Theorem 6 with equality.

Theorem 11. A permutation π^* is uniformly efficient for a nondegenerate code $\mathcal{C}$ if and only if $\mathcal{C}\pi^*$ meets the DLP bounds, Eqs. (5) and (6), with equality, i.e.,

$$p_i(\mathcal{C}\pi^*) = K_i(\mathcal{C}) \text{ and } f_i(\mathcal{C}\pi^*) = K_{n-i}(\mathcal{C}) \text{ for all } i.$$

This guarantees that $\mathcal{C}\pi^*$ meets all of the lower bounds on complexity, Eqs. (9) through (14), with equality. Conversely, if $\mathcal{C}\pi^*$ meets any one of the lower bounds, Eqs. (11) through (13), with equality, then π^* is a uniformly efficient permutation for $\mathcal{C}$.

Theorem 11 shows that uniformly efficient permutations, which are defined in terms of trellis comparability, turn out to be the same as "efficient" [3] or "strictly optimum" [4] orderings, which were defined in terms of the DLP bounds. Note that a code may not have a permutation that meets these conditions.

A uniformly efficient permutation, if it exists, is not unique: If π^* is uniformly efficient for $\mathcal{C}$, then so is the reverse of π^*, and in fact the number of uniformly efficient permutations must be at least as large as the automorphism group of the code. There may also be different permutations that are uniformly efficient and produce distinct MSGMs for the code.

Even though uniform efficiency is a very strong property to require of a trellis, there are many codes that have uniformly efficient permutations. For example, the standard permutation of any Reed-Muller code is uniformly efficient [4, Theorem 2]. Additional examples of uniformly efficient codes are given in Section III.C, which lists trellises that are both uniformly efficient and uniformly concise.

We now give some theoretical results that impose necessary conditions on uniformly efficient permutations.

Theorem 12. Suppose $\mathcal{C}$ is a code that has some uniformly efficient permutation π^*. Then for any i, j such that $i + j \leq n$,

$$K_{i+j}(\mathcal{C}) \geq K_i(\mathcal{C}) + K_j(\mathcal{C})$$

Theorem 13. If π^* is a uniformly efficient permutation for an $(n, k, d, d^\perp)$ code $\mathcal{C}$, then $\mathcal{C}\pi^*$ contains codewords of the form $X^d 0^{n-d}$, $0^{n-d}X^d$, and $\mathcal{C}^\perp\pi^*$ contains codewords of the form $X^{d^\perp}0^{n-d^\perp}$, $0^{n-d^\perp}X^{d^\perp}$, where 0^j denotes j consecutive zeros, and X^j denotes some sequence of j nonzero symbols from $GF(q)$.

Corollary 1. If $\mathcal{C}$ is a binary $(n, k, d, d^\perp)$ code that has some uniformly efficient permutation π^*, then $\min(d, d^\perp)$ must be even.

By Corollary 1, the (23,12,7,8) Golay code has no uniformly efficient permutation; neither does the $(2^m - 1, 2^m - m - 1, 3, 2^{m-1})$ Hamming code for any $m \geq 3$. Consequently, no nontrivial perfect binary linear code has a uniformly efficient permutation.

Although many codes lack uniformly efficient permutations, there may be some permutation that simultaneously minimizes all of the trellis complexity measures. For example, the (7,4) Hamming code is sufficiently small that we can verify by exhaustive search that there are permutations that are optimal with respect to all of the complexity measures despite not being uniformly efficient.

For self-dual codes, [6, Theorem 3] tells us that there is always a single permutation that simultaneously minimizes E, V, and M. We suspect that not every code has a permutation that simultaneously minimizes all of the complexity measures, though we do not yet know of an example that confirms this conjecture.

C. Best Codes

Uniformly concise codes are optimum in a rather strong sense. Not only do they have an efficient permutation, but they also minimize all of the trellis complexity measures compared to all codes with the same parameters. The following theorem shows that codes that achieve the bounds in Theorems 5 and 7 with equality are uniformly concise.

Theorem 14. An $(n, k, d, d^\perp)$ code $\mathcal{C}^*$ is uniformly concise if the dimensions of its past and future subcodes meet the bounds in Theorem 5 with equality, i.e.,

$$p_i(\mathcal{C}^*) = \overline{K}_i(n, k, d, d^\perp) \text{ and } f_i(\mathcal{C}^*) = \overline{K}_{n-i}(n, k, d, d^\perp) \text{ for all } i$$

In this case, $\mathcal{C}^*$ meets all of the lower bounds on complexity, Eqs. (15) through (20), with equality. Conversely, if $\mathcal{T}(\mathcal{C}^*)$ meets any of the bounds of Eqs. (17) through (19) with equality, then $\mathcal{C}^*$ is uniformly concise.

Table 1 lists known uniformly concise binary codes. In each case, the complexity values listed are the lowest possible for any code with the same parameters. From Theorem 10, the dual of each code is also uniformly concise. Generator matrices for many of these codes are given in Kiely et al.[5] All of the rate 1/2 codes in the table are either self-dual or have duals that are permuted versions of the original code.

Theorem 15. All $(2^m, m+1, 2^{m-1}, 4)$ first-order Reed–Muller codes and their duals, the $(2^m, 2^m - m - 1, 4, 2^{m-1})$ extended Hamming codes, are uniformly concise.

There are also examples of code parameters $(n, k, d, d^{\perp})$ for which no uniformly concise trellis can exist. The $\mathcal{R}(r, m)$ Reed–Muller codes when $(m = 6, r = 2, 3)$, $(m = 7, r = 2, 3, 4)$ are codes that do not meet the UDLP bounds. This is established by comparing the UDLP bounds to the known optimal permutations for the Reed–Muller codes.

Results such as the examples above and Theorems 12 and 13 illustrate that in many instances the UDLP bounds on complexity are not tight. An area of further research is to produce tighter bounds on trellis complexity based on the code parameters $(n, k, d, d^{\perp})$.

D. Worst Minimal Trellises

The following theorems show that uniformly inefficient and uniformly full minimal trellises are the same as the trellises that achieve the LDLP bounds with equality.

Theorem 16. An (n, k) code $\mathcal{C}$ is uniformly full if and only if the dimensions of the past and future subcodes of $\mathcal{C}$ meet the bounds of Eqs. (7) and (8) with equality, i.e.,

$$p_i(\mathcal{C}) = \max(0, k - n + i) \text{ and } f_i(\mathcal{C}) = \max(0, k - i) \text{ for all } i$$

In this case, $\mathcal{C}$ meets all of the upper bounds on complexity, Eqs. (21) through (26), with equality. Conversely, if $\mathcal{C}$ meets any one of the upper bounds, Eqs. (23) through (25), with equality, then $\mathcal{C}$ is uniformly full.

Theorem 17. A minimal trellis $\mathcal{T}(\mathcal{C}\pi^*)$ is uniformly full if and only if π^* is a uniformly inefficient permutation of $\mathcal{C}$.

Many codes have uniformly inefficient trellises in their standard permutations. For example, the minimal trellises for all cyclic, extended cyclic, and shortened cyclic codes are uniformly inefficient [5,7]. However, not every code has a uniformly inefficient permutation.

Additional examples of codes with uniformly inefficient trellises are given in the following two theorems.

Theorem 18. A self-dual code always has a uniformly inefficient permutation.

Theorem 19. If and only if a code is maximum distance separable (MDS), every permutation π is uniformly inefficient and the corresponding trellis complexity measures equal the upper bounds in Eqs. (21) through (26).

[5] Kiely et al., op cit.

Table 1. Some known uniformly concise binary codes.[a]

Code (parameters)	E	V	M	$s_{\max}$	$e_{\max}$	ε
Minimal span codes[b] $(d+(k-1)\left\lceil \frac{d}{2} \right\rceil, k, d, 2)$	$2kd$	$2+2k(d-1)$	$2k-1$	$\min\{2,$ $d-2\}$	2	kd
Dual	$4k(d-2)+4$	$2+2k(d-1)$	$2k(d-3)+3$	2	2	$k(d-2)+n$
Reed–Muller[c] $\mathcal{R}(1,m)$ $(2^m, 1+m, 2^{m-1}, 4)$	$(2^{2m+1}+2^4)/3$ -4	$(2^{2m+1}+2^4)/3$ $-3(2^{m-1})-2$	$3(2^{m-1})-1$	m	m	$(m-1)2^m$ $+2$
Extended Hamming Dual	$(2^{2m+2}+2^5)/3$ $-4-3(2^{m+1})$	$(2^{2m+1}+2^4)/3$ $-3(2^{m-1})-2$	$(2^{2m+1}+2^4)/3$ $-9(2^{m-1})-1$	m	$m+1$	$m(2^m-2)$
Extended Golay $\mathcal{G}_{24}$ $(24, 12, 8, 8)$ self-dual	3580	2686	895	9	9	136
Reed–Muller $\mathcal{R}(2,6)$ $(32, 16, 8, 8)$ self-dual	6396	4798	1599	9	9	202
Quadratic residue $(48, 24, 12, 12)$ self-dual	860156	645118	115039	16	16	502
$(10, 5, 4, 4)$[d] Formally self-dual	60	46	15	3	3	24
$(12, 6, 4, 4)$[d] Formally self-dual	76	58	19	3	3	30
$(16, 4, 8, 2)$[d]	88	78	11	3	3	36
Dual	132	78	55	3	4	44
$(20, 6, 8, 4)$[d]	236	206	31	4	4	66
Dual	348	206	143	4	5	74
$(24, 7, 8, 4)$[d]	300	262	39	4	4	82
Dual	444	262	183	4	5	92
$(24, 8, 8, 4)$[d]	364	302	63	5	5	86
Dual	476	302	175	5	5	94
$(40, 7, 16, 4)$[d]	940	878	63	5	5	170
Dual	1628	878	751	5	6	196
$\mathcal{R}(1,3) \oplus \mathcal{R}(1,3)$ $(16, 8, 4, 4)$ self-dual	88	67	22	3	3	36
$\mathcal{G}_{24} \oplus \mathcal{G}_{24}$ $(48, 24, 8, 8)$ self-dual	7160	5371	1790	9	9	272

[a] Codes are grouped with their duals, which are also uniformly concise.

[b] $d > 2$, $k \leq 3$.

[c] Complexity expressions for first-order Reed-Muller and extended Hamming codes are valid for $m \geq 3$, except $e_{max} = 3$ when $m = 3$.

[d] See Kiely et al., op cit.

This theorem follows from the fact that a code is MDS if and only if every subset of k columns of its generator matrix is linearly independent. A peculiar consequence of Theorem 19 is that every permutation of an MDS code is also uniformly efficient, as noted by Forney [3]. This observation emphasizes that uniform efficiency is only a relative measure of trellis complexity.

IV. Conclusion

In this article, we extended the analysis of [6] to consider permutations of a code that minimize the complexity of a trellis representation that can be used for encoding or decoding. The analysis for a fixed code generalizes naturally to similar results for codes allowed to vary over a domain of optimization.

We identified two useful domains, the set of permutations of a given code and the set of all codes with given code parameters. Within each domain, we defined uniformly best and worst minimal trellises that are guaranteed to simultaneously minimize or maximize all of the complexity measures. We showed that it is easy to generalize the bounds on maximum state complexity derived by other authors from the dimension/length profile of a code to similar bounds on all the complexity measures over each optimization domain. Furthermore, if a minimal trellis attains the bounds for some of the complexity measures, it must necessarily be uniformly extremal, but this is not true for the simpler measures of maximum state or edge dimension considered by other authors. This lends further credence to the argument that a measure of total complexity (such as the total number of edges) is more useful than a measure of maximum complexity [10].[6]

Unlike the case of a fixed permutation of a given code, uniformly best and worst minimal trellises are not guaranteed to exist within the larger domains of optimization. However, we demonstrated the usefulness of the concepts by presenting several examples of uniformly best trellises, most notably the optimum permutation of the (48,24) quadratic residue code [2], heretofore unknown. Conversely, by deriving some necessary existence conditions, we also identified some cases for which uniformly extremal minimal trellises cannot exist.

We showed that the useful relationships between the trellis complexity of a code and that of its dual developed in [6] extend naturally to optimizations over larger code domains. This approach yields many of the same results obtained by other authors for dimension/length profiles or generalized Hamming weights, but it emphasizes that all the duality results stem from fundamental minimal trellis relationships valid for a fixed permutation of a code. In fact, we have argued that the symmetry of the constraints imposed by the code and its dual on trellis complexity is so fundamental that the minimum distance of the dual code should be included as one of the intrinsic code parameters that limits achievable complexity.

Acknowledgment

A portion of W. Lin's contribution was sponsored by AFOSR grant no. F4960-94-1-005. Part of this work was presented at the 32nd Annual Allerton Conference on Communication, Control, and Computing in Allerton, Illinois, in October 1994.

References

[1] A. E. Brouwer and T. Verhoeff, "An Updated Table of Minimum-Distance Bounds for Binary Linear Codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 662–677, 1993.

[2] S. Dolinar, L. Ekroot, A. Kiely, W. Lin, and R. J. McEliece, "The Permutation Trellis Complexity of Linear Block Codes," *Proc. 32nd Annual Allerton Conference on Communication, Control, and Computing*, Allerton, Illinois, October 1994.

[6] McEliece, op cit.

[3] G. D. Forney, "Dimension/Length Profiles and Trellis Complexity of Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 6., pp. 1741–1752, November 1994.

[4] T. Kasami, T. Takata, T. Fujiwara, and S. Lin "On the Optimum Bit Orders With Respect to the State Complexity of Trellis Diagrams for Binary Linear Codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 242–245, 1993.

[5] T. Kasami, T. Takata, T. Fujiwara, and S. Lin, "On Complexity of Trellis Structure of Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1057–1064, 1993.

[6] A. B. Kiely, S. Dolinar, R. J. McEliece, L. Ekroot, and W. Lin, "Minimal Trellises for Linear Block Codes and Their Duals," *The Telecommunications and Data Acquisition Progress Report 42-121, January-March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 148–158, May 15, 1995.

[7] F. R. Kschischang and V. Sorokine, "On the Trellis Structure of Block Codes," *Proc. 1994 IEEE International Symposium on Information Theory*, Trondheim, Norway, p. 337, June 27–July 1, 1994.

[8] A. Lafourcade and A. Vardy, "Asymptotically Good Codes Have Infinite Trellis Complexity," *IEEE Trans. Inform. Theory*, vol. 41, no. 2, pp. 555–559, March 1995.

[9] J. L. Massey, "Foundations and Methods of Channel Coding," in *Proc. of the Int. Conf. on Info. Theory and Systems*, vol. 65, NTG-Fachberichte, September 1978.

[10] R. J. McEliece, "The Viterbi Decoding Complexity of Linear Block Codes," *Proc. 1994 IEEE International Symposium on Information Theory*, Trondheim, Norway, p. 341, June 27–July 1, 1994.

[11] A. Vardy and Y. Be'ery, "Maximum-Likelihood Soft Decision Decoding of BCH Codes," *IEEE Trans. Inform. Theory*, vol. 40, pp. 546–554, 1994.

[12] V. K. Wei, "Generalized Hamming Weights for Linear Codes," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1412–1418, 1991.

[13] J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, 1978.

N95- 32235

TDA Progress Report 42-121

May 15, 1995

Residual and Suppressed-Carrier Arraying Techniques for Deep-Space Communications

M. Shihabi, B. Shah, S. Hinedi, and S. Million
Communications Systems Research Section

Three techniques that use carrier information from multiple antennas to enhance carrier acquisition and tracking are presented. These techniques in combination with baseband combining are analyzed and simulated for residual and suppressed-carrier modulation. It is shown that the carrier arraying using a single carrier loop technique can acquire and track the carrier even when any single antenna in the array cannot do so by itself. The carrier aiding and carrier arraying using multiple carrier loop techniques, on the other hand, are shown to lock on the carrier only when one of the array elements has sufficient margin to acquire the carrier on its own.

I. Introduction

Combining or arraying signals from multiple antennas has the advantage of increasing the signal-to-noise ratio (SNR) of the received signal. For example, it is well known [1] that ideally the SNR of the combined signal is the sum of the SNRs corresponding to the individual antennas. Practically, the achievable gain depends on the type of scheme being implemented as well as on the characteristics of the received signal. This article is mainly concerned with three similar techniques that first use information from multiple antennas to acquire and track the carrier, and then use baseband combining (BBC) [2] on the carrier demodulated signals to demodulate the subcarrier and detect the symbols. The three techniques, which work in conjunction with BBC, are carrier arraying using a single carrier loop, carrier arraying using multiple carrier loops, and carrier aiding. As will be shown shortly, the second and third techniques are usable for both residual and suppressed-carrier modulation. The carrier arraying with a single carrier loop followed by the baseband combining technique, however, is not practical for suppressed-carrier modulation. Practical implementations that demodulate an arrayed suppressed-carrier signal using a single carrier loop are the full-spectrum combining and/or complex symbol combining techniques described in [3].

The main difference between the techniques under consideration is that the first, carrier arraying using a single carrier loop, does not require any single antenna in the array to acquire and track the carrier by itself. The other two techniques, on the other hand, require at least one antenna in the array to lock the carrier on its own. The use of these techniques is best illustrated through an example. Consider an array of one 70-m and two standard (STD) 34-m antennas operating at S-band frequencies (2.2–2.3 GHz) [4]. A typical radio frequency spectrum of the received signal is shown in Fig. 1 in the absence of noise.

C-3

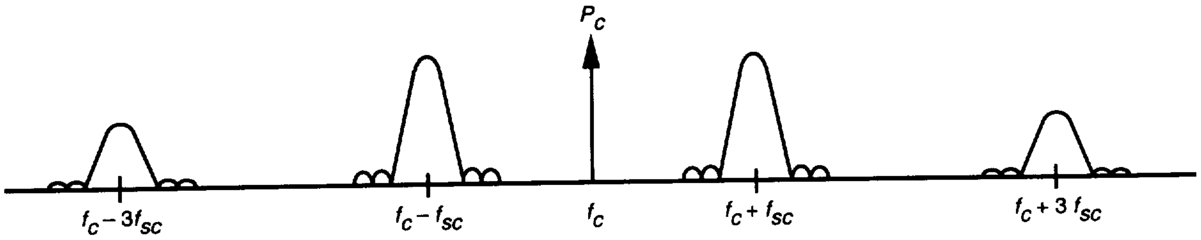


Fig. 1. PCM/PSK/PM square-wave subcarrier signal model.

Assume that P_T/N_0, the ratio of the total received power to the one-sided noise power spectral density (PSD) level, at the 70-m is 15 dB-Hz; the modulation index is 58 deg, and the symbol rate is 20 symbols per second (sps). Then, since the ratio of P_T/N_0 at the STD 34-m to that at the 70-m is $\gamma = 0.17$ [1], the $(P_T/N_0)_{34-m} = 7.3$ dB-Hz. (The ratios of P_T/N_0 of typical 34-m antennas in the DSN to the P_T/N_0 of the 70-m are shown in Table 1.) The corresponding P_C/N_0 are 9.5 dB-Hz for the 70-m and 1.8 dB-Hz for the 34-m. For this scenario, suppose that the minimum bandwidth required to track the carrier is 1 Hz, and the minimum loop SNR needed to reliably track the residual carrier is 7 dB [5]. Then the 70-m antenna with a carrier loop SNR of 9.5 dB can acquire the carrier, but the two 34-m antennas with loop SNRs of 1.8 dB are unable to do so. Applying the techniques described in this article, however, still enables us to make use of the information at the smaller antennas.

Table 1. Gamma factors for DSN antennas.

Antenna size	Frequency band	γ_i
70-m	S-band	1.00
34-m STD	S-band	0.17
34-m HEF	S-band	0.07
70-m	X-band	1.00
34-m STD	X-band	0.13
34-m HEF	X-band	0.26

Let us discuss the techniques one at a time. Carrier aiding is shown in Fig. 2. Here the 70-m (or master) antenna in the array first locks the carrier and then passes its reference to the other (34-m) antennas. At the 34-m antenna, the received signal is first delayed to time align it with the 70-m signal, then open-loop downconverted to baseband using the 70-m reference, and subsequently coherently demodulated using a baseband phase-locked loop (PLL). (Note that we arbitrarily assume the signal at the 34-m antenna to be delayed relative to the 70-m antenna.) When the antennas in the array are colocated, the baseband PLLs can operate at bandwidths much narrower than otherwise possible, because most of the signal dynamics are removed by the master reference signal in the downconversion to baseband. In the case of the example given, the baseband PLL would be able to use a bandwidth much narrower than 1 Hz, because it must only track the residual Doppler between the 70-m antenna and 34-m antennas. The narrow bandwidth results in an increased loop SNR, which allows the 34-m antennas to lock the carrier. In this example, if the modulation index were changed to 90 deg so that the carrier is fully suppressed, the technique in Fig. 2 could still be used by using a Costas loop instead of a PLL to track the carrier.

Note that carrier aiding is only useful when at least one antenna is able to acquire the carrier on its own. If this requirement is not met, a different technique, such as carrier arraying using a single carrier loop, is needed. We begin with the implementation shown in Fig. 3. Here the time-aligned residual carrier

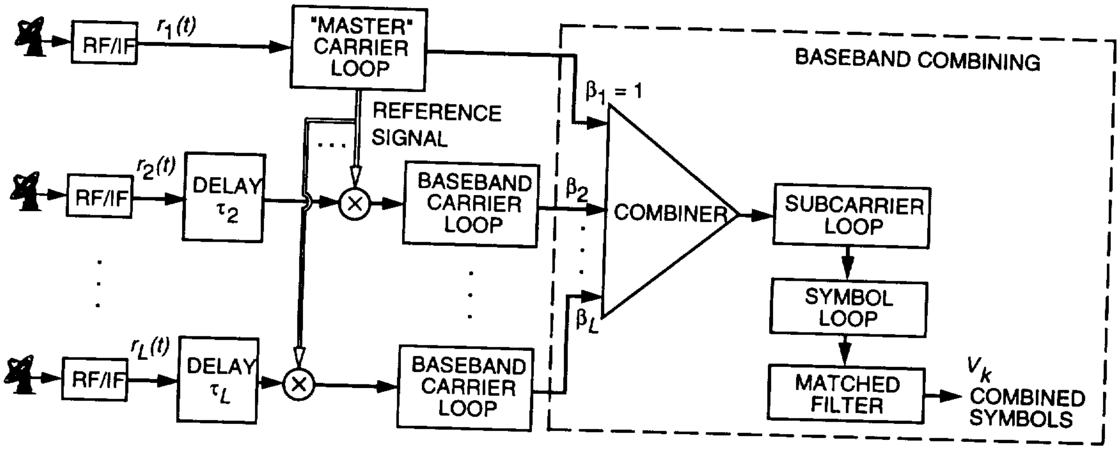


Fig. 2. Carrier aiding/BBC: system overview.

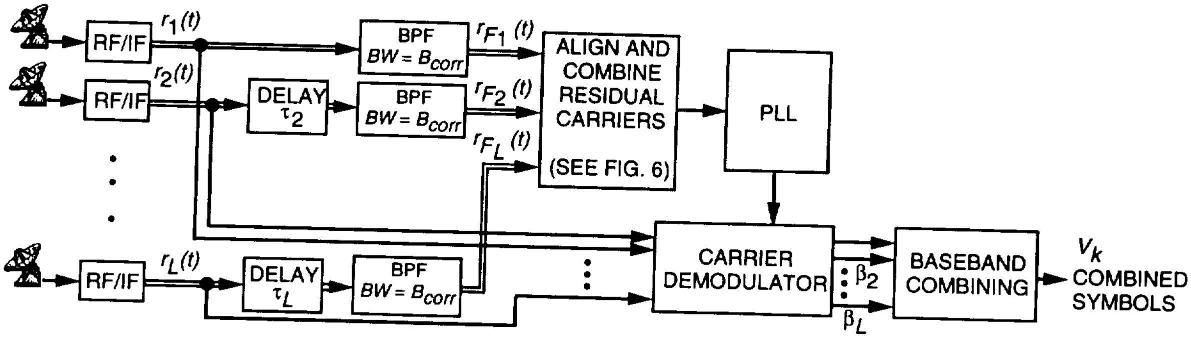


Fig. 3. CA/single PLL: system overview.

component at each antenna is filtered and transmitted to a central location, phase aligned, combined, and input to a single carrier loop. As a result, for a given bandwidth, the loop will lock the carrier provided the combined signal has sufficient P_C/N_0. Ideally, the combined P_C/N_0 is the sum of P_C/N_0 at the individual antennas. Consider the same scenario as before but with the 70-m antenna replaced by two additional 34-m STD antennas. Under this scenario, carrier aiding cannot be implemented using a 1-Hz loop, as none of the four 34-m antennas has sufficient P_C/N_0 to lock the carrier. However, carrier arraying using a single PLL with a 1-Hz bandwidth can be implemented since the combined P_C/N_0 of the four 34-m antennas is 7.8 dB-Hz. When there is no residual component at $f = f_c$ in Fig. 1, the implementation shown in Fig. 3 cannot be used without modification. The simplest way to handle this case would be to widen the bandwidth of the bandpass filter (BPF) in Fig. 3 so that it passes the first N harmonics of the telemetry signal. The harmonics from each antenna would then be transmitted to a central location, aligned, combined, and tracked by replacing the PLL in Fig. 3 with a Costas loop. Note that the modified implementation is impractical because it requires the signal to be combined twice: first, as just described, for carrier tracking and then for baseband demodulation. A more practical implementation along these lines is full-spectrum combining (FSC) [3], where the signal is combined at IF and then tracked using a single receiver. An altogether different approach that also uses a single carrier loop but multiple subcarrier and symbol loops is complex symbol combining (CSC) [3].

Finally, we turn to the carrier-arraying with multiple PLLs technique shown in Fig. 4. As will shortly be shown, this technique can be viewed as a hybrid of the techniques in Figs. 2 and 3. Here, as in Fig. 2, the received signal at each antenna (except the master) is first downconverted to baseband using the master antenna carrier reference and coherently tracked using a baseband PLL. As before, due to rate aiding by the master, the baseband PLL operates at narrower bandwidths and a higher loop SNR than

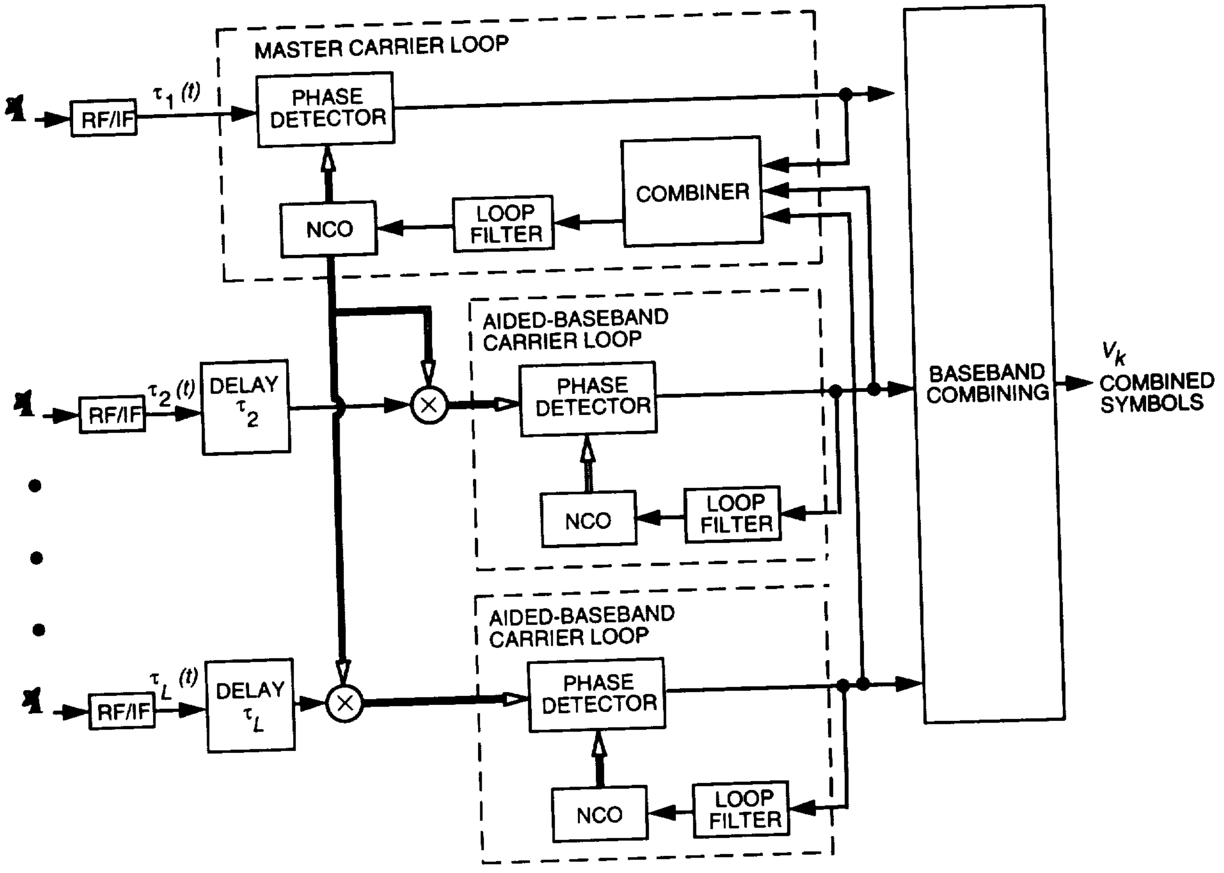


Fig. 4. CA/multiple PLLs: system overview.

in the absence of rate aiding. However, now the master antenna also benefits, because the error signal from each of the other antennas is added to its error signal. Hence, when all the loops are tracking, the master PLL also operates at a loop SNR that is improved. In the upper limit, when all the error signals add coherently, the loop SNR of the master is equal to the ideal loop SNR of the carrier-arraying with the single PLL technique in Fig. 3. In practice, we can expect the performance of this scheme to be better than carrier aiding but not as good as carrier-arraying with a single PLL. Note that if the master cannot acquire the signal on its own, it cannot rate aid the other antennas, and this scheme is unusable. In the examples considered earlier, this technique would work well for an array of one 70-m and two STD 34-m antennas, but would not be implementable for an array of four STD 34-m antennas that cannot lock individually. This scheme can be used for suppressed-carrier modulation by replacing the PLL with the Costas loop.

In this article, the tracking performance of all three techniques is measured in terms of SNR degradation and symbol SNR loss. Both performance measures have been explained in detail earlier [3]. Briefly, SNR degradation is defined as the ratio of the SNR at the matched filter output in the presence of nonideal synchronization to the SNR in the presence of ideal synchronization. Symbol SNR loss is defined as the additional symbol SNR needed by a system with synchronization errors to achieve the same symbol error rate (SER) as one with no synchronization errors. In the following sections, analytical expressions are derived to describe the performances of carrier arraying using a single PLL and carrier aiding. The performances of these systems were also obtained via simulations and seen to agree closely with the theory. Performance for carrier arraying using multiple PLLs is obtained via simulation only.

II. Single Receiver Performance

We begin with the performance of a single receiver, as it is the basis for the analysis of the schemes in Figs. 1 through 3. In deep-space communications, the downlink symbols are first modulated onto a square-wave subcarrier that, in turn, modulates an RF carrier [6]. As shown in Fig. 1, this has the advantage of transmitting a residual carrier component whose frequency does not coincide with the data spectrum. In general, the downlink deep-space signal can be represented as [6]

$$r(t) = \sqrt{2P_T} \sin\left[\omega_c t + \delta\ d(t)\ \mathrm{Sqr}(\omega_{sc} t + \theta_{sc}) + \theta_c\right] + n(t) \tag{1}$$

where P_T is the total received power in watts (W), and ω_c and θ_c are the carrier angular frequency in radians per second (rad/s) and phase in rad, respectively. The $\mathrm{Sqr}(\omega_{sc}t + \theta_{sc}) = \mathrm{sgn}(\sin(\omega_{sc}t + \theta_{sc}))$ is the square-wave subcarrier with angular frequency ω_{sc} rad/s and phase θ_{sc} rad. The signum function sgn (x) equals $+1$ when its argument is positive and -1 otherwise. The modulation index, δ, ranges from 0 to $\pi/2$. The carrier power $P_C = P_T \cos^2 \delta$, and the data power $P_D = P_T \sin^2 \delta$. When $\delta = \pi/2$, the signal is "suppressed-carrier" modulated. In this case, the downlink signal spectrum is as given in Fig. 1, but without the residual carrier at f_c. The symbol stream, $d(t)$, is given by

$$d(t) = \sum_{k=-\infty}^{\infty} d_k p(t - kT) \tag{2}$$

where d_k is the ± 1 binary data for the kth symbol and T is the symbol period in seconds. The baseband pulse, $p(t)$, is unity in $[0,T)$ and zero otherwise. The bandpass noise, $n(t)$, can be written as

$$n(t) = \sqrt{2} n_c(t) \cos(\omega_c t) - \sqrt{2} n_s(t) \sin(\omega_c t) \tag{3}$$

where $n_c(t)$ and $n_s(t)$ are statistically independent, stationary, band-limited, white Gaussian low-pass noise processes with one-sided PSD level N_0 (W/Hz) and one-sided bandwidth W_n (Hz).

As shown in Fig. 5, the deep-space signal is demodulated using a receiving chain consisting of a carrier-tracking loop, a subcarrier-tracking loop, and a symbol-synchronizer loop. If $\delta < \pi/2$, a PLL is used for carrier tracking. When $\delta = \pi/2$, however, carrier tracking is achieved using a Costas loop. Computation of the degradation and loss begins with the expression for the soft symbols, v_k, in Fig. 5. From [1,6],

$$v_k = \begin{cases} \sqrt{P_D} C_c C_{sc} d_k + n_k & d_k = d_{k-1} \\ \sqrt{P_D} C_c C_{sc} \left(1 - \dfrac{|\phi_{sy}|}{\pi}\right) d_k + n_k & d_k \neq d_{k-1} \end{cases} \tag{4}$$

where the noise n_k is Gaussian with variance $\sigma_n^2 = N_0/(2T)$. The signal reduction functions C_c and C_{sc} are due to imperfect carrier and subcarrier synchronization and are given as [1,6]

$$C_c = \cos \phi_c \tag{5}$$

$$C_{sc} = 1 - \frac{2}{\pi} |\phi_{sc}| \tag{6}$$

where ϕ_c and ϕ_{sc} denote the carrier and subcarrier phase tracking errors, respectively. The symbol timing error, ϕ_{sy}, which affects the output only when there is a symbol transition (i.e., when $d_k \neq d_{k+1}$), reduces

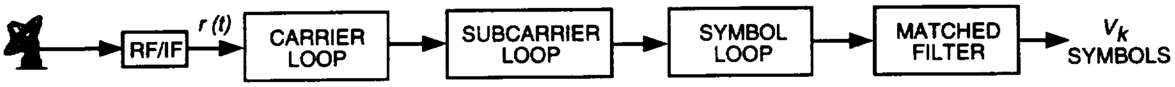


Fig. 5. Single receiver: overview.

the signal amplitude by $1 - (|\phi_{sy}|/\pi)$. Ideally, $\phi_c = \phi_{sc} = \phi_{sy} = 0$ and Eq. (4) reduces to the familiar matched filter output $v_{k,ideal} = \sqrt{P_D}d_k + n_k$, as expected. In writing Eq. (4), it is assumed that the carrier, subcarrier, and symbol loop bandwidths are much smaller than the symbol rate so that the phase errors ϕ_c, ϕ_{sc}, and ϕ_{sy} can be modeled as constant over several symbols.

Throughout this article, the density function of ϕ_c is assumed to be Tikhonov,[1] that is,

$$p_c(\phi_c) = \begin{cases} \dfrac{\exp(\rho_c \cos\phi_c)}{2\pi I_0(\rho_c)} & |\phi_c| \le \pi \quad \text{residual-carrier case} \\ \dfrac{\exp((1/4)\rho_c \cos 2\phi_c)}{\pi I_0((1/4)\rho_c)} & |\phi_c| \le \dfrac{\pi}{2} \quad \text{suppressed-carrier case} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $I_k(x) = 1/\pi \int_0^\pi e^{x\cos\theta}\cos(k\theta)d\theta$ is the modified Bessel function of order k, and ρ_c is the carrier loop SNR. From [7],

$$\rho_c = \begin{cases} \dfrac{P_C/N_0}{B_c} & \text{residual-carrier case} \\ \dfrac{P_D/N_0}{B_c}\left(1 + \dfrac{1}{2E_s/N_0}\right)^{-1} & \text{suppressed-carrier case} \end{cases} \tag{8}$$

where the symbol SNR $E_s/N_0 = P_D T/N_0$ and B_c Hz is the carrier loop bandwidth. The subcarrier and symbol densities, $p_{sc}(\phi_{sc})$ and $p_{sy}(\phi_{sy})$, are assumed to be Gaussian. Hence,

$$p_i(\phi_i) = \frac{\exp(-\phi_i^2/2\sigma_i^2)}{\sqrt{2\pi}\sigma_i}, \qquad i = sc, sy \tag{9}$$

where σ_{sc}^2 is the reciprocal of the subcarrier loop SNR, ρ_{sc}, and σ_{sy}^2 is the reciprocal of the symbol loop SNR, ρ_{sy}. The subcarrier [7] and symbol [8] loop SNRs are respectively given as

$$\rho_{sc} = \left(\frac{2}{\pi}\right)^2 \frac{P_D/N_0}{W_{sc}B_{sc}}\left(1 + \frac{1}{2E_s/N_0}\right)^{-1} \tag{10}$$

$$\rho_{sy} = \frac{P_D/N_0}{2\pi^2 W_{sy}B_{sy}}$$

$$\times \frac{\left(\operatorname{erf}\left(\sqrt{E_s/N_0}\right) - (W_{sy}/(2\sqrt{\pi}))\sqrt{E_s/N_0}\exp\left(-(E_s/N_0)\right)\right)^2}{\left(1 + (E_s/N_0)(W_{sy}/2) - (W_{sy}/2)\left[(1/\sqrt{\pi})\exp\left(-(E_s/N_0)\right) + \sqrt{E_s/N_0}\operatorname{erf}\left(\sqrt{E_s/N_0}\right)\right]^2\right)} \tag{11}$$

[1] It is assumed that the Costas loop locks at zero phase error. The π lock point can be handled by an appropriate transformation [6].

where $\mathrm{erf}(x) = (2/\sqrt{\pi})\int_0^x \exp(-v^2)dv$ is the error function, and B_{sc} and B_{sy} (in Hz) denote the single-sided subcarrier and symbol loop bandwidths, respectively. The parameters W_{sc} and W_{sy}, which denote the subcarrier and symbol window, are unitless and limited to $(0, 1]$.

A useful quantity needed to compute degradation and loss is the symbol SNR conditioned on ϕ_c, ϕ_{sc}, and ϕ_{sy}. The conditional symbol SNR, denoted by $SSNR'$, is defined as the square of the conditional mean of v_k divided by the conditional variance of v_k, i.e.,

$$\begin{aligned} SSNR' &= \frac{\overline{(v_k/\phi_c, \phi_{sc}, \phi_{sy})}^2}{\sigma_n^2} \\ &= \begin{cases} \dfrac{2P_D T}{N_0} C_c^2 C_{sc}^2 & d_k = d_{k-1} \\ \dfrac{2P_D T}{N_0} C_c^2 C_{sc}^2 \left(1 - \dfrac{|\phi_{sy}|}{\pi}\right)^2 & d_k \neq d_{k-1} \end{cases} \end{aligned} \tag{12}$$

where $\overline{(x/y)}$ denotes the statistical expectation of x conditioned on y, and v_k and σ_n^2 are defined earlier.

A. Degradation

The symbol SNR degradation is defined as the symbol SNR at the matched filter output in the presence of imperfect synchronization divided by the ideal matched filter output SNR. The nonideal symbol SNR, denoted as $SSNR$, is found by first averaging Eq. (12) over the symbol transition probability and then over the carrier, subcarrier, and symbol phases. It can be shown that [1]

$$SSNR = \frac{2P_D T}{N_0} \overline{C_c^2}\, \overline{C_{sc}^2}\, \overline{C_{sy}^2} \tag{13}$$

where the signal amplitude reduction due to symbol timing errors is denoted C_{sy} and given as

$$C_{sy} = 1 - \frac{|\phi_{sy}|}{2\pi} \tag{14}$$

for a transition probability of one-half. The average of the signal reduction functions is [1]

$$\overline{C_c^2} = \begin{cases} \dfrac{1}{2}\left[1 + \dfrac{I_2(\rho_c)}{I_0(\rho_c)}\right] & \text{residual-carrier case} \\ \dfrac{1}{2}\left[1 + \dfrac{I_1((1/4)\rho_c)}{I_0((1/4)\rho_c)}\right] & \text{suppressed-carrier case} \end{cases} \tag{15}$$

$$\overline{C_{sc}^2} = 1 - \sqrt{\frac{32}{\pi^3}}\frac{1}{\sqrt{\rho_{sc}}} + \frac{4}{\pi^2}\frac{1}{\rho_{sc}} \tag{16}$$

$$\overline{C_{sy}^2} = 1 - \sqrt{\frac{2}{\pi^3}}\frac{1}{\sqrt{\rho_{sy}}} + \frac{1}{4\pi^2}\frac{1}{\rho_{sy}} \tag{17}$$

Ideally, when there are no phase errors (i.e., when $\rho_c = \rho_{sc} = \rho_{sy} = \infty$), $\overline{C_c^2} = \overline{C_{sc}^2} = \overline{C_{sy}^2} = 1$ and Eq. (13) reduces to $SSNR_{ideal} = 2P_D T/N_0$, as expected. The degradation, D, for a single antenna is thus given by

$$D = 10\log_{10}\left(\frac{SSNR}{SSNR_{ideal}}\right) = \log_{10}\overline{C_c^2}\;\overline{C_{sc}^2}\;\overline{C_{sy}^2} \tag{18}$$

Note that the degradation defined in this way is a negative number.

B. Loss

The SER for the single receiver in Fig. 5, denoted $P_s(E)$, is defined as [2,3]

$$P_s(E) = \int\limits_{\phi_c}\int\limits_{\phi_{sc}}\int\limits_{\phi_{sy}} P_s'(E)p_c(\phi_c)p_{sc}(\phi_{sc})p_{sy}(\phi_{sy})d\phi_{sy}d\phi_{sc}d\phi_c = f\left(\sqrt{\frac{E_s}{N_0}}\right) \tag{19}$$

where $f(\cdot)$ is the functional relationship between SER and $\sqrt{E_s/N_0}$. The quantity $P_s'(E)$ is the SER conditioned on the phase errors ϕ_c, ϕ_{sc}, and ϕ_{sy}. Following similar steps as in [9], the conditional SER can be shown to be

$$P_s'(E) = \frac{1}{4}\text{ erfc }\left(\sqrt{SSNR' \text{ when } d_k \neq d_{k-1}}\right) + \frac{1}{4}\text{ erfc }\left(\sqrt{SSNR' \text{ when } d_k = d_{k-1}}\right) \tag{20}$$

where

$$\text{erfc }(x) = \frac{2}{\sqrt{\pi}}\int\limits_x^\infty \exp(-v^2)dv = 1 - \text{erf}(x) \tag{21}$$

is the complementary error function. Substituting Eq. (12) for $SSNR'$ in Eq. (20) yields

$$P_s'(E) = \frac{1}{4}\text{ erfc }\left[\sqrt{\frac{E_s}{N_0}}C_c C_{sc}\left(1 - \frac{|\phi_{sy}|}{\pi}\right)\right] + \frac{1}{4}\text{ erfc }\left[\sqrt{\frac{E_s}{N_0}}C_c C_{sc}\right] \tag{22}$$

Ideally, when there are no timing errors, Eq. (19) reduces to the well-known binary phase shift keyed (BPSK) error rate, $P_s(E) = 1/2 \text{ erfc } (\sqrt{E_s/N_0})$.

Symbol SNR loss is defined as the additional symbol SNR needed in the presence of imperfect synchronization to achieve the same SER as in the presence of perfect synchronization. Mathematically, the SNR loss due to imperfect carrier, subcarrier, and symbol timing references is given in dB as

$$L = 20\log\left[f^{-1}(P_s(E))\right]\Big|_{\text{(infinite loop SNR)}} - 20\log\left[f^{-1}(P_s(E))\right]\Big|_{\text{(finite loop SNR)}} \tag{23}$$

where $f(\cdot)$ and $P_s(E)$ are as defined by Eq. (19). The first term in Eq. (23) is the value of E_s/N_0 required at a given value of $P_s(E)$ in the presence of perfect synchronization, whereas the second term is the value of E_s/N_0 required for imperfect synchronization. Note that loss defined in this way is a negative number.

III. Carrier Array Using a Single PLL

Carrier arraying using a single PLL followed by BBC is shown in Fig. 3. This scheme is similar to the single receiver in that signal demodulation uses a single PLL, subcarrier loop, and symbol loop. Two main differences, however, are (1) the IF residual carrier signals are combined so that the PLL operates at a higher loop SNR than in the single receiver case, and (2) after carrier demodulation, the baseband signals are also combined so the subcarrier and symbols operate at a higher loop SNR as well.

Due to different path lengths, the received signal at antenna i is delayed by τ_i s relative to antenna 1. After complex downconversion to an appropriate IF, the signal at antenna i can be represented as [1]

$$\begin{aligned} r_i(t) &= r_1(t-\tau_i) \\ &= \sqrt{P_{T_i}} \exp\left\{j\left[\omega_I t - \omega_c \tau_{i1} + \delta d(t-\tau_i) \ \mathrm{Sqr}[\omega_{sc}(t-\tau_i) + \theta_{sc_i}] + \theta_{c_i}\right]\right\} \\ &\quad + n_i(t) \exp\left\{j\left[\omega_I t + \theta_{c_i}\right]\right\} \end{aligned} \tag{24}$$

where for an L-antenna array, $i = 1, 2, \cdots, L$. The carrier phase of the ith signal is $\theta_{c_i}(t) = \theta_{c_1}(t) + \Delta\theta_i(t)$ where $\Delta\theta_i$ represents the differential Doppler between the signal i and the signal 1. (Antenna 1 has arbitrarily been chosen as the reference antenna.) All other parameters in Eq. (24) are as defined in Eq. (1), except for ω_I, which denotes the carrier IF frequency. Here the noise $n_i(t)$ is a complex noise process with a one-sided PSD level equal to $2N_0$ (W/Hz). As shown in Fig. 3, each IF signal is first filtered to extract the carrier component and then transmitted to a central location where it is phase aligned and combined with carrier signals from other antennas. The phase alignment and combining algorithms are shown in Figs. 6 and 7. Note that the combining algorithm here is almost identical to that used for the full-spectrum combining technique described in [1,3], the difference being that here the output of the bandpass filter in Fig. 3 is the residual carrier component, whereas in [1,3] it was the first N harmonics of the telemetry signal. The filter output, $r_{F_i}(t)$ in Fig. 6, is given as

$$r_{F_i}(t) = \sqrt{P_{C_i}} \exp\left[j(\omega_I t + \theta_{c_i})\right] + n_{F_i}(t) \exp\left[j(\omega_I t + \theta_{c_i})\right] \tag{25}$$

for $i = 1, \cdots, L$. Here P_{C_i} is the received carrier power at antenna i, and the noise $n_{F_i}(t)$ is a complex bandpass Gaussian noise. The signals $r_{F_i}(t)$ $(i \neq 1)$ are phase aligned with $r_{F_1}(t)$, scaled by the optimum weighting factors [2,10], $\beta_i = (\sqrt{P_{C_i}} N_{01})/(\sqrt{P_{C_1}} N_{0i})$, and then combined. Combining the carrier signals in this way maximizes the combining gain [10].

Let $\theta_{i1} = \Delta\theta_i$ denote the phase difference between signal i and the reference signal before phase alignment. Then the signal $r_{F_i}(t)$ is aligned with the reference $r_{F_1}(t)$ by rotating $r_{F_i}(t)$ by $e^{-j\hat{\theta}_{i1}}$ for $i = 2, \cdots, L$. The estimate [11], $\hat{\theta}_{i1}$, is obtained using the algorithm in Fig. 7. Denote the phase alignment error $\Delta\phi_{i1} = \theta_{i1} - \hat{\theta}_{i1}$. Then the variance of $\Delta\phi_{i1}$ is related to the SNR of the phase difference estimator by [1,3,11]

$$\sigma^2_{\Delta\phi_{i1}} \approx \frac{1}{2\ SNR_{i1}} \tag{26}$$

where [11]

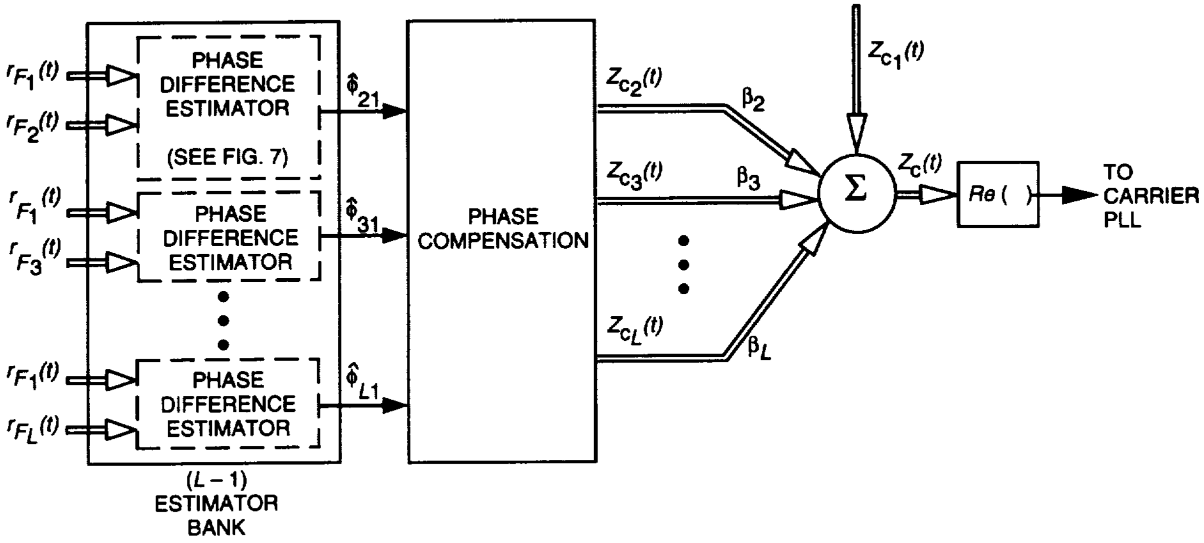


Fig. 6. The carrier maximal-ratio combining bank.

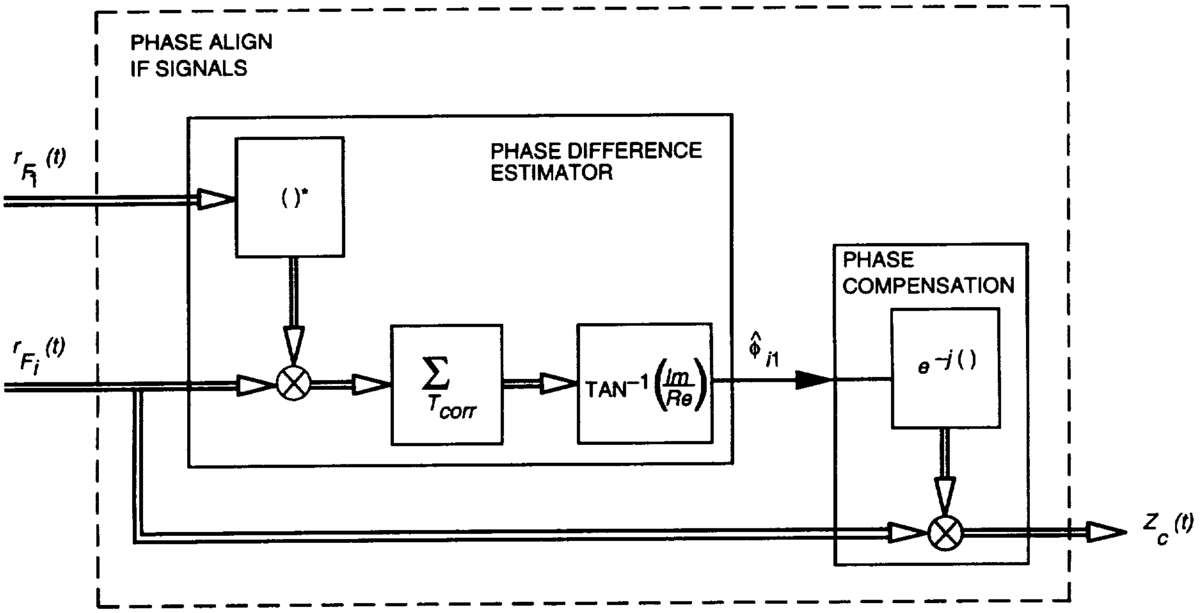


Fig. 7. Phase alignment and combining of two carrier signals.

$$SNR_{i1} = \frac{2T_{corr}((P_{C_1})/(N_{01}))}{1 + (1/\gamma_i) + B_{corr}[1/((P_{C_i})/(N_{oi}))]} \tag{27}$$

The parameter B_{corr} denotes the single-sided bandwidth of the BPF in Fig. 3, T_{corr} denotes the estimation interval, and the ratio $\gamma_i = (P_{C_i}/P_{C_1})(N_{01}/N_{0i})$ is called the antenna gamma factor. These ratios are shown in Table 1 for several DSN antennas operating in S-band or X-band (8.4–8.5 GHz).

The IF carrier signals after phase compensation, denoted $Z_{C_i}(t)$ in Fig. 6, are given as

$$Z_{C_i}(t) = \sqrt{P_{C_i}} e^{j[\omega_I t+\theta_1(t)+\Delta\phi_{i1}(t)]} + n_i(t) e^{j[\omega_I t+\theta_1(t)+\Delta\phi_{i1}(t)]} \tag{28}$$

The combined signal, $Z_C(t)$, obtained by taking the weighted sum of $Z_{C_i}(t)$ is a complex tone plus noise. Namely,

$$Z_C(t) = \sum_{i=1}^{L} \beta_i Z_{C_i}(t) \tag{29}$$

Following the same steps as in [1,3], the power of the complex tone in Eq. (29) averaged over $\Delta\phi_{i1}$ can be shown to be

$$P_{C_{comb}} = P_{C_1} \sum_{i=1}^{L} \sum_{j=1}^{L} \gamma_i \gamma_j \overline{C_{ij}} \tag{30}$$

where $\overline{C_{ij}}$, the average signal reduction function due to phase misalignment between the signal i and the signal j, is given as [1,3]

$$\overline{C_{ij}} = \begin{cases} e^{-(1/2)[\sigma^2_{\Delta\phi_{i1}} + \sigma^2_{\Delta\phi_{j1}}]} & m \neq n \\ 1 & m = n \end{cases} \tag{31}$$

Similarly, the one-sided PSD level of the combined noise at the carrier loop input is given by [2]

$$2N_{0_{eff}} = 2\ N_{01} \sum_{i=1}^{L} \gamma_i \tag{32}$$

Referring to Fig. 6, the PLL input is formed by taking the real part of the combined signal $Z_C(t)$. Consequently, the PLL loop SNR is given by

$$\begin{aligned} \rho_c &= \frac{P_{C_{comb}}/N_{0_{eff}}}{B_c} \\ &= \frac{P_{C_1}/N_{0_1}}{B_c} \left[\frac{\sum_{i=1}^{L} \gamma_i^2 + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \gamma_i \gamma_j \overline{C_{ij}}}{\sum_{i=1}^{L} \gamma_i} \right] \end{aligned} \tag{33}$$

where the bracketed term is the improvement in loop SNR due to arraying.

A. Carrier Demodulation

Since the PLL input is formed by aligning the phase of signals 2 through L with the phase of signal 1, the PLL reference is tuned to signal 1 and can be used without modification to demodulate the carrier at antenna 1. Carrier demodulation at antenna i (for $i \neq 1$), however, can be performed only after aligning the phase of the PLL reference to that of the carrier at antenna i. That is, carrier demodulation at antenna i is performed after rotating the PLL reference by $e^{j\hat{\theta}_{i1}}$. Also note that since the carrier reference at all antennas is derived from a single carrier loop, the SNR degradation and loss due to imperfect carrier synchronization is the same for all antennas. That is, in the telemetry channel, the carrier signal reduction function for antenna i, denoted by C_{c_i}, is given by

$$C_{c_i} = \cos\phi_c \qquad i = 1, 2, \cdots, L \tag{34}$$

where $\sigma^2_{\phi_c} = 1/\rho_c$, and ρ_c is given by Eq. (33).

Assume that the baseband combiner in Fig. 3 perfectly time aligns the signals before combining them;[2] then, following the same steps as in [1,11], it can be shown that the combined symbol stream at the matched filter output can be written as

$$v_k = \begin{cases} \sqrt{P_{D_1}\, g_{comb}}\, C_c C_{sc} d_k + n_k & d_k = d_{k-1} \\ \sqrt{P_{D_1}\, g_{comb}}\, C_c C_{sc}\left(1 - \dfrac{|\phi_{sy}|}{\pi}\right) d_k + n_k & d_k \neq d_{k-1} \end{cases} \tag{35}$$

where the conditional gain factor, denoted g_{comb}, is given by

$$g_{comb} = \sum_{n=1}^{L} \gamma_n^2 + \sum_{n=1}^{L} \sum_{\substack{m=1 \\ n \neq m}}^{L} \gamma_n \gamma_m C_{nm} \tag{36}$$

and the noise n_k is a Gaussian random variable with variance $\sigma_n^2 = N_{0_{eff}}/2T$. Defining the conditional symbol SNR as before yields

$$SSNR' = \begin{cases} \dfrac{2P_{D_1}T}{N_{01}} C_{comb} C_c^2 C_{sc}^2 & d_k = d_{k-1} \\ \dfrac{2P_{D_1}T}{N_{01}} C_{comb} C_c^2 C_{sc}^2 \left(1 - \dfrac{|\phi_{sy}|}{\pi}\right)^2 & d_k \neq d_{k-1} \end{cases} \tag{37}$$

where

$$C_{comb} = \frac{\sum_{n=1}^{L} \gamma_n^2 + \sum_{n=1}^{L} \sum_{\substack{m=1 \\ n \neq m}}^{L} \gamma_n \gamma_m C_{nm}}{\sum_{n=1}^{L} \gamma_n} \tag{38}$$

is the degradation due to imperfect phase alignment. The last equation is useful in computing the symbol SNR degradation and SER loss as shown below.

B. Degradation

The SSNR degradation is defined as the ratio of the SSNR in the presence of imperfect phase alignment and synchronization to the ideal SSNR (no phase errors). The degradation is obtained by computing the SSNR in the presence of phase errors (averaging Eq. (37) over $\Delta\phi_{i1}$, ϕ_c, ϕ_{sc}, and ϕ_{sy}) and then dividing that result by the ideal SSNR ($SSNR_{ideal} = ((2P_{D_1}T)/(N_{01}))\sum_{i=1}^{L}\gamma_i$). Hence,

[2] This assumption simplifies the analysis without affecting the relative performance of the schemes. Note that the uncombined signals are not assumed to be perfectly phase aligned.

$$D = 10\log_{10}\left[\overline{C_c^2}\;\overline{C_{sc}^2}\;\overline{C_{sy}^2}\;\left(\frac{\sum_{m=1}^{L}\gamma_m^2 + \sum_{m=1}^{L}\sum_{\substack{n=1\\n\neq m}}^{L}\gamma_m\gamma_n\overline{C_{mn}}}{(\sum_{m=1}^{L}\gamma_m)^2}\right)\right] \tag{39}$$

where $\overline{C_{nm}}$ is given by Eq. (31). The quantities $\overline{C_c^2}$, $\overline{C_{sc}^2}$, and $\overline{C_{sy}^2}$ are given by Eqs. (15) through (17) with the modification that the loop SNRs ρ_c, ρ_{sc}, and ρ_{sy} presented in Eqs. (8) through (11) are now computed using the combined power-to-noise level, or $P_C/N_{0_{eff}}$, which is found from Eqs. (30) and (32).

C. Loss

The SER for the array in Fig. 3 is computed using the same procedure as in the single receiver case. Therefore, the SER is given by averaging the conditional SER over all the phase errors. Assuming that the phase alignment errors, $\Delta\phi_{i1}$, are independent for $i = 1, \cdots, L$ we have [3]

$$P_s(E) = \int_{\phi_c}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\underbrace{\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}}_{L-1} P_s'(E) \times \left[p(\phi_c)p(\phi_{sc})p(\phi_{sy}) \times \left(\prod_{n=2}^{L} p(\Delta\phi_{n1})\right)\right] \mathbf{d\Delta\phi} d\phi_{sy} d\phi_{sc} d\phi_c \tag{40}$$

where $\mathbf{\Delta\phi} = (\Delta\phi_{21}, \cdots, \Delta\phi_{L1})$ are the resulting $L-1$ phase alignment errors. The $\mathbf{\Delta\phi}$ are independent and identically distributed Gaussian random variables with variance given by Eq. (26). The statistics of the error processes ϕ_c, ϕ_{sc}, and ϕ_{sy} were described earlier. After substituting Eq. (37) in Eq. (20), the conditional SER becomes

$$P_s'(E) = \frac{1}{4}\,\mathrm{erfc}\left[\sqrt{\frac{E_{s1}}{N_{01}}}C_{comb}C_cC_{sc}\left(1 - \frac{|\phi_{sy}|}{\pi}\right)\right] + \frac{1}{4}\,\mathrm{erfc}\left[\sqrt{\frac{E_{s1}}{N_{01}}}C_{comb}C_cC_{sc}\right] \tag{41}$$

where $E_{s1}/N_{01} = P_{D_1}T/N_{01}$ is the symbol SNR at antenna 1. Ideally, when there is no combining and the synchronization errors $C_{comb} = C_c = C_{sc} = 1 - |\phi_{sy}|/\pi = 1$, the SER given in Eq. (40) reduces to the well known BPSK symbol error rate, $P_s(E) = 1/2\ \mathrm{erfc}\left(\sqrt{E_{s1}/N_{01}(\sum_{n=1}^{L}\gamma_n)}\right)$, where $(\sum_{n=1}^{L}\gamma_n)$ is the ideal combining gain. The SNR loss is given by Eq. (23) after using Eq. (40) for $P_s(E)$.

D. Numerical Examples

The use of Eqs. (39) and (40) is illustrated here by computing the degradation and loss for the system in Fig. 3 when $L = 2$ and 4.

1. Array of One 70-m and One STD 34-m Antenna. Consider again an array of one 70-m and one STD 34-m antenna operating at S-band. Then from Table 1, with $\gamma_1 = 1$ and $\gamma_2 = 0.17$, the ideal gain $10\log_{10}(\gamma_1 + \gamma_2) = 0.68$ dB. The degradation to the ideal gain versus the 70-m symbol SNR (E_{s1}/N_{01}) is shown in Fig. 8 for a symbol rate of 200 sps and a modulation index of 70 deg. In Fig. 8, the degradation for the end-to-end system in Fig. 3 is shown by the solid line and obtained by evaluating Eq. (23). The degradation due to the individual components is shown by the broken lines. For example, the degradation due to the carrier loop, shown by the top line (CA) in Fig. 8 is found by assuming that all the other components in the array have ideal operation, that is, by evaluating Eq. (23) as follows:

$$D|_{[SNR_{n1}=\rho_{sc}=\rho_{sy}=\infty]} = 10\log_{10}\overline{C_c^2} \tag{42}$$

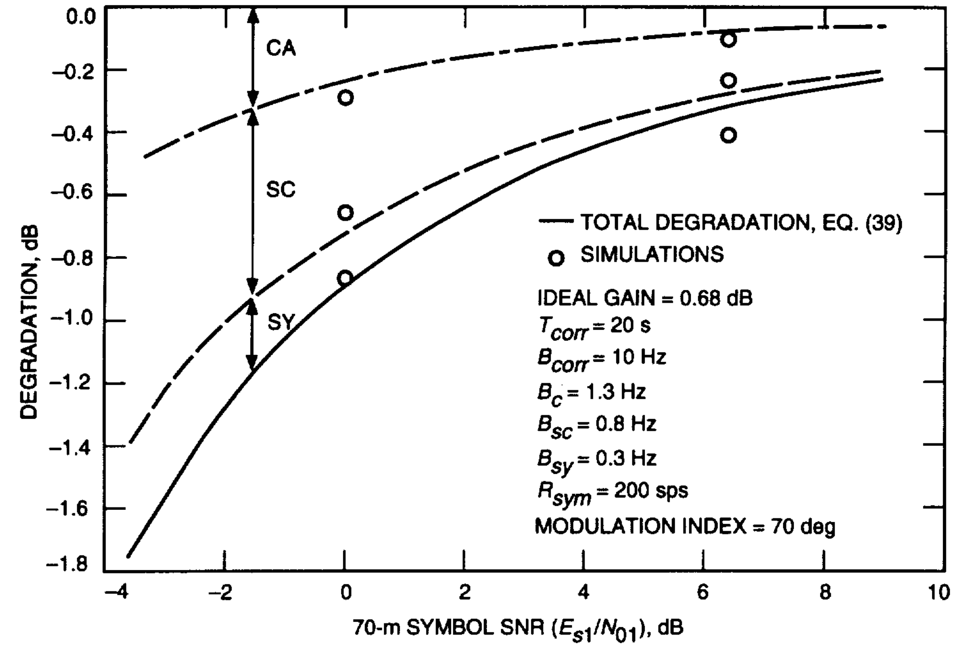


Fig. 8. SSNR degradation for an array of two different antennas.

The second line from the top (SC) is the degradation due to the carrier and subcarrier, and the bottom line (SY) is the carrier, subcarrier, and symbol or total degradation. The IF carrier combining and baseband telemetry combining degradations are not shown individually because they are negligible. Note that it was shown in [1] that the total degradation in dB is approximately equal to the sum of the individual degradations. Results obtained by simulating the system in Fig. 3 are indicated by the circles.

SER curves needed to compute the loss are shown in Fig. 9. The bottom curve is the SER assuming an array with ideal gain (0.68 dB). The SER for nonideal gain, Eq. (40), is shown by the curve in the middle. Simulation results for a nonideal array are shown as circles. At the top is the nonideal performance for a single 70-m antenna, Eq. (19). In the example, the conditional SNR, $P_s'(E)$ in Eq. (40), is given by Eq. (41) with

$$C_{comb} = \frac{\gamma_1^2 + \gamma_2^2}{\gamma_1 + \gamma_2} + \frac{2\gamma_1\gamma_2}{\gamma_1 + \gamma_2}\cos(\Delta\phi_{21}) \tag{43}$$

where $\gamma_1 = 1$ and $\gamma_2 = 0.17$.

The degradation and loss for various SERs are given in Table 2. The second column in the table is the symbol SNR needed (at antenna 1) for an ideal array to achieve the SER in column 1. The loss in the third column is the additional SNR needed by a nonideal system to achieve the same SER as an ideal one. For example, to achieve an SER of 10^{-2}, an ideal array requires that $E_{s1}/N_{01} = 3.7$ dB, whereas a practical system would require that $E_{s1}/N_{01} = (3.7 + 0.5)$ dB. The degradation in the fourth column is the reduction in the ideal SNR gain observed at the matched filter output. For instance, in our two-antenna example, since the symbol SNR at the 70-m antenna is ideally equal to 3.7 dB, then the observed or measured combined symbol SNR would be $(3.7 + 0.68 - 0.5)$ dB.

2. Array of Four 34-m Antennas. Analytical and simulation results for the symbol SNR degradation of an array of four 34-m STD antennas (i.e., $L = 4$ in Fig. 3) are shown in Fig. 10. In this

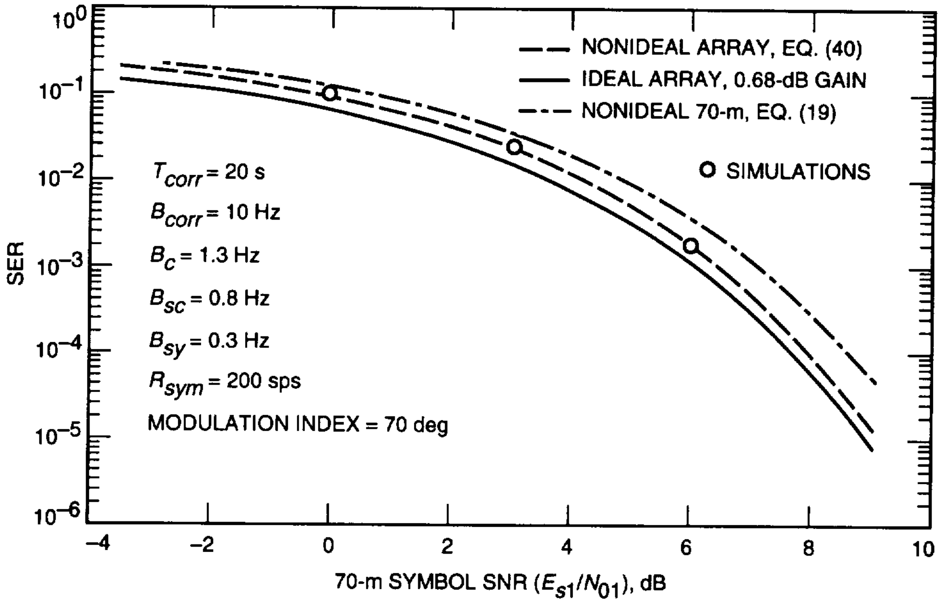


Fig. 9. SER for an array of two different antennas.

Table 2. SNR loss versus SSNR degradation (array of one 34-m STD and one 70-m antenna).

SER	E_{s1}/N_{01}	Loss, dB	Degradation, dB
10^{-1}	−1.5	−1.3	−1.2
10^{-2}	3.7	−0.5	−0.5
10^{-3}	6.1	−0.4	−0.4
10^{-4}	7.7	−0.3	−0.3

case, because all the antennas have the same efficiency and aperture, $\gamma_i = 1$ for all i. The analytical degradation is computed as before, using Eq. (39) with C_{comb} given by Eq. (38) as follows:

$$\begin{aligned} C_{comb} &= 1 + \frac{1}{2}[\cos(\Delta\phi_{21}) + \cos(\Delta\phi_{31}) + \cos(\Delta\phi_{41}) \\ &\quad + \cos(\Delta\phi_{31} - \Delta\phi_{21}) + \cos(\Delta\phi_{41} - \Delta\phi_{21}) + \cos(\Delta\phi_{41} - \Delta\phi_{31})] \end{aligned} \tag{44}$$

SER for this example is shown in Fig. 11. Curves are obtained for an array with ideal gain ($10\log_{10}(4) = 6$ dB), nonideal gain [Eq. (40)], and a single receiver with nonideal synchronization [Eq. (19)]. Degradation and loss for various SER values are tabulated in Table 3.

IV. Carrier Aiding

In carrier aiding, the "master antenna" is assumed to lock on the carrier and, subsequently, rate aid the other antennas. As shown in Fig. 2, the received signal at antenna $i = (2, \cdots, L)$ is first downconverted using the carrier reference from the master antenna and then tracked using a baseband PLL. If we assume that all the elements in the array are colocated, the ith PLL can operate at much narrower bandwidths

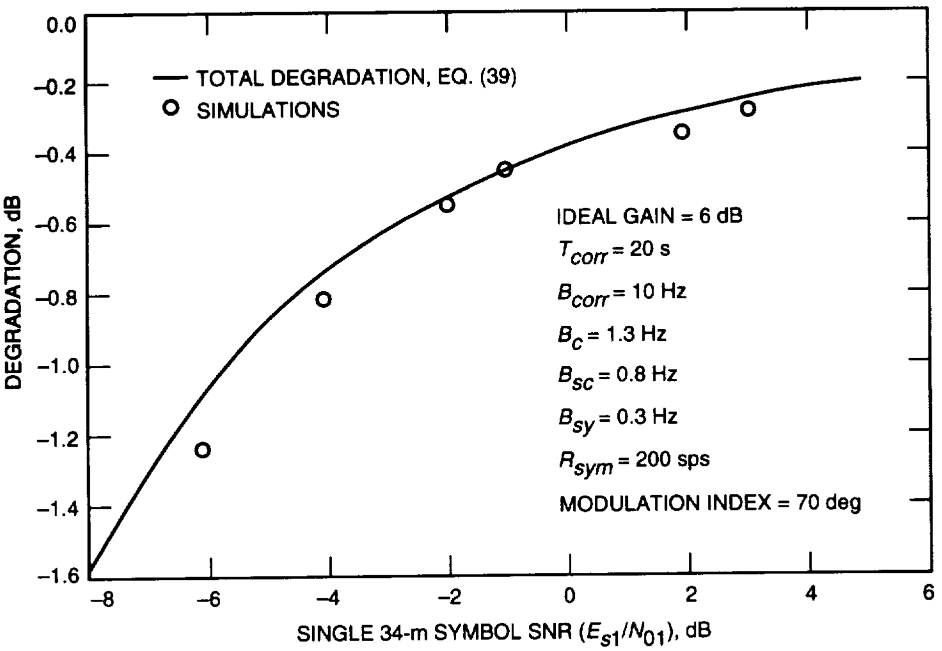


Fig. 10. SSNR degradation for an array of four identical antennas.

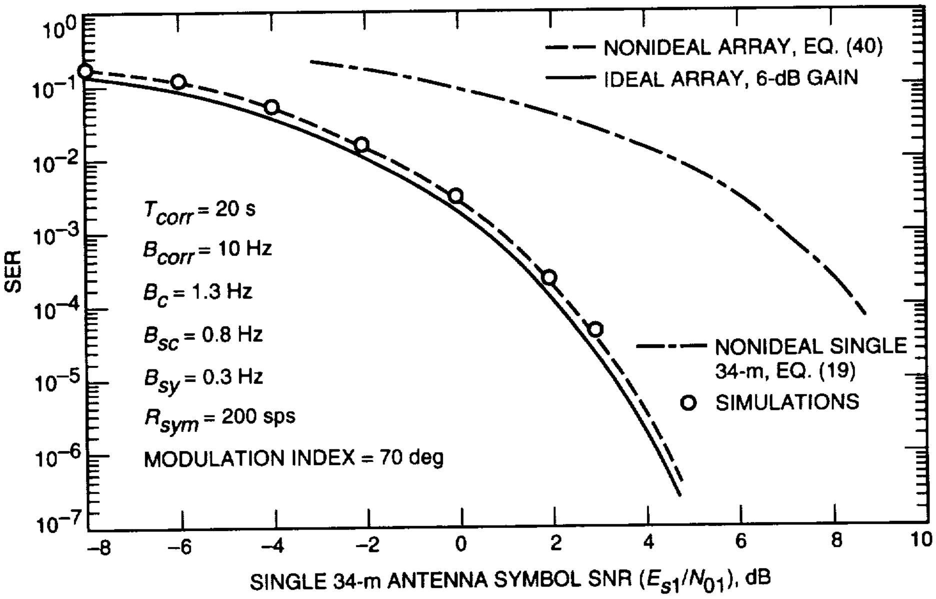


Fig. 11. SER for an array of four identical antennas.

than in the absence of rate aiding, because it need only track the Doppler dynamics relative to the master antenna. After carrier demodulation, the signals from each antenna are sent to a central location where they are time delayed, weighted, combined, and then passed through a chain of subcarrier loop, symbol loop, and matched filter. Degradation and loss for this scheme are derived as before. However, now the degradation and loss are a function of the phase error of L carrier loops. Two quantities that are needed to derive the performance of this system are the loop SNR of the ith carrier loop, ρ_{c_i}, and the joint probability density function of the carrier phase errors $\boldsymbol{\phi}_c = (\phi_{c_1}, \phi_{c_2}, \cdots, \phi_{c_L})$.

Table 3. SNR loss versus SSNR degradation (array of four 34-m STD antennas).

SER	E_{s1}/N_{01}	Loss, dB	Degradation, dB
10^{-1}	−6.9	−1.3	−1.3
10^{-2}	−1.7	−0.5	−0.5
10^{-3}	0.77	−0.4	−0.4
10^{-4}	2.4	−0.35	−0.3

A. Derivations of ρ_{c_i} and Joint Probability Density Function of ϕ_c

Since the operation of the master PLL in Fig. 2 is unaffected by the PLLs at the other antennas, its loop SNR, ρ_{c_1}, is given by Eq. (8). The aided loop, on the other hand, is directly affected by the performance of the master PLL, so its loop SNR can be expected to be related to the loop SNR (and bandwidth) of the master antenna. For residual and suppressed-carrier modulation, the aided-loop loop SNR, denoted ρ_{c_i}, is shown in the Appendix, using Fokker–Planck, to be

$$\rho_{c_i} = \left[\frac{1}{\rho'_{c_i}} + \frac{\xi_{1i}}{3\rho_{c_1}} \frac{2 + 4\xi_{1i} + 5\xi_{1i}^2 + 3\xi_{1i}^3}{1 + 2(\xi_{1i} + \xi_{1i}^2 + \xi_{1i}^3) + \xi_{1i}^4}\right]^{-1} \tag{45}$$

where, for residual carrier modulation, $\rho'_{c_i} = (P_{C_i}/N_{0i})/B_{c_i}$, and, for suppressed-carrier modulation, $\rho'_{c_i} = (P_{D_i}/N_{0i})/B_{c_i}\left(1 + (1/2E_{s_i}/N_{0_i})\right)^{-1}$. The parameter ξ_{1i} denotes the ratio of the loop bandwidth and is given by

$$\xi_{1i} = \frac{B_{c_1}}{B_{c_i}} \tag{46}$$

Some insight into the last equation can be given by examining the relationship between the master and aided loops in the following four cases: (1) $B_{c_i} \to \infty$, B_{c_1} fixed, (2) $B_{c_1} \to 0$, B_{c_i} fixed, (3) $B_{c_i} \to 0$, B_{c_1} fixed, and (4) $B_{c_1} \to \infty$, B_{c_i} fixed. Note that cases (3) and (4) are of most interest because, in practice, $B_{c_i} \ll B_{c_1}$ and, equivalently, $\xi_{1i} \gg 1$.

Case (1): In the limit $B_{c_i} \to \infty$, the loop SNR $\rho_{c_i} \to 0$, as expected. Case (2): Recall that in our model of the IF signals [see Eq. (24)], the phase at antenna i is given by $\theta_i = \theta_1 + \Delta\theta_{i1}$, where θ_1 is the phase of the master antenna and $\Delta\theta_{i1}$ is the phase at antenna i relative to antenna 1. If the master loop is tracking, the phase input to the ith loop is $\phi_{c_1} + \Delta\theta_{i1}$, where ϕ_{c_1} is the tracking error at antenna 1. Now suppose that the master loop is tracking θ_1 perfectly (i.e, $\phi_{c_1} \to 0$, or alternatively, $\rho_{c_1} \to \infty$ and $B_{c_1} \to 0$); then intuitively we can expect the master loop not to degrade the tracking performance of the aided loop. Letting $B_{c_1} \to 0$ in Eq. (45), we find that $\rho_{c_i} \to \rho'_{c_i}$, which is independent of ρ_{c_1}. Case (3): As $B_{c_i} \to 0$, $\rho_{c_i} \to \rho_{c_1}$, as shown in Fig. 12 for the case of a 70-m and a 34-m antenna. The broken line in the figure is obtained by evaluating Eq. (45), whereas the circles represent simulation results for two PLLs in cascade. One way to view this result is by letting the received phase at both antennas be the same (i.e., $\Delta\theta_{i1} = 0$ for $i \neq 1$). Then, the input to the second loop is the noise process ϕ_{c_1}. Intuitively, we would not expect the second loop to be able to reduce the phase error or noise from the first loop. Hence, it seems reasonable that even for loop bandwidths approaching zero, the loop SNR of the ith loop can never be greater than ρ_{c_1}. Case (4): The limit $B_{c_1} \to \infty$ implies that loop 1 is not tracking the carrier and, therefore, the signal into the cascaded loop is one mixed by an incoherent reference. Hence, in this case, we can expect the cascaded loop not to track its input either. The inability of the cascaded loop to track the signal is shown in Fig. 13, where, in the limit, ρ_{c_2} approaches zero. From the above cases, we can conclude that $\rho_{c_i} \leq \rho_{c_1}$.

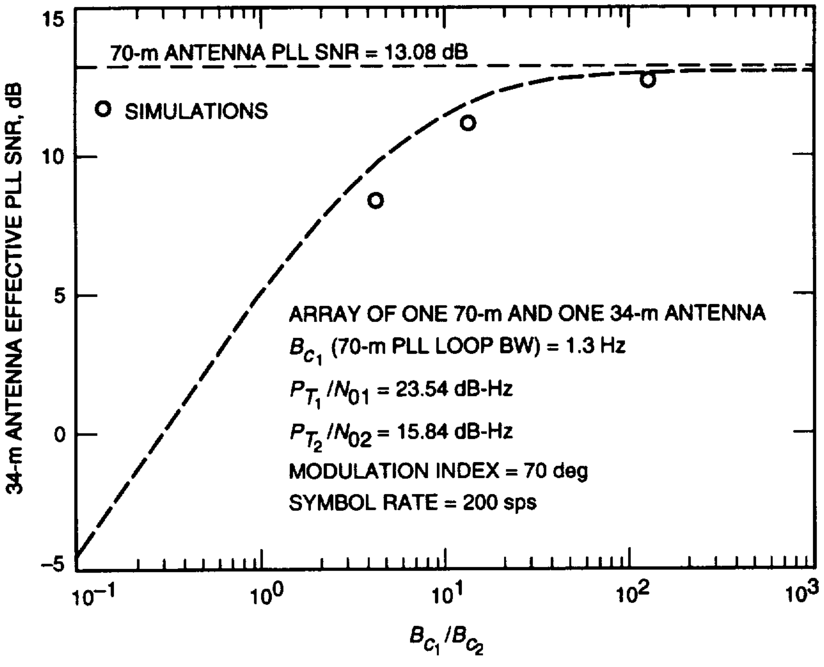


Fig. 12. Loop SNR limit case (3).

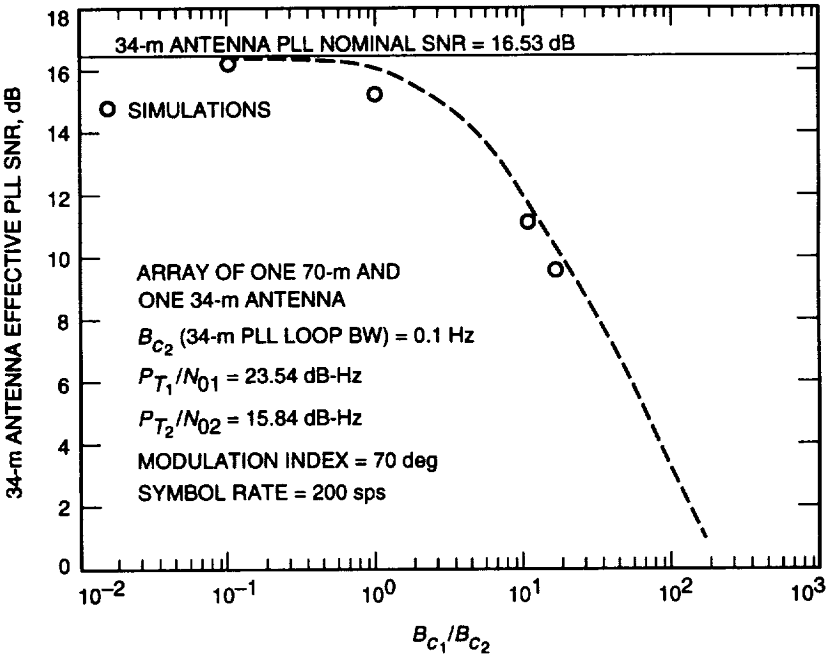


Fig. 13. Loop SNR limit case (4).

Next we turn to the derivation of $p(\phi_{c_1}, \phi_{c_2}, \cdots, \phi_{c_L})$, which is needed to determine the SER and loss. We begin with the derivation of $p(\phi_{c_i}, \phi_{c_1})$. Note that, from $\theta_i = \theta_1 + \Delta\theta_{i1}$, it is clear that for $i \neq 1$, θ_i and θ_1 are not independent. Assuming that $p(\phi_{c_1})$ is Thikonov distributed as in Eq. (7), the joint density $p(\phi_i, \phi_1)$ is derived in the Appendix to be

$$p(\phi_{c_1}, \phi_{c_i}) = \begin{cases} \dfrac{\exp\left[\alpha_i \cos\left(\phi_{c_i} - \eta_{1i}\sqrt{\rho_{c_1}/\rho_{c_i}}\phi_{c_1}\right) + \rho_{c_1}\cos(\phi_{c_1})\right]}{(2\pi)^2 I_0(\alpha_i) I_0(\rho_{c_1})} & \text{residual-carrier case} \\ \dfrac{\exp\left[(\alpha_i/4)\cos\left[2\left(\phi_{c_i} - \eta_{1i}\sqrt{\rho_{c_1}/\rho_{c_i}}\phi_{c_1}\right)\right] + (\rho_{c_1}/4)\cos(2\phi_{c_1})\right]}{(\pi)^2 I_0(\alpha_i/4) I_0(\rho_{c_1}/4)} & \text{suppressed-carrier case} \end{cases} \tag{47}$$

where $\alpha_i = \rho_{c_i}/(1 - \eta_{1i}^2)$, and where the correlation coefficient, η_{1i}, is shown in the Appendix to be

$$\eta_{1i} = \sqrt{\frac{\rho_{c_i}}{\rho_{c_1}}} \left[\frac{\xi_{1i}^2}{3} \; \frac{1 + 4\xi_{1i} + 3\xi_{1i}^2}{1 + 2(\xi_{1i} + \xi_{1i}^2 + \xi_{1i}^3) + \xi_{1i}^4}\right] \tag{48}$$

Some insight into Eq. (47) can be given by once again considering the extreme cases when $B_{c_i} \rightarrow 0$ and $B_{c_i} \rightarrow \infty$. We have already seen that when B_{c_1} is fixed and $B_{c_i} \rightarrow \infty$, then $\rho_{c_i} \rightarrow 0$. Hence, in this limit, the loop is unable to track, and we can expect $p(\phi_{c_i})$ to be uniformly distributed in the interval $[-\pi, \pi]$ for the residual-carrier case and in the interval $[-(\pi/2), \pi/2]$ for the suppressed-carrier case, respectively. It can be shown that

$$\begin{aligned} p(\phi_{c_i})|_{B_{c_i}=\infty} &= \lim_{B_{c_i}\rightarrow\infty} \int_{\phi_{c_1}} p(\phi_{c_1}, \phi_{c_i}) d\phi_{c_1} \\ &= \begin{cases} \dfrac{1}{2\pi} & \text{residual-carrier case} \\ \dfrac{1}{\pi} & \text{suppressed-carrier case} \end{cases} \end{aligned} \tag{49}$$

for both cases in Eq. (47). Similarly, it can be shown that when B_{c_1} is fixed and $B_{c_i} \rightarrow 0$, the density is given as

$$\begin{aligned} p(\phi_{c_i})|_{B_{c_i}=0} &= \lim_{B_{c_i}\rightarrow 0} \int_{\phi_{c_1}} p(\phi_{c_1}, \phi_{c_i}) d\phi_{c_1} \\ &= \begin{cases} \dfrac{\exp[\rho_{c_1}\cos(\phi_{c_i})]}{2\pi I_0(\rho_{c_1})} & \text{residual-carrier case} \\ \dfrac{\exp[(\rho_{c_1}/4)\cos(2\phi_{c_i})]}{\pi I_0(\rho_{c_1}/4)} & \text{suppressed-carrier case} \end{cases} \end{aligned} \tag{50}$$

Notice that the last equation is a function of the master-loop loop SNR ρ_{c_1}, not ρ_{c_i}. This is consistent with our earlier result, where we concluded that the upper limit of the aided-loop loop SNR (i.e., as $B_{ci} \rightarrow 0$) is equal to ρ_{c_1}.

The joint probability density function (pdf) $p(\phi_{c_1}, \phi_{c_2}, \cdots, \phi_{c_L})$ is found by applying Bayes Theorem, namely,

$$\begin{aligned} p(\phi_{c_1}, \phi_{c_2}, \cdots, \phi_{c_L}) &= p(\phi_{c_2}, \cdots, \phi_{c_L} | \phi_{c_1}) p(\phi_{c_1}) \\ &= p(\phi_{c_1}) p(\phi_{c_2} | \phi_{c_1}) p(\phi_{c_3} | \phi_{c_1}) \cdots p(\phi_{c_L} | \phi_{c_1}) \\ &= p(\phi_{c_1}) \prod_{i=2}^{L} \left[\frac{p(\phi_{c_1}, \phi_{c_i})}{p(\phi_{c_1})} \right] \end{aligned} \tag{51}$$

where $p(\phi_{c_1})$ and $p(\phi_{c_1}, \phi_{c_i})$ are given by Eqs. (7) and (47), respectively. The last equation simplifies to its final form because $p(\phi_{c_i}/\phi_{c_1})$ and $p(\phi_{c_j}/\phi_{c_1})$ are independent for $i \neq j$.

One more quantity needed to describe the performance of carrier aiding is the joint pdf of ϕ_m and ϕ_n for $m \neq n$ and $m, n \neq 1$. We start with the identity

$$p(\phi_{c_m}, \phi_{c_n}) = \int\limits_{\phi_{c_1}} p(\phi_{c_1}, \phi_{c_m}, \phi_{c_n}) d\phi_{c_1} \tag{52}$$

Using Eq. (51) for $p(\phi_{c_1}, \phi_{c_m}, \phi_{c_n})$, we have

$$p(\phi_{c_m}, \phi_{c_n}) = \int\limits_{\phi_{c_1}} \frac{p(\phi_{c_1}, \phi_{c_m}) p(\phi_{c_1}, \phi_{c_n})}{p(\phi_{c_1})} d\phi_{c_1} \tag{53}$$

B. Performance of Carrier Aiding

Assuming as before that the time delay for each antenna is perfectly estimated, then following the same steps as in [1,2], the samples of the combined signal at the output of the matched filter are given by

$$v_k = \begin{cases} \sqrt{P_{D_1}} \left(\sum\limits_{i=1}^{L} \gamma_i C_{c_i} \right) C_{sc} d_k + n_k & d_k = d_{k-1} \\ \sqrt{P_{D_1}} \left(\sum\limits_{i=1}^{L} \gamma_i C_{c_i} \right) C_{sc} \left(1 - \dfrac{|\phi_{sy}|}{\pi} \right) d_k + n_k & d_k \neq d_{k-1} \end{cases} \tag{54}$$

where $C_{c_i} = \cos(\phi_{c_i})$, and all other terms are as defined earlier. The symbol SNR conditioned on ϕ_{c_i}, ϕ_{sc}, and ϕ_{sy} is given from Eq. (12) as

$$SSNR' = \begin{cases} \dfrac{2P_{D_1}T}{N_{01}} C_{comb} C_{sc}^2 & d_k = d_{k-1} \\ \dfrac{2P_{D_1}T}{N_{01}} C_{comb} C_{sc}^2 \left(1 - \dfrac{|\phi_{sy}|}{\pi} \right)^2 & d_k \neq d_{k-1} \end{cases} \tag{55}$$

where

$$C_{comb} = \frac{\left[\sum_{i=1}^{L} \gamma_i C_{c_i} \right]^2}{\sum_{i=1}^{L} \gamma_i} \tag{56}$$

1. Degradation. Proceeding as in Section III, the SSNR degradation for this case is determined by averaging Eq. (55) over all the phase errors and then dividing the result with the ideal combined SNR. Hence,

$$D = 10\log_{10}\left[\overline{C_{sc}^2 C_{sy}^2}\left(\frac{\sum_{m=1}^{L}\gamma_m^2\overline{C_{c_m}^2} + \sum_{m=1}^{L}\sum_{\substack{n=1\\n\neq m}}^{L}\gamma_m\gamma_n\overline{C_{c_m,c_n}}}{(\sum_{m=1}^{L}\gamma_m)^2}\right)\right] \tag{57}$$

where $\overline{C_{c_m}^2}$ is given by using the appropriate loop SNR in Eq. (15), and $\overline{C_{sc}^2}$ and $\overline{C_{sy}^2}$ are as defined earlier. The first moment of the joint carrier degradation, $\overline{C_{c_m,c_n}}$, is defined as

$$\overline{C_{c_m,c_n}} = \int\limits_{\phi_{c_n}}\int\limits_{\phi_{c_m}} \cos(\phi_{c_m})\cos(\phi_{c_n})p(\phi_{c_m},\phi_{c_n})d\phi_{c_m}d\phi_{c_n} \tag{58}$$

After substituting Eq. (53) for the joint pdf, we have the following equation that must be computed numerically:

$$\overline{C_{c_m,c_n}} = \int\limits_{\phi_{c_n}}\int\limits_{\phi_{c_m}}\int\limits_{\phi_{c_1}} \cos(\phi_{c_m})\cos(\phi_{c_n})\left[\frac{p(\phi_{c_1},\phi_{c_m})p(\phi_{c_1},\phi_{c_n})}{p(\phi_{c_1})}\right]d\phi_{c_1}d\phi_{c_m}d\phi_{c_n} \tag{59}$$

Ideally, when there are no phase errors (i.e., when $\rho_{c_i} = \rho_{sc} = \rho_{sy} = \infty$), $\overline{C_{c_m}^2} = \overline{C_{c_m,c_n}} = \overline{C_{sc}^2} = \overline{C_{sy}^2} = 1$ and Eq. (57) becomes zero, as expected.

2. Loss. The carrier-aiding SER for an L antenna array is defined as

$$P_s(E) = \int\limits_{\phi_{sc}}\int\limits_{\phi_{sy}}\int\limits_{\phi_{c_1}}\int\limits_{\phi_{c_2}}\cdots\int\limits_{\phi_{c_L}} P_s'(E) \times [p(\phi_{sc})p(\phi_{sy}) \times p(\phi_{c_1},\phi_{c_i},\cdots,\phi_{c_L})]\,\mathbf{d}\boldsymbol{\phi}_\mathbf{c} d\phi_{sy}d\phi_{sc} \tag{60}$$

where $\mathbf{d}\boldsymbol{\phi}_\mathbf{c} = d\phi_{c_1}d\phi_{c_2}\cdots d\phi_{c_L}$. The conditional SER, $P_s'(E)$, is obtained by substituting Eq. (55) in Eq. (20). After some algebra, we have

$$P_s'(E) = \frac{1}{4}\text{erfc}\left[\sqrt{\frac{E_{s1}}{N_{01}}}C_{comb}C_{sc}\left(1 - \frac{|\phi_{sy}|}{\pi}\right)\right] + \frac{1}{4}\text{ erfc}\left[\sqrt{\frac{E_{s1}}{N_{01}}}C_{comb}C_{sc}\right] \tag{61}$$

where $E_{s1}/N_{01} = P_{D_1}T/N_{01}$ is the symbol SNR at the "master" antenna and C_{comb} was defined earlier in Eq. (56). Again, as a check, we note that, when there are no timing errors, Eq. (61) reduces to the well known BPSK error rate for an ideal array of L antennas, namely, $P_s(E) = 1/2 \text{ erfc } (\sqrt{\sum_{i=1}^{L}(E_{si}/N_{0i})})$.

C. Example: Array of One 70-m and One 34-m Antenna

The degradation and loss for carrier aiding using residual carrier and suppressed-carrier modulation are presented here for a two-element array of one 70-m antenna and one STD 34-m antenna. As in the carrier-arraying with a single PLL case, the 70-m antenna is chosen as the reference antenna so $\gamma_1 = 1$ and $\gamma_2 = 0.17$. Furthermore, the symbol rate is 200 sps, and the modulation index for the residual carrier case is 70 deg.

The analytical results for residual carrier modulation are obtained by using the PLL loop SNR in Eqs. (57) and (60), whereas the results for the suppressed case use the same equations with the Costas loop SNR instead. The analytical [Eq. (57)] and simulated degradation results for residual and suppressed-carrier modulation are shown in Figs. 14 and 15, respectively. The individual degradations due to the carrier (CA), subcarrier (SC), and symbol (SY) tracking error are shown by the broken lines. As before, the individual degradations are obtained by using infinite loop SNR in Eq. (57) for all the loops except the one whose degradation contribution is desired.

The SER performance for the residual case is depicted in Fig. 16 and in Fig. 17 for the suppressed case. In both figures, the curves shown are for an array with an ideal gain of 0.68 dB; an array with nonideal gain, Eq. (60); and the nonideal performance of a single 70-m antenna, Eq. (19). Simulated SER results for the nonideal array are shown as circles. Note that the conditional SER in Eq. (60) for this example is given as

$$P_s'(E) = \frac{1}{4}\left[\operatorname{erfc}\left[\sqrt{\frac{E_{s1}}{N_{01}}\,\frac{(C_{c_1}+\gamma_2 C_{c_2})^2}{\gamma_1+\gamma_2}}\,C_{sc}\left(1-\frac{|\phi_{sy}|}{\pi}\right)\right]+\operatorname{erfc}\left[\sqrt{\frac{E_{s1}}{N_{01}}\,\frac{(C_{c_1}+\gamma_2 C_{c_2})^2}{\gamma_1+\gamma_2}}\,C_{sc}\right]\right] \qquad (62)$$

For the residual carrier case, degradation and loss at specific SER values are shown in Table 4.

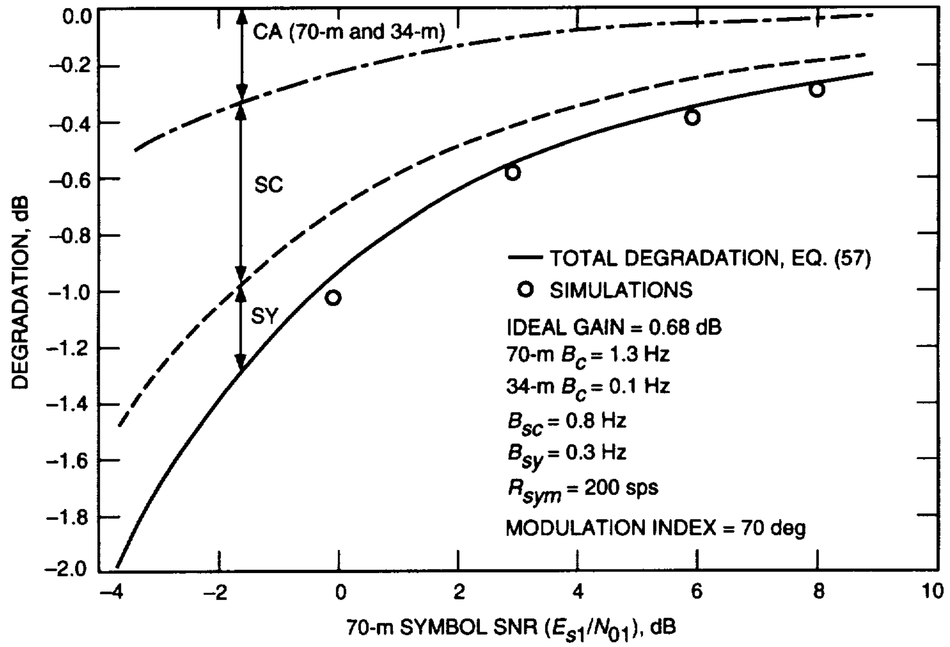


Fig. 14. SSNR degradation for an array of two different antennas (CA-aid).

V. Carrier Arraying Using Multiple Carrier Loops

Carrier arraying using multiple carrier loops is shown in Fig. 4. As explained earlier, this scheme is an improvement over carrier aiding because feedback from the aided loops enables the master loop to operate at a higher loop SNR than in the absence of feedback. The disadvantage of this scheme is that, for the array to get started, at least one of the antennas seems to require to lock on the carrier. For residual carrier modulation, this technique has been partially analyzed [12,13] and also demonstrated [13]. In [12], analytical expressions for the phase error variance (due to thermal noise) of the master loop, as

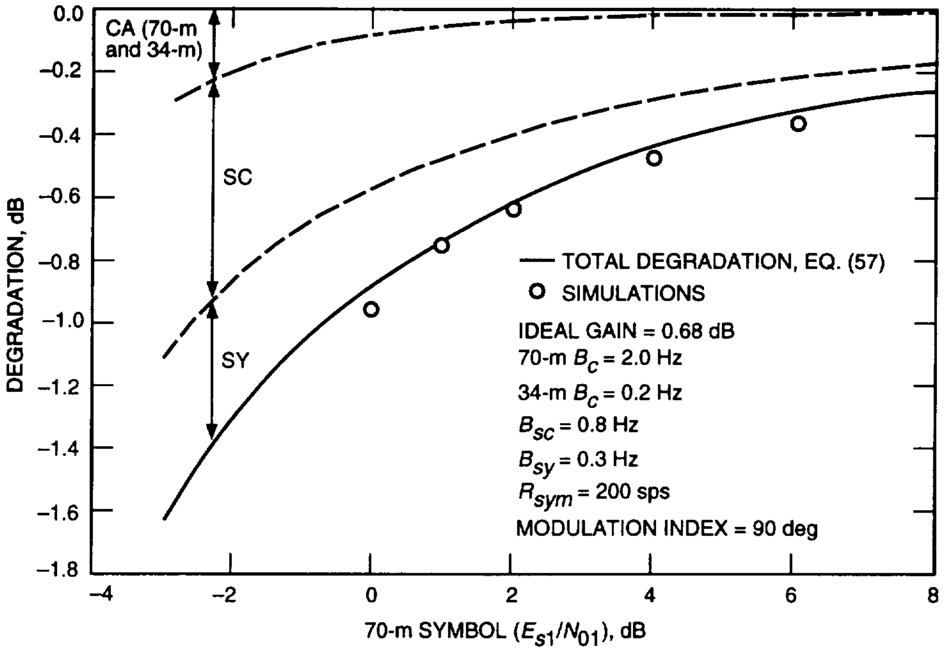


Fig. 15. SSNR degradation for an array of two different antennas, suppressed carriers (CA-aid).

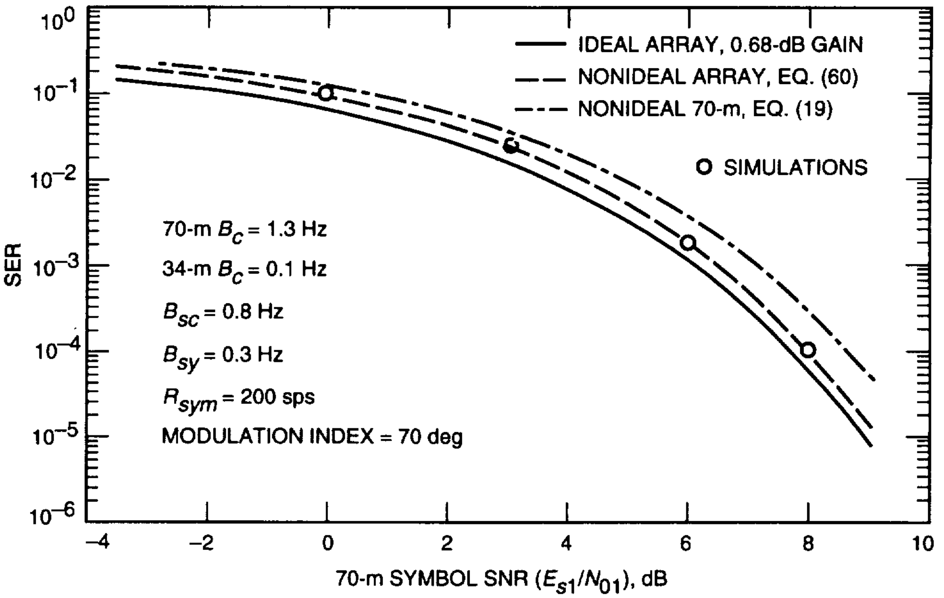


Fig. 16. SER for an array of two different antennas (CA-aid).

well as the aided (slave) loops, were presented. An extension of this theory that included the effects of oscillator phase noise on loop jitter was given in [13]. Analytical expressions for degradation and loss for the end-to-end system have yet to be presented. In our study, we obtained results for the degradation and loss by simulating Fig. 6. We would like to note that we were not able to match certain intermediate simulation results with the theory presented in [12]. Specifically, we found that the loop SNR of the aided loop obtained via simulations differed substantially from the theory presented in [12]. The cause of this discrepancy, we believe, is due to neglecting all the terms (including first-order terms) involving the carrier loop bandwidth ratio, B_{c_i}/B_{c_1}, in evaluating the integral [12, Eq. (60)].

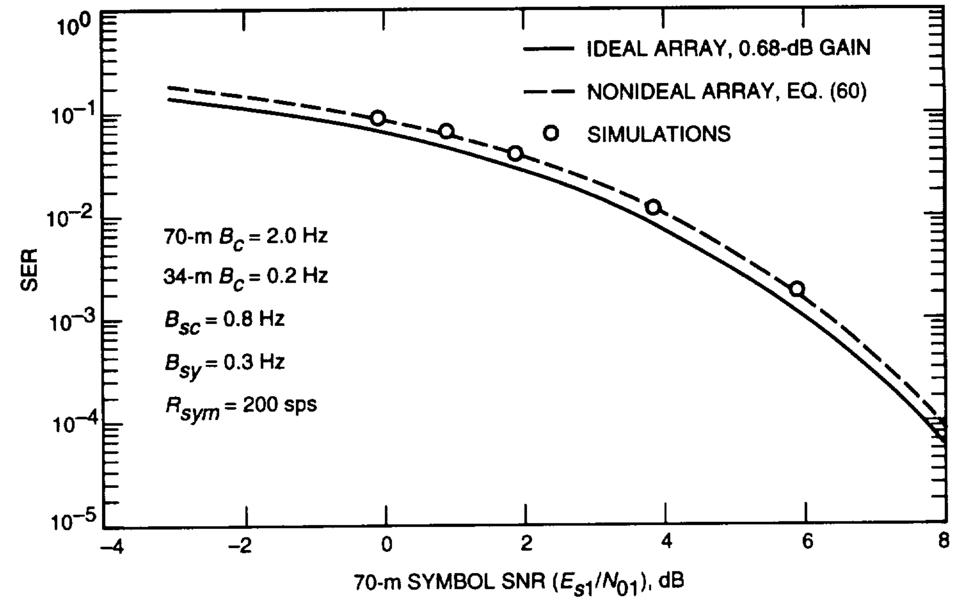


Fig. 17. SER for an array of two different antennas, suppressed carrier (CA-aid).

Table 4. SNR loss versus SSNR degradation (carrier aiding: array of one 34-m STD and one 70-m antenna).

SER	E_{s1}/N_{01}	Loss, dB	Degradation, dB
10^{-1}	-1.5	-1.4	-1.3
10^{-2}	3.7	-0.6	-0.5
10^{-3}	6.1	-0.4	-0.4
10^{-4}	7.7	-0.3	-0.3

The deviation between the existing theory for residual carrier modulation and our simulation results is illustrated using an array of one 34-m high efficiency (HEF) antenna and one 34-m STD antenna operating at S-band. Let the 34-m STD be the master antenna; then, from Table 1, $\gamma_1 = 1$ and $\gamma_2 = 0.07/0.17 = 0.41$. The ideal gain is $10\log_{10}(\gamma_1 + \gamma_2) = 1.5$ dB. For simulation purposes, we set $(P_C/N_0)_{STD} = 10$ dB-Hz, $(P_C/N_0)_{HEF} = 6.1$ dB-Hz, and $B_{c,STD} = 1$ Hz. Hence, without arraying, the master-PLL loop SNR is 10 dB. The master-PLL loop SNR in the arrayed system, denoted $\rho_{c,STD}$, should be higher than 10 dB, due to error signal feedback from the aided loop. Note that the improvement in the master-PLL loop SNR, which is maximum when the error signals add coherently, can be expected to be an upper bound on the ideal arraying gain $(1 + \gamma_2)$, or 1.5 dB. The loop SNR, $\rho_{c,STD}$, is shown in Fig. 18 as a function of the ratio between the master loop bandwidth and the aided-loop bandwidth, $B_{c,HEF}$. The bottom solid line in Fig. 18 is the loop SNR of the master loop predicted by the analysis in [12]; applying our example to the result in [12, Eq. (26)] yields

$$\rho_{STD} = \frac{1}{\sigma^2_{\phi_{c_1}}} = \frac{3\rho'_{STD}\delta}{\lambda} \tag{63}$$

where $\rho'_{STD} = ((P_C/N_o)_{STD})/B_{c,STD} = 10$ dB is the nominal master loop SNR, and

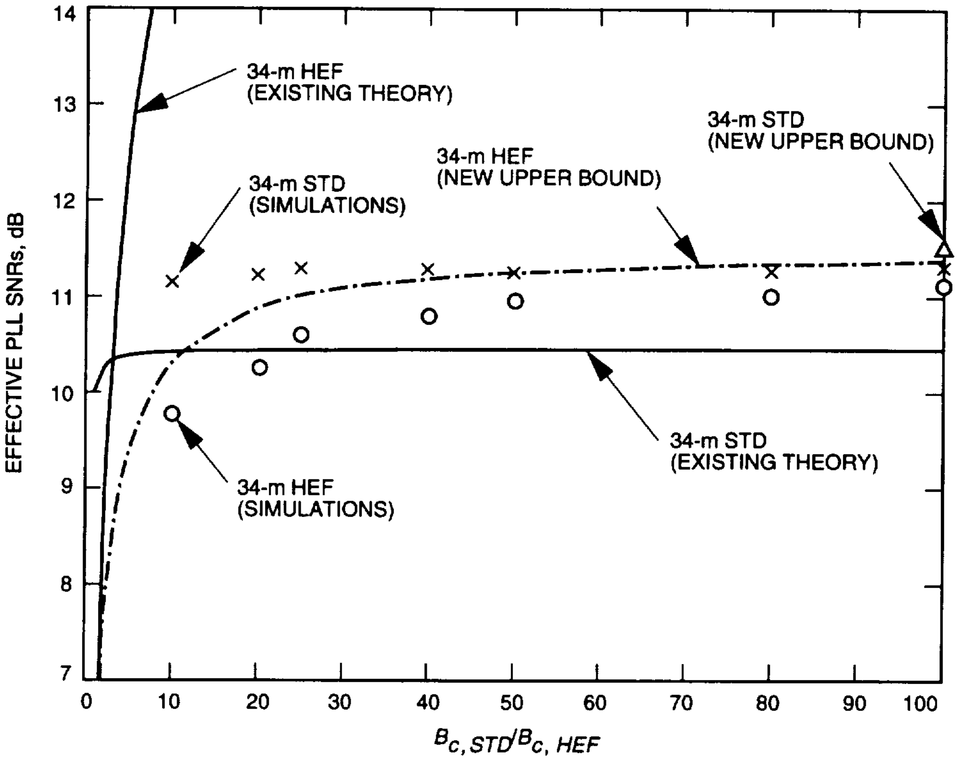


Fig. 18. Effective loop SNRs.

$$\lambda = 4G(1+2G) + 4G(5+G)\xi + 4(7G-1)\xi^2 + 4(1+5G)\xi^3 + 12\xi^4 \qquad (64)$$

$$\delta = 4G^2 + 4(3G-1)\xi + 8G\xi^2 + 4(1+G)\xi^3 + 4\xi^4 \qquad (65)$$

where $\xi = B_{c,HEF}/B_{c,STD}$, and $G = \gamma_1 + \gamma_2$ is the ideal gain. Note that the above expressions are for a carrier loop with a second-order loop filter with the damping parameter $r = 2$. The maximum gain or improvement predicted by Eq. (63) can be found by keeping $B_{c,STD}$ fixed and letting $B_{c,HEF} \rightarrow 0$. For the example given, the upper limit of the master PLL loop SNR is the value $\rho|_{B_{c,HEF}=0}$, shown in Fig. 18. Hence, the theory seems to predict that the maximum improvement is less than the ideal arraying gain. Notice in Fig. 18 that as $B_{c,HEF} \rightarrow 0$, the simulated loop SNR (shown as ×) approaches the maximum achievable loop SNR of (10 + 1.5) dB, denoted by Δ in the figure. Next we turn to the aided-PLL SNR, ρ_{HEF}, which is also shown in Fig. 18 versus $B_{c,STD}/B_{c,HEF}$. The aided-loop SNR as predicted by [12, Eq. (61)], namely,

$$\rho_{HEF} = \frac{1}{\sigma^2_{\phi_{c_2}}} = \left\{ \frac{\xi}{3\,G\,\rho'_{STD}} \left[G + 2 + \frac{\gamma_2(4G+10)}{G^2+2G+5} \right] + \frac{1}{\rho'_{HEF}} \left[1 - \frac{2\gamma_2(3G+5)}{3(G^2+2G+5)} \right] \right\}^{-1} \qquad (66)$$

is shown by the top solid line in Fig. 18. The quantity ρ'_{HEF} in Eq. (66) is the nominal carrier loop SNR of the aided antenna and is equal to $((P_C/N_o)_{HEF})/B_{c,HEF}$. Keeping $B_{c,STD}$ fixed, and letting $B_{c,HEF} \rightarrow 0$, we find that $\rho_{HEF} \rightarrow \infty$, whereas the simulated results (shown as circles) approach the master-loop SNR. The simulation results for the aided loop are consistent with the theory and results for the carrier-aiding scheme in Fig. 2. Recall that in Section III we concluded that the loop SNR of the aided loop is upper bounded by that of the master loop. Interestingly, if we assume that there is perfect feedback from the aided loop so that the master loop is operating with a 1.5-dB improvement, then using Eq. (45), we can determine the upper bound on the second loop SNR, which is represented by (-·-) in Fig. 18.

A. Example: Simulating an Array of One 70-m and One 34-m Antenna

As in the two previous schemes, we present the degradation and loss for a two-element array of one 70-m and one STD 34-m antenna. The results are obtained by simulations. For comparison purposes, we use the same exact parameters used before. The symbol SNR degradation results are shown in Fig. 19, and the SER performance is presented in Fig. 20. It is observed that the degradation and loss results are better than the carrier aiding and worse than the carrier-array with a single PLL example.

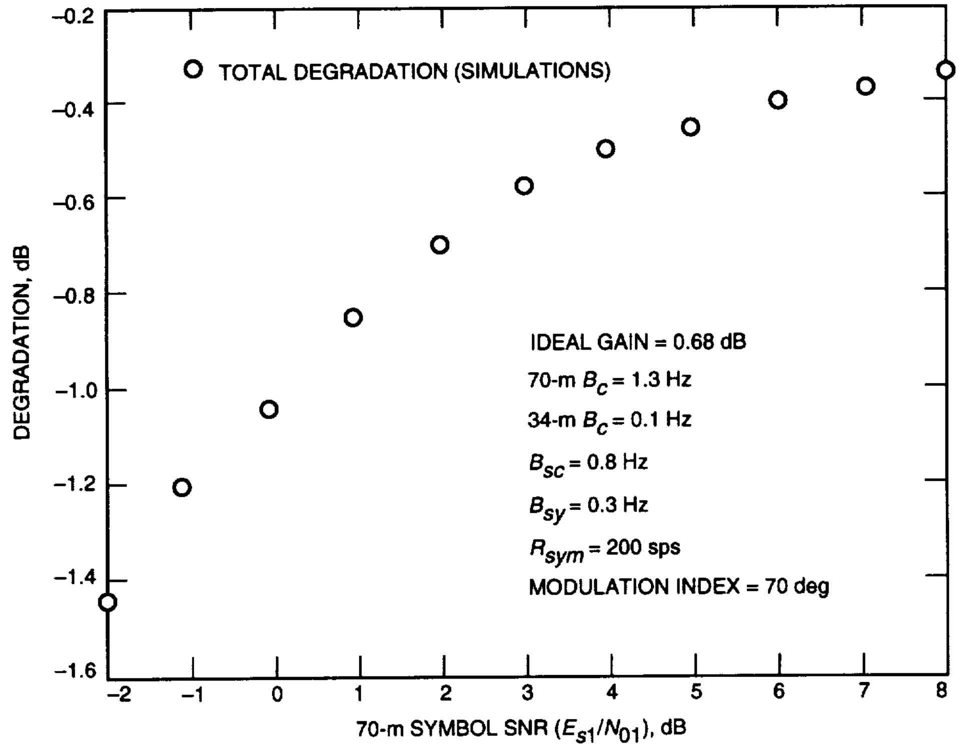


Fig. 19. Degradation (simulations).

VI. Conclusion

Three similar techniques that use carrier information from multiple antennas to enhance carrier acquisition and tracking were presented in conjunction with baseband combining. It was shown that the carrier arraying using a single carrier loop technique can acquire and track the carrier, even when any single antenna in the array cannot do so by itself. The carrier aiding and carrier arraying using multiple carrier loops techniques, on the other hand, were shown to lock the carrier only when one of the array elements has sufficient margin to acquire the carrier on its own. The tracking performance of these techniques was shown to be almost equal for medium and high data rates. For low data rates, however, carrier arraying using a single PLL has the best performance, followed by carrier arraying using multiple PLLs, and then carrier aiding.

The analytical expressions for degradation and loss of the carrier arraying using a single PLL and the carrier aiding schemes were confirmed by simulations of the end-to-end system. The carrier arraying using multiple carrier loops technique was evaluated by simulation alone.

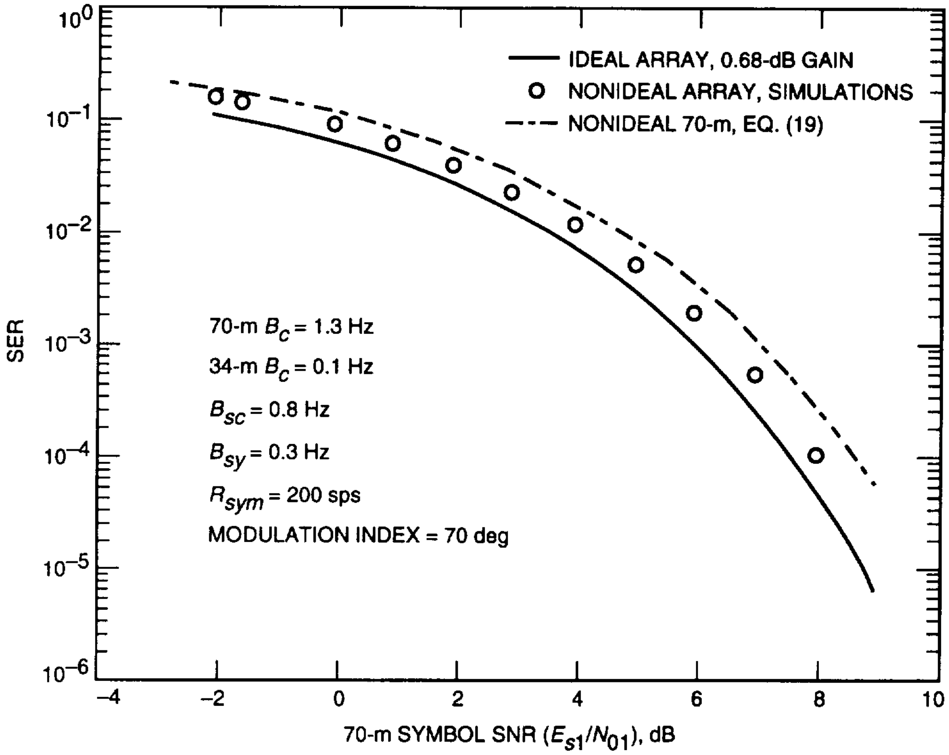


Fig. 20. SER (simulations).

Acknowledgments

We would like to thank Dr. D. Divsalar for reviewing this article and Prof. H. Tan for many helpful discussions.

References

[1] A. Mileant and S. Hinedi, "Overview of Arraying Techniques for Deep Space Communications," *IEEE Trans. on Comm.*, vol. 42, nos. 2/3/4, pp. 1856–1865, February/March/April 1994.

[2] D. Divsalar, "Symbol Stream Combining Versus Baseband Combining for Telemetry Arraying," *The Telecommunications and Data Acquisition Progress Report 42-74, April–June 1983*, Jet Propulsion Laboratory, Pasadena, California, pp. 13–28, August 15, 1983.

[3] S. Million, B. Shah, and S. Hinedi, "A Comparison of Full-Spectrum and Complex Symbol Combining Techniques for the Galileo S-Band Mission," *The Telecommunications and Data Acquisition Progress Report 42-116, October–December 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 128–162, February 15, 1994.

[4] W. Rafferty, S. Slobin, C. Stelzried, and M. Sue, "Ground Antennas in NASA's Deep Space Telecommunications," *Proceedings of the IEEE*, vol. 82, no. 5, pp. 636–645, May 1994.

[5] S. Aguirre, "Acquisition Times of Carrier Tracking Sampled Data Phase Locked Loops," *The Telecommunications and Data Acquisition Progress Report 42-84, October–December 1985*, Jet Propulsion Laboratory, Pasadena, California, pp. 88–93, February 15, 1986.

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

[7] W. J. Hurd and S. Aguirre, "A Method to Dramatically Improve Subcarrier Tracking," *IEEE Trans. on Commun.*, vol. 36, pp. 238–243, February 1988.

[8] M. K. Simon, "Analysis of the Steady-State Phase Noise Performance of a Digital Data-Transition Tracking Loop," *Space Program Summary 37-55*, vol. 3, Jet Propulsion Laboratory, Pasadena, California, pp. 54–62, February 1969.

[9] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques–Signal Design and Detection*, New Jersey: Prentice-Hall Inc., 1994.

[10] H. Wilck, "A Signal Combiner for Antenna Arraying," *JPL Deep Space Network Progress Report 42-25*, Jet Propulsion Laboratory, Pasadena, California, pp. 111–117, February 15, 1975.

[11] M. Shihabi and H. Tan, "Open Loop Residual Carrier Arraying with Baseband Combining," *IEEE GLOBECOM '94*, San Francisco, California, pp. 1040–1044, November 27–December 1, 1994.

[12] D. Divsalar and J. Yuen, "Carrier Arraying with Coupled Phase-Locked Loops for Tracking Improvement," *IEEE Trans. on Comm.*, vol. 30, no. 10, pp. 2319–2328, October 1982.

[13] T. Pham, M. Simon, T. Peng, M. H. Brockman, S. Kent, and R. Weller, "A Carrier-Arraying Demonstration at Goldstone for Receiving Pioneer 11 Signals," *The Telecommunications and Data Acquisition Progress Report 42-106, April–June 1991*, Jet Propulsion Laboratory, Pasadena, California, pp. 307–334, August 15, 1991.

[14] W. C. Lindsey and C. L. Weber, "On the Theory of Automatic Phase Control," *Stochastic Optimization and Control*, edited by H. F. Karreman, New York: John Wiley & Sons, Inc., 1968.

[15] C. L. Weber and J. J. Stein, "Cascaded Phase Locked Loops," *Proceedings of National Electronic Conference*, vol. 24, pp. 181–186, December 1968.

[16] J. Yuen, *Theory of Cascaded and Parallel Tracking Systems with Applications*, Ph.D. Dissertation, University of Southern California, Los Angeles, June 1971.

Appendix

Performance of Two Cascaded Phase-Locked Loops

The analysis of cascaded loops was considered in the past by several authors [14–16] for the purpose of determining accurate two-way Doppler and phase measurements between an antenna and a spacecraft in order to determine the relative position and velocity of the spacecraft. Here, in the carrier-aiding scheme, we are interested in determining accurately the loop SNR of the aided loop and the joint pdf of the two carrier phase error processes. Therefore, to accomplish that, we can apply the results of [14], keeping in mind that, in our case, the two cascaded loops are both in the downlink.

The proposed solution in [15], which is based on Fokker–Planck techniques and verified by simulation, takes on the following form:

$$p(x_1, x_2) = \frac{\exp\{a_2 \cos[(x_2 - m_2) - a(x_1 - m_1)] + a_1 \cos(x_1 - m_1)\}}{(2\,\pi)^2\, I_0(a_2)\, I_0(a_1)} \tag{A-1}$$

where

$$\left.\begin{aligned} a_1 &= \frac{1}{\sigma_1^2} \\ a_2 &= [\sigma_2^2(1-\rho^2)]^{-1} \\ a &= \frac{\eta\sigma_2}{\sigma_1} \end{aligned}\right\} \tag{A-2}$$

within the region

$$-\pi \le x_i \le \pi \text{ for } i = 1, 2$$

and

$$(x_1, x_2) = \begin{cases} (\hat{\theta}_1, \hat{\theta}_2) & \text{then } (m_1, m_2) = (\theta_1, \theta_2 + \hat{\theta}_1) \\ (\hat{\phi}_1, \hat{\phi}_2) & \text{then } (m_1, m_2) = (0, 0) \end{cases}$$

The σ_1^2, σ_2^2, ρ, m_1, and m_2 are the parameters of the two-dimensional Gaussian density to which either $p(\hat{\theta}_1, \hat{\theta}_2)$ or $p(\hat{\phi}_1, \hat{\phi}_2)$ converge at high SNR, which must be determined in terms of the cascaded loop system parameters in order to characterize the joint density function as given in Eq. (A-1). The results that are stated here are specialized to second-order loops with imperfect integrators and damping parameters equal to 2.

A High-Speed Photonic Clock and Carrier Regenerator

X. S. Yao and G. Lutes
Communications Systems Research Section

As data communications rates climb toward 10 Gbits/s, clock recovery and synchronization become more difficult, if not impossible, using conventional electronic circuits. The high-speed photonic clock regenerator described in this article may be more suitable for such use. This photonic regenerator is based on a previously reported photonic oscillator capable of fast acquisition and synchronization. With both electrical and optical clock inputs and outputs, the device is easily interfaced with fiber-optic systems. The recovered electrical clock can be used locally and the optical clock can be used anywhere within a several kilometer radius of the clock/carrier regenerator.

I. Introduction

In high-speed fiber-optic communications systems, the ability to recover the clock from the incoming random data is essential. The recovered clock must be in precise synchronism with the incoming data and is used in further signal processing systems, such as regenerative repeaters, time division switching systems, and demultiplexers.

Conventional clock recovery devices are generally based on electronic phase-locked loops (PLLs) [1]. These devices may not be suited for the high-speed fiber-optic communications system because of their relatively slow speed, slow acquisition time, narrow tracking range, inability to be tuned over a wide range of frequencies, and non-optical inputs and outputs. Having optical inputs and outputs is important because it makes interfacing with a fiber-optic system easier.

All optical clock recovery schemes proposed by many authors [2–6] are based on injection locking a pulsed laser with the incoming data stream, wherein the pulsed laser has a nominal pulsation rate close to the incoming data rate. In one scheme, the pulsed laser [2–4] is a mode-locked fiber ring laser, and the input data modulates the laser cavity length or loss via the optical nonlinear effect. Because optical nonlinearity is used, the intensity of the injection data has to be high and is, therefore, not practical in many applications. In another scheme, the pulsed laser is a self-pulsating semiconductor laser [5,6] where the self-pulsation is caused by self-Q-switching within the device. The pulsation rate can be controlled by varying the current to the device. The problems associated with such a device are the relatively low speed (a few GHz) and relatively high noise.

Although the concept of all optical systems is attractive, the majority of present and future systems will be hybrid, meaning that the system can be controlled and accessed both optically and electronically.

Therefore, a clock recovery device having such a hybrid capability is important, and in this article we report such a device—the photonic clock regenerator. We also show that the same device can be used for high-frequency carrier recovery and will be useful in fiber-optic analog communications systems.

The photonic clock and carrier regenerator is based on the photonic oscillator described in an earlier article [7]. As shown in Fig. 1, functionally it is a six-port device with an optical and an electrical injection port, an optical and an electrical output port, and two voltage-controlling ports for tuning frequency. The incoming data are injected into the photonic oscillator either optically or electrically. The free-running photonic oscillator is tuned to oscillate at a nominal frequency close to the clock frequency of the incoming data. With the injection of the data, the photonic oscillator will be quickly phase locked to the clock frequency of the data stream while rejecting other frequency components (harmonics and subharmonics) associated with the data. Consequently, the output of the locked photonic oscillator is a continuous periodic wave synchronized with the incoming data, or simply the recovered clock.

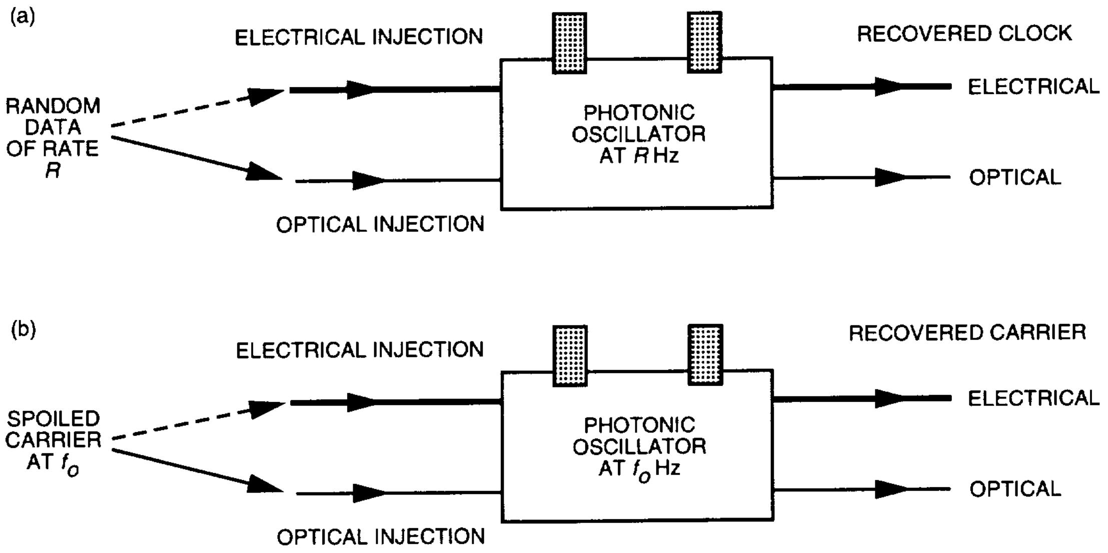


Fig. 1. Functions of the photonic clock and carrier regenerator: (a) clock recovery and (b) carrier recovery.

II. Clock Recovery Demonstration

Figure 2 shows the clock recovery experiment setup. An HP 8080 Word Generator System was used to generate a stream of repetitive 64-bit words at 100 Mbits/s, and the photonic oscillator was tuned to oscillate at 100 MHz. The data were injected into the bias port of the electro-optical (E/O) modulator through a filter and a bias T. The filter was centered at 100 MHz with a 3-dB bandwidth of 10 MHz. It was used to reduce unwanted frequency components of the input data. The output of the photonic oscillator was fed either into a spectrum analyzer (HP 8562) or an oscilloscope (Tektronix 2465B). When using the oscilloscope, the first bit of each word was used to trigger the sweep so the whole word could be displayed.

Note that although a photonic clock regenerator is capable of recovering a clock at much higher frequencies (up to 70 GHz), due to equipment constraints we chose to demonstrate clock recovery at 100 MHz to make our measurements easier. With the HP 8080 system, the data pattern can easily be selected to be either return-to-zero (RZ) or non-return-to-zero (NRZ), so both types of data were tested in our experiments. The selected 64-bit word was 0010101101101001 0110001110101101 1011001010001010 0010101100101000. The clock recovery is independent of the word chosen, as long as it is balanced.

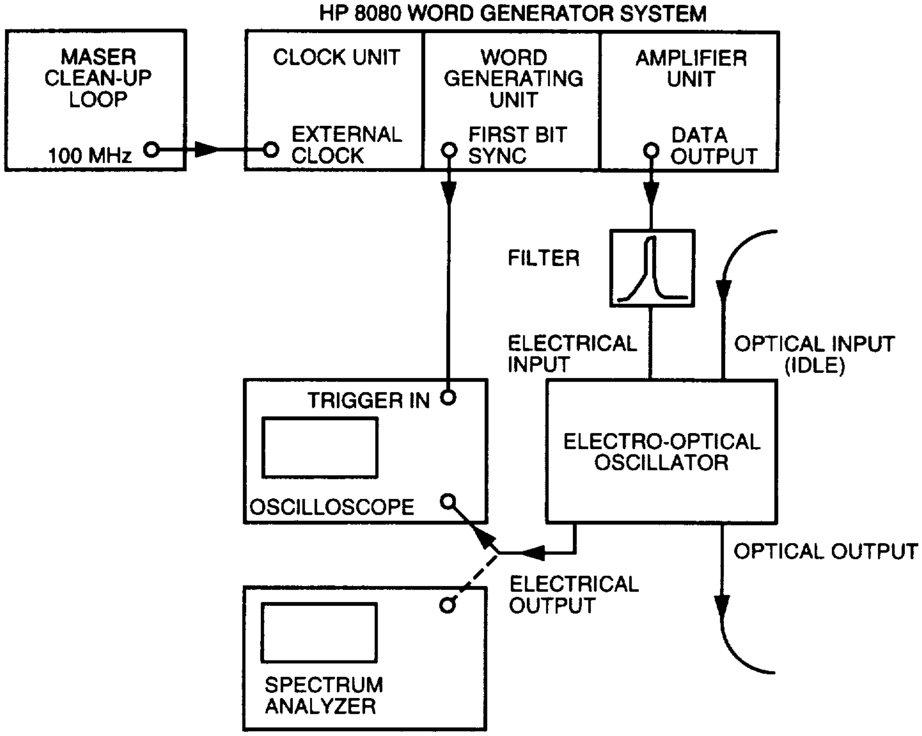


Fig. 2. Clock recovery experiment setup.

Figure 3 shows the experiment results in the frequency domain that demonstrated successful clock recovery from an NRZ data stream. The frequency spectrum of the input data is measured with the 100-MHz filter and is identical to the injected signal. As one can see, the selected NRZ data stream has some frequency components stronger than the clock frequency. After clock regeneration, the recovered clock is 62-dB stronger than the strongest harmonic component. Figure 4 shows the same experiment results in the time domain. Figure 4(a) shows the traces of the input data (lower trace) and the trigger signal (upper trace). Figure 5 contains the same information as Fig. 4, except that the time span is reduced 10 times so that the details of the traces can be seen. It is evident that the recovered clock is a perfect sine wave. The fact that the recovered clock can be clearly displayed on the oscilloscope when the first bit of data is used as the trigger indicates that the recovered clock is synchronized with the data. If the photonic oscillator is not locked to the data (free running), its phase wanders relative to the data bits. As a result, the display of the photonic oscillator's output signal on the oscilloscope is smeared when any data bit is used to trigger the oscilloscope, as shown in Fig. 6. Note the recovered clock level is almost independent of the input signal level, a feature that is desirable for clock recovery and is inherent in injection-locked oscillators. Other proposed high-speed clock recovery circuits use automatic gain control and limiting amplifiers to achieve constant amplitude [8].

We have also successfully demonstrated photonic clock recovery from RZ formatted data. Because the RZ data have a higher level of the clock frequency component, recovering the clock is more straightforward than recovering the clock from NRZ data. Similar results are expected for optical injection since the data in the optical domain will be automatically converted by the internal photodetector into the electrical domain before affecting the photonic oscillator. Note that for infinitely long NRZ random data, the clock frequency component is zero. In order to recover the clock from such a data stream, a procedure to convert NRZ data format to RZ format is required [1].

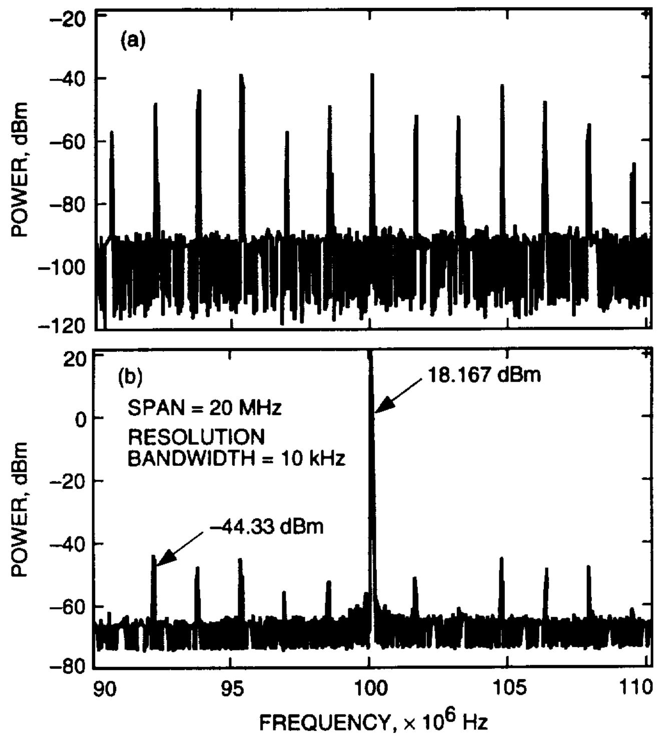


Fig. 3. Clock recovery from an NRZ data stream measured in the frequency domain: (a) random data input and (b) recovered clock.

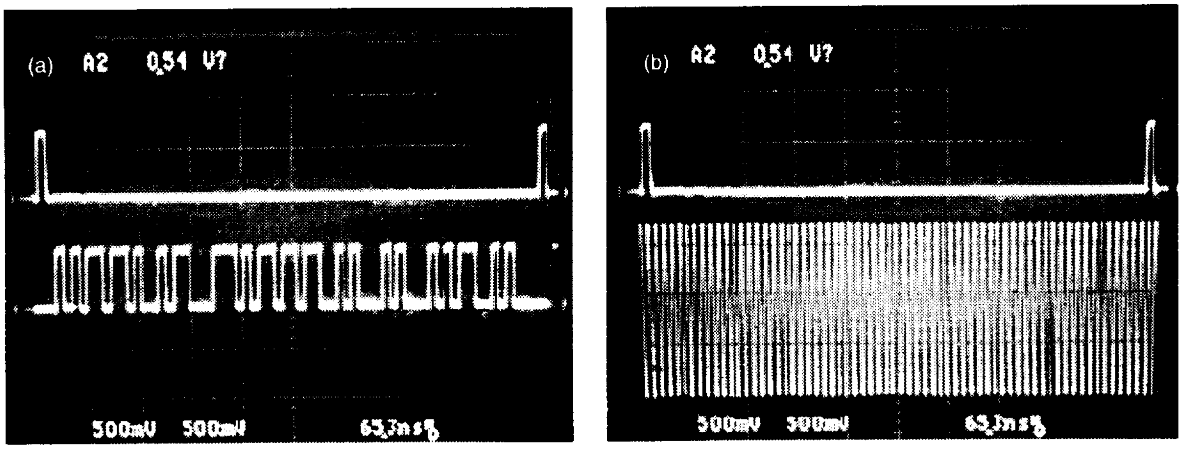


Fig. 4. Clock recovery from an NRZ data stream measured in the time domain: (a) random data input and (b) recovered clock.

III. Carrier Recovery Demonstration

Similar to clock recovery, a carrier buried in noise can also be recovered by the photonic oscillator. To do so, we simply inject the spoiled carrier into the photonic oscillator that has a free-running frequency close to the carrier frequency and an output power level $N(N >> 1)$ dB higher than the carrier level. The injected carrier forces the photonic oscillator to be locked with the carrier and results in an equivalent carrier gain of N dB. Because the open-loop gain of the photonic oscillator is only n dB ($n \approx 1$), the noise of the input is amplified by only n dB and the signal-to-noise ratio of the carrier is then increased by $(N - n)$ dB.

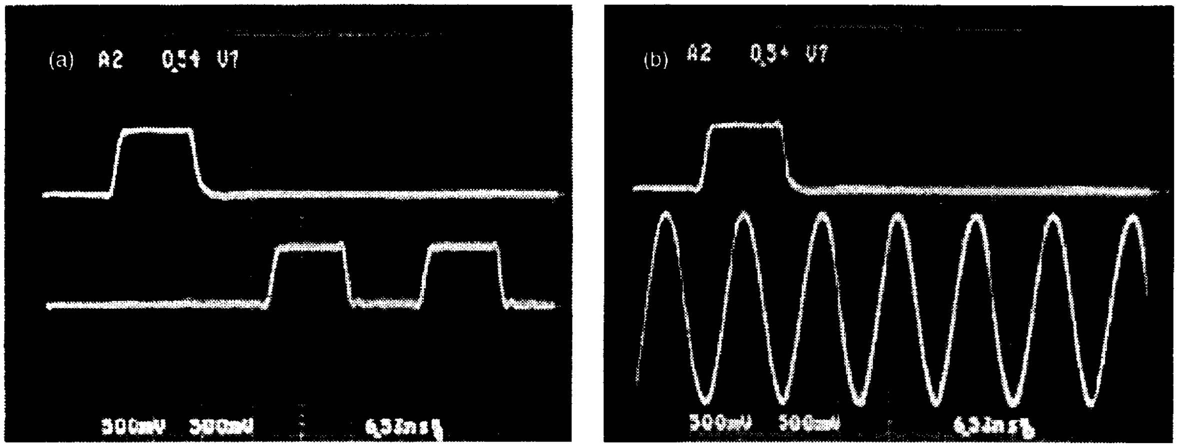


Fig. 5. The data trace of Fig. 4, with the time span reduced 10 times: (a) random data input and (b) recovered clock.

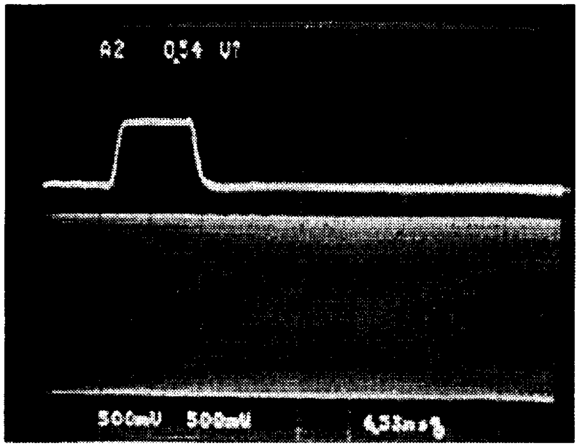


Fig. 6. The data trace where the photonic oscillator is not locked to the data.

Figure 7 is the experiment setup for demonstrating the photonic carrier recovery. In the experiment, a clean 100-MHz carrier from an H-maser frequency standard and a clean-up loop is combined with a noise source consisting of two noisy amplifiers in series. The resulting spoiled carrier was measured using the spectrum analyzer and is shown in Fig. 8(a). Figure 8(b) shows the spectrum of the recovered carrier, and it is evident from the figure that the signal-to-noise ratio of the carrier is increased by more than 50 dB. We also measured the spoiled carrier and recovered carrier in the time domain with an oscilloscope, and the results are shown in Fig. 9. In both Fig. 9(a) and Fig. 9(b), the upper trace (a square pulse) is the trigger signal and the lower trace is the carrier. Comparison of the two figures clearly demonstrates the effectiveness of the photonic oscillator as a carrier recovery device.

IV. Attractive Properties of the Photonic Clock and Carrier Regenerator

Our experiment results and analysis indicate that the photonic clock and carrier regenerator described above has the following attractive properties:

(1) High-speed or high-frequency operation. The speed of the device can be as high as 70 GHz and is limited only by the speed of the photodetector and the E/O modulator used. We have demonstrated a photonic oscillator operating as high as 9.2 GHz.

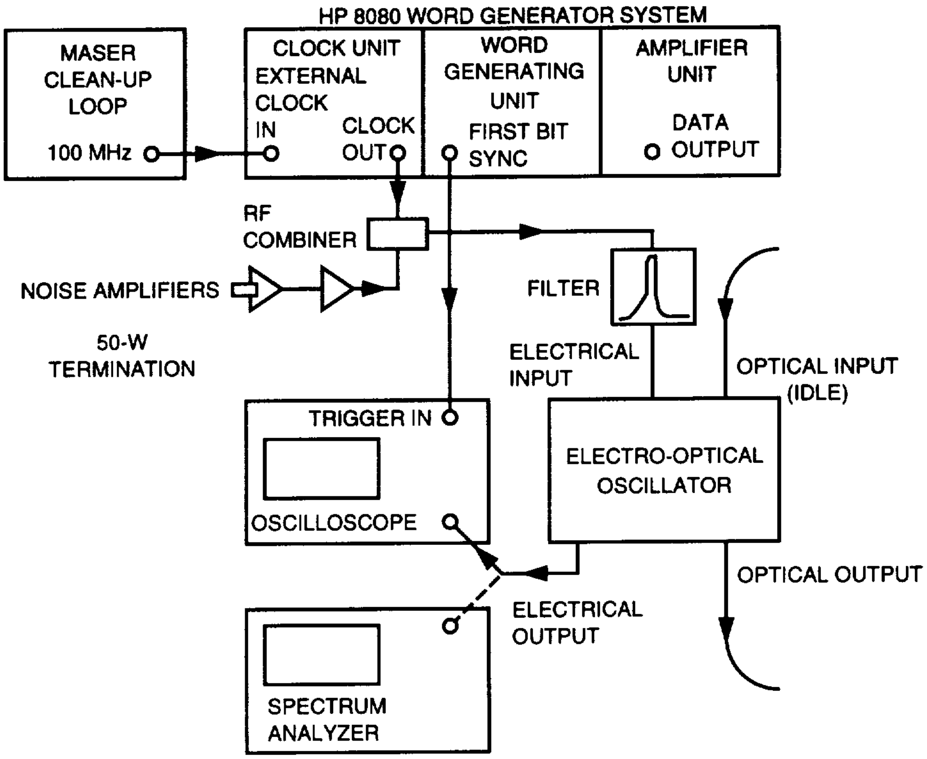


Fig. 7. Experiment setup for demonstrating photonic carrier recovery.

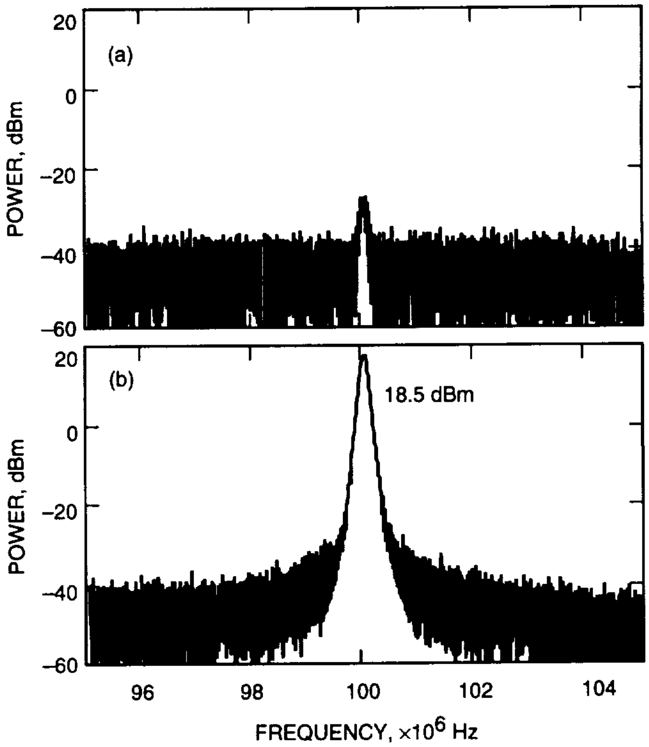


Fig. 8. Carrier recovery measurement in the frequency domain: (a) spoiled carrier and (b) recovered carrier.

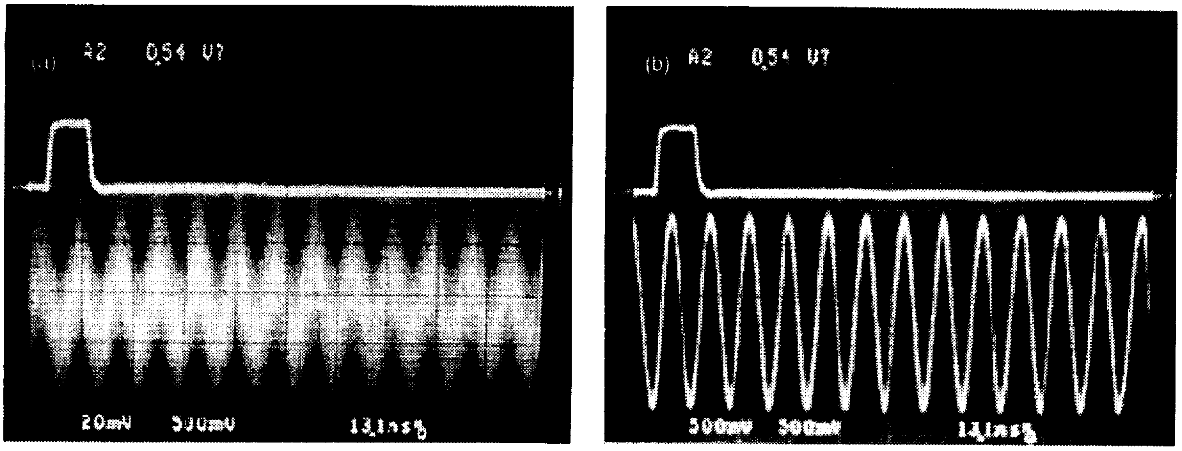


Fig. 9. Carrier recovery measurement in the time domain: (a) spoiled carrier and (b) recovered carrier.

The reason for choosing 100 MHz to demonstrate the clock and carrier recovery in the experiments above is because the measurement equipment we have (word generator, oscilloscope, and reference clock) operate around 100 MHz.

(2) The amplitude of the recovered signal (clock or carrier) is constant. It is independent of the input power of the signal to be recovered. This feature is especially important in clock recovery because the clock component contained in the received data stream varies with time and with sender (in a time division multiplexing system). The photonic clock regenerator ensures that the recovered clock has a constant power level at all times.

(3) The photonic clock and carrier regenerator can be accessed both optically and electronically. It has both electrical and optical inputs and outputs. This feature makes the device attractive in terms of easy interfacing with a complex fiber-optic communication system.

(4) Fast acquisition time for phase locking. Because the photonic clock and carrier regenerator is based on injection locking, its acquisition time is much faster than that of a clock recovery device based on a phase-locked loop [9]. Fast acquisition is important for high-speed telecommunications, especially for burst-mode communication. The estimated acquisition time is on the order of a microsecond or faster.

(5) Wide tracking range. The tracking range of the photonic clock and carrier regenerator is on the order of a few percent of the clock frequency, compared to a few tens of Hz for a clock recovery device based on a phase-locked loop. Having a wide tracking range makes the implementation of the device easier because the device does not have to be tuned precisely to match the incoming data rate.

(6) Frequency tunability. Unlike many other kinds of oscillators that can be tuned in only a narrow frequency band, the photonic oscillator can be tuned over many tens of MHz by changing the filter in the feedback loop and fine tuned by simply changing the loop delay or bias point of the E/O modulator. Delay line oscillators maintain high Q in spite of their ability to be tuned over a wide frequency range. This feature makes the device flexible in accommodating different systems, designs, and signal conditions.

(7) The device can be integrated on a chip. All of the key components of the device, such as the laser, the amplifier, the E/O modulator, and the photodetector can all be based on the GaAs technology and can be fabricated on the same substrate.

V. Applications

Figure 10 shows a clock recovery, synchronization, and signal recovery system based on the clock regenerator described here. An optical carrier containing high data-rate digital information arrives from a remote location and is split into two paths. One of these signals is injected into the photonic clock regenerator and the other signal is delayed in an optical delay line. The delay line is used to delay the received signal long enough for the clock regenerator to lock up so no data bits will be lost from the leading edge of the digital data stream. The recovered electrical clock is applied to the data recovery device in synchronization with the received signal, permitting the digital data to be recovered.

The recovered optical clock can be transmitted over optical fiber to be used by other devices within a several-kilometers area. This negates the need to have multiple clock recovery systems in a complex. Because of the high loss and dispersion of metallic transmission lines, it is not practical to use them to distribute a recovered 10-GHz clock over more than a few tens of meters.

In Fig. 11, the carrier regenerator is used as a clean-up loop for an analog frequency reference signal transmitted from a remote frequency reference. Again the regenerator has both an electrical and an optical output, so once the frequency reference is regenerated, it can be distributed locally over optical fiber.

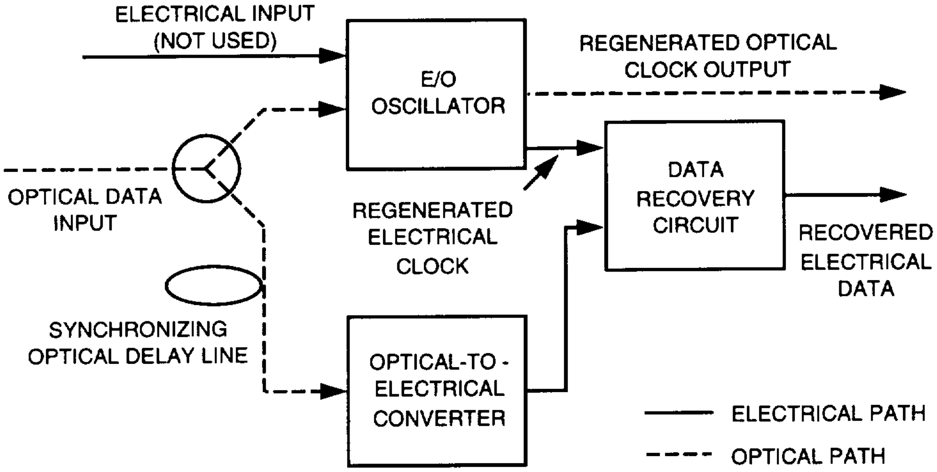


Fig. 10. Clock regenerator and data recovery system.

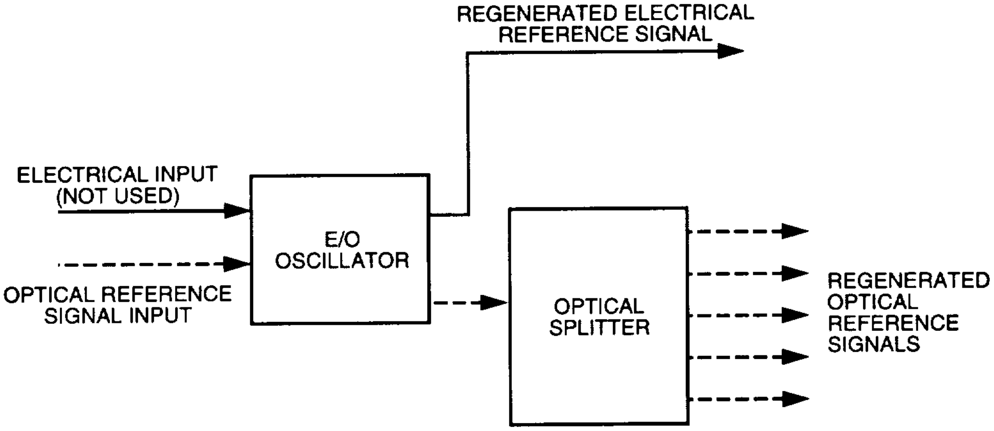


Fig. 11. Optical frequency reference regeneration and distribution.

References

[1] D. Wolever, *Phase-Locked Circuit Design*, Englewood Cliffs, New Jersey: Prentice Hall, 1991.

[2] K. Smith and J. K. Lucek, "All-Optical Clock Recovery Using a Mode-Locked Laser," *Electronic Letters*, vol. 28, no. 19, pp. 1814–1816, 1992.

[3] A. D. Ellis, K. Smith, and D. M. Patrick, "All Optical Clock Recovery at Bit Rates Up to 40 Gb/s," *Electronic Letters*, vol. 29, no. 15, pp. 1323–1324, 1993.

[4] D. M. Patrick and R. J. Manning, "20 Gb/s All-Optical Clock Recovery Using Semiconductor Nonlinearity," *Electronic Letters*, vol. 30, no. 2, pp. 15–152, 1994.

[5] P. E. Barnsley, H. J. Wicks, G. E. Wickens, and D. M. Spivit, "All-Optical Clock Recovery From 5 Gb/s RZ Data Using a Self-Pulsating 1.56 μm Laser Diode," *IEEE Photonics Technology Letters*, vol. 3, no. 10, pp. 942–945, 1991.

[6] M. Jinno and T. Matsumoto, "All-Optical Timing Extraction Using a 1.5 μm Self-Pulsating Multielectrode DFB LD," *Electronic Letters*, vol. 24, no. 23, pp. 1426–1427, 1988.

[7] X. S. Yao and L. Maleki, "High Frequency Optical Subcarrier Generator," *Electronic Letters*, vol. 30, no. 18, pp. 1525–1526, 1994.

[8] H. Ichino, M. Togashi, M. Ohhata, Y. Imai, N. Ishihata, and G. Sano, "Over 10 Gb/s ICs for Future Light Wave Communications," *J. Lightwave Technology*, vol. 12, no. 2, pp. 308–319, 1994.

[9] V. Vzunoglu and M. H. White, "The Synchronous Oscillator: A Synchronization and Tracking Network," *IEEE J. Solid State Circuits*, SC-2016, pp. 1214–1226, 1985.

TDA Progress Report 42-121

May 15, 1995

Effects of Correlated Noise on the Full-Spectrum Combining and Complex-Symbol Combining Arraying Techniques

P. Vazirani
Communications Systems Research Section

The process of combining telemetry signals received at multiple antennas, commonly referred to as arraying, can be used to improve communication link performance in the Deep Space Network (DSN). By coherently adding telemetry from multiple receiving sites, arraying produces an enhancement in signal-to-noise ratio (SNR) over that achievable with any single antenna in the array. A number of different techniques for arraying have been proposed and their performances analyzed in past literature [1,2]. These analyses have compared different arraying schemes under the assumption that the signals contain additive white Gaussian noise (AWGN) and that the noise observed at distinct antennas is independent.

In situations where an unwanted background body is visible to multiple antennas in the array, however, the assumption of independent noises is no longer applicable. A planet with significant radiation emissions in the frequency band of interest can be one such source of correlated noise. For example, during much of Galileo's tour of Jupiter, the planet will contribute significantly to the total system noise at various ground stations. This article analyzes the effects of correlated noise on two arraying schemes currently being considered for DSN applications: full-spectrum combining (FSC) and complex-symbol combining (CSC). A framework is presented for characterizing the correlated noise based on physical parameters, and the impact of the noise correlation on the array performance is assessed for each scheme.

I. Introduction

Arraying spacecraft telemetry has a number of desirable applications in the Deep Space Network. By combining signals from multiple antennas, arraying has the benefit of increasing the signal-to-noise ratio (SNR) of the combined signal over that achievable with any individual antenna in the array. Arraying may be used to coherently track signals that are too weak to be tracked by a single antenna or to allow an increase in the supportable data rate for stronger signals. Several different schemes for performing arraying have been proposed and analyzed in past literature [1,2]. These schemes differ in the synchronization processes that are used to combine and demodulate the signals. Thus, a benchmark used to compare different arraying schemes is symbol SNR degradation, which is a measure of the SNR reduction due to imperfect synchronization for a particular scheme.

Previous analyses that have compared arraying techniques in terms of symbol SNR degradation have used an additive white Gaussian noise (AWGN) model to describe the deep-space channel and have assumed the noise waveforms received at distinct antennas are independent. However, if a strong radio source is within the antenna pattern of multiple antennas in the array, the noise observations at different antennas become correlated. For a substantial fraction of Galileo's encounter with Jupiter, for example, the planet will have an angular separation from the spacecraft that is less than the beamwidth of a 70-m antenna at S-band (2.3 GHz).[1] Further analysis is thus needed to characterize the performance of arraying schemes in cases where correlated noise is present.

Prior work has been conducted on this subject but has not exhausted research possibilities. A study by Dewey [3] examines correlated noise effects due to planetary sources, focusing mainly on physical considerations. A correlated noise model is presented, taking into account properties of the source and the array geometry. The impact of the background source on arrayed symbol SNR relative to a case of uncorrelated noise is then analyzed. The results obtained are applied to observation of the Galileo spacecraft from a four-element array in the DSN's Australia complex. However, Dewey's study does not take into account the effects of imperfect synchronization in telemetry arraying, which are dependent on the specific arraying technique used. Thus, the analysis does not identify the relative advantages and disadvantages of different arraying schemes under conditions of correlated noise.

The purpose of this article is to analyze the effects of correlated noise on the full-spectrum combining (FSC) and complex-symbol combining (CSC) arraying schemes. In Section II, background material needed to understand the physics underlying background noise in receiving systems is presented. Parameters used to characterize the noise correlation properties will be introduced and explained. Sections III and IV then apply this model to the FSC and CSC techniques and compute the symbol SNR degradation for each scheme. Section V applies the results of the previous sections to the Galileo mission. Predicts for the signal and noise parameters are used to evaluate the performance of both arraying schemes in this scenario. Finally, Section VI summarizes the main results of the work.

II. Background Noise Properties

Here we present basic terminology used to describe broadband sources that will be used for the remainder of the analysis. The discussion that follows is included only to summarize major results from previous work; readers interested in a more thorough treatment of the subject material may refer to a text on radio astronomy, such as [4], or the work performed by Dewey alluded to earlier [3].

Consider first the effect of a background source on a single receiving system. The noise observed at an antenna consists of both thermal noise due to front-end receiver electronics and radiation due to any radio sources in the antenna's field of view. Such sources typically have an emission spectrum that varies very slowly with frequency and can, therefore, be considered white over the bandwidth of interest.[2] The increase in total system temperature due to the background source is found by integrating the source's brightness distribution over the antenna's reception pattern, i.e.,

$$T_s = \frac{A_e}{2k} \int\int B(\hat{s}) P_N(\hat{s})\, d\hat{s} \qquad (1)$$

where A_e is the effective receiving area of the antenna in m^2; k is Boltzmann's constant, 1.379×10^{-23} W/K/Hz; $B(\hat{s})$ is the brightness of the source in W/m^2/Hz/sr (sr stands for steradian, a measure of solid angle); $P_N(\hat{s})$ is the normalized antenna reception pattern; and $\hat{s}$ is a unit vector specifying direction.

[1] G. Resch, "Jupiter's Contribution to the Total System Temperature at S-Band During the Galileo Mission," JPL Interoffice Memorandum 335.3-92.02 (internal document), Jet Propulsion Laboratory, Pasadena, California, June 23, 1993.

[2] Ibid.

The one-sided power spectral density of the noise due to the source is then given by $N_s = kT_s$. Note that in the upper limit, when the source is concentrated in the peak of the antenna's reception pattern, the temperature increase is given by

$$T_s = \frac{A_e}{2k} \int\int B(\hat{s})\, d\hat{s} \tag{2}$$

$$= \frac{A_e}{2k} S \tag{3}$$

where S is the total flux density of the source in $W/m^2/Hz$. As the angular separation between the source and the spacecraft increases, the background source moves out of the peak of the antenna pattern, and its temperature contribution diminishes. In addition, the flux density for a particular source is dependent on its distance to Earth; the greater the range, the smaller the observed flux is. Thus, the temperature contribution for a body depends on both its strength and its position.

Now consider a pair of antennas physically separated by a baseline vector $\vec{B}_{ik}$ observing a common source. The cross-correlation function for the baseband (BB) noise processes $\tilde{n}_i(t)$ and $\tilde{n}_k(t)$ can be written as

$$R_{\tilde{n}_i,\tilde{n}_k}(\tau) \triangleq E[\tilde{n}_i(t)\tilde{n}_k^*(t-\tau)] = \alpha\, \frac{\sin(2\pi B\tau)}{\pi\tau} \tag{4}$$

where B is the one-sided bandwidth of the noise waveforms, and α is their cross-power spectral density. If the bandwidth B is wider than the telemetry bandwidth, then the cross-spectrum is white over the bandwidth of interest, and the "sinc" function $\sin(2\pi B\tau)/(\pi\tau)$ can be approximated by an impulse function, i.e.,

$$R_{\tilde{n}_i,\tilde{n}_k}(\tau) = \alpha\, \delta(\tau) \tag{5}$$

It can be shown [3,4] that the cross-power spectral density level is given by

$$\alpha = \frac{\sqrt{A_{e_i}A_{e_k}}}{2} \int\int B(\hat{s})\sqrt{P_{N_i}(\hat{s})P_{N_k}(\hat{s})}e^{j2\pi f_o \vec{B}_{ik}\cdot\hat{s}/c}\, d\hat{s}$$

$$= \frac{\sqrt{A_{e_i}A_{e_k}}}{2}|V|e^{j\phi_v} \tag{6}$$

where f_o is the observation frequency, and c is the speed of light, 3×10^8 m/s. In radio interferometry applications, the quantity $|V|e^{j\phi_v}$ is known as the complex visibility of the source. A few important observations regarding Eq. (6) are made here. First, note that the exponential term $e^{j2\pi f_o \vec{B}_{ik}\cdot\hat{s}/c}$ produces a sinusoidal variation over the spatial extent of the source. This variation is known as the fringe pattern formed by a particular pair of antennas. The period of these fringe oscillations is given in radians/cycle by $c/f_oB_{ik_p}$, where B_{ik_p} is the projected baseline length in the direction of the source. If a source has an angular size much greater than the fringe period, the cross-correlation magnitude then tends to zero due to the averaging effect of the sinusoid. Thus, in the long baseline limit (i.e., $B_{ik_p} \gg c/(f_oR_s)$, with R_s being the angular radius of the source), $|\alpha| \to 0$, and the noise observations due to the source become uncorrelated. By contrast, for $B_{ik_p} \ll c/(f_oR_s)$, the magnitude of the cross-power spectral density achieves its upper limit, namely

$$|\alpha| \rightarrow \frac{\sqrt{A_{e_i} A_{e_k}}}{2} S \tag{7}$$

Thus, the degree of noise correlation observed by an array of antennas depends heavily on the geometry of the array. This point is stressed in [3], where it is stated that the more compact the array configuration, the greater the impact of a background body on the array.

Finally, we introduce the correlation coefficient, describing the degree of correlation that exists between the noise at two antennas, defined as

$$\rho_{ik} \triangleq \frac{|\alpha|}{\sqrt{N_{o_i} N_{o_k}}} \tag{8}$$

Note that in the upper limit (i.e., source size small compared to fringe period), the correlation coefficient becomes

$$\rho_{ik} \rightarrow \sqrt{\frac{T_{s_i} T_{s_k}}{T_i T_k}} \tag{9}$$

where T_{s_i}, T_{s_k} are the source temperatures at antennas i, k, and T_i, T_k are the *total* system temperatures at the two antennas. Thus, the greater the contribution of the source to the total system temperature, the higher the correlation coefficient, as is intuitively expected.

Combining Eqs. (5), (6), and (8), the cross-correlation function for the noise observed at two antennas can be expressed as

$$R_{\tilde{n}_i,\tilde{n}_k}(\tau) = \rho_{ik} \sqrt{N_{o_i} N_{o_k}}\; e^{j\phi^n_{ik}}\; \delta(\tau) \tag{10}$$

where ϕ^n_{ik} is used to express the correlation phase, denoted by ϕ_v in Eq. (6).

III. Full-Spectrum Combining Performance

Given a mathematical description of noise correlation properties, we now apply the model to analyzing correlated noise effects on arraying. Full-spectrum combining is described in detail in [2] and summarized here briefly. Assume the array consists of L antennas, where antenna 1 is taken to be the "master" antenna (i.e., the antenna with the highest G/T.) As shown in Fig. 1, each signal is first downconverted to baseband[3] by local oscillators in phase quadrature. Each signal pair, which can be thought of as a single complex signal, is then shifted in time by some amount $\hat{\tau}_i$ to compensate for differing arrival times of the spacecraft signal at the various antennas. The complex baseband signals are then aligned in phase, multiplied by prespecified weighting factors, and added. Finally, the combined signal is processed by a single carrier, subcarrier, and symbol loop.

Two quantities used to describe arraying performance are the ideal arraying gain, denoted by G_A, and the symbol SNR degradation, denoted by D. The arraying gain is defined as the ratio of the ideal symbol SNR of the arrayed signal to the ideal symbol SNR of antenna 1 [1]. Here, "ideal" means the

[3] Analysis presented in [2] actually assumes all processing is done at an intermediate frequency, rather than at baseband. A baseband system was assumed here to simplify the analysis. This represents no loss of generality, since final results are not dependent on what frequency processing is done at.

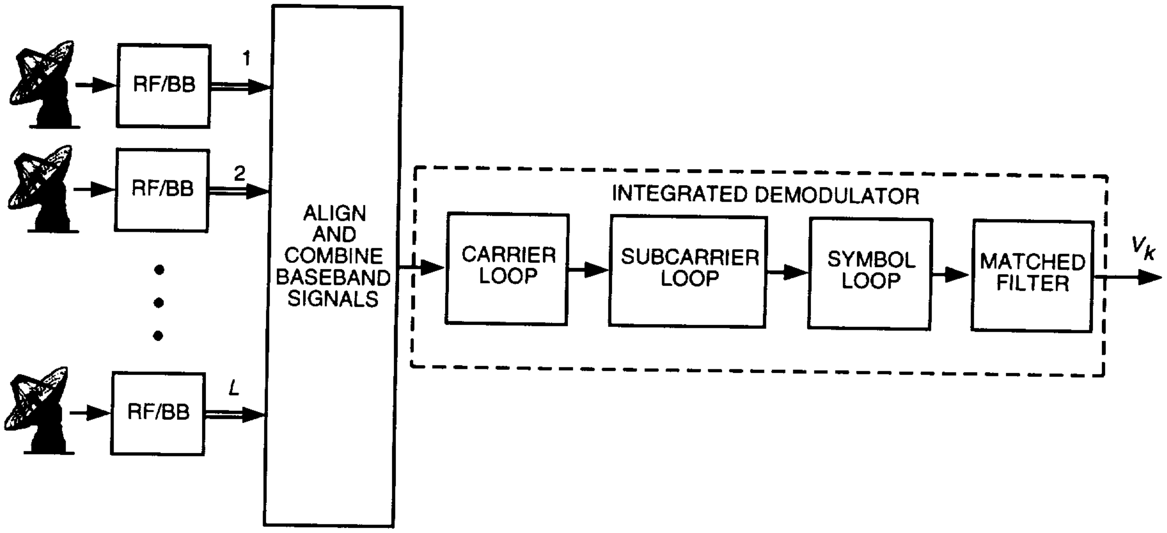


Fig. 1. Full-spectrum combining.

symbol SNR that would be achieved in the absence of synchronization errors (i.e., perfect signal combining and perfect carrier, subcarrier, and symbol references.) Note that the arraying gain G_A is *independent* of which arraying technique is used, since synchronization losses are ignored. Thus, G_A describes the maximum SNR enhancement that can be achieved by arraying, but is not useful for evaluating the relative performance of one arraying scheme over another. The ideal arraying gain is computed in [1] for a set of antennas observing independent noise waveforms. Our first step in evaluating the impact of a background body on arraying will be to compute G_A for the case of correlated noise. This analysis is analogous to that found in [3], although the notation adopted here is different.

Degradation is defined as the ratio of the actual SNR of the arrayed telemetry to that achieved with perfect synchronization (i.e., the ideal SNR). Clearly, degradation is dependent on which arraying scheme is used, since synchronization losses depend on the specific processing used to combine and demodulate the signals. Degradation for full-spectrum combining and complex-symbol combining was computed in [2], also under the assumption of independent noises. Thus, the second step in analyzing correlated noise effects will be to derive degradation expressions for the two schemes.

A. Ideal Arraying Gain

The signal format for deep-space telemetry is binary phase shift keyed (BPSK) employing a squarewave subcarrier. After time alignment, the IF signal from the ith antenna can be expressed as [1]

$$\begin{aligned} y_i(t) &= s_i(t) + n_i(t) \\ &= \sqrt{2P_{T_i}} \cos\left(\omega_{IF}t + \theta_i + \Delta d(t)\ \mathrm{sqr}(\omega_{sc}t + \theta_{sc})\right) + n_i(t) \\ &= \sqrt{2P_{C_i}} \cos\left(\omega_{IF}t + \theta_i\right) - \sqrt{2P_{D_i}} d(t)\ \mathrm{sqr}(\omega_{sc}t + \theta_{sc}) \sin(\omega_{IF}t + \theta_i) + n_i(t) \end{aligned} \tag{11}$$

where P_{T_i} is the total signal power in watts; ω_{IF} is the intermediate frequency in radians/s; θ_i is the carrier phase in radians; Δ is the modulation index in radians; $d(t)$ is the binary data, taking on values of ± 1; $\mathrm{sqr}(x)$ is the squarewave function, given by $\mathrm{sqr}(x) = \mathrm{sgn}(\sin x)$; ω_{sc} is the subcarrier frequency in radians/s; θ_{sc} is the subcarrier phase in radians; P_{C_i} is the carrier power in watts, given by $P_{C_i} = P_{T_i} \cos^2 \Delta$; P_{D_i} is the data power in watts, given by $P_{D_i} = P_{T_i} \sin^2 \Delta$; and $n_i(t)$ is an additive white

Gaussian noise process with one-sided power spectral density N_{o_i} W/Hz. The corresponding complex baseband signal is given by

$$\tilde{y}_i(t) = \tilde{s}_i(t) + \tilde{n}_i(t) \tag{12}$$

$$= \sqrt{P_{C_i}} e^{j(\omega_b t + \theta_i)} + j\sqrt{P_{D_i}} d(t) \ \mathrm{sqr}(\omega_{sc} t + \theta_{sc}) e^{j(\omega_b t + \theta_i)} + \tilde{n}_i(t) \tag{13}$$

where ω_b is the baseband frequency (which, by definition, is close to zero), and $\tilde{n}_i(t)$ is the complex baseband noise, the real and imaginary parts of which each has one-sided power spectral density N_{o_i}. The spectrum of the baseband telemetry is shown in Fig. 2.

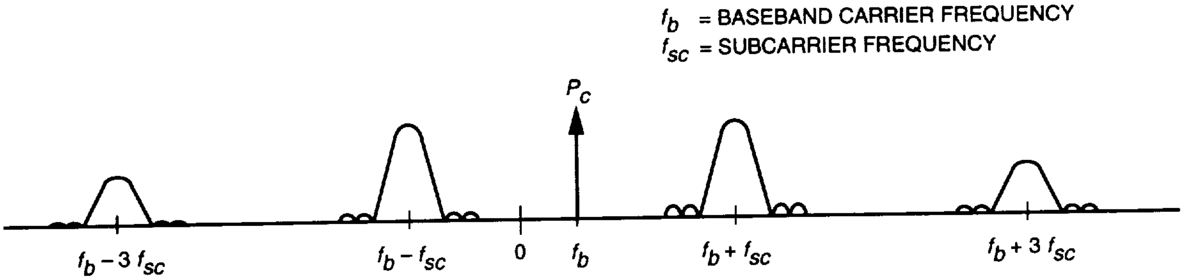


Fig. 2. Spectrum of the baseband telemetry signal.

Note that the bandwidth needed to transmit the signals $\tilde{y}_i(t)$ to a common location for combining is determined by the subcarrier frequency, $f_{sc} = \omega_{sc}/2\pi$, and is much greater than the actual data rate. As an alternative to the method described in [2], a version of FSC that only transmits portions of the spectrum containing signal energy can be used to reduce this bandwidth requirement. Each signal can be passed through a bank of matched filters separately, passing the subcarrier harmonics with the data modulation; the total transmission bandwidth is then proportional to the data rate. This alternative is mentioned briefly in [1]. However, the drawback of such a system is that the processing required is dependent on the subcarrier frequency and data rate and must be modified for each mission. For simplicity, we will focus on the more basic implementation of FSC described in [2], keeping in mind that a more bandwidth-economizing option also exists.

Let the phase difference between the 1st and ith signal be denoted by $\phi_{1i} = \theta_1 - \theta_i$. In the algorithm described in [1], signals 2 through L are phase rotated by an estimate of this quantity, $\hat{\phi}_{1i}$, to align them with signal 1. The aligned signals are then multiplied by prespecified weighting factors, β_i, and summed. The combined signal is thus given by

$$\tilde{y}_{comb}(t) = \tilde{s}_{comb}(t) + \tilde{n}_{comb}(t) \tag{14}$$

$$= \sum_{i=1}^{L} \beta_i e^{j\hat{\phi}_{1i}} \tilde{s}_i(t) + \sum_{i=1}^{L} \beta_i e^{j\hat{\phi}_{1i}} \tilde{n}_i(t) \tag{15}$$

$$= \sum_{i=1}^{L} \beta_i e^{j\hat{\phi}_{1i}} \left(\sqrt{P_{C_i}} e^{j(\omega_b t + \theta_i)} + j\sqrt{P_{D_i}} d(t) \ \mathrm{sqr}(\omega_{sc} t + \theta_{sc}) e^{j(\omega_b t + \theta_i)} \right)$$

$$+ \sum_{i=1}^{L} \beta_i e^{j\hat{\phi}_{1i}} \tilde{n}_i(t) \tag{16}$$

where the weights β_i are chosen to satisfy the condition

$$\beta_i = \sqrt{\frac{P_{T_i}}{P_{T_1}} \frac{N_{o1}}{N_{oi}}} \tag{17}$$

for $i = 1, \cdots, L$. It is shown in [1] that these weights maximize the combined SNR when the noises $\tilde{n}_i(t)$ are independent. Note that this is *not* necessarily the optimal choice of weights for the correlated noise case, as pointed out in [3]. Furthermore, the optimal choice of phases used to array the signals is not necessarily the relative signal phases, ϕ_{1i}. Using the phases ϕ_{1i} will certainly maximize the arrayed signal power, but not necessarily the *ratio* of signal to noise power, which is the relevant criteria for optimization. The problem of optimal combining weights and phases for signals with correlated noise has been analyzed in [5], where the results are applied to an array of antenna feed elements. However, computation of these weights requires knowledge of the pairwise correlations between the noises, $\alpha_{ij}e^{j\phi^n_{ij}}$, for all i, j pairs. A scheme can be devised to estimate the required parameters in real time and modify the weights accordingly, but would significantly complicate the problem. Our goal, instead, is to determine the performance impact of the correlated noise assuming the traditional combining scheme is used.

The total combined signal power, P_T, is given by

$$P_T \triangleq E\left[\tilde{s}_{comb}(t)\right]\ E\left[\tilde{s}^*_{comb}(t)\right] \tag{18}$$

If the relative signal phases are estimated perfectly (i.e., $\hat{\phi}_{1i} = \phi_{1i}$ for $i = 2, \cdots, L$), the combined signal power becomes

$$P_T = P_{T_1}\left(\sum_{i=1}^{L}\gamma_i^2 + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\gamma_i\gamma_j\right) \tag{19}$$

where $\gamma_i \triangleq [(P_{T_i})/(P_{T_1})][(N_{o_1})/(N_{o_i})]$.

The one-sided power spectral density of the real and imaginary parts of the combined noise is given by

$$N_o \triangleq \frac{1}{2B}E\left[\tilde{n}_{comb}(t)\ \tilde{n}^*_{comb}(t)\right] \tag{20}$$

where B is the one-sided bandwidth of the noise waveforms. Note that the factor of two in the denominator of Eq. (20) results from the fact that the real and imaginary parts of the noise each has half the power of the complex noise. From the definitions of power spectral density and cross-power spectral density, it follows that

$$E\left[\tilde{n}_i(t)\tilde{n}^*_i(t)\right] = 2N_{o_i}B \tag{21}$$

$$E\left[\tilde{n}_i(t)\tilde{n}^*_j(t)\right] = 2\alpha_{ij}e^{j\phi^n_{ij}}B \tag{22}$$

Equations (20), (21), and (22) can be combined to find the power spectral density of the combined noise, yielding

$$N_o = N_{o_1}\left(\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\sqrt{\gamma_i\gamma_j}\rho_{ij}e^{j(\phi_{ij}^n-\phi_{ij})}\right) \qquad (23)$$

The P_T/N_o of the combined signal is thus given by

$$\frac{P_T}{N_o} = \frac{P_{T_1}}{N_{o_1}}\,\frac{\left(\sum_{i=1}^{L}\gamma_i\right)^2}{\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\sqrt{\gamma_i\gamma_j}\;\rho_{ij}\;e^{j\psi_{ij}}} \qquad (24)$$

where $\psi_{ij} \triangleq \phi_{ij}^n - \phi_{ij}$. The parameters ρ_{ij} and ψ_{ij} describe the relevant statistics for the noise correlations between the various antenna pairs and determine the correlated noise impact on the ideal arraying gain.

The combined signal is finally processed by a single carrier, subcarrier, and symbol loop. Assuming perfect references at each of these three stages, the symbol SNR of the arrayed signal becomes

$$\begin{aligned} SNR_{ideal} &= \frac{2P_{D_1}}{N_{o_1}R_{sym}}\,\frac{\left(\sum_{i=1}^{L}\gamma_i\right)^2}{\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}(\gamma_i\gamma_j)^{1/2}\rho_{ij}\;e^{j\psi_{ij}}} \\ &= \frac{2P_{D_1}}{N_{o_1}R_{sym}}\,G_A \end{aligned} \qquad (25)$$

where G_A is the ideal arraying gain due to combining the signals. Note that setting all the noise correlation coefficients ρ_{ij} to zero results in $G_A = \sum_{i=1}^{L}\gamma_i$, which is the ideal arraying gain in the case of uncorrelated noises, as discussed in [1].

Further note that the ideal arraying gain in the presence of correlated noise can be higher *or* lower than the uncorrelated noise case, depending on the phases ψ_{ij}. This point can be understood by considering an array of two equal antennas (i.e., $\gamma_1 = \gamma_2 = 1$.) Figure 3 shows values for G_A for two equal antennas as a function of ρ and ψ. For $\rho = 0$, the ideal arraying gain is a constant 3 dB, as expected. Now suppose the noises have some nonzero correlation coefficient, ρ, and some correlation phase, ϕ^n. If $\psi = 0$ deg, then the phase difference of the spacecraft signal as observed by antennas 1 and 2, ϕ, is equal to the noise correlation phase ϕ^n. Thus, phase aligning the two signals *also* phase aligns the correlated component of the noise. The noise from the background source adds maximally in phase, and the combined noise power increases. Thus, the combined SNR decreases, and hence the arraying gain falls below 3 dB. By contrast, if $\psi = 180$ deg, phase aligning the signal results in combining the correlated component of the noise 180 deg out of phase. Thus, the noise combines destructively in this case, and the arraying gain is now *greater* than 3 dB. For intermediate values of ψ, the arraying gain varies continuously from its minimum value at $\psi = 0$ deg to its maximum at $\psi = 180$ deg.

B. Symbol SNR Degradation

In practice, perfect phase alignment and ideal carrier, subcarrier, and symbol references are not available. Some degradation in the arrayed symbol SNR is, therefore, incurred due to synchronization errors. To quantify the degradation, we first find the set of density functions for the phase alignment errors $\Delta\phi_{1i} \triangleq \hat{\phi}_{1i} - \phi_{1i}$, $i = 2, \cdots, L$. This set of functions is then used to compute the P_T/N_o of the arrayed signal. Adding in losses due to carrier, subcarrier, and symbol tracking, the symbol SNR at the matched-filter output can be computed. Finally, comparing the actual symbol SNR to the ideal symbol SNR given by Eq. (25) yields the degradation for full-spectrum combining.

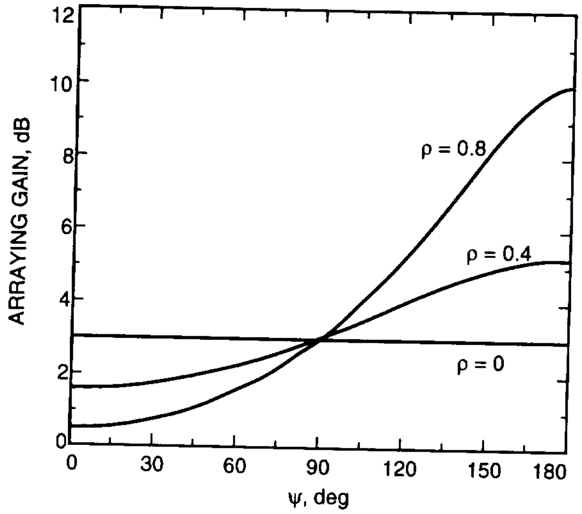


Fig. 3. Ideal arraying gain G_A for various ρ,ψ.

1. Antenna Phasing. A set of phase estimates $\hat{\phi}_{1i}$ for $i = 2, \cdots, L$ are needed to align signals $2, \cdots, L$ with signal 1. In the description of FSC given in [2], the phase difference between $\tilde{s}_i(t)$ and $\tilde{s}_1(t)$ is estimated by filtering the two signals to some lowpass bandwidth B_{lp} Hz, multiplying them, and averaging their product over T_{corr} s. The phase of this complex quantity is then computed by taking the inverse tangent of the ratio of the imaginary to real parts. A block diagram of this scheme is shown in Fig. 4.

The complex product of the baseband signals after averaging, Z, is given by

$$Z = \frac{1}{T_{corr}} \int (\tilde{s}_{lp_1}(t) + \tilde{n}_{lp_1}(t))(\tilde{s}^*_{lp_i}(t) + \tilde{n}^*_{lp_i}(t))\, dt$$

$$= (\sqrt{P_{C_1}P_{C_i}} + \sqrt{P_{D_1}P_{D_i}}H)e^{j\phi_{1i}} + \frac{1}{T_{corr}} \int \left(\tilde{n}_{s,n}(t) + \tilde{n}_{lp_1}(t)\tilde{n}^*_{lp_i}(t)\right)\, dt \qquad (26)$$

where H is given by

$$H = \left(\frac{4}{\pi}\right)^2 \sum_{\substack{i=1 \\ i \text{ odd}}}^{M} \frac{1}{2i^2} \qquad (27)$$

and M is the highest harmonic of the subcarrier passed by the lowpass filter. The term $\tilde{n}_{s,n}(t)$ is composed of signal-noise terms in the product and has zero mean. Note, however, that the noise-noise term, $\tilde{n}_{lp_1}(t)\tilde{n}_{lp_i}(t)^*$, does not necessarily have zero mean, due to a possible correlation that exists between the two noise waveforms. The expected value of this noise product can easily be computed from the cross-power spectral density of $\tilde{n}_1(t)$ and $\tilde{n}_i(t)$; thus,

$$E[z] = (\sqrt{P_{C_1}P_{C_i}} + \sqrt{P_{D_1}P_{D_i}}H)e^{j\phi_{1i}} + 2\rho_{1,i}\sqrt{N_{o_1}N_{o_i}}B_{lp}e^{j\phi^n_{1i}} \qquad (28)$$

Since ϕ^n_{1i} is not necessarily equal to ϕ_{1i}, the noise product introduces a "bias" to the estimate of the relative signal phase. This situation is represented pictorially in Fig. 5. The complex quantity $E[z]$ can

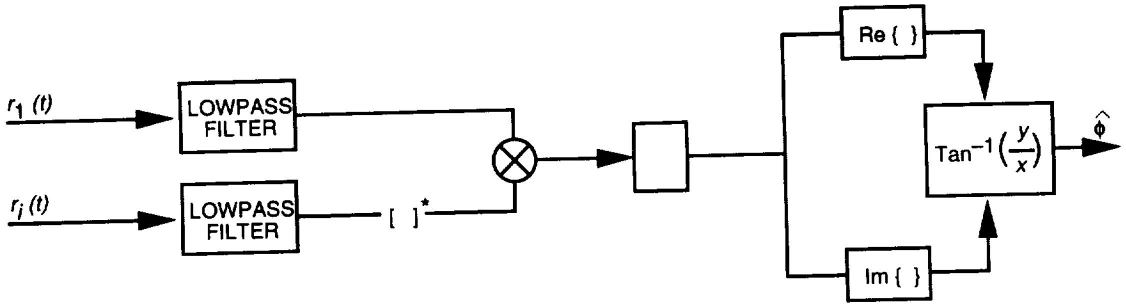


Fig. 4. Conventional phase estimator.

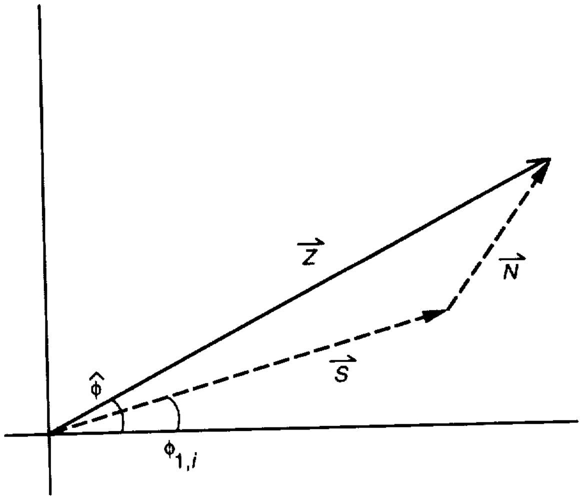


Fig. 5. Complex correlation vector.

be thought of as a vector sum of a signal-to-signal correlation, $\vec{S}$, and a noise-to-noise correlation, $\vec{N}$. Note how the presence of the noise vector biases the measurement of the phase of the complex correlation. The relative magnitude of these vectors is given by

$$\frac{|\vec{S}|}{|\vec{N}|} = \frac{1}{2\rho_{1i}\, B_{lp}} \left(\left(\frac{P_{C_1}}{N_{o_1}} \frac{P_{C_i}}{N_{o_i}} \right)^{1/2} + H \left(\frac{P_{D_1}}{N_{o_1}} \frac{P_{D_i}}{N_{o_i}} \right)^{1/2} \right) \tag{29}$$

For typical parameters, even relatively modest levels of noise correlation can lead to a substantial biasing effect in estimating the relative signal phase. For example, consider correlating two signals, each having a P_T/N_o of 20 dB-Hz with a 1-kHz correlation bandwidth. Even if all subcarrier harmonics are included in the correlation, making $H = 1$, a correlation coefficient as low as $\rho = 0.1$ makes the ratio in Eq. (29) equal to 0.5. The phase estimates are then influenced more by the relative *noise* phases ϕ^n_{1i} than the desired quantities ϕ_{1i}, leading to a high amount of degradation in combining the signals. If the alternative method described in Section III.A is used, where the subcarrier harmonics are filtered individually prior to combining, the effective correlation bandwidth can be lessened, thus reducing the impact of the noise bias. Nevertheless, a practical implementation of full-spectrum combining requires a modified phase estimation algorithm if correlation levels encountered will generate significant biases.

The method of phase estimation shown in Fig. 6 can be used for this purpose. Here, each signal is filtered to some bandpass bandwidth B_{bp}, and an additional complex correlation is performed between

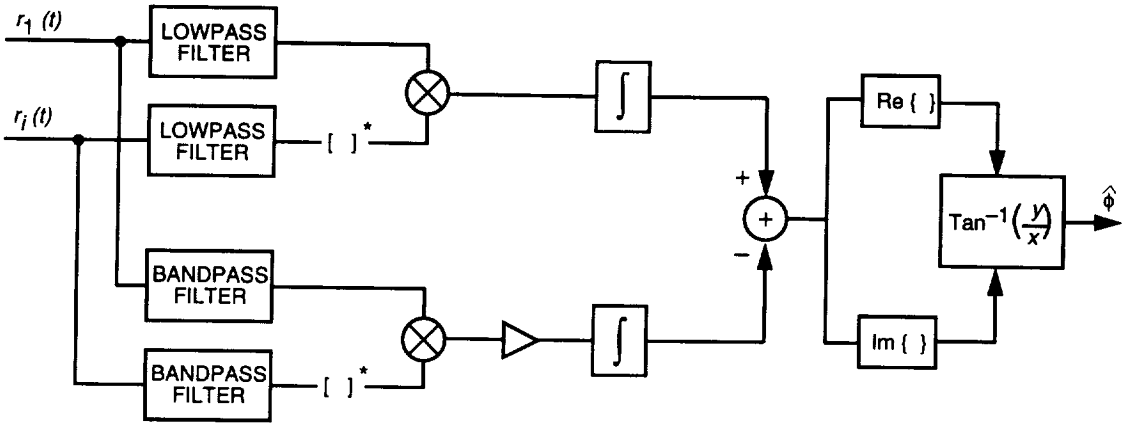


Fig. 6. Modified phase estimator.

the resulting waveforms. The center frequency of this filter is chosen so as to *not* capture any energy from the telemetry; this can be accomplished by locating the filter at an even multiple of the subcarrier frequency, for example. After scaling the noise-only correlation by the ratio of the lowpass-to-bandpass bandwidths, this quantity provides an estimate of the contribution of the noise to the total correlation. The bandpass correlation can then be subtracted from the lowpass correlation to compensate for the mean correlation vector $|\vec{N}|$. The compensated correlation can thus be expressed as

$$
\begin{aligned}
Z &= \left(\sqrt{P_{C_1}P_{C_i}} + \sqrt{P_{D_1}P_{D_i}}H\right) e^{j\phi_{1i}} + \frac{1}{T_{corr}} \int \left(\tilde{n}_{s,n}(t) + \tilde{n}_{lp_1}(t)\tilde{n}^*_{lp_i}(t)\right)\, dt \\
&\quad - \frac{B_{lp}}{B_{bp}} \frac{1}{T_{corr}} \int \tilde{n}_{bp_1}(t)\tilde{n}^*_{bp_i}(t)\, dt \\
&= (\sqrt{P_{C_1}P_{C_i}} + \sqrt{P_{D_1}P_{D_i}}H)e^{j\phi_{1i}} + \tilde{N}
\end{aligned} \tag{30}
$$

where the the noise term $\tilde{N}$ now has zero mean. The phase estimate is then found by taking the inverse tangent of the ratio of the imaginary-to-real part of Eq. (30), i.e.,

$$
\hat{\phi}_{1i} = \tan^{-1}\left[\frac{(\sqrt{P_{C_1}P_{C_i}} + \sqrt{P_{D_1}P_{D_i}}H)\sin\phi_{1i} + N_Q}{(\sqrt{P_{C_1}P_{C_i}} + \sqrt{P_{D_1}P_{D_i}}H)\cos\phi_{1i} + N_I}\right] \tag{31}
$$

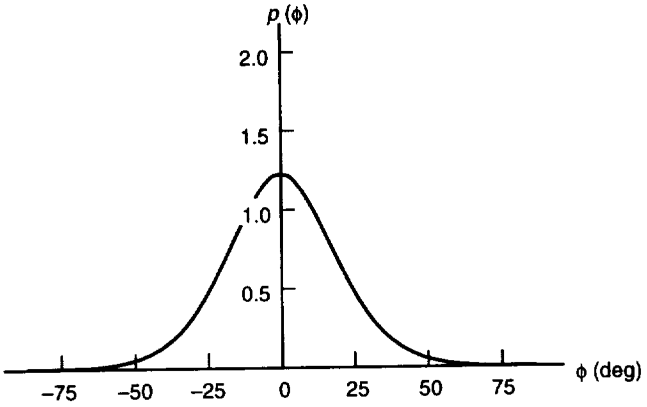
where N_I and N_Q are the real and imaginary parts of $\tilde{N}$, respectively. Note that although N_I and N_Q have zero mean, their joint statistics are *still* influenced by the correlation between $\tilde{n}_1(t)$ and $\tilde{n}_i(t)$. These statistics are analyzed in Appendix A, and the density function for the phase estimation error $\Delta\phi_{1i} \overset{\Delta}{=} \hat{\phi}_{1i} - \phi_{1i}$ is derived.

In [2], a quantity known as the *correlator SNR* is introduced, defined as

$$
SNR_{corr} = \frac{E[Z]E^*[Z]}{E[ZZ^*] - E[Z]E^*[Z]} \tag{32}
$$

The correlator SNR is a measure of the spread of the phase error density $p_\phi(\Delta\phi_{1i})$ and is inversely related to the variance of the phase error. In [1], where FSC is analyzed for independent noises, it is shown that the phase error density can be expressed solely in terms of the correlator SNR. For the correlated noise case, the density is given in Appendix A in terms of the correlator SNR and the correlation parameters ρ_{1i} and ψ_{1i}.

Figures 7 through 9 show the density function $p_\phi(\Delta\phi)$ for various values of ρ and ψ. The signal parameters chosen for these curves are $(P_T/N_o)_1 = (P_T/N_o)_2 = 25$ dB-Hz, $\Delta = 90$ deg, with seven subcarrier harmonics included in the correlation. The correlator parameters are $B_{lp} = B_{bp} = 15$ kHz, and $T_{corr} = 3$ s. Note that even for a noise correlation as high as 0.4, the density function looks remarkably like that of the uncorrelated noise case. Simulations were performed for the same parameters and densities collected for the measured phase estimates. These results are shown with the analytical curves in Fig. 10.



ρ = 0

Fig. 7. Phase estimate density.

2. Arrayed Symbol SNR and Symbol SNR Degradation. Using the set of estimated phases to align the signals, the combined signal becomes

$$\tilde{y}_{comb}(t) = \tilde{s}_{comb}(t) + \tilde{n}_{comb}(t) \tag{33}$$

$$= \sum_{i=1}^{L} \beta_i \, e^{j\hat{\phi}_{1i}} \tilde{s}_i(t) + \sum_{i=1}^{L} \beta_i \, e^{j\hat{\phi}_{1i}} \tilde{n}_i(t) \tag{34}$$

$$= \sum_{i=1}^{L} \beta_i \left(\sqrt{P_{C_i}} - j\sqrt{P_{D_i}} \, \mathrm{sqr}(\omega_{sc}t + \theta_{sc}) \right) e^{j(\omega_b t + \theta_1 + \Delta\phi_{1i})} + \sum_{i=1}^{L} \beta_i e^{j\hat{\phi}_{1i}} \tilde{n}_i(t) \tag{35}$$

The combined signal power conditioned on the set of phase errors $\Delta\phi_{1i}$ is thus given by

$$P'_T = E\left[\tilde{s}_{comb}(t)\right] E\left[\tilde{s}^*_{comb}(t)\right] \tag{36}$$

$$= P_{T_1} \left(\sum_{i=1}^{L} \gamma_i^2 + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \gamma_i \gamma_j e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})} \right) \tag{37}$$

Similarly, the conditional noise power spectral density is given by

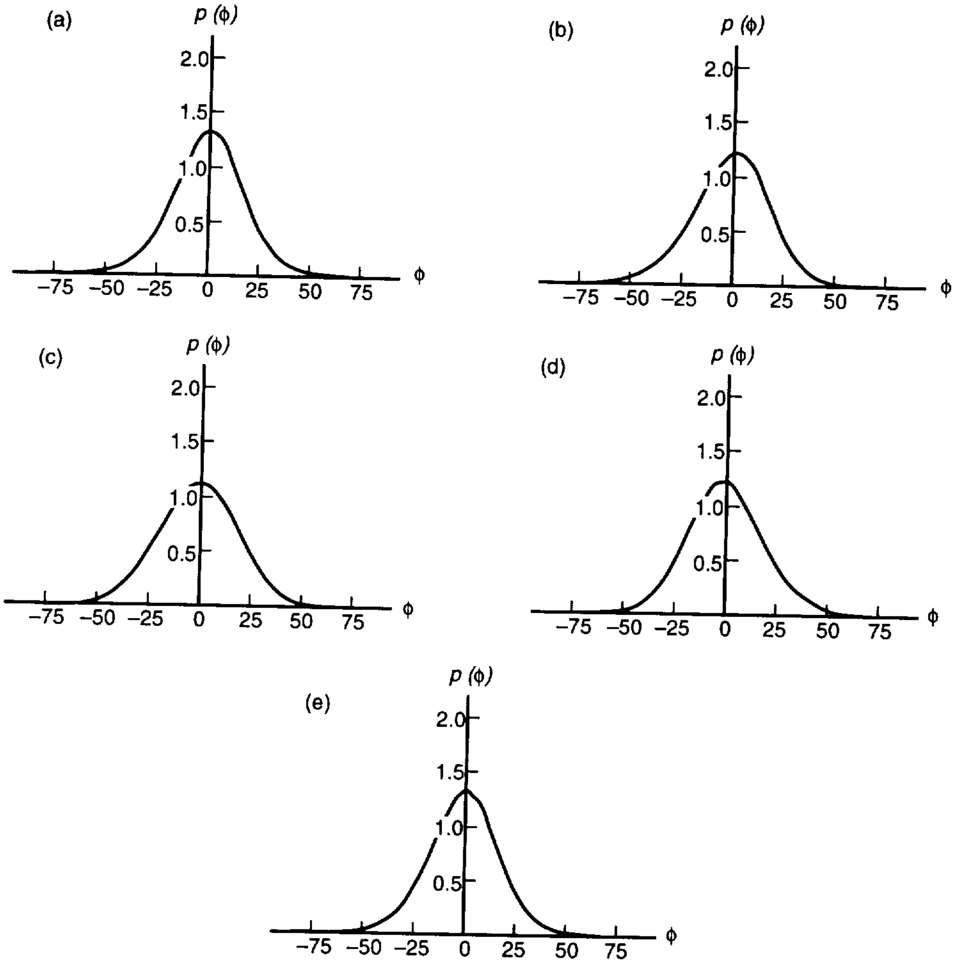


Fig. 8. Phase estimate densities: (a) ρ = 0.4, Ψ = 0 deg, (b) ρ = 0.4, Ψ = 45 deg, (c) ρ = 0.4, Ψ = 90 deg, (d) ρ = 0.4, Ψ = 135 deg, and (e) ρ = 0.4, Ψ = 180 deg.

$$N_o' = \frac{1}{2B} E\left[\tilde{n}_{comb}(t)\tilde{n}_{comb}^*(t)\right] \tag{38}$$

$$= N_{o_1}\left(\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}(\gamma_i\gamma_j)^{1/2}\rho_{ij}\ e^{j\psi_{ij}}\ e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}\right) \tag{39}$$

Taking the ratio of Eq. (37) to Eq. (39) yields the conditional P_T/N_o of the combined signal, i.e.,

$$\left(\frac{P_T}{N_o}\right)' = \frac{P_{T1}}{N_{o1}}\ \frac{\sum_{i=1}^{L}\gamma_i^2 + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\gamma_i\gamma_j e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}}{\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}(\gamma_i\gamma_j)^{1/2}\ \rho_{ij}\ e^{j\psi_{ij}}\ e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}} \tag{40}$$

After carrier and subcarrier demodulation and matched filtering, the conditional symbol SNR of the arrayed signal is given by

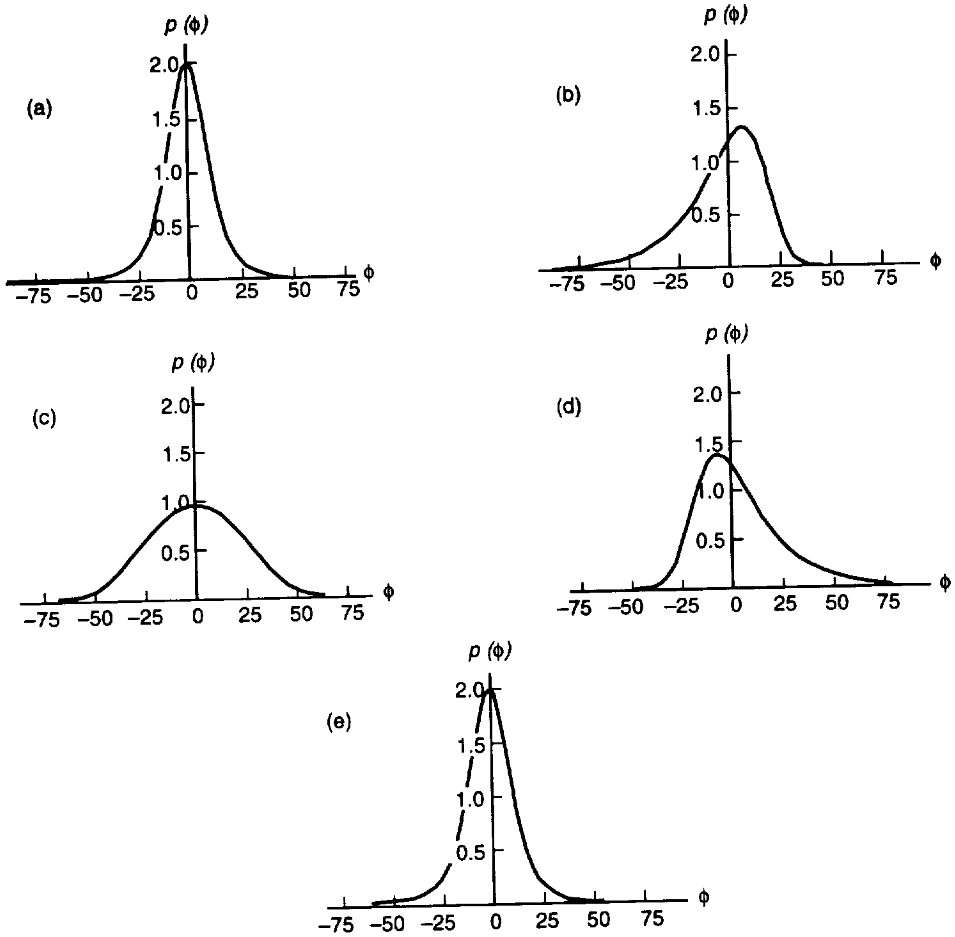


Fig. 9. Phase estimate densities: (a) $\rho = 0.8$, $\psi = 0$ deg, (b) $\rho = 0.8$, $\psi = 45$ deg, (c) $\rho = 0.8$, $\psi = 90$ deg, (d) $\rho = 0.8$, $\psi = 135$ deg, and (e) $\rho = 0.8$, $\psi = 180$ deg.

$$SNR' = \frac{2P_{T1}}{N_{o1}R_{sym}} \frac{\sum_{i=1}^{L} \gamma_i^2 + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i\neq j}}^{L} \gamma_i \gamma_j e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})}}{\sum_{i=1}^{L} \gamma_i + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i\neq j}}^{L} (\gamma_i\gamma_j)^{1/2} \rho_{ij} e^{j\psi_{ij}} \; e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})}} C_c^2 C_{sc}^2 C_{sy}^2 \tag{41}$$

where C_c, C_{sc}, and C_{sy} are the carrier, subcarrier, and symbol reduction functions, respectively. The unconditional symbol SNR is obtained by integrating Eq. (41) over the density functions for $\hat{\phi}_{21}, \cdots, \hat{\phi}_{L1}$ and the loop errors ϕ_c, ϕ_{sc}, and ϕ_{sy}. In order to simplify this computation, the loop errors and phase estimates are generally assumed to be independent. Taking expectation with respect to each of these quantities separately yields an expression for the unconditional symbol SNR, namely

$$\begin{aligned} SNR &= \frac{2P_{T1}}{N_{o1}R_{sym}} \overline{C_c^2}\; \overline{C_{sc}^2}\; \overline{C_{sy}^2} \\ &\times \int_{-\pi}^{\pi} \cdots \int_{-\pi}^{\pi} \left[\frac{\sum_{i=1}^{L} \gamma_i^2 + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i\neq j}}^{L} \gamma_i \gamma_j e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})}}{\sum_{i=1}^{L} \gamma_i + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i\neq j}}^{L} (\gamma_i\gamma_j)^{1/2} \rho_{ij} \; e^{j\psi_{ij}} \; e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})}} p(\Delta\phi_{1i}) \cdots p(\Delta\phi_{1L}) \right] \\ &\times d\Delta\phi_{12} \cdots d\Delta\phi_{1L} \end{aligned} \tag{42}$$

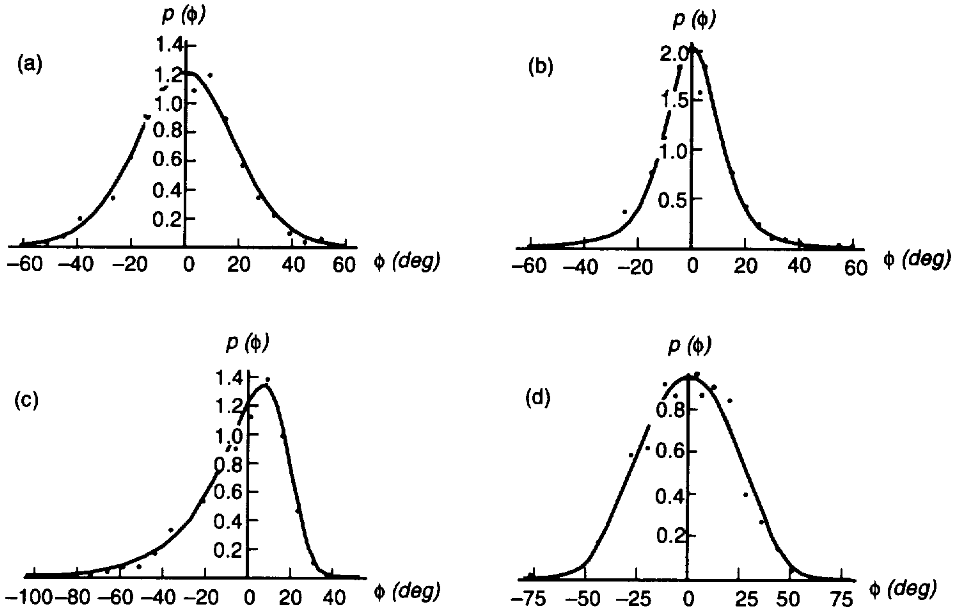


Fig. 10. Phase estimate densities with simulation points: (a) ρ = 0.0, (b) ρ = 0.8, ψ = 0 deg, (c) ρ = 0.8, ψ = 45 deg, and (d) ρ = 0.8, ψ = 90 deg.

where the density functions $p_\phi(\Delta\phi_{1i})$ are as given in Appendix A. Finally, taking the ratio of Eq. (42) to the ideal SNR, Eq. (25), yields the degradation for full-spectrum combining:

$$
\begin{aligned}
D_{fsc} &= \overline{C_c^2}\;\overline{C_{sc}^2}\;\overline{C_{sy}^2} \\
&\times \int_{-\pi}^{\pi}\cdots\int_{-\pi}^{\pi}\left[\frac{\sum_{i=1}^{L}\gamma_i^2+\sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\gamma_i\gamma_j e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}}{\sum_{i=1}^{L}\gamma_i+\sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}(\gamma_i\gamma_j)^{1/2}\rho_{ij}\; e^{j\psi_{ij}}\; e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}}\,p(\Delta\phi_{1i})\cdots p(\Delta\phi_{1L})\right] \\
&\times d\Delta\phi_{12}\cdots d\Delta\phi_{1L}\; G_A^{-1} \qquad (43)
\end{aligned}
$$

Note that D_{fsc} is equal to one in the upper limit, where $\Delta\phi_{1i} = 0$ for $i = 2, \cdots, L$ and $\overline{C_c^2} = \overline{C_{sc}^2} = \overline{C_{sy}^2} = 1$. The second moments of the reduction functions $\overline{C_c^2}, \overline{C_{sc}^2}$, and $\overline{C_{sy}^2}$ can be expressed in terms of the loop SNRs of the three loops, and are given in [1].

C. Simulation Results

A simple two-antenna array was simulated under conditions of correlated noise to verify the analysis given above. The symbol SNR of the combined data was measured using the split-symbol moments estimator and divided by the ideal symbol SNR to obtain measured degradations. The signal parameters used were $P_{T_1}/N_{o_1} = P_{T_2}/N_{o_2} = 25$ dB-Hz, $R_{sym} = 200$ symbols per second (sps), and $\Delta = 90$ deg. The carrier, subcarrier, and symbol loops were operated with bandwidths of 3.5, 0.75, and 0.15 Hz, respectively, with a symbol window of 1/2. The correlation coefficient between the noises, ρ, and the relative noise phase, ψ, were varied over a range of values.

Figure 11 shows simulation values along with curves describing analytical results for a "high" correlator SNR. The correlation bandwidths and integration time were chosen so that degradation resulting from

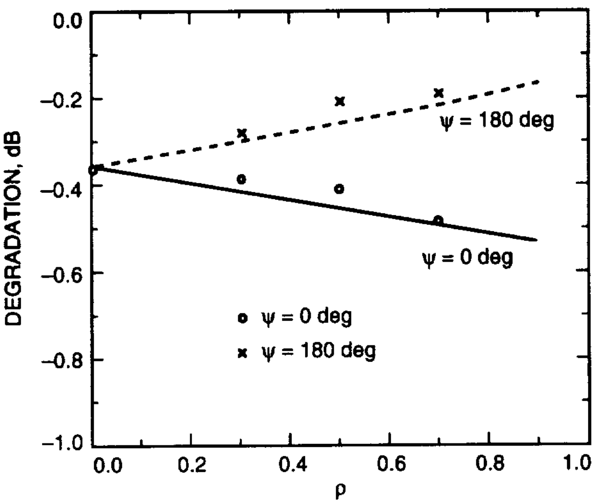


Fig. 11. FSC degradation, high correlator SNR—theory and simulation.

imperfect phasing is negligible compared to the carrier, subcarrier, and symbol losses. The curves show that more degradation is incurred with increasing noise correlation for $\psi = 0$ deg, and that degradation decreases as ρ increases for $\psi = 180$ deg.[4] This can be explained by noting the effect of varying ρ and ψ on the arraying gain. For $\psi = 0$ deg, increasing ρ causes a decrease in arrayed symbol SNR, as explained in Section III.A. The loop SNR of the three loops, therefore, decreases, resulting in more carrier, subcarrier, and symbol loss. By contrast, when $\psi = 180$ deg, increasing ρ increases the combined P_T/N_o and raises the three loop SNRs. This results in *less* degradation in demodulating the signal. Since the correlator SNR is high in this example, the demodulation losses are the dominant source of degradation, and the trend shown in Fig. 11 is thus explained.

Figure 12 shows the same results performed for a relatively "low" correlator SNR. Here, the degradation curve for $\psi = 180$ deg actually lies *below* the curve for $\psi = 0$ deg. This result, although seemingly counterintuitive, can nevertheless be explained qualitatively. Note from Eq. (41) that the phase error terms $\Delta\phi_{1i}$ appear in both the numerator *and* the denominator of the SNR expression; the phase errors affect both the arrayed signal power and the arrayed noise power. This is in contrast to the uncorrelated noise case, where only the numerator depends on the phase errors $\Delta\phi_{1i}$; since the noises are uncorrelated, the choice of phases used in combining them does not affect their arrayed power. The phase errors $\Delta\phi_{1i}$ always decrease the arrayed signal power, but can decrease *or* increase the arrayed noise power, depending on the phase parameter ψ. For $\psi = 180$ deg, the noise power is increased by errors in estimating ϕ_{1i}, since phasing the array perfectly results in maximum noise cancellation. Therefore, estimating the phase imperfectly results in a twofold penalty: The combined signal power is lessened, and the combined noise power increases. This results in increased degradation due to imperfect phase alignment. On the other hand, when $\psi = 0$ deg, phase misalignment *decreases* the arrayed noise power. Since $\phi_{1i} = \phi^n_{1i}$ in this case, aligning the signals imperfectly also lessens the constructive addition of the noise. The reduced noise power due to phasing errors, therefore, has a mitigating effect on the degradation incurred.

It should be noted that the fact that the $\psi = 180$-deg case has more degradation than the $\psi = 0$-deg case in this example does *not* mean that the overall performance of the array is worse for $\psi = 180$ deg. Recall that degradation is defined as the deviation from the ideal arraying gain, G_A. In the above example,

[4] The phrase "decreasing degradation" is used loosely to mean decreased synchronization losses; in actuality, numerically lower degradation implies *greater* losses incurred.

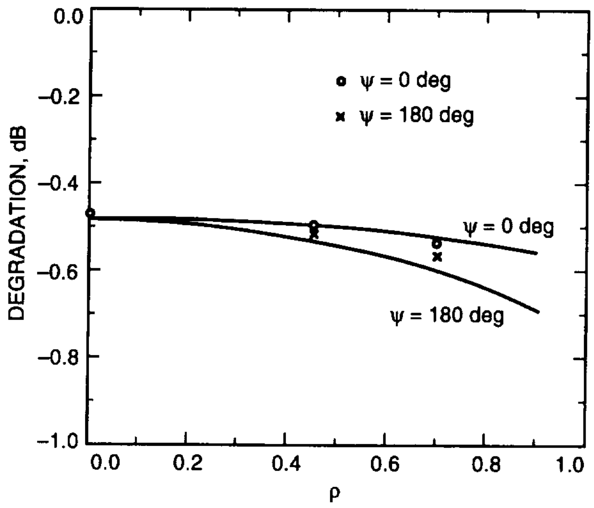


Fig. 12. FSC degradation, low correlator SNR—theory and simulation.

although the degradation for $\psi = 180$ deg is slightly higher, the ideal gain is substantially higher than it is for $\psi = 0$ deg. Thus, to determine the absolute performance for the array in terms of total combined SNR, both the ideal gain and the degradation must be accounted for.

IV. Complex-Symbol Combining Performance

A block diagram of the complex-symbol combining arraying scheme is shown in Fig. 13. Each signal is open-loop downconverted to baseband with quadrature tones and tracked by separate subcarrier and symbol loops. Since the carrier is not tracked coherently, each signal consists of both an "I" and "Q" component, which can be thought of as a single complex signal. Furthermore, since subcarrier and symbol tracking are performed in the absence of carrier lock, the loop SNRs of these loops are different from the case where the carrier is tracked first. Two types of subcarrier and symbol loops that may be used in complex-symbol combining are discussed in [2]: the conventional, or "I" loop, which uses only one of the two signals in the complex pair to track, and the "IQ" loop, which uses both real and imaginary channels. We will assume the IQ loops are used, since they have higher loop SNRs.

The matched filter outputs consist of data modulated by complex baseband tones. These complex symbols are transmitted to a central location for combining. As in the case of full-spectrum combining, correlations are performed to phase align the carriers, after which the signals are weighted and summed coherently. A baseband Costas loop is finally used to demodulate the carrier.

Since the ideal arraying gain G_A is independent of which arraying technique is used, the expression computed in Section III.A is valid for complex-symbol combining also. Thus, it is only necessary to evaluate the degradation for CSC, taking into account combining and demodulation losses. Once again, the presence of correlated noise creates complications in phasing the array. A technique similar to the one used for FSC can be employed to reduce the biases in estimating the relative signal phases, as discussed below.

A. Antenna Phasing

The complex-symbol stream from the ith antenna is given by

$$\tilde{Y}_i(k) = \sqrt{P_{D_i}} C_{sc_i} C_{sy_i} \, d(k) \, e^{j(\omega_b T_s k + \theta_i)} + \tilde{N}_i(k) \tag{44}$$

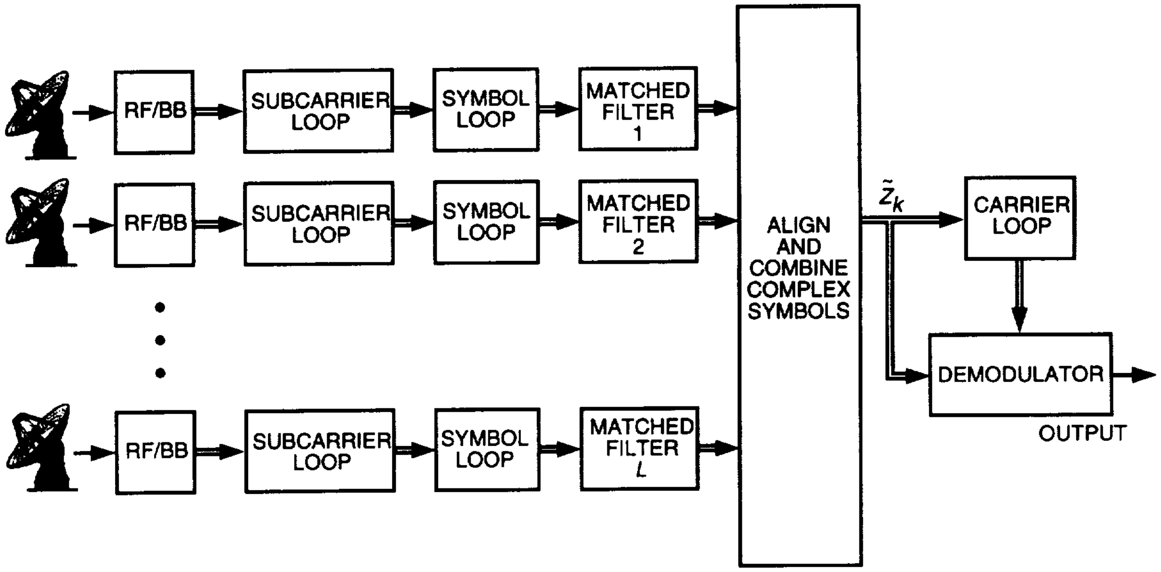


Fig. 13. Complex-symbol combining.

where C_{sc_i} and C_{sy_i} are the subcarrier and symbol reduction functions for the ith receiver, T_s is the symbol time, and $\tilde{N}_i(k)$ is the noise output from the ith matched filter. Taking the complex product between the 1st and ith streams yields

$$Z = \tilde{Y}_1(k)\tilde{Y}_i^*(k) = \sqrt{P_{D_1}P_{D_i}}C_{sc_1}C_{sc_i}C_{sy_1}C_{sy_i} + \tilde{N}_{s,n}(k) + \tilde{N}_1(k)\tilde{N}_i^*(k) \tag{45}$$

where the signal-noise term $\tilde{N}_{s,n}(k)$ has zero mean. Once again, the complex-noise product $\tilde{N}_1(k)\tilde{N}_i^*(k)$ has nonzero mean if the correlation coefficient is nonzero and introduces a bias to the signal correlation vector. Note, however, that the spectrum of the signals at the point of combining, $\tilde{Y}_i(k)$, does not contain empty bands as in the case of full-spectrum combining. Demodulating the subcarrier collapses all the data sidebands to baseband, allowing a much narrower combining bandwidth. Since the shared information rate for CSC is equal to the symbol rate, there is *no excess bandwidth that can be used to measure the correlation of the noise alone.* This problem may be solved by adding an extra matched filter for each receiver to capture noise only. Before investigating this possibility, however, we calculate the expectation of the noise product, $E[\tilde{N}_i(k)\tilde{N}_j^*(k)]$.

Consider the block diagram of Fig. 14, which shows the processing for complex-symbol combining up to the matched filter outputs. The signal $s_i(t)$ is the subcarrier reference from the ith subcarrier loop, given by

$$s_i(t) = \mathrm{sqr}(\omega_{sc}t + \theta_{sc} + \phi_{sci}) \tag{46}$$

where θ_{sc} is the instantaneous subcarrier phase and ϕ_{sc_i} is the instantaneous phase error in the ith loop, for $i = 1, \cdots, L$. The limits of integration for the ith matched filter are given by

$$t_{l_i} = kT_s + \tau_i \tag{47}$$

$$t_{u_i} = (k+1)T_s + \tau_i \tag{48}$$

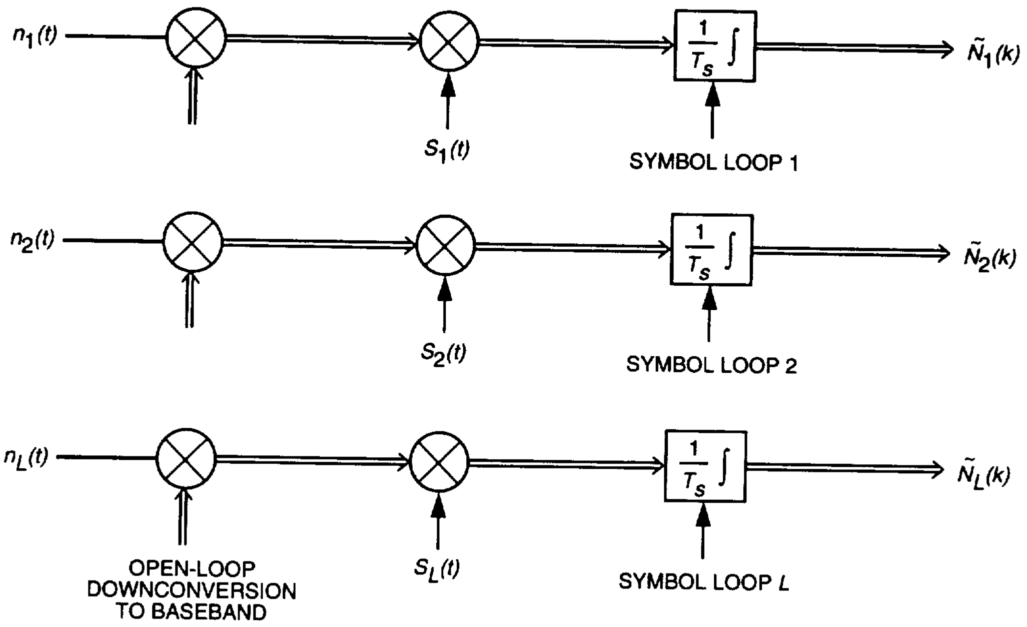


Fig. 14. Matched filter noise outputs for CSC.

where τ_i is the timing error in the ith symbol loop. The matched-filter noise samples are, therefore, given by

$$\tilde{N}_i(k) = \frac{1}{T_s} \int_{kT_s+\tau_i}^{(k+1)T_s+\tau_i} \tilde{n}_i(t)\ \mathrm{sqr}(\omega_{sc}t + \theta_{sc} + \phi_{sc_i})\ dt \tag{49}$$

$$\tilde{N}_j(k) = \frac{1}{T_s} \int_{kT_s+\tau_j}^{(k+1)T_s+\tau_j} \tilde{n}_j(t)\ \mathrm{sqr}(\omega_{sc}t + \theta_{sc} + \phi_{sc_j})\ dt \tag{50}$$

The conditional expectation of $\tilde{N}_i(k)\tilde{N}_j^*(k)$ given the subcarrier and symbol timing errors can then be calculated by combining the above expressions with the cross-correlation function for the complex baseband noises, i.e.,

$$R_{\tilde{n}_i,\tilde{n}_j}(u,v) = E\left[\tilde{n}_i(u)\tilde{n}_j^*(v)\right] = \alpha_{ij}\ e^{j\phi^n_{ij}}\ \delta(u-v) \tag{51}$$

yielding

$$\begin{aligned}
E\left[\tilde{N}_i(k)\tilde{N}_j^*(k)\right] &= \frac{1}{T_s^2} E\left[\int_{kT_s+\tau_i}^{(k+1)T_s+\tau_i} \int_{kT_s+\tau_j}^{(k+1)T_s+\tau_j} \tilde{n}_i(u)s_i(u)\ \tilde{n}_j^*(v)s_j(v)\ du\ dv\right] \\
&= \frac{\alpha_{ij}e^{j\phi^n_{ij}}}{T_s^2} \int_{kT_s+\tau_i}^{(k+1)T_s+\tau_i} \int_{kT_s+\tau_j}^{(k+1)T_s+\tau_j} \delta(u-v)\ s_i(u)s_j(v)\ du\ dv \\
&= \frac{\alpha_{ij}e^{j\phi^n_{ij}}}{T_s^2} \int_{t_{min}}^{t_{max}} s_i(v)s_j(v)\ dv
\end{aligned} \tag{52}$$

where the limits of integration of v are given by

$$t_{min} = \max(kT_s + \tau_i,\ kT_s + \tau_j) \tag{53}$$

$$t_{max} = \min\left((k+1)T_s + \tau_i,\ (k+1)T_s + \tau_j\right) \tag{54}$$

Finally, integrating with respect to v yields

$$E\left[\tilde{N}_i(k)\tilde{N}_j^*(k)\right] = \frac{\alpha_{ij}e^{j\phi^n_{ij}}}{T_s^2}\left(1 - \frac{2}{\pi}|\phi_{sc_i} - \phi_{sc_j}|\right)\left(T_s - |\tau_i - \tau_j|\right) \tag{55}$$

$$= \frac{\alpha_{ij}e^{j\phi^n_{ij}}}{T_s}\left(1 - \frac{2}{\pi}|\phi_{sc_i} - \phi_{sc_j}|\right)\left(1 - \frac{1}{2\pi}|\phi_{sy_i} - \phi_{sy_j}|\right) \tag{56}$$

$$= \alpha_{ij}e^{j\phi^n_{ij}}\ R_{sym}\ C_{sc_{ij}}\ C_{sy_{ij}} \tag{57}$$

Note that, in the absence of phase errors in any of the loops, Eq. (57) reduces to $\alpha_{ij}e^{j\phi^n_{ij}}R_{sym}$, which is simply the cross-power spectral density of the noises $\tilde{n}_i(t)$ and $\tilde{n}_j(t)$ times the effective bandwidth of the matched filter. Thus, in addition to reducing the effective signal power at the matched filter output, the subcarrier and symbol phase errors *also* reduce the noise correlation at this point.

Calculating the unconditional covariance of the matched filter noises requires taking the expectation of Eq. (57) with respect to the phase errors $\phi_{sc_i}, \phi_{sc_j}, \phi_{sy_i}$, and ϕ_{sy_j}. Two approximations are made to perform this computation. First, the densities of the phase errors are assumed to be Gaussian. This condition is nearly satisfied for loop SNRs above 10 dB and is consistent with the approximation made in [1]. Second, the phase errors of all loops are assumed to be mutually independent. This statement is not strictly justifiable, since the subcarrier and symbol loops from a single receiver are affected by the same noise and, furthermore, because the noises viewed by separate receivers are correlated. Nevertheless, it is invoked for the purpose of making a first-order approximation to evaluating the unconditional covariance. The quantities $\phi_{sc_i} - \phi_{sc_j}$ and $\phi_{sy_i} - \phi_{sy_j}$ are then Gaussian-distributed with known mean and variance, and the unconditional expectation $E[\tilde{N}_i(k)\tilde{N}_j^*(k)]$ becomes

$$E\left[\tilde{N}_i(k)\tilde{N}_j^*(k)\right] = \alpha_{ij}e^{j\phi^n_{ij}}R_{sym}\overline{C_{sc_{ij}}}\ \overline{C_{sy_{ij}}} = \rho_{ij}\sqrt{N_{o_i}N_{o_j}}e^{j\phi^n_{ij}}\ R_{sym}\ \left(1 - \frac{2}{\pi}\sqrt{\frac{2}{\pi}}(\sigma^2_{\phi_{sc_i}} + \sigma^2_{\phi_{sc_j}})^{1/2}\right)$$

$$\times\left(1 - \frac{1}{2\pi}\sqrt{\frac{2}{\pi}}\left(\sigma^2_{\phi_{sy_i}} + \sigma^2_{\phi_{sy_j}}\right)^{1/2}\right) \tag{58}$$

Equations (58) and (45) can be combined to calculate the ratio of the signal-to-noise correlation magnitude, analogous to that computed in (29):

$$\frac{|\vec{S}|}{|\vec{N}|} = \frac{\sqrt{P_{D_1}P_{D_i}}\ \overline{C_{sc_1}}\ \overline{C_{sy_1}}\ \overline{C_{sc_i}}\ \overline{C_{sy_i}}}{\rho_{ij}\sqrt{N_{o_1}N_{o_i}}R_{sym}\ \overline{C_{sc_{ij}}}\ \overline{C_{sy_{ij}}}} \approx \frac{1}{\rho_{ij}}\left(\frac{E_{s_1}}{N_{o_1}}\frac{E_{s_i}}{N_{o_i}}\right)^{1/2} \tag{59}$$

where $E_s/N_o = P_DT_s/N_o$ is the bit SNR. In making the approximation of Eq. (59), the effects of synchronization have been ignored for simplicity. This result provides a useful rule of thumb for determining if the noise correlation is a significant bias in estimating the relative signal phase. If $|\vec{S}|/|\vec{N}|$ is much less than 1, then an extra correlation is needed to compensate for the noise vector, as mentioned earlier. On the other hand, if this quantity is much greater than 1, then it is unnecessary to add the extra matched filter channel to perform the noise-only correlation. Note that collapsing all the data sidebands to baseband and performing matched filtering *before* the correlation takes place substantially decreases the correlation bandwidth relative to that of the FSC scheme described in [2]. The full-spectrum combining scheme can optionally be modified to employ a similar strategy by using a series of matched filters for each subcarrier harmonic, as discussed earlier. Estimating the degree of correlation ρ that will be observed for a particular antenna pair and applying the rule described above will indicate whether or not the noise contribution to the total correlation is substantial and must be compensated for by performing an additional correlation.

Here we briefly describe how the extra matched-filter outputs can be used to measure the noise correlation: The complex baseband signal from each antenna can be shifted in frequency so that an empty portion of the spectrum is located at baseband. This may be accomplished by shifting by an even multiple of the subcarrier frequency, i.e.,

$$\tilde{y}_i'(t) = \left(\sqrt{P_{C_i}}e^{j(\omega_b t+\theta_i)} + j\sqrt{P_{D_i}}d(t)\ \mathrm{sqr}(\omega_{sc}t+\theta_{sc})e^{j(\omega_b t+\theta_i)} + \tilde{n}_i(t)\right) e^{jN\omega_{sc}t} = \tilde{s}_i'(t) + \tilde{n}_i'(t) \qquad (60)$$

where N is an even integer. The shifted signal can then be multiplied by the subcarrier reference from the ith antenna and passed through a matched filter using timing from the ith symbol loop, as shown in Fig. 14. Thus,

$$\tilde{N}_i'(t) = \frac{1}{T_s}\int_{kT_s+\tau_i}^{(k+1)T_s+\tau_i} \tilde{n}_i'(t)\ \mathrm{sqr}(\omega_{sc}t+\theta_{sc}+\phi_{sc_i})\ dt \qquad (61)$$

From the above analysis, it is clear that $E[\tilde{N}_i'(k)\tilde{N}_j'(k)]$ will be given by Eq. (58). Correlating the two noise-only matched filter outputs then yields a quantity that can be subtracted from the total correlation, Z, to compensate for the noise bias. The density function for the phase estimate computed using this technique is similar to the FSC case and is analyzed in Appendix B. Note, however, that performing this compensation requires increasing the combining bandwidth beyond what is required for CSC in the uncorrelated noise case, as well as additional hardware to process the extra channel containing noise only. A tradeoff in performance versus complexity must, therefore, be made to determine if complex-symbol combining is an attractive option when correlated noise is present.

B. Arrayed Symbol SNR and Symbol SNR Degradation

An expression for the conditional arrayed symbol SNR can be obtained in a similar manner as is the full-spectrum combining case. The combined signal for complex-symbol combining is given by

$$\tilde{Y}_{comb}(k) = \tilde{S}_{comb}(k) + \tilde{N}_{comb}(k) = \sum_{i=1}^{L}\beta_i e^{j\hat{\phi}_{1i}}\left(\sqrt{P_{D_i}}C_{sc_i}C_{sy_i}d(k)e^{j(\omega_b T_s k+\theta_i)} + \tilde{N}_i(k)\right) \qquad (62)$$

The conditional signal power, defined as $E[\tilde{S}_{comb}(k)]E[\tilde{S}_{comb}^*(k)]$, is given by

$$P_{comb} = P_{D_1}\left(\sum_{i=1}^{L}\gamma_i^2C_{sc_i}^2C_{sy_i}^2 + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\gamma_i\gamma_jC_{sc_i}C_{sc_j}C_{sy_i}C_{sy_j}\ e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}\right) \qquad (63)$$

where, as before, $\Delta\phi_{1i}$ is defined as the error in estimating the phase difference between the 1st and ith signal, $\hat{\phi}_{1i} - \phi_{1i}$. The one-sided power spectral density of the real and imaginary parts of $\tilde{N}_{comb}(k)$ is given by

$$N_o = T_s \; Var \; \left(\tilde{N}_{comb}(k)\right) = T_s \; E\left[\left(\sum_{i=1}^{L} \beta_i e^{j\hat{\phi}_{1i}} \tilde{N}_i(k)\right) \times \left(\sum_{j=1}^{L} \beta_j e^{-j\hat{\phi}_{1j}} \tilde{N}_j^*(k)\right)\right] \tag{64}$$

Using the relations

$$E[\tilde{N}_i(k)\tilde{N}_i^*(k)] \;=\; \frac{N_{o_i}}{T_s} \tag{65}$$

$$E[\tilde{N}_i(k)\tilde{N}_j^*(k)] \;=\; \frac{\rho_{ij}\sqrt{N_{o_i}N_{o_j}}\; e^{\phi_{ij}^n}}{T_s} \;\overline{C_{sc_{ij}}}\;\overline{C_{sy_{ij}}} \tag{66}$$

Eq. (64) can be shown to be equal to

$$N_o' = N_{o_1}\left(\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\sqrt{\gamma_i\gamma_j}\rho_{ij}C_{sc_{ij}}C_{sy_{ij}}e^{j\psi_{ij}}e^{j\left(\Delta\phi_{1i}^n - \Delta\phi_{1j}^n\right)}\right) \tag{67}$$

Taking the ratio of Eq. (63) to Eq. (67) then yields the combined P_D/N_o for CSC. The combined signal is finally processed by a baseband Costas loop, and the conditional SNR adding in carrier losses is given by

$$SNR' = \frac{2P_{D_1}}{N_{o_1}R_{sym}}\;\frac{\sum_{i=1}^{L}\gamma_i^2 C_{sc_i}^2 C_{sy_i}^2 + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\gamma_i\gamma_j C_{sc_i}C_{sc_j}C_{sy_i}C_{sy_j}e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}}{\sum_{i=1}^{L}\gamma_i + \sum_{i=1}^{L}\sum_{\substack{j=1\\ i\neq j}}^{L}\sqrt{\gamma_i\gamma_j}\rho_{ij}C_{sc_{ij}}C_{sy_{ij}}e^{j\psi_{ij}}e^{j(\Delta\phi_{1i}-\Delta\phi_{1j})}}\;C_c^2 \tag{68}$$

Computing the unconditional symbol SNR requires taking the expectation of the above quantity with respect to the phase errors ϕ_{sc_i} and ϕ_{sy_i} for $i = 1, \cdots, L$, the phase estimates $\hat{\phi}_{1i}$ for $i = 2, \cdots, L$, and the carrier phase error ϕ_c. Once again, we assume all loop phase errors and phase-aligning errors are mutually independent. Thus, integration over the carrier phase error ϕ_c is accomplished easily by considering the carrier reduction function C_c^2 separately. However, unlike the case of full-spectrum combining, the subcarrier and symbol phase errors appear in both the numerator *and* the denominator. The expectation with respect to the subcarrier and symbol phase errors, therefore, cannot be given in closed form. Calculating the unconditional symbol SNR for even a simple two-element array would thus require a fifth-order numerical integration. Rather than resort to such brute-force tactics, we make further simplifying assumptions to allow evaluation of some of the integrals in closed form.

In taking the expectation with respect to the ϕ_{sc_i} and ϕ_{sy_i} terms, we apply the approximation

$$E\left[\frac{x}{y}\right] \approx \frac{E[x]}{E[y]} \tag{69}$$

to the ratio of Eq. (68), yielding

$$\begin{aligned} SNR = & \frac{2P_{d1}}{N_{o1}R_{sym}} \\ & \times E_{\hat{\Phi}} \left[\frac{E_{\Phi_{sc},\Phi_{sy}|\hat{\Phi}} \left[\sum_{i=1}^{L} \gamma_i^2 C_{sc_i}^2 C_{sy_i}^2 + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \gamma_i \gamma_j C_{sc_i} C_{sc_j} C_{sy_i} C_{sy_j} e^{j(\Delta\phi_{i1} - \Delta\phi_{j1})} \right]}{E_{\Phi_{sc},\Phi_{sy}|\hat{\Phi}} \left[\sum_{i=1}^{L} \gamma_i + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \sqrt{\gamma_i \gamma_j} \rho_{ij} C_{sc_{ij}} C_{sy_{ij}} \, e^{j\psi_{ij}} e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})} \right]} \right] \overline{C_c^2} \end{aligned} \tag{70}$$

where Φ_{sc} is the set of subcarrier phase errors ϕ_{sc_i} for $i = 1, \cdots, L$, Φ_{sy} is the set of symbol phase errors ϕ_{sy_i} for $i = 1, \cdots, L$, and $\hat{\Phi}$ is the set of phase estimates $\hat{\phi}_{1i}$ for $i = 2, \cdots, L$. The approximation of Eq. (69) is reasonable if the mean of y squared is much greater than the variance of y (i.e., if y is nearly a constant). This condition is met for the case under consideration, since it is implicitly assumed that the loop SNRs of the subcarrier and symbol loops are high enough to maintain lock, with 13 dB being a typical threshold. Thus, the variances of the reduction functions $C_{sc_{ij}}$ and $C_{sy_{ij}}$, which contain the loop phase errors, will be small compared to the mean of the entire denominator term.

By the above argument, the unconditional SNR can be evaluated as

$$\begin{aligned} SNR = & \frac{2P_{D_1}}{N_{o_1}R_{sym}} \, \overline{C_c^2} \\ & \times \int_{-\pi}^{\pi} \cdots \int_{-\pi}^{\pi} \left[\frac{\sum_{i=1}^{L} \gamma_i^2 \overline{C_{sc_i}^2} \; \overline{C_{sy_i}^2} + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \gamma_i \gamma_j \overline{C_{sc_i}} \; \overline{C_{sc_j}} \; \overline{C_{sy_i}} \; \overline{C_{sy_j}} e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})}}{\sum_{i=1}^{L} \gamma_i + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \sqrt{\gamma_i \gamma_j} \rho_{ij} \overline{C_{sc_{ij}}} \; \overline{C_{sy_{ij}}} e^{j\psi_{ij}} e^{j(\Delta\phi_{1i} - \Delta\phi_{1j})}} \right. \\ & \left. \times \, p(\Delta\phi_{12}) \cdots p(\Delta\phi_{1L}) \right] d\Delta\phi_{12} \cdots d\Delta\phi_{1L} \end{aligned} \tag{71}$$

The ideal symbol SNR for complex-symbol combining is identical to that for full-spectrum combining; since SNR_{ideal} is defined as the SNR that would be obtained in the absence of synchronization errors, its value is independent of the order in which combining and demodulation occur. Thus, the degradation for complex-symbol combining is found by combining the results of Eq. (71) with Eq. (25), yielding

$$\begin{aligned} D_{csc} = & \overline{C_c^2} \\ & \times \int_{-\pi}^{\pi} \cdots \int_{-\pi}^{\pi} \left[\frac{\sum_{i=1}^{L} \gamma_i^2 \overline{C_{sc_i}^2} \; \overline{C_{sy_i}^2} + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \gamma_i \gamma_j \overline{C_{sc_i}} \; \overline{C_{sc_j}} \; \overline{C_{sy_i}} \; \overline{C_{sy_j}} e^{j(\Delta\phi_{i1} - \Delta\phi_{j1})}}{\sum_{i=1}^{L} \gamma_i + \sum_{i=1}^{L} \sum_{\substack{j=1 \\ i \neq j}}^{L} \sqrt{\gamma_i \gamma_j} \rho_{ij} \overline{C_{sc_{ij}}} \; \overline{C_{sy_{ij}}} e^{j\psi_{ij}} e^{j(\Delta\phi_{i1}^{n} - \Delta\phi_{j1}^{n})}} \right. \\ & \left. \times \, p(\Delta\phi_{12}) \cdots p(\Delta\phi_{1L}) \right] d\Delta\phi_{12} \cdots d\Delta\phi_{1L} \; G_A^{-1} \end{aligned} \tag{72}$$

C. Simulation Results

Simulations of a two-antenna complex-symbol combining system were performed. The signal parameters used were the same as those used for the full-spectrum combining simulations: $P_{T_1}/N_{o_1} = P_{T_2}/N_{o_2} = 25$ dB-Hz, $R_{sym} = 200$ sps, and $\Delta = 90$ deg. The loop bandwidths were also set as before; the carrier, subcarrier, and symbol loop bandwidths were 3.5, 0.75, and 0.15 Hz, respectively, with a symbol window of 1/2. Both the compensating and noncompensating methods of estimating the signal phase difference were implemented. In Figs. 15 and 16, simulated and analytical degradation values are shown for various values of ρ and ψ.

For the uncompensated case, the degradation curve drops down sharply for ψ = 90 deg and ψ = 180 deg. One cause for this is the bias in the complex correlation used to estimate the relative signal phase. For the parameters being used, $|\vec{S}|/|\vec{N}|$, given by Eq. (59), is equal to 3.15 for $\rho = 0.5$. Thus, the noise vector is of comparable but lesser magnitude to that of the signal in estimating the phase. Note that for ψ = 0 deg, the noise correlation phase is equal to the relative signal phase ($\phi = \phi^n$), and the vectors $\vec{S}$ and $\vec{N}$ are colinear (see Fig. 5). The noise vector, therefore, does not bias the measurement away from the desired quantity, and the downward trend is not present.

For the compensated case, less overall degradation is observed. However, the ψ = 180-deg curve still drops down with increasing ρ. Recall from Section IV.A that imperfect subcarrier and symbol tracking tend to decrease the power levels of the individual signals at the matched filter output *and* decrease the correlation of the matched filter noises. When ψ = 0 deg, this has a beneficial effect on the arrayed SNR, since it reduces the coherent addition of the noise. By contrast, when ψ = 180 deg, a high degree of correlation between the noises is desirable, so that the noise cancels maximally. Thus, decreasing this correlation lessens the arrayed SNR and causes more degradation. This explains the fact that the ψ = 0-deg curve tends upwards with increasing ρ, while the ψ = 180-deg tends downward. Note, however, that the reverse trend is true of the ideal arraying gain, G_A. For example, for $\rho = 0.8$, G_A = 10 dB for ψ = 180 deg, but only 0.46 dB for ψ = 0 deg.

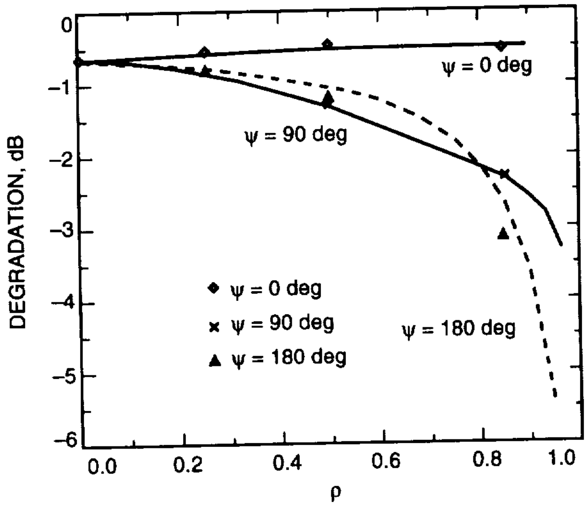


Fig. 15. CSC degradation, phase uncompensated: theory and simulation.

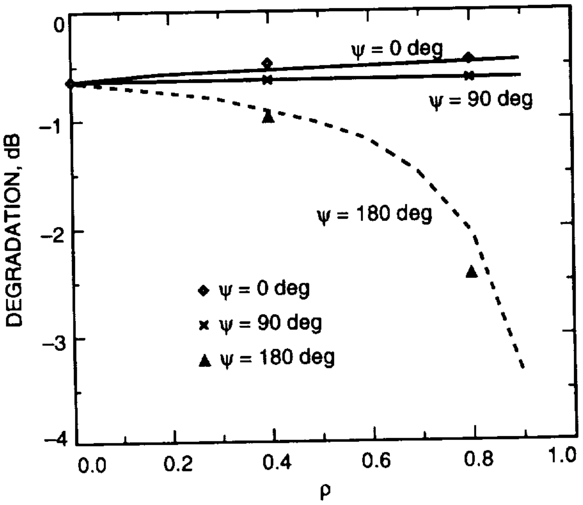


Fig. 16. CSC degradation, phase compensated: theory and simulation.

V. Example: Galileo Scenario

In order to illustrate the major concepts presented in this article, the performance of full-spectrum combining and complex-symbol combining is analyzed for the Galileo signal. An array of DSS 14, which is a 70-m antenna, and DSS 15, a 34-m high-efficiency (HEF) antenna, is chosen for this example. First, predicts for physical parameters describing the signal strength and degree of noise correlation are developed. These quantities are then used to calculate the arraying gain and degradation for each of the two schemes.

A. Signal Parameters

In the case of the Galileo spacecraft, correlated noise will be contributed by Jupiter being in the beam of both antennas. As discussed in Section II, the contribution of a background body to total system noise depends on its angular separation from the spacecraft and on its total flux, which varies with its distance from Earth. Values for the Jupiter–Earth probe (JEP) angle and Jupiter–Earth distance can be found from ephemeris information for the Galileo tour. For the purpose of this example, we select values that maximize the noise contribution of the planet to estimate the impact of correlated noise in a worst-case scenario. Thus, we assume the JEP angle is zero and that the Jupiter–Earth range is at its minimum value during the tour, which is $R_j = 4.0$ AU. Using these values, the temperature contribution of Jupiter for DSS 14 and DSS 15 are $T_{s_1} = 6.6$ K and $T_{s_2} = 1.4$ K, respectively. Note that the temperature contribution is higher for DSS 14 due to the greater aperture size and antenna efficiency.

The predicted signal parameters are as follows: $(P_T/N_o)_1 = 22.0$ dB-Hz and $(P_T/N_o)_2 = 11.6$ dB-Hz for the 70- and 34-m antennas, respectively; $\Delta = 90$ deg; and $R_{sym} = 200$ sps. Note that since we are assuming that the planet and spacecraft are at their closest range, the spacecraft signal is *also* at its peak strength, in addition to the noise contribution of Jupiter. The total system temperatures predicted for DSS 14 and DSS 15 are 22.6 and 42.2 K, respectively.[5]

To determine the degree to which the source is resolved on this array baseline, we must compare the fringe spacing to the angular size of the source. In our example, the observing frequency f_o is 2.3×10^9 Hz, and the maximum possible projected baseline is the physical separation between the two antennas, which is approximately 500 m. Thus, the smallest possible fringe spacing is 2.5×10^{-4} rad. At a range of 4.0 AU, Jupiter has an angular size on the order of 1×10^{-3} rad. Since these values are comparable, we cannot use either the long baseline limit or the short baseline limit in evaluating ρ (see Section I). However, for the purpose of determining the impact of the correlated noise in the most extreme case, we overestimate the degree of noise correlation using the upper bound on ρ, given by

$$\rho = \sqrt{\frac{T_{s_1} T_{s_2}}{T_1 T_2}} \approx 0.1 \tag{73}$$

B. Arraying Performance

Using the two P_T/N_o levels and correlation coefficient ρ found above, the ideal arraying gain G_A can be computed as a function of ψ using Eq. (25). A graph showing this relationship is shown in Fig. 17. Note that the arraying gain in this example is much smaller compared to our previous examples of two equal antennas, since the signal level of one antenna is approximately 10 dB lower than the other. For $\psi = 0$ deg, the correlated component of the noise adds maximally in phase, thus decreasing the arraying gain. By contrast, the background noise interferes destructively for $\psi = 180$ deg, resulting in greater arraying gain. Since the correlation coefficient is relatively low in this example, the difference between the best-case and worst-case scenarios is only about 0.45 dB.

Representative values for the carrier, subcarrier, and symbol loop bandwidths were chosen as 1.5, 0.4, and 0.07 Hz, respectively. For full-spectrum combining, a correlation bandwidth of $B_{corr} = 2$ kHz was used, with a correlation time of 15 s. The total degradation for FSC as a function of ψ is shown in Fig. 18, along with simulation points. Because the correlation coefficient ρ is relatively low in this example, the degradation is almost constant with respect to the phase parameter ψ. The combined P_T/N_o only varies by roughly 0.4 dB as ψ ranges from 0 to 180 deg; thus, the loop SNRs of the three loops also do not change much, and synchronization losses remain essentially constant.

[5] Predicts for noise and signal parameters were obtained from the Galileo S-Band Analysis Program (GSAP).

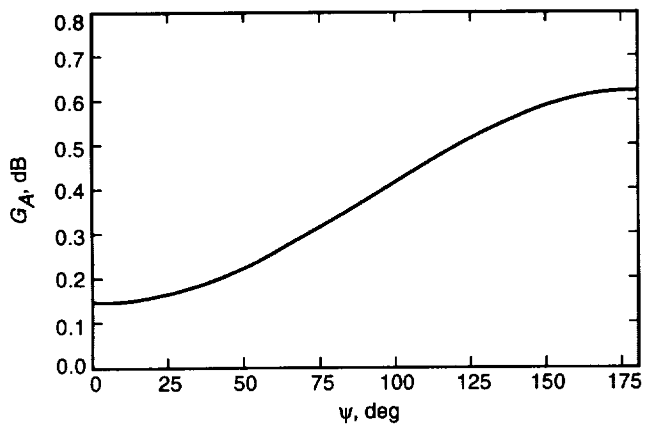


Fig. 17. Ideal arraying gain for Galileo signal parameters.

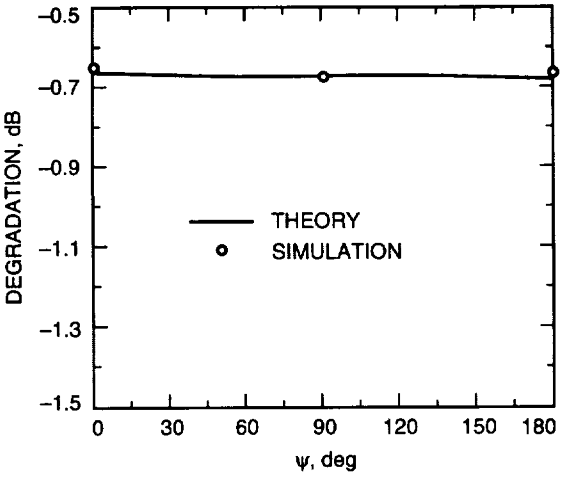


Fig. 18. FSC degradation for Galileo signal parameters.

The same signal parameters and loop bandwidths were used to simulate the complex-symbol combining case. A slight variation of the basic scheme, known as complex-symbol combining with aiding (CSCA), was implemented. This scheme is discussed in [2] as an option for arraying the Galileo signal. In CSCA, the subcarrier and symbol references from the receiver tracking the stronger signal are used to track the signal from the 34-m antenna as well. This technique can be used to perform complex-symbol combining even if the 34-m antenna signal is too weak to achieve subcarrier and symbol lock on its own. Thus, the loop SNRs for the 34-m antenna subcarrier and symbol loops are equal to the corresponding 70-m antenna loop SNRs.

Equation (59) can be applied to determine whether or not the "noise-only" channel is needed to phase the array. Substituting in values from above, we find

$$\frac{1}{\rho}\left(\frac{E_{s_1}}{N_{o_1}}\frac{E_{s_2}}{N_{o_2}}\right)^{1/2} = \frac{1}{\rho}\frac{1}{R_{sym}}\left(\frac{P_{T_1}}{N_{o_1}}\frac{P_{T_2}}{N_{o_2}}\right)^{1/2} \tag{74}$$

$$= 2.39 \tag{75}$$

Thus, the magnitude of the noise correlation vector is less than but comparable to that of the signal correlation vector. To illustrate the impact of the phase bias in aligning the signals, CSCA was simulated with both the compensating and uncompensating method for estimating the relative signal phase. In Fig. 19, we show the degradation for CSCA for these two cases. The correlation time used to estimate the relative signal phase was 2 s. Note that a shorter estimation interval than the full-spectrum combining case can be used here since the effective correlation bandwidth is equal to the data bandwidth of 200 Hz as opposed to 2 kHz for FSC. For the compensated case, the degradation is essentially constant since, once again, the noise correlation does not affect synchronization losses much. For the uncompensated case, the degradation becomes greater as the difference between the noise and signal phase ψ grows larger, since the noise correlation begins to bias the phase estimate further away from the relative signal phase. This effect can be seen graphically by referring once again to Fig. 5, where the complex-signal and noise correlations are represented as vectors.

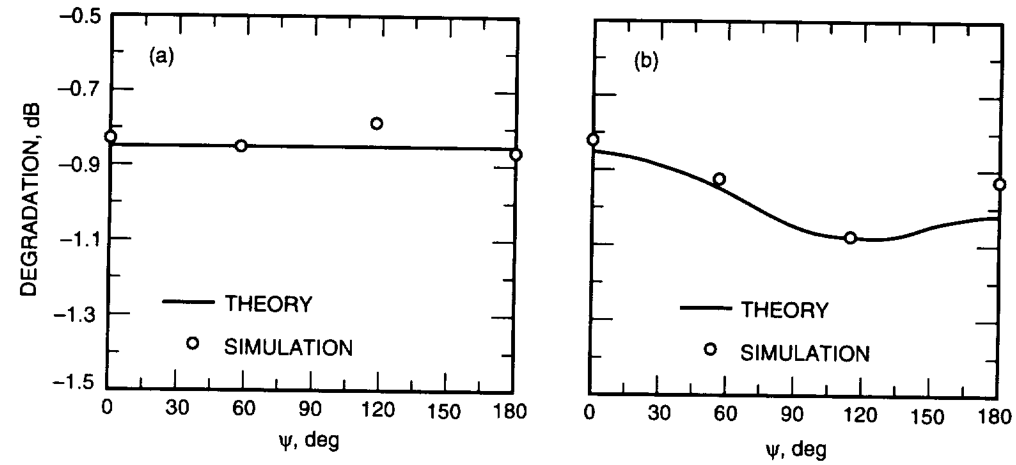


Fig. 19. CSC degradation for Galileo signal parameters: (a) phase compensated and (b) phase uncompensated.

VI. Conclusion

The effects of correlated noise on the full-spectrum combining and complex-symbol combining arraying schemes have been analyzed. As seen in Section II, accurate modeling of the noise correlation properties for a given antenna pair requires detailed analysis of factors such as the source structure and position, the antenna gain patterns, and the geometry of the array. However, the correlation coefficient can be determined easily in cases where the baseline is either very short or very long. These two extreme cases can be used to obtain a rough idea of what degree of noise correlation can be expected for a given scenario.

Describing the correlation between the various antenna pairs in an array by the parameters ρ_{ij} and ψ_{ij}, expressions for the ideal arraying gain and arraying degradation were derived. Several important differences from the uncorrelated noise case were noted. For a given set of signal levels (P_{T_i}/N_{o_i}), the ideal arraying gain when the noise is correlated may be higher *or* lower than when the noise waveforms are independent. This reflects the fact that the noise may add constructively or destructively, depending on the relative signal and noise phases (i.e., the ψ_{ij} parameters).

In addition, correlated noise can have a significant impact on the synchronization processes used to combine and demodulate the signals, which vary with the specific arraying technique used. Most notably, a bias due to the noise correlation is present in the conventional method of estimating the relative signal phases. Since the magnitude of this bias is proportional to the correlation bandwidth used, full-spectrum combining is potentially more sensitive to this problem than complex-symbol combining, depending on the specific method used to correlate the signals. A modified method of phase estimation, where the correlation due to the noise alone is measured and compensated for, can optionally be employed for both FSC and CSC, as necessary.

Acknowledgments

The contributions of numerous individuals to the work described in this article are gratefully acknowledged. I would like to thank David Rogstad and David Meier for their informative discussions on radio interferometry, Sami Hinedi for his insights on arraying techniques, Samson Million for his comments on software simulation methods, and Biren Shah and Mazen Shihabi for their assistance in editing and organizing the article.

References

[1] A. Mileant and S. Hinedi, "Overview of Arraying Techniques for Deep Space Communications," *IEEE Transactions on Communications*, vol. 42, nos. 2/3/4, pp. 1856–1865, February/March/April 1994.

[2] S. Million, B. Shah, and S. Hinedi, "A Comparison of Full-Spectrum and Complex-Symbol Combining Techniques for the Galileo S-Band Mission," *The Telecommunications and Data Acquisition Progress Report 42-116, October–December 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 128–162, February 15, 1994.

[3] R. Dewey, "The Effects of Correlated Noise in Intra-Complex DSN Arrays for S-Band Galileo Telemetry Reception," *The Telecommunications and Data Acquisition Progress Report 42-111, July–September 1992*, Jet Propulsion Laboratory, Pasadena, California, pp. 129–152, November 15, 1992.

[4] A. Thompson, J. Moran, and G. Swenson, *Interferometry and Synthesis in Radio Astronomy*, New York: John Wiley & Sons, 1986.

[5] H. Tan, "Performance of Residual Carrier Array-Feed Combining in Correlated Noise," *The Telecommunications and Data Acquisition Progress Report 42-121, January–March 1995*, Jet Propulsion Laboratory, Pasadena, California, pp. 131–147, May 15, 1995.

Appendix A

Performance of the FSC Correlator

For full-spectrum combining, the phase difference between two signals is estimated by performing one lowpass and one bandpass correlation, as described in Section III.B. After being filtered to some lowpass bandwidth, B_{lp} Hz, the signals from antenna 1 and antenna i are given by

$$\tilde{y}_{lp_1}(t) = \left[\sqrt{P_{C_1}} + j\sqrt{P_{D_1}}d(t)\left(\frac{4}{\pi}\right)\sum_{\substack{k=1\\ k\ odd}}^{M}\frac{\sin k\omega_{sc}t}{k}\right]e^{(j\omega t+\theta_1)} + \tilde{n}_{lp_1}(t) \tag{A-1}$$

$$\tilde{y}_{lp_i}(t) = \left[\sqrt{P_{C_i}} + j\sqrt{P_{D_i}}d(t)\left(\frac{4}{\pi}\right)\sum_{\substack{k=1\\ k\ odd}}^{M}\frac{\sin k\omega_{sc}t}{k}\right]e^{(j\omega t+\theta_i)} + \tilde{n}_{lp_i}(t) \tag{A-2}$$

where the subcarrier is expressed in terms of its sinusoidal components that are passed by the lowpass filter. The two signals passed through the bandpass filter of bandpass B_{bp} Hz contain only noise and are given by

$$\tilde{y}_{bp_1}(t) = \tilde{n}_{bp_1}(t) \tag{A-3}$$

$$\tilde{y}_{bp_i}(t) \ = \tilde{n}_{bp_i}(t) \tag{A-4}$$

The complex quantity used to estimate the relative signal phase $\phi_{1i} = \theta_1 - \theta_i$ is given by

$$\begin{aligned}
Z \ &= I + jQ \\
&= \frac{1}{T_{corr}} \int \tilde{y}_{lp_1} \tilde{y}^*_{lp_i} \ dt - \frac{B_{lp}}{B_{bp}} \frac{1}{T_{corr}} \int \tilde{y}_{bp_1} \tilde{y}^*_{bp_i} \ dt \\
&= \left(\sqrt{P_{C_1} P_{C_i}} + \sqrt{P_{D_1} P_{D_i}} H \right) e^{j\phi_{1i}} + \frac{1}{T_{corr}} \int (\tilde{n}_{s,n} + \tilde{n}_{lp_1} \tilde{n}^*_{lp_i}) \ dt - \frac{B_{lp}}{B_{bp}} \frac{1}{T_{corr}} \int \tilde{n}_{bp_1} \tilde{n}^*_{bp_i} \ dt \\
&= \left(\sqrt{P_{C_1} P_{C_i}} + \sqrt{P_{D_1} P_{D_i}} H \right) e^{j\phi_{1i}} + \tilde{N}
\end{aligned} \tag{A-5}$$

In most cases, the contribution of the signal–noise term $\tilde{n}_{s,n}(t)$ to the total noise power is much smaller than that of the noise–noise terms, and can be ignored. This is especially true if the P_T/N_o levels of the two signals are very low, or if large correlation bandwidths are used. By the Central Limit Theorem, the complex noise $\tilde{N}$ can be approximated as Gaussian if the correlation extends over many independent samples (i.e., if T_{corr} is much greater than the inverse correlation bandwidths). After averaging, the variance of the real and imaginary parts of $\tilde{N}$ can be shown to be equal to

$$\lambda_I \ = \ Var(N_I) \ = \frac{1}{T_{corr}} \left(B_{lp} + \frac{B^2_{lp}}{B_{bp}} \right) (N_{o_1} N_{o_i} + \alpha^2_{1i} \cos 2\phi^n_{1i}) \tag{A-6}$$

$$\lambda_Q \ = \ Var(N_Q) \ = \frac{1}{T_{corr}} \left(B_{lp} + \frac{B^2_{lp}}{B_{bp}} \right) (N_{o_1} N_{o_i} - \alpha^2_{1i} \cos 2\phi^n_{1i}) \tag{A-7}$$

where N_I and N_Q are the real and imaginary parts of $\tilde{N}$, respectively. The covariance of N_I and N_Q can be shown to be equal to

$$\lambda_{IQ} = Cov(N_I, \ N_Q) = \frac{1}{T_{corr}} \left(B_{lp} + \frac{B^2_{lp}}{B_{bp}} \right) \alpha^2_{1i} \sin 2\phi^n_{1i} \tag{A-8}$$

Furthermore, it is clear from Eq. (A-5) that the means of the real and imaginary parts of Z are given by

$$m_I \ = \ \left(\sqrt{P_{C_1} P_{C_i}} + \sqrt{P_{D_1} P_{D_i}} H \right) \cos \phi_{1i} \tag{A-9}$$

$$m_Q \ = \ \left(\sqrt{P_{C_1} P_{C_i}} + \sqrt{P_{D_1} P_{D_i}} H \right) \sin \phi_{1i} \tag{A-10}$$

Equations (A-6), (A-7), (A-9), and (A-10) can be combined to compute the correlator SNR as defined in [1], i.e.,

$$
\begin{aligned}
SNR_{corr,fsc} &= \frac{E[Z]E^*[Z]}{E[ZZ^*] - E[Z]E^*[Z]} \\
&= \frac{m_I^2 + m_Q^2}{\lambda_I + \lambda_Q} \\
&= \frac{T_{corr}}{2(B_{lp} + (B_{lp}^2/B_{bp}))} \left(\sqrt{\frac{P_{C_1}}{N_{o_1}}\ \frac{P_{C_i}}{N_{o_i}}} + \sqrt{\frac{P_{D_1}}{N_{o_1}}\ \frac{P_{D_i}}{N_{o_i}}\ H} \right)^2
\end{aligned}
\tag{A-11}
$$

Equations (A-9), (A-10), and (A-6) through (A-8) can be used to determine the joint density function $p_{I,Q}(I,Q)$. Since the density of $\hat{\phi}_{1i} = \tan^{-1}(Q/I)$ is the desired quantity, we express the joint density function in terms of polar coordinates, using the variable definitions

$$r \triangleq \sqrt{I^2 + Q^2} \tag{A-12}$$

$$\phi \triangleq \tan^{-1}\left(\frac{Q}{I}\right) \tag{A-13}$$

The density function for jointly Gaussian random variables is given in polar form by

$$
\begin{aligned}
f_{r,\phi}(r,\phi) &= \frac{r}{2\pi(\lambda_I\lambda_Q - \lambda_{IQ})} \\
&\times \exp\left(- \frac{\lambda_I(r\cos\phi - m_I)^2 - 2\lambda_{IQ}(r\cos\phi - m_I)(r\sin\phi - m_Q) + \lambda_Q(r\sin\phi - m_Q)^2}{2(\lambda_I\lambda_Q - \lambda_{IQ})} \right)
\end{aligned}
\tag{A-14}
$$

Integrating Eq. (A-14) with respect to r yields the marginal density of ϕ alone. Expressing the phase estimate density in terms of the estimation error $\Delta\phi = \hat{\phi}_{1i} - \phi_{1i}$ yields

$$f_\phi(\Delta\phi) = G_1 \exp\left(-SNR_{corr,fsc} \frac{1 - \rho^2\cos 2\psi}{1 - \rho^4} \right) \left[1 + \sqrt{\pi}\ G_2\ e^{G_2^2} \operatorname{erf}\ G_2 + 1) \right] \tag{A-15}$$

where

$$G_1 = \frac{1 - \rho^4}{2\pi\,(1 - \rho^2\cos(2\psi - \Delta\phi))} \tag{A-16}$$

$$G_2 = \sqrt{SNR_{corr,fsc}} \frac{\cos\Delta\phi - \rho^2\cos(2\psi - \Delta\phi)}{(1 - \rho^4)(1 - \rho^2\cos(2\psi - \Delta\phi))} \tag{A-17}$$

Appendix B
Performance of the CSC Correlator

The method of estimating the relative signal phases for complex-symbol combining is analogous to the full-spectrum combining algorithm; using the extra correlation to compensate for the noise bias, the complex correlation can be expressed as

$$
\begin{aligned}
Z &= \frac{1}{N}\sum_{k=1}^{N}\tilde{Y}_1(k)\tilde{Y}_i^*(k) - \frac{1}{N}\sum_{k=1}^{N}\tilde{N}_1(k)\tilde{N}_i^*(k) \\
&= \sqrt{P_{D_1}P_{D_i}}\ \overline{C_{sc_1}}\ \overline{C_{sy_1}}\ \overline{C_{sc_i}}\ \overline{C_{sy_i}}e^{j\phi_{1i}} + \frac{1}{N}\sum_{k=1}^{N}\sqrt{P_{D_1}}C_{sc_1}C_{sy_1}e^{j\theta_1}\tilde{N}_i^*(k) \\
&\quad + \frac{1}{N}\sum_{k=1}^{N}\sqrt{P_{D_i}}C_{sc_i}C_{sy_i}e^{-j\theta_i}\tilde{N}_1^*(k) + \frac{1}{N}\sum_{k=1}^{N}\tilde{N}_1(k)\tilde{N}_i^*(k) - \frac{1}{N}\sum_{k=1}^{N}\tilde{N}_1'(k)\tilde{N}_i'^*(k) \\
&= \sqrt{P_{D_1}P_{D_i}}\ \overline{C_{sc_1}}\ \overline{C_{sy_1}}\ \overline{C_{sc_i}}\ \overline{C_{sy_i}}e^{j\phi_{1i}} + \tilde{N}
\end{aligned}
\tag{B-1}
$$

where N is the number of symbols averaged over, given by $N = T_{corr}/T_{sym}$, and the noise term $\tilde{N}$ has zero mean. The statistics of this noise can be analyzed in the same manner as before; here, the effective correlation bandwidth for both the lowpass and the bandpass correlation is $R_{sym}/2$. Using the definition given by Eq. (32), the correlator SNR can be shown to be equal to

$$
SNR_{corr,csc} = \frac{P_{D_1}}{N_{o_1}}\frac{T_{corr}\overline{C_{sc_1}}^2\ \overline{C_{sy_1}}^2\ \overline{C_{sc_i}}^2\ \overline{C_{sy_i}}^2}{\overline{C_{sc_i}^2}\ \overline{C_{sy_i}^2} + \overline{C_{sc_1}^2}\ \overline{C_{sy_1}^2}(1/\gamma_i) + (N_{o_i}/P_{D_i})2R_{sym}}
\tag{B-2}
$$

The density function for the phase estimation error can be found in a manner analogous to that applied in Appendix A. The only difference is in the expression for the correlator SNR; otherwise, both problems are inherently governed by the same mathematics. The density function for the phase estimation error $\Delta\phi_{1i}$ is thus given by Eq. (A-15), with $SNR_{corr,fsc}$ replaced by $SNR_{corr,csc}$.

N95- 32238

TDA Progress Report 42-121

May 15, 1995

A Seismic Data Compression System Using Subband Coding

A. B. Kiely and F. Pollara
Communications Systems Research Section

This article presents a study of seismic data compression techniques and a compression algorithm based on subband coding. The algorithm includes three stages: a decorrelation stage, a quantization stage that introduces a controlled amount of distortion to allow for high compression ratios, and a lossless entropy coding stage based on a simple but efficient arithmetic coding method. Subband coding methods are particularly suited to the decorrelation of nonstationary processes such as seismic events. Adaptivity to the nonstationary behavior of the waveform is achieved by dividing the data into separate blocks that are encoded separately with an adaptive arithmetic encoder. This is done with high efficiency due to the low overhead introduced by the arithmetic encoder in specifying its parameters. The technique could be used as a progressive transmission system, where successive refinements of the data can be requested by the user. This allows seismologists to first examine a coarse version of waveforms with minimal usage of the channel and then decide where refinements are required. Rate-distortion performance results are presented and comparisons are made with two block transform methods.

I. Introduction

A typical seismic analysis scenario involves collection of data by an array of seismometers, transmission over a channel offering limited data rate, and storage of data for analysis. Seismic data analysis is performed for monitoring earthquakes and for planetary exploration, as in the planned study of seismic events on Mars. Seismic data compression systems are required to cope with the transmission of vast amounts of data over constrained channels and must be able to accurately reproduce both low-energy seismic signals and occasional high-energy seismic events.

We describe a compression algorithm that includes three stages: a decorrelation stage based on subband coding, a uniform quantization stage, and a lossless entropy coding stage based on arithmetic coding. Rate-distortion performance results are presented and comparisons are made with two block transform methods: the discrete cosine transform (DCT) and the Walsh–Hadamard transform (WHT).

Subband coding methods are particularly suited to the decorrelation of nonstationary processes such as seismic events. For most seismic data, signal energy is more concentrated in the low-frequency subbands, which suggests the use of nonuniform subband decomposition. The decorrelation stage is implemented by quadrature mirror filters using a lattice structure. Adaptivity to the nonstationary behavior of the waveform is achieved by dividing the data into blocks that are separately encoded.

The compression technique described in this article can be used as a progressive transmission system, where successive refinements of the data can be requested by the user. This allows reconstruction of a low-resolution version of the waveform after receiving only a small portion of the compressed data. This could allow seismologists to make a preliminary examination of the waveform with minimal usage of the channel and then decide where high-resolution refinements are desired.

In general, given a fixed transmission rate, lossy compression algorithms applied to high-accuracy instruments deliver higher scientific content than lossless compression methods applied to lower accuracy instruments.

II. Subband Decomposition

In the analysis stage of subband coding, a signal is filtered to produce a set of subband components, each having smaller bandwidth than the original signal. Because of this limited bandwidth, each component is downsampled, so that the subband transformed data contain as many data points as the original signal. The subband components are then quantized and compressed. In the synthesis stage, the reconstructed signal is formed by adding together the subbands obtained by applying the inverse filters to upsampled versions of the subband components.

The analysis and synthesis filters used here are finite impulse response (FIR) quadrature mirror filters (QMF) implemented using the lattice structures shown in Figs. 1 and 2, which are described in [7,1]. Analysis and synthesis quadrature mirror filters of order $2M$ are implemented using an M-stage lattice structure. Suitable lattice filters can be found in [1, p. 267] and [7, p. 310].

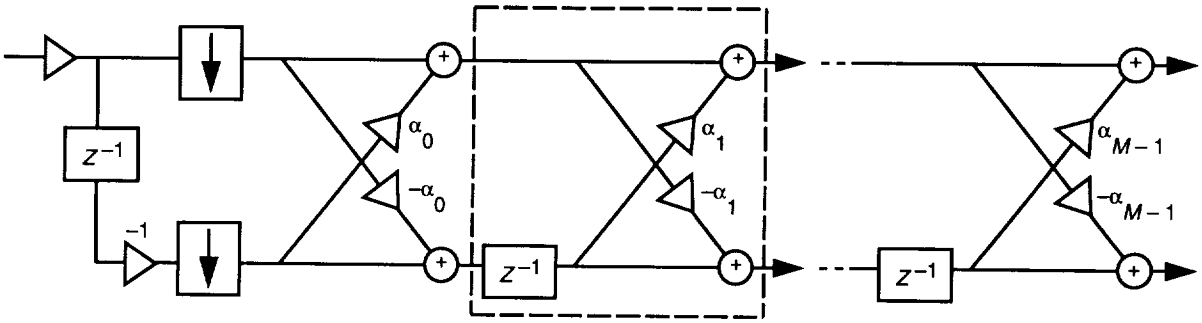


Fig. 1. Analysis filter structure. (The stage inside the box is repeated.)

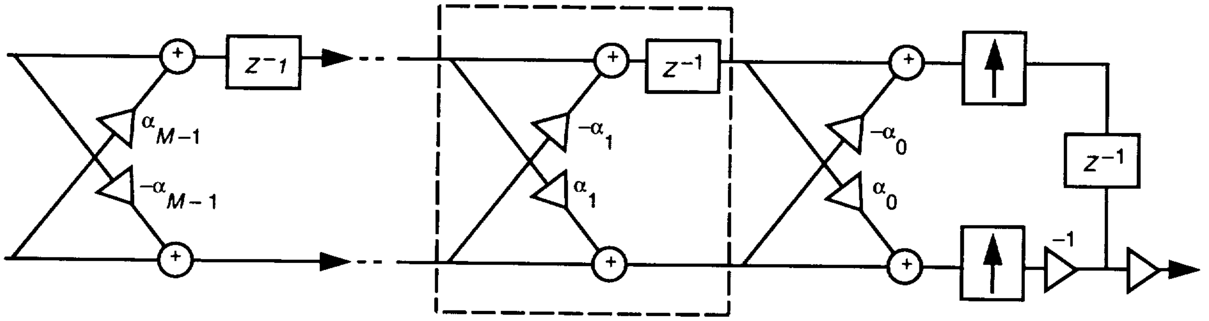


Fig. 2. Synthesis filter structure.

For most seismic data samples, signal energy is concentrated primarily in the low subbands.[1] Figures 3 and 4 give two periodograms (power spectral density estimates [4]) for seismic data. The uneven distribution of spectral energy in seismic signals provides the basis for subband coding source-compression techniques. For effective signal coding, subspectra containing more energy deserve higher priority for further processing.

A subband decomposition that tends to work well for seismic data is the dyadic tree decomposition shown in Fig. 5. The signal is first split into low- and high-frequency components in the first level. A two-band subband decomposition uses high-pass and low-pass digital filters to decompose a data sequence into high (H) and low (L) subbands, each containing half as many points as the original sequence. The filter is repeated to further decompose the low subband. This process may be repeated several levels.

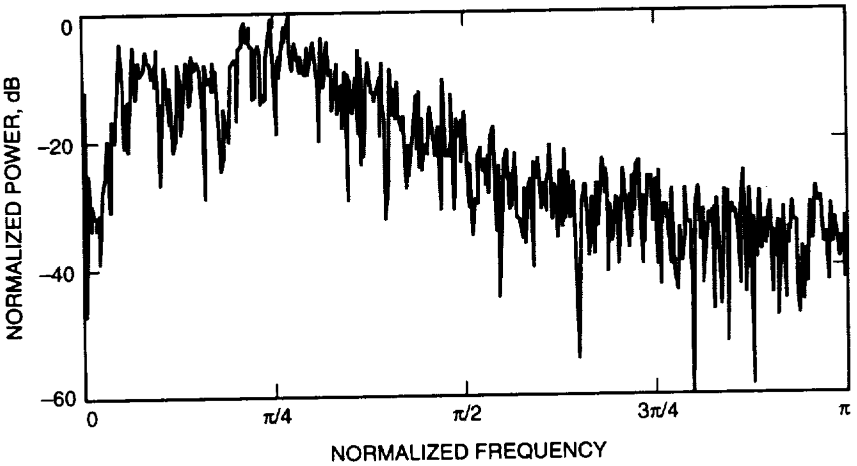


Fig. 3. Periodogram of 1024-point EHZ (100 samples/s) data sample containing seismic event.

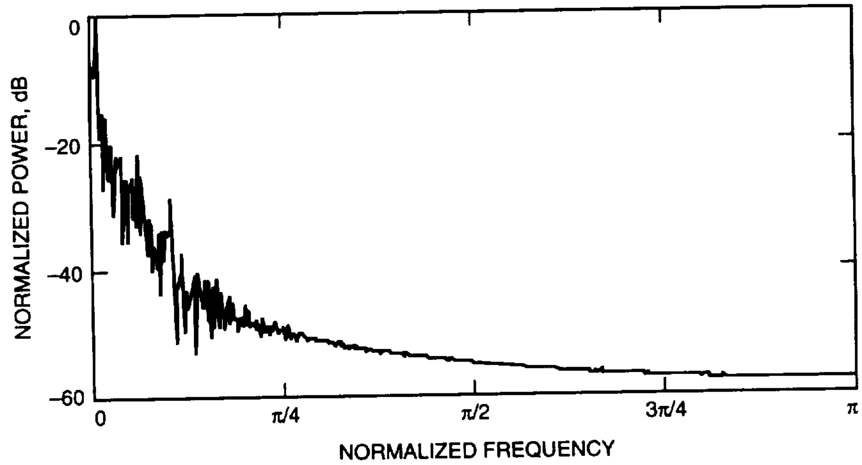


Fig. 4. Periodogram of 1024-point BHZ (20 samples/s) data sample containing seismic event.

[1] This generally applies to the event (EHZ) and broadband (BHZ) seismic data components, which have sample rates of 100 and 20 samples/s, respectively. Energy in long-period (LHZ) data, which has a sample rate of only 1 sample/s, is typically not as concentrated in the low frequencies. However, because of the much lower sample rate, compression of this component is not as important as the others. A different subband decomposition could be implemented to accommodate this type of data.

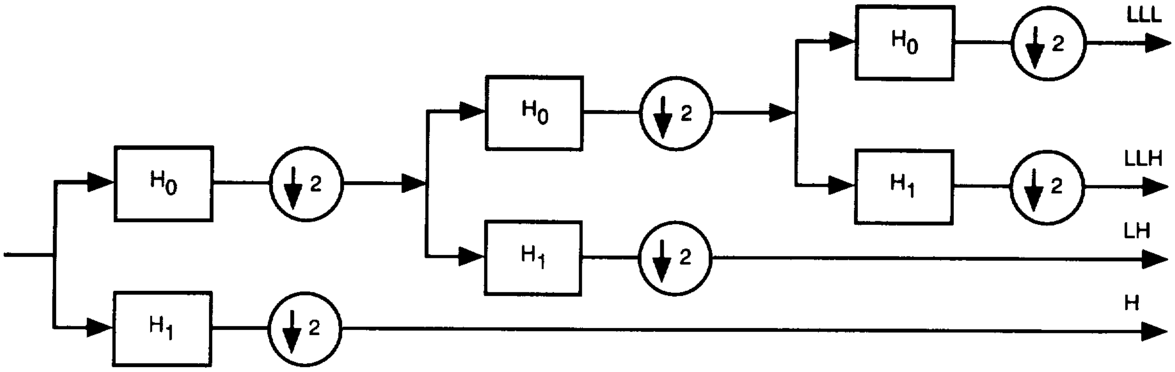


Fig. 5. Subband decompositions.

Increasing the number of subbands produces diminishing rate-distortion returns, with gains often observable only at very high compression ratios. One reason for this is that, after several decompositions, the energy is no longer so highly concentrated in the lowest subband.

So that a filtered block has the same length as the original, each block is periodically extended (i.e., repeated in time) before filtering, and the components corresponding to a single period of the filtered extended signal are taken as the filtered signal. If this operation were not performed, the length of the filtered signal would exceed the original block length. An unfortunate side effect of periodic extension is that it often produces high-frequency components at the edges of data blocks, an effect whose impact increases with filter length. These components are not as easily compressed as the rest of the subband data and are separated for compression purposes. Longer filters are also more likely to introduce noticeable spurious effects at the onset of a high-energy seismic event, as we shall see in Section VI. It is also worth noting that longer filters generally do not dramatically outperform shorter filters, as we will see in the following section.

III. Comparing Subband Coding to Block Transforms

For comparison purposes, we also examined the discrete cosine transform (DCT), a popular technique used in the compression of two-dimensional data (e.g., images). A general description of the DCT as used in the Joint Photographic Experts Group (JPEG) compression algorithm can be found in [5, pp. 113–128]. The DCT can also be applied to one-dimensional data, as is done here.

The data are partitioned into blocks of length 8, the DCT of each block is computed using the 8 × 8 DCT matrix, and these transformed values are uniformly quantized. A different quantizer step size could be used for each coefficient, but in practice, for most seismic data samples, near-optimum performance is obtained when all quantizers use the same step size. The quantized coefficients are arranged in groups of 8 blocks for subsequent coding, so that 64 transformed coefficients are encoded at a time. In this way, the procedure is similar to a one-dimensional version of the JPEG algorithm. The lowest frequency (dc) quantized coefficients are encoded using differential pulse-code modulation (DPCM) and Huffman coding, except at very low rates, when a run-length code is used. The remaining (ac) coefficients are run-length encoded, in order of increasing frequency. The run-length encoding used is the same as that described in [5, pp. 114–115].

We also used the same algorithm with an 8 × 8 WHT in place of the DCT, separately encoding each coefficient. The WHT performed uniformly worse (see Fig. 6). To make a fair comparison with subband coding, we compared the block transform compression methods to subband coding combined with Huffman coding of the quantizer output, rather than the arithmetic coding procedure to be described in the next section.

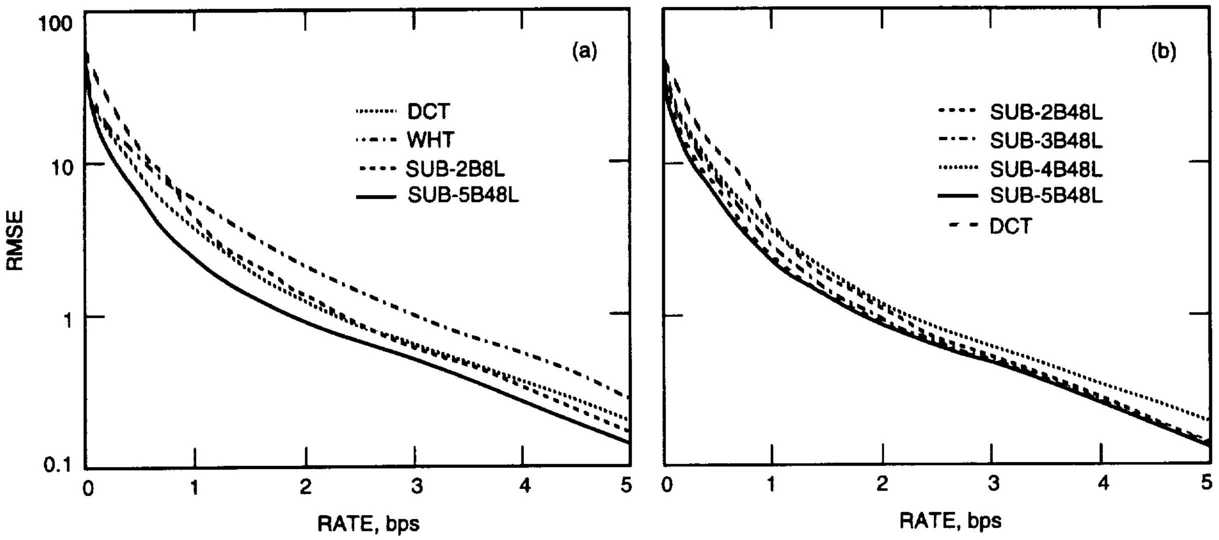


Fig. 6. Rate-distortion performance for various compression techniques applied to a seismic data sample: (a) comparison with block transform methods and (b) comparison of different subband decompositions.

Rate-distortion curves for a seismic data sample using these different techniques are shown in Fig. 6. The labels on the curves corresponding to subband coding identify the number of subbands and the particular filters used. For example, "3B8L" refers to a three-band decomposition using an order-8 FIR filter. In terms of root-mean-square error (RMSE), subband coding is able to outperform the DCT and WHT with only moderate complexity.

IV. Entropy Coding Stage: Arithmetic Coding

Anyone who has experienced an earthquake knows that the energy present in a seismic signal can vary tremendously over time. Consequently, seismometers have a large dynamic range, and it is desirable to have an adaptive compression system capable of transmitting low-energy and high-energy signals reliably.

A block of m data samples produces m subband coded samples. Because of the downsampling operation, half of these are high-subband samples, one-fourth are low–high-subband samples, etc. All of the samples from a particular subband are quantized and encoded together block adaptively. Because this is a block-to-block encoding procedure, the effects of a channel error are confined to the block during which that error occurs. The block encoding provides the additional benefit of adaptivity.

The output of the subband coding stage is a sequence of real numbers that are quantized and then compressed. For seismic data, as with many other types of data, these components are generally zero-mean, roughly symmetric, and have a probability density that is decreasing as we move away from the origin. This is illustrated in Fig. 7, which gives an empirical probability density function (pdf) of signal amplitude from a low-pass-filtered seismic data sample.

The compression scheme we use is bit-wise arithmetic coding [2]. A high-resolution quantizer is used, and the quantized values are mapped into fixed-length binary codewords. Figure 8 illustrates the bit assignment for a four-bit quantizer: The first bit indicates the sign of the quantizer reconstruction point, and each successive bit gives progressively higher resolution information. Because the pdf is zero mean and decreasing as we move away from the origin, a zero will be more likely than a one in every bit position. This redundancy is exploited using a binary arithmetic encoder to achieve compression.

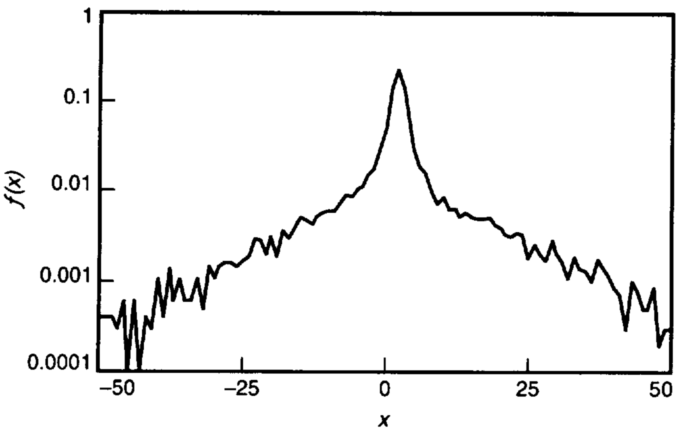


Fig. 7. Empirical pdf for low-pass subband filtered data.

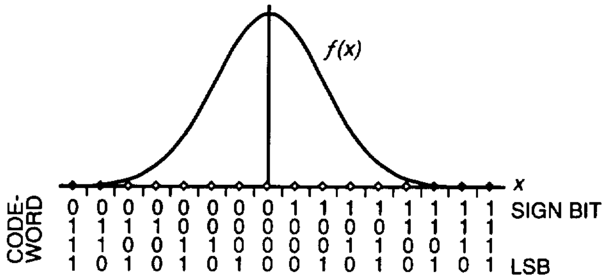


Fig. 8. Codeword assignment for the four-bit quantizer.

Codewords corresponding to each subband are grouped together. The sign bits of the codeword sequence are encoded using a block-adaptive binary-input binary-output arithmetic encoder described in [2]. The next most significant bits are similarly encoded, and so on. Each bit sequence (or layer) is encoded independently— at the ith stage the arithmetic coder calculates (approximately) the *unconditional* probability that the ith codeword bit is a zero.

The obvious loss is that we lose the benefit of interbit dependency. For example, the probability that the second bit is a zero is not in general independent of the value of the first bit, though the encoding procedure acts as if it were. Traditional Huffman coding of the quantized samples does not suffer from this loss. However, for many sources, such as Gaussian and Laplacian sources, this loss is quite small [2]. In fact, for many practical sources with low entropy, this technique has lower redundancy than Huffman coding, because the arithmetic coder is not required to produce an output symbol for every input symbol.

Because the interbit dependencies are ignored, very little overhead information is required (i.e., long tables of Huffman codewords are unnecessary). The overhead required for bit-wise arithmetic encoding increases linearly in the number of codeword bits. By contrast, the overhead of block-adaptive Huffman coding increases exponentially in the number of codeword bits unless we are able to cleverly exploit additional information about the source [3].

Another advantage is that, as we will see in the next section, this technique is naturally progressive. In a progressive transmission system, each successive data segment transmitted provides higher-resolution information about the signal. Using a buffer, we can choose to transmit only some of the data segments. This provides a convenient method for trading rates between blocks, so that more resources can be devoted to reproducing the high-energy signal blocks.

V. Progressive Transmission Behavior

In designing a compression system to be used in progressive transmission or in situations where rate constraints may result in the loss of data, it is important to consider the rate-distortion behavior of the system when only portions of the compressed data have been received. Such performance can be improved simply by careful choice of the order in which the compressed data are transmitted.

The typical characteristics of subband-filtered seismic data motivate our transmission strategy. Because the probability density for subband-filtered seismic data is generally zero mean (see Fig. 7), the sign bit layers of each subband usually have high entropy. Because the energy in seismic waveforms is often quite small, the high-order bit layers (excluding the sign bit) often consist entirely of zeros or can be readily compressed using the block-adaptive arithmetic encoder. Finally, as mentioned in Section II, periodic extension of the data is required in the subband filtering stage, which often produces high-frequency components at the start of data segments. These initial values, which we call transients, are encoded separately from the rest of the data. All but the lowest subband contain these transients.

Generally speaking, we transmit compressed data ordered from the most significant bit layer to the least significant bit (LSB) layer, and within this order, proceeding from the lowest frequency to the highest frequency subband. Initially, we skip the sign bit layer and begin with the next most significant bit layer. If this layer consists entirely of zeros (which is usually the case), a single "0" is transmitted and we move on to the same layer in the next higher subband. For every subband, a "0" is transmitted for each layer consisting entirely of zeros until a "1" is transmitted at some layer ℓ, denoting that the ℓth layer is not all zeros. At this point, we transmit the sign bits (using the block-adaptive arithmetic coding procedure already described). Then the transients for the subband are transmitted using run-length encoding of the leading zeros, and then the (compressed) ℓth bit layer is transmitted. Then we proceed to the ℓth layer for the next higher subband. Each subsequent bit layer of the subband is sent, compressed by arithmetic coding.

Because the order of transmission is determined using a rather simple decision procedure, the additional overhead required to describe the transmission order is quite small—it consists only of occasional one-bit flags. As an example, Fig. 9 shows a seismic data sample along with waveforms reconstructed from only small portions of compressed data for a 51.2-s (1024-point) block.

The rate-distortion progressive transmission performance of this system for one seismic data sample can be seen in Fig. 10. The highest rate point of each curve is the final design goal, and the rest of the curve shows the rate-distortion performance when the signal is reconstructed using only portions of the data. It is remarkable that the curves are nearly indistinguishable. Note that a system designed to transmit at a rate of 5 bits per sample (bps) but cut off at only 2.5 bps performs almost as well as a system designed to operate at 2.5 bps.

VI. Distortion Measures and Artifacts

In the previous sections, we have been mostly concerned with the mean-square error (MSE) distortion measure. However, mean-square distortion may not be a sufficient indicator of fidelity for seismic analysis purposes. For example, Spanias et al. [6] examined the effect of transform data compression methods on estimation of the body wave magnitude, which they call "the key parameter used in seismic analysis." Other distortion measures may be more relevant, depending on the interests of the seismologists who will ultimately analyze the data. Unfortunately, we do not know of a distortion measure that seismologists will widely accept as the most useful.

Artifacts are erroneous features that may appear in the reconstructed waveform. Different algorithms create different artifacts depending on their modes of operation. For example, "blockiness" is an artifact commonly associated with block transforms such as the DCT, while "ringing" may be produced by

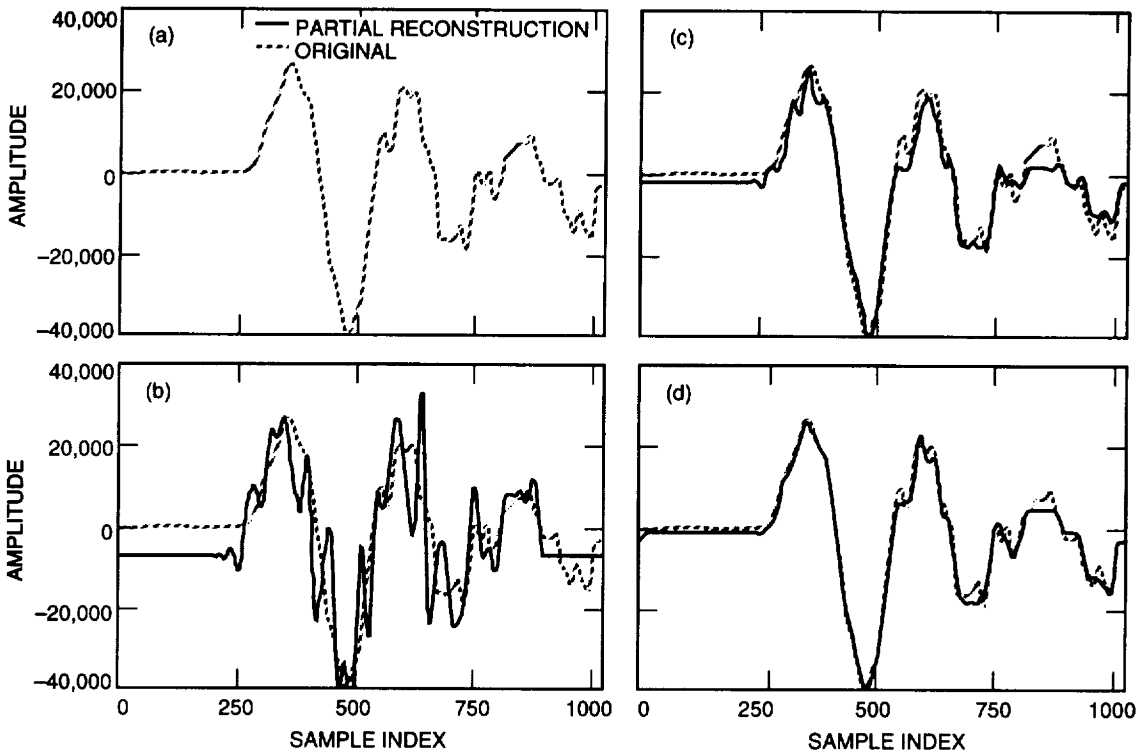


Fig. 9. Progressive transmissioin example: (a) original, (b) 0.1 bps, (c) 0.2 bps, and (d) 0.3 bps.

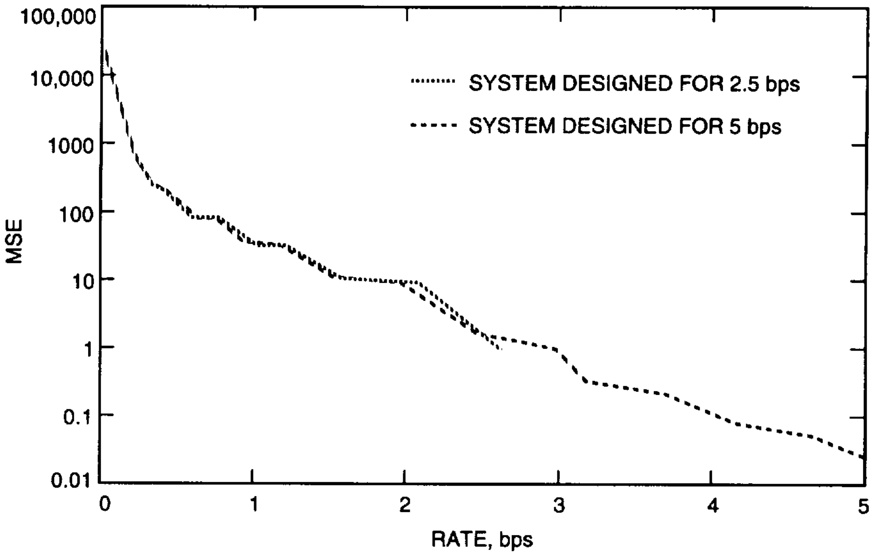


Fig. 10. Progressive transmission performance.

subband coding using a filter with a too sharp response. Even a given algorithm may exhibit different artifacts depending on the bit rate at which it is operated. Some artifacts may be more objectionable than others for correct waveform interpretation.

In this section, we illustrate two artifacts that may be observable in subband coding depending on the mode of operation and the compression ratio. Understanding the causes and cures for such artifacts allows seismologists to give meaningful feedback to engineers in deciding what features of a compression system are most important.

We are actively trying to engage the seismology community to characterize any essential artifacts produced by the proposed method [8]. One of the results of this interaction was the objection of seismologists to the precursor artifact created by a particular subband filter, as shown in Fig. 11(b). After determining that such an artifact was due to a filter with a too sharp response, we experimented with different, shorter filters, producing the result shown in Fig. 11(c), which reduces the precursor problem while preserving essentially the same compression ratio.

A different artifact is introduced when the quantizer step size is quite large (this equivalent effect may occur if the waveform is reconstructed using only a portion of the data). In this case, each subband will have low resolution, and because most of the energy is contained in the low frequencies, the high-frequency subbands may all be zeroed out. This may produce the interesting smoothing effect that can be observed in the periodogram of the reconstructed waveform shown in Fig. 12. If this frequency range has more significance than the others, the corresponding subbands could be assigned higher priority in the transmission and quantization stages.

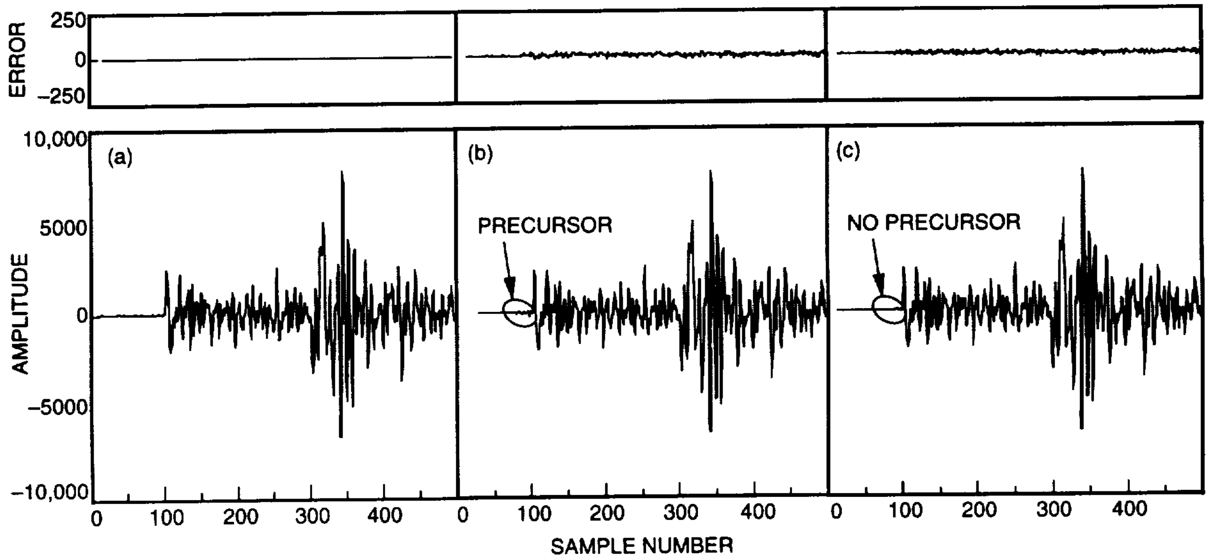


Fig. 11. Original and reconstructed waveforms for two different filters: (a) original, 24 bps, (b) reconstructed, 0.8 bps, and (c) reconstructed, 0.8 bps (improved filter).

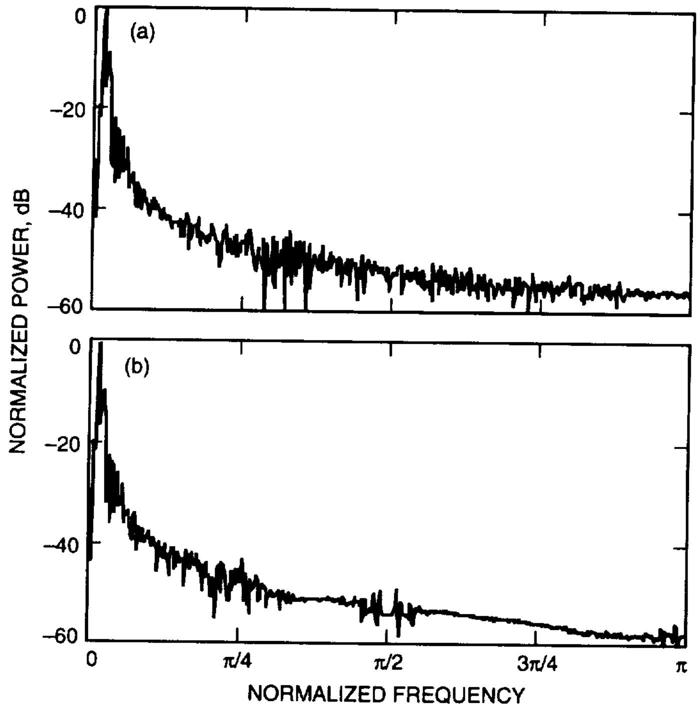


Fig. 12. Periodograms of 1024-point BHZ (20 samples/s) background (i.e., nonevent) data constructed from (a) the original and (b) the reconstructed waveform with low-resolution quantizer.

References

[1] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Processing*, San Diego California: Academic Press, 1992.

[2] A. B. Kiely, "Bit-Wise Arithmetic Coding for Data Compression," *The Telecommunications and Data Acquisition Progress Report 42-117, January–March 1994*, Jet Propulsion Laboratory, Pasadena, California, pp. 145–160, May 15, 1994.

[3] R. J. McEliece and T. H. Palmatier, "Estimating the Size of Huffman Code Preambles," *The Telecommunications and Data Acquisition Progress Report 42-114, April–June 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 90–95, August 15, 1993.

[4] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*, Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

[5] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*, Bellingham, Washington: SPIE Press, 1991.

[6] A. S. Spanias, S. B. Jonsson, and S. D. Stearns, "Transform Methods for Seismic Data Compression," *IEEE Trans. Geoscience and Remote Sensing*, vol. 29, no. 3, pp. 407-416, May 1991.

[7] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Englewood Cliffs, New Jersey: PTR Prentice-Hall, 1993.

[8] *Workshop on the Use of Data Compression in Seismic Data Proceedings*, Jet Propulsion Laboratory, Pasadena, California, March 2, 1994.

TDA Progress Report 42-121

May 15, 1995

DSS-24 Microwave Holography Measurements

D. J. Rochblatt, P. M. Withington, and H. J. Jackson
Ground Antennas and Facilities Engineering Section

The JPL DSN Microwave Antenna Holography System (MAHST) was applied to the newly constructed DSS-24 34-m beam-waveguide antenna at Goldstone, California. The application of MAHST measurements and corrections at DSS 24 provided the critical RF performance necessary to not only meet the project requirements and goals, but to surpass them. A performance increase of 0.35 dB at X-band (8.45 GHz) and 4.9 dB at Ka-band (32 GHz) was provided by MAHST, resulting in peak efficiencies of 75.25 percent at X-band and 60.6 percent at Ka-band (measured from the Cassegrain focus at f1). The MAHST enabled setting the main reflector panels of DSS 24 to 0.25-mm rms, making DSS 24 the highest precision antenna in the NASA/JPL DSN. The precision of the DSS-24 antenna (diameter/rms) is 1.36×10^5*, and its gain limit is at 95 GHz.*

I. Introduction

The JPL Microwave Antenna Holography System (MAHST) (Fig. 1) [1] has become the leading technique for increasing the performance of the large NASA/JPL DSN antennas, especially at the shorter wavelengths (X-band (8.45 GHz) and Ka-band (32 GHz)). The MAHST provides an efficient and low-cost technique to optimize and maintain the performance and operation of the large DSN antennas, providing far-field amplitude and phase pattern measurement with a 90-dB dynamic range, and enabling high-resolution and high-precision antenna imaging with a standard deviation of 100 μm. The panel setting/unbending screw adjustment is provided with an accuracy of 10 to 20 μm. Fast subreflector position optimization is provided, which increases the antenna performance capacity and pointing accuracy. The MAHST is a portable system that can be shipped to any DSN antenna around the world and can be easily interfaced with its encoders and antenna drive systems. The MAHST was designed utilizing many off-the-shelf commercially available components. The remaining parts were designed and built at JPL. The MAHST has been successfully tested and demonstrated at the NASA/JPL DSN [1,2].

The microwave holography technique utilizes the Fourier transform relationship between the complex far-field radiation pattern of an antenna and the complex aperture field distribution. Resulting aperture phase and amplitude distribution data are used to derive various crucial performance parameters, including panel alignment, subreflector position, antenna aperture illumination, directivity at various frequencies, and gravity deformation effects [3,4]. Strong continuous wave (CW) signals obtained from geostationary satellite beacons are utilized as far-field sources. Strong CW beacon signals are available on nearly all satellites at Ku-band (10.7 to 12.7 GHz), X-band (7.0 to 7.8 GHz), and C-band (3.7 to 4.2 GHz). A portable 2.8-m reference antenna (Fig. 1) is used as a phase reference and provides the signal to the receiver phase-lock-loop (PLL) channel. The intermediate-frequency (IF) section of a Hewlett Packard Microwave Receiver (HP8530A) and an external JPL-designed and -built PLL enable

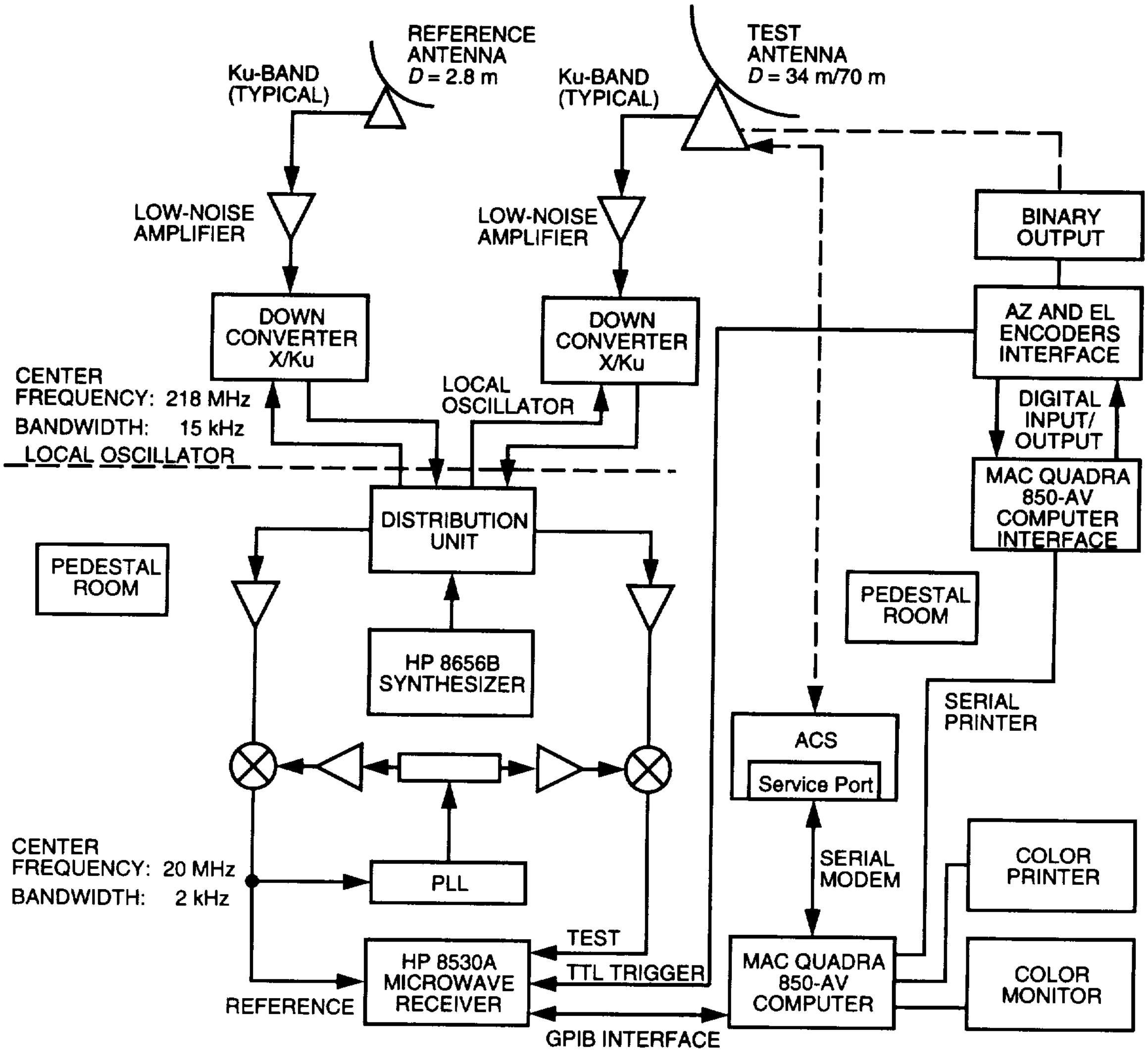


Fig. 1. MAHST block diagram.

precision amplitude and phase measurements of the ground antenna sidelobes with a 90-dB dynamic range. The far-field data are collected by continuously scanning the test antenna against the signal from a geosynchronous satellite, sampling a two-dimensional grid directly on the u, v (direction cosine) space. Each subscan start position is updated in real time to track the predicted orbit position of the geosynchronous satellite. The angular extent of the response that must be recorded is inversely proportional to the size of the required resolution cell in the processed holographic maps. The data processing provided with the system computes the desired information.[1] It is the information in the surface error map that is used to compute the adjustments of the individual panels in an overall main reflector best-fit reference frame. The amplitude map provides valuable information about the energy distribution in the antenna aperture. A short summary of the theory is presented in Appendix A.

[1] D. J. Rochblatt, *A User Manual, Data Processing Software for Microwave Antenna Holography: Computer Programs for Diagnostics, Analysis, and Performance Improvement of Large Reflector and Beam Waveguide Antennas*, JPL D-10237 (internal document), Jet Propulsion Laboratory, Pasadena, California, January 15, 1993.

II. Holographic Measurements and Results

The holographic measurements of DSS 24 were conducted during May 13 through 23, 1994 (Table 1). Four high-resolution (33.7-cm), four medium-resolution (84.8-cm), and one low-resolution (172-cm) measurements were performed (for a total of nine). Diagnostics, analysis, subreflector position, and panel setting listing were all derived on site. The antenna panels were reset on May 19, 1994 (excluding panels under the shadow areas of the quadripod). Eight measurements were made at the rigging angle of 46.3 deg, from the antenna Cassegrain focus at f1, utilizing the beacon signal at Ku-band (11.9225 GHz) from the GSTAR-1 satellite. Only one medium-resolution measurement at the low-elevation angle of 12.7 deg (f1 focus) was made due to the short time allocated for the holographic measurements. The beacon signal from the INTELSAT-V (307) satellite at Ku-band (11.7009 GHz) was utilized for the low-elevation measurement.

The data acquisition time for the high-resolution maps required for panel setting was 6.5 h. The data processing for obtaining panel setting information took 8 h. It took an additional 8 h to actually reset the panels of the antenna. The measurement and data processing time required for subreflector position correction for a 34-m antenna is approximately 2 h (two iterations).

Table 1. DSS-24 holographic measurements.

Date	File no.	EL angle, deg	Array size	Remarks
5/13/94	DSN006	46.4	25 × 25	Subreflector correction
5/13/94	DSN007	46.3	51 × 51	Verification
5/14/94	DSN008	46.3	127 × 127	Panel setting derivation
5/16/94	—[a]	—[a]	—[a]	Briefing at JPL
5/17/94	DSN009	46.3	51 × 51	Geometry confirmation
5/18/94	DSN010	46.3	121 × 121	Repeatability verification
5/19/94	—[a]	—[a]	—[a]	Panel setting
5/19/94	DSN011	46.3	51 × 51	After panel setting
5/20/94	DSN012	46.3	127 × 127	After panel setting and touch up
5/22/94	DSN013	46.3	127 × 127	Bad scan
5/23/94	DSN014	12.7	51 × 51	Low-elevation map

[a] No measurement taken.

A. Subreflector Position Correction

Appendix B summarizes the theory of subreflector position correction via holography as applied at DSS 24 (for a 70-m antenna, the processing is slightly different). The subreflector correction is derived from the low-order phase distortions in the antenna aperture function derived from low-resolution (25 × 25 array for a 34-m antenna, or 51 × 51 for a 70-m antenna) holographic imaging. Since the derivation is based on an iteration algorithm, two low-resolution measurements are required. The time required for a single low-resolution measurement is approximately 45 min, and data processing time is 16 min. Figure 2 shows the far-field amplitude pattern of DSS 24 as found in the initial stage of the holographic measurements, and Fig. 3 shows the same information after holographic corrections were applied. The corrections that were derived and applied to the subreflector positioner are 0.516 in. in the −X direction, 0.375 in. in the +Y direction and 0.135 in. in the +Z direction. From observing the far-field patterns in Figs. 2 and 3, it is clear that the antenna went through a transformation from being unfocused to focused. The performance improvement obtained by setting the subreflector is 0.25 dB at X-band and 3.6 dB at Ka-band. The

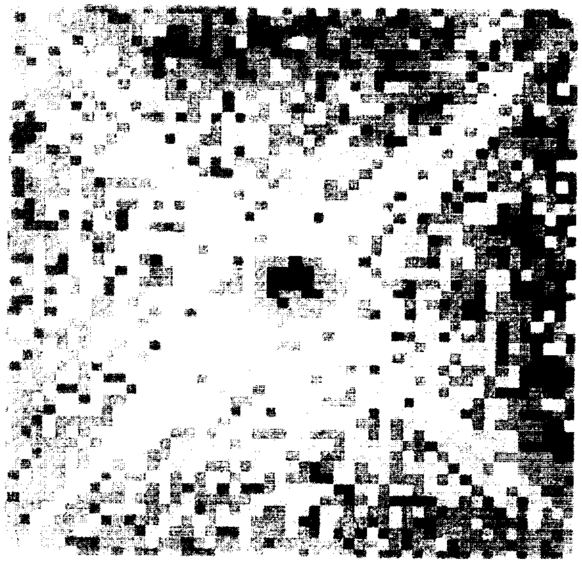


Fig. 2. Far-field pattern recorded on May 13, 1994, indicating an unfocused antenna. (Color image available electronically.)

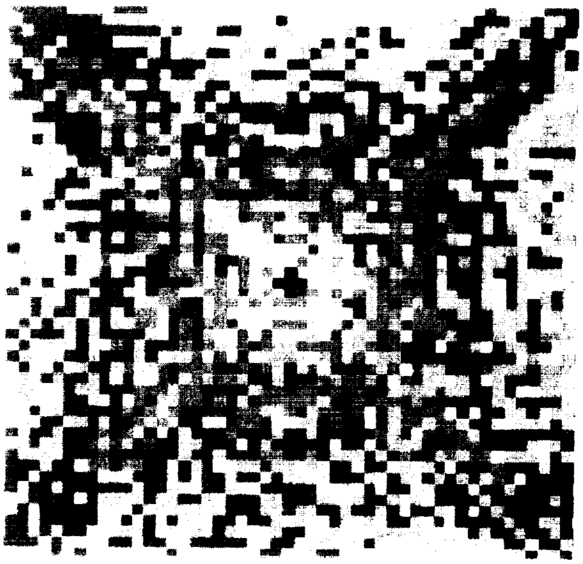


Fig. 3. Far-field pattern recorded on May 14, 1994, after correcting the subreflector position, indicating a focused antenna. (Color image available electronically.)

derivation of the subreflector correction in the X-direction was especially critical since no servo drive but only manual mechanical adjustment is available for this axis (for DSS 24), and therefore the traditional trial-and-error methods are not efficient. Figures 4 and 5 show a one-dimensional elevation cut of the far-field amplitude pattern (11.9225 GHz) before and after corrections, respectively, that were made to the subreflector. Figures 6 and 7 show a one-dimensional azimuth cut of the far-field amplitude pattern (11.9225 GHz) before and after corrections, respectively, that were made to the subreflector.

Holography can derive the subreflector (X, Y, Z) position at any observation angle from which geostationary satellites can be viewed. For the 70-m antennas, two tilt-angle corrections are also included. In practice, usually three elevation angles are readily available from Goldstone (approximately 45-, 37-, and 12-deg elevation). However, it is shown here that when the finite element model for the subreflector offset is accurate (as is the case for DSS 24), adding to it a constant term derived at a single elevation (e.g., 45 deg) creates a new model that is accurate over all elevation angles. Since the time allocated for holographic measurement was minimal, only this derivation was possible. Derivation of the subreflector offsets from the f3 focus position will compensate for any misalignment of the beam-waveguide (BWG) mirrors, and thus may cause peak antenna gain to occur at different elevation angles, and away from the rigging angle for different feed positions.

Equation (1) was derived[2] using a finite element modeling of DSS 24 for the subreflector offsets (X, Y, Z) as a function of the elevation angle (EL):

$$\left.\begin{aligned} X &= 0 \\ Y &= -0.008\{\sin(45) - \sin(\mathrm{EL})\} + (-1.485)\{\cos(45) - \cos(\mathrm{EL})\} \\ Z &= -0.164\{\sin(45) - \sin(\mathrm{EL})\} + (-0.004)\{\cos(45) - \cos(\mathrm{EL})\} \end{aligned}\right\} \tag{1}$$

[2] R. Levy, "DSS-24 Subreflector Positioner Offsets," JPL Interoffice Memorandum 3323-94-032 (internal document), Jet Propulsion Laboratory, Pasadena, California, February 16, 1994.

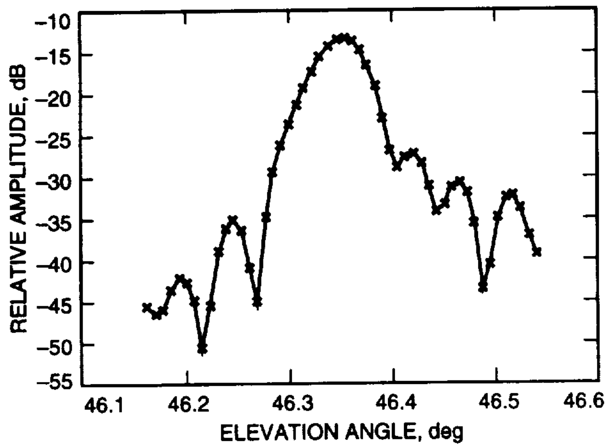


Fig. 4. Far-field elevation cut as found (dB-relative scale).

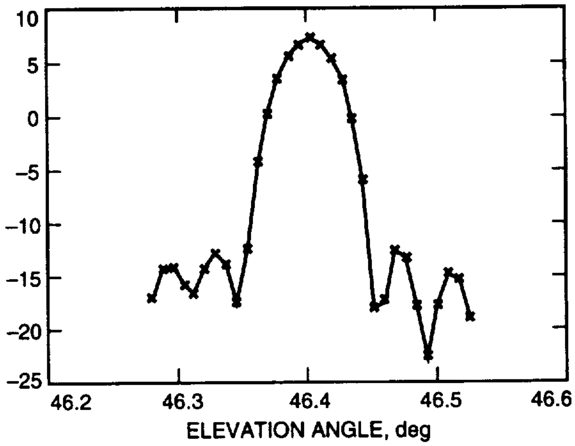


Fig. 5. Far-field elevation cut after holography (dB-relative scale).

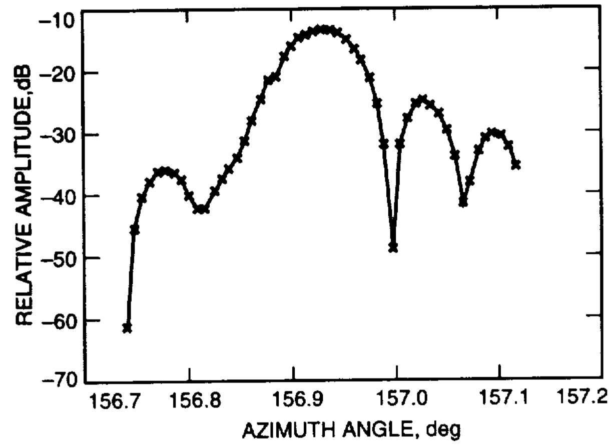


Fig. 6. Far-field azimuth cut as found (dB-relative scale).

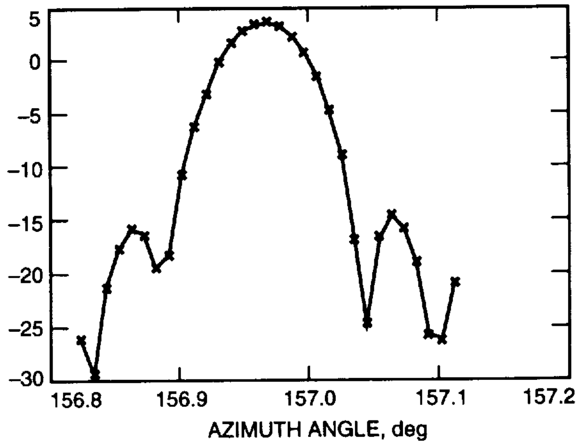


Fig. 7. Far-field azimuth cut after holography (dB-relative scale).

In Eq. (2), a constant term derived by holography at 46.3-deg elevation (and one iteration) is added to Eq. (1):

$$\left.\begin{aligned} X &= -0.516 \\ Y &= 0.375 - 0.008\{\sin(45) - \sin(\mathrm{EL})\} + (-1.485)\{\cos(45) - \cos(\mathrm{EL})\} \\ Z &= 0.135 - 0.164\{\sin(45) - \sin(\mathrm{EL})\} + (-0.004)\{\cos(45) - \cos(\mathrm{EL})\} \end{aligned}\right\} \qquad (2)$$

Holography and radiometry should derive the same subreflector offsets at approximately 45-deg elevation. (Note that holography did not optimize the subreflector position after panel setting due to time constraints imposed on the project.) Under these conditions, the maximum deviation in the equation for the Z-axis is 0.03 in. at 10-deg elevation, which translates to 0.045 dB at Ka-band. The remaining terms in the equation for the Y-axis deviate by 0.07 in. at 80-deg elevation, which translates to 0.02 dB at Ka-band (Fig. 8).

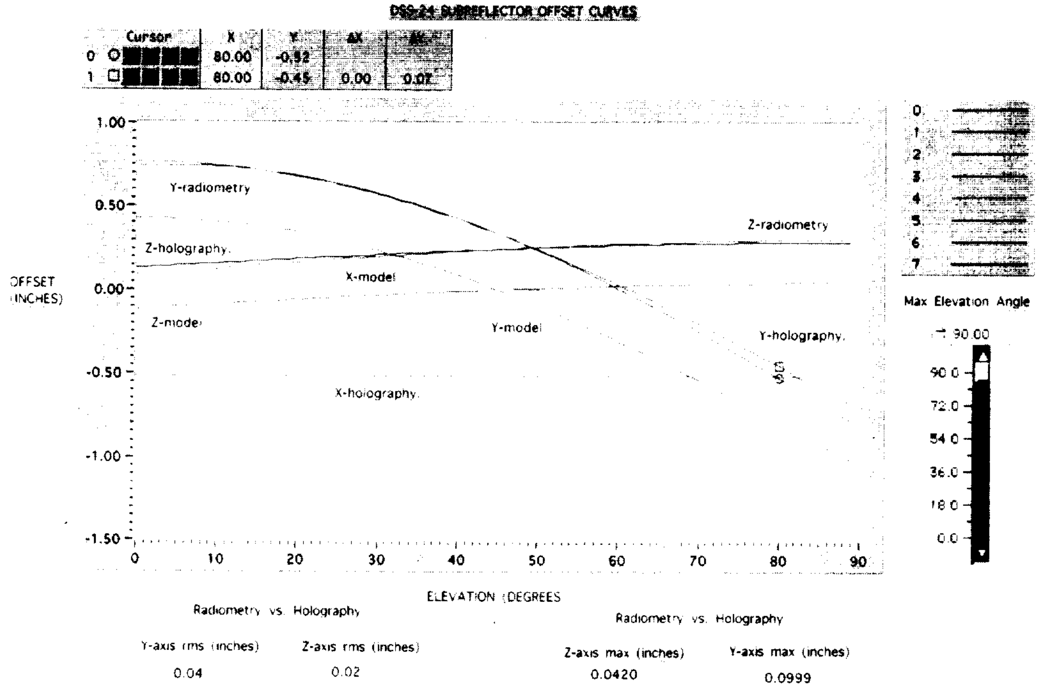


Fig. 8. Subreflector offsets versus elevation angle. (Color image available electronically.)

B. Panel Setting

The theory of panel setting as used at DSS 24 is described in [5]. Figure 9 is the mechanical surface error map of DSS 24 derived from the measurement on May 14, 1994 (DSN008). The normal rms surface error of the inner 32-m diameter at a resolution of 33.7 cm is 0.50 mm. Panel settings were derived from this scan (DSN008) after verifying repeatability (scan DSN010) and confirming coordinate geometry and pixel registering accuracy. Panels 1, 7, 13, and 19 in ring 2 (counting 1 from the center and 9 as the outermost ring) were installed last and can easily be distinguished (they are 90 deg apart). Figure 10 is the mechanical surface error map of DSS 24 derived from the measurement on May 20, 1994 (DSN012) after panel setting. The normal rms surface error of the inner 32-m diameter at a resolution of 33.7 cm is 0.258 mm, and the infinite resolution axial error is 0.25 mm. The precision of DSS 24 (diameter/rms) is 1.36 × 10^5, the highest of the NASA/JPL DSN antennas. The performance improvements achieved via holography by resetting the DSS-24 surface and positioning the subreflector are 0.35 dB at X-band and 4.9 dB at Ka-band; these improvements are summarized in Table 2. The efficiency of DSS 24 at the nominal elevation angle of 45 deg was increased from 68.83 percent to 74.61 percent at X-band (f3 referenced to horn aperture) and from 19.83 percent to 61.29 percent at Ka-band (f3 referenced to horn aperture). Figure 11 shows the gain loss of DSS 24 due to main reflector surface errors (using the Ruze equation [6]) before and after panel setting. Figure 12 is a plot of DSS-24 gain (from f3) versus frequency, indicating that its gain limit is at 95 GHz. As can be seen from Table 3,[3] the MAHST provided the critical RF performance necessary not only to meet the project requirements and goals, but to surpass them.

Figure 13 is the predicted surface error map of DSS 24 derived from the measurement on May 14, 1994 (DSN008), indicating that an rms surface error of 0.20 mm could have been achieved if the panel

[3] The "expected" values in this table were supplied from notes by W. Veruttipong, Ground Antennas and Facilities Engineering Section, and D. A. Bathker, DSN Advanced Planning Office, "DSS-24 RF Optics Design Detailed Gain/Noise Budgets for S/X Ka-Bands," Jet Propulsion Laboratory, Pasadena, California, February 7, 1992.

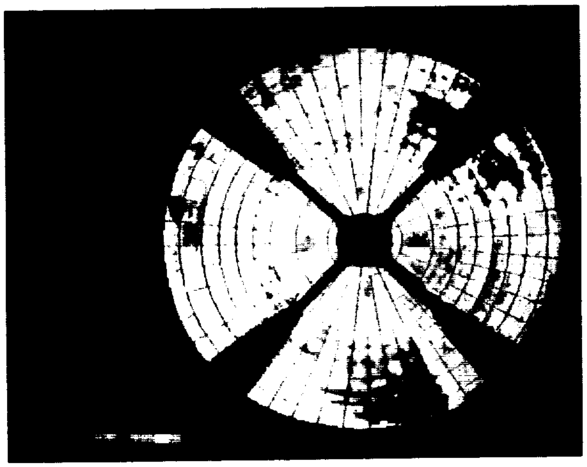


Fig. 9. High-resolution (33.7-cm) error map of the central 32 m of the antenna surface at 46.3-deg elevation, before panel setting, as derived from scan DSN008 (May 14, 1994). The normal, axial, and infinite resolution axial rms errors are 0.50, 0.44, and 0.475 mm, respectively. (Color image available electronically.)

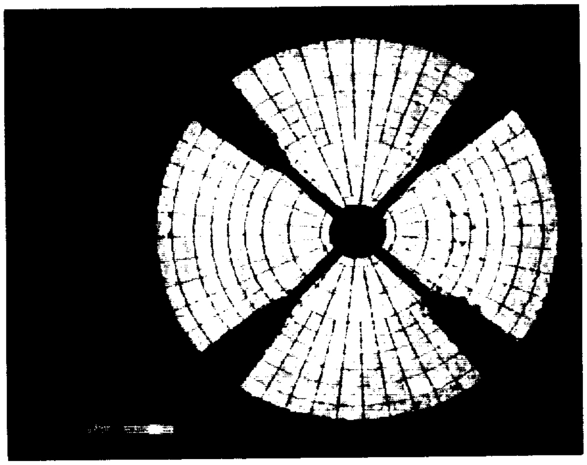


Fig. 10. High-resolution (33.7-cm) error map of the central 32 m of the antenna surface at 46.3-deg elevation, after panel setting, as derived from scan DSN012 (May 20, 1994). The normal, axial, and infinite resolution axial rms errors are 0.26, 0.23, and 0.25 mm, respectively. (Color image available electronically.)

Table 2. Performance improvement by microwave holography at approximately 45-deg elevation.

Frequency, GHz	Panel setting, dB	Subreflector, dB	Total, dB
X-band, 8.45	0.1	0.25	0.35
Ka-band, 32	1.27	3.6	4.87

setting listing were executed precisely (the accuracy of the panel setting listing is 35 μm). DSS 24 has 348 panels and 1716 adjusting screws. The rms surface of the individual panels is 0.127 mm and the rms surface error of the subreflector is 0.125 mm. Since a precision panel adjusting tool was not available, and in order to reduce the panel setting time, the panel listings were rounded to the nearest ±1/8 of a screw turn (±0.16 mm). This enabled resetting the entire dish in an 8-h period. The inferred panel setting accuracy is therefore 0.175-mm rms.

Figure 14 is a map differencing (DSN010 − DSN008) that verified repeatability and confirmed coordinate geometry and pixel registering accuracy. Before scan DSN010 was recorded, two panels were intentionally moved as targets. Panel 23 in ring 3 and panel 23 in ring 5 were translated −1.00 mm. In the map differencing of Fig. 14, the two panels appear in the correct location (within the boundaries of the panel masking) and with the correct polarity and within the expected range (the blue color next to the last in Fig. 14 corresponds to −1.07 mm). (Color images are available electronically.)

Figures 15 and 16 are the far-field amplitude and phase functions, respectively. The figures show 127 × 127 samples to the 51st sidelobe, recorded on May 20, 1994, after panel setting. The samples are separated by 34 mdeg (in the u, v space), forming a window of ±2.14 deg relative to the antenna main beam at Ku-band. The far-field amplitude (Fig. 15) shows a well-concentrated and symmetrical pattern, and the far-field phase (Fig. 16) shows a symmetrical pattern with well-concentric rings as expected. Figure 17 is the derived DSS-24 aperture amplitude function, indicating a well-uniform illuminating antenna, while the energy rolls off −15 dB just over the edge of the antenna (the last 2 m of the diameter).

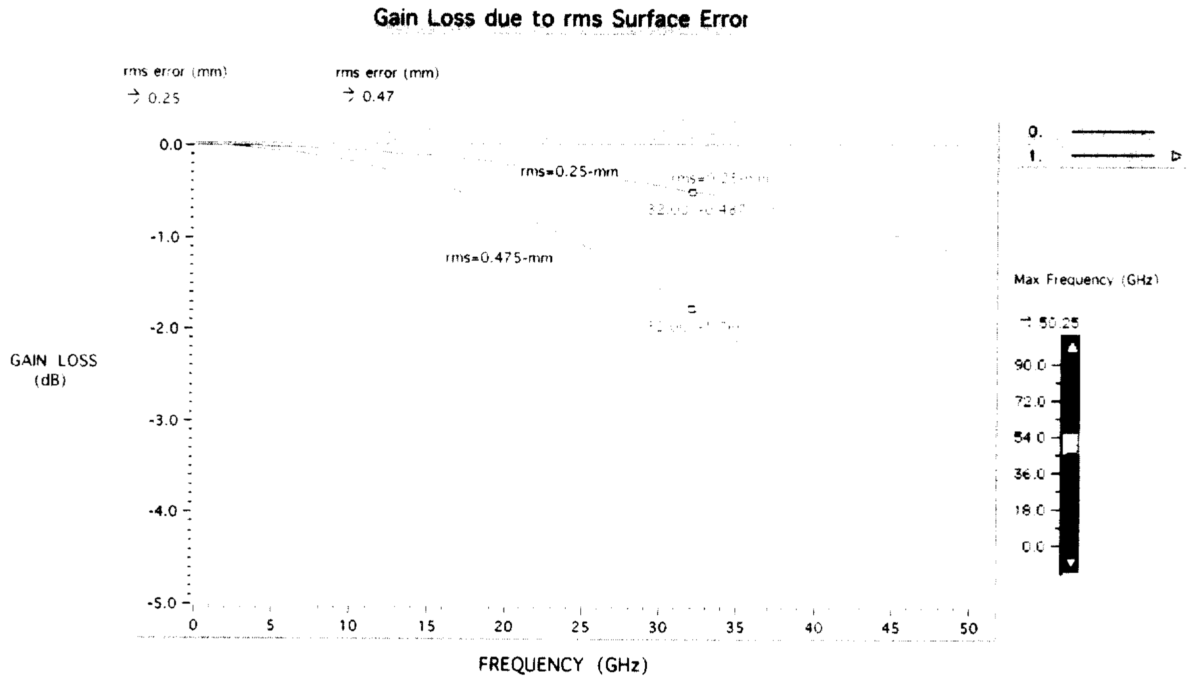


Fig. 11. Gain loss due to main reflector surface error (based on [6]). (Color image available electronically.)

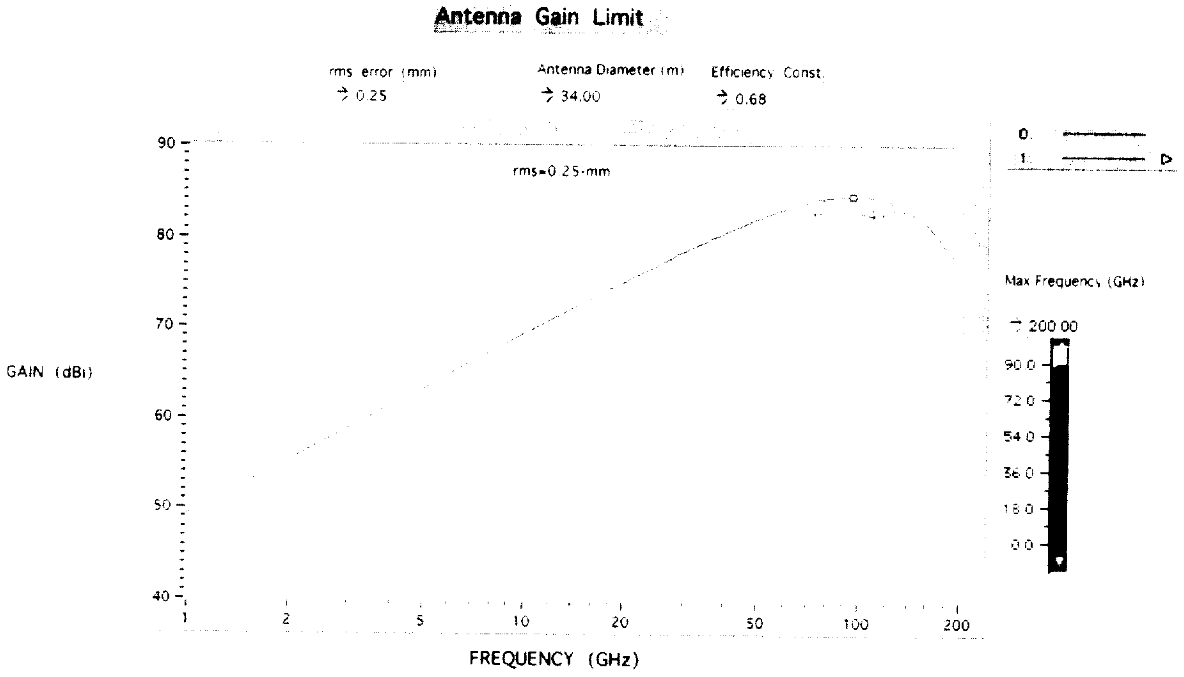


Fig. 12. Gain versus frequency. (Color image available electronically.)

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

Table 3. Maximum aperture efficiency at rigging elevation angles referenced to horn aperture.

Parameter	Percent aperture efficiency at X-band		Percent aperture efficiency at Ka-band	
	f1[a]	f3[b]	f1[c]	f1[d]
Expected[e]	78.9 ± 1.5	77.6 ± 2.5	68.2 ± 3.0	59.9 ± 4.0
Specified	—[f]	72.0	—[f]	41.0
As built	71.2 ± 3.0	68.83 ± 3.0	21.07 ± 4.0	19.83 ± 4.0
Measured post-holography	77.2 ± 2.0	74.61 ± 2.0	65.14 ± 2.3	61.29 ± 2.7

[a] 42.2 deg.
[b] 51.5 deg.
[c] 44.5 deg.
[d] 40.8 deg.
(These elevation angles were supplied by L. S. Alvarez, "Aperture Efficiency Measurements," *DSS-24 Antenna RF Performance Measurements*, JPL D-12277 (internal document), Jet Propulsion Laboratory, Pasadena, California, February 1, 1995.)
[e] W. Veruttipong and D. A. Bathker, op cit.
[f] Not specified.

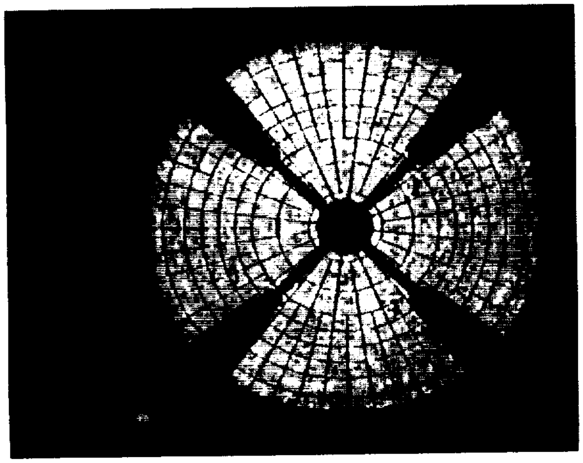


Fig. 13. Predicted surface error, map derived from DSN008. (Note: This represents the best achievable rigging angle surface that would have resulted if the 1716 screws were adjusted precisely as specified by the software.) (Color image available electronically.)

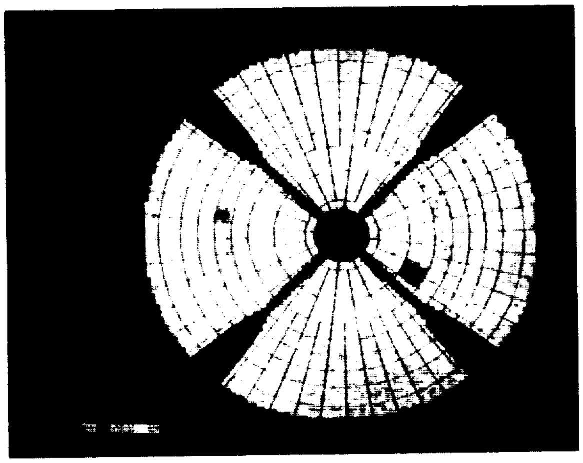


Fig. 14. Map differencing (DSN010 − DSN008) that verified repeatability and confirmed coordinate geometry and pixel registering accuracy. Before scan DSN010 was recorded, two panels were intentionally moved as targets. Panel 23 in ring 3 and panel 23 in ring 5 were translated ≅ 1.00 mm. (Color image available electronically.)

C. Gravity Deformation

Only one medium-resolution (84.8-cm) holographic measurement was recorded at a low elevation angle of 12.5 deg (Table 1). The normal rms surface error of 0.39 mm was computed at a resolution of 84.8 cm and is presented in Fig. 18. The systematic component of the antenna deformation was derived by fitting the data to a set of radial and circumferential polynomials (modified Jacobi polynomials [7], which are similar to Zernike polynomials, which are more common in optics). The first 18 terms of the modified Jacobi polynomial are tabulated in Table 4 and are shown in Fig. 19, indicating an rms surface error of 0.29 mm. A slight structural "twist" at the low elevation angle of 12.5 deg is noticed in the result. The low-order gravity deformation of DSS 24 is predominately astigmatic (80.3 percent), and its symmetrical

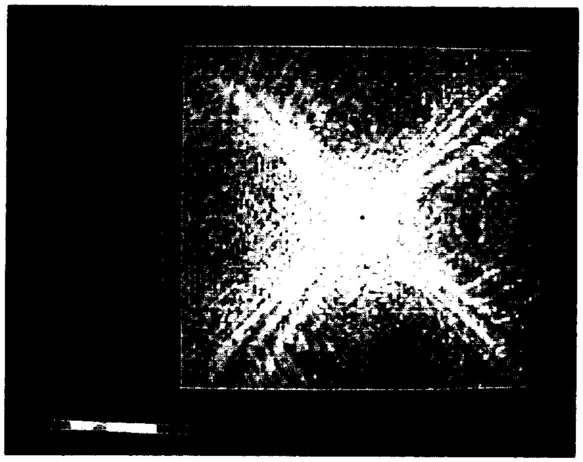


Fig. 15. Far-field (DSN012) amplitude pattern after panel setting. (Color image available electronically.)

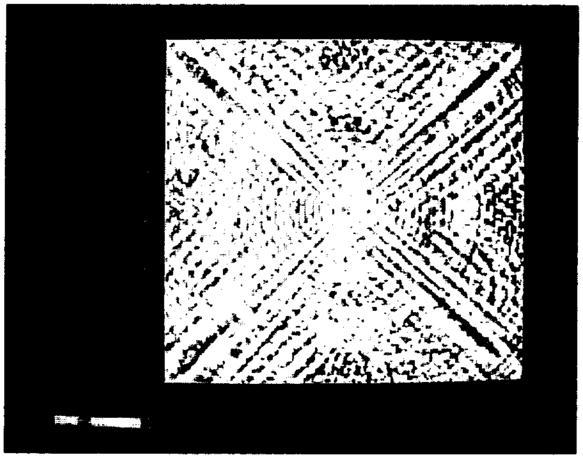


Fig. 16. Far-field (DSN012) phase pattern after panel setting. (Color image available electronically.)

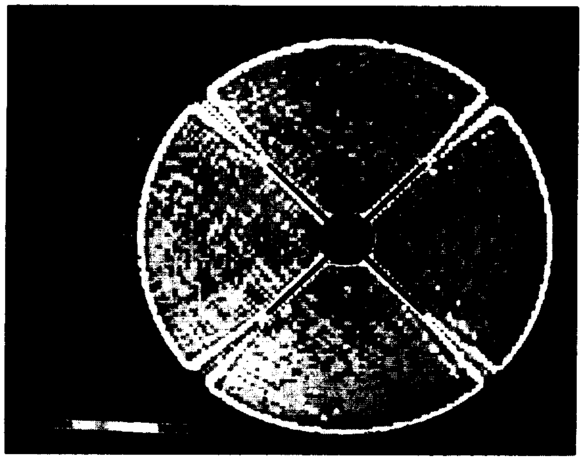


Fig. 17. Derived antenna aperture amplitude illumination (DSN008). (Color image available electronically.)

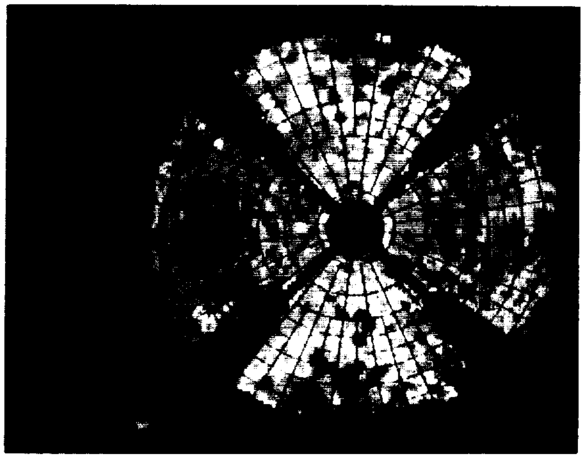


Fig. 18. Medium-resolution (84.8-cm) error map of the central 32 m of the antenna surface at 12.5-deg elevation after panel setting, derived from scan DSN014 (May 23, 1994). The normal rms error is 0.39 mm. (Color image available electronically.)

(top–down/left–right) component is shown in Fig. 20 with an rms error of 0.26 mm. Figure 21 is the map-differencing of Fig. 19 from Fig. 18, indicating that the remaining gravity distortion components of the antenna structure are of higher order or "random." The root sum squares (rss) of the systematic component and the random component agree well with the total distortion. The predicted gain loss at angles 33.8 deg away from the rigging angle is estimated at −0.046 dB at X-band and −0.65 dB at Ka-band. Efficiency measurements at X-band and Ka-band from the f3 focus indicate a gain loss of −0.042 dB and −0.575 dB at 33.8 deg from a peak gain at 51.43 deg and 40.8 deg, respectively, agreeing well with the holography predictions.

The gravity performance of DSS 24 was greatly improved relative to the gravity performance of DSS 13. It was characterized and analyzed by holography:[4] gravity distortion of DSS 13 causes 2.27-dB gain loss at 32 GHz at 33.8 deg from the rigging angle.

[4] D. J. Rochblatt and B. L. Seidel, *Holographic Measurements of DSS-13 Beam Waveguide Antenna, December 2, 1991 Through February 6, 1992*, JPL D-9910 (internal document), Jet Propulsion Laboratory, Pasadena, California, July 15, 1992.

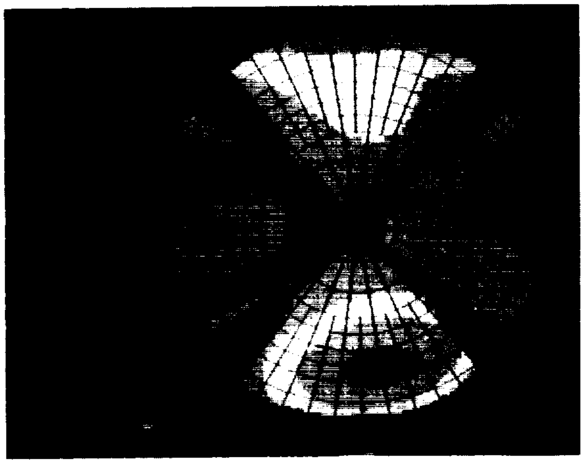


Fig. 19. Systematic component of the low-elevation error map represented by the first 18 terms of the modified Jacobi polynomials. The normal rms surface is 0.29 mm. (Color image available electronically.)

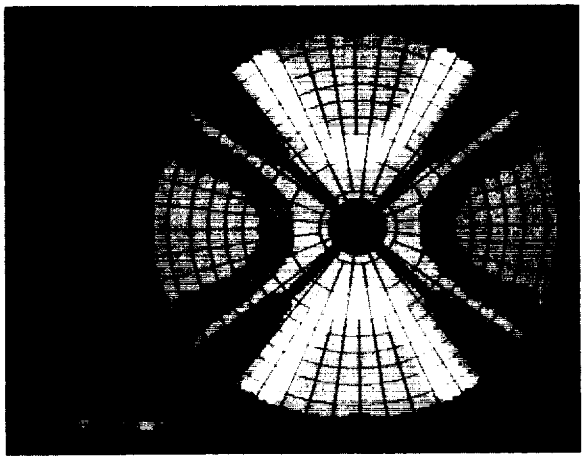


Fig. 20. Astigmatic component of the gravity distortion represents 80.3 percent of the total systematic distortion due to gravity. The normal rms surface is 0.26 mm. (Color image available electronically.)

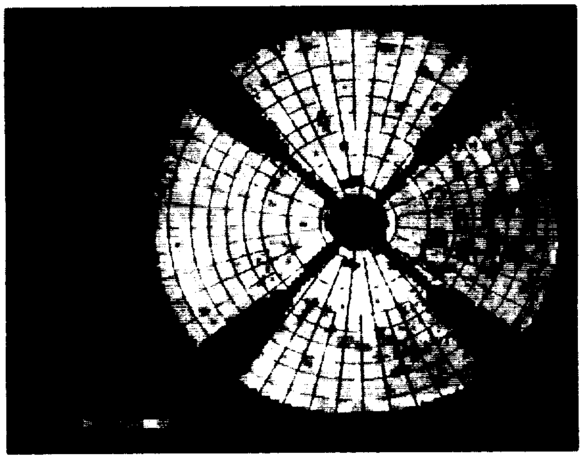


Fig. 21. Random component surface at 12.5-deg elevation. The normal rms surface error is 0.27 mm. (Color image available electronically.)

Table 4. Modified Jacobi polynomial decomposition for gravity deformation characterization.

n	m	C, in.	D, in.
0	0	−0.006389	0.000000
0	1	−0.000450	0.000000
0	2	−0.001264	0.000000
1	0	0.001766	0.00243
1	1	0.001961	−0.003240
1	2	0.000329	0.001786
2	0	−0.009939	0.000196
2	1	−0.002407	−0.001486
2	2	−0.001709	0.000767

III. Conclusions and Recommendations

The JPL MAHST provided DSS 24 with the critical RF performance necessary not only to meet the project requirements and goals, but to surpass them, transforming DSS 24 to the highest precision antenna in the DSN. The main reflector panels were set to 0.25-mm rms and the subreflector was positioned in its focus location as seen from f1 at 46.3-deg elevation. New offset curves were derived for the subreflector position at all elevation angles as seen from f1. Unfortunately, time was not allocated for holographic measurements from the f3 focus.

It is recommended that in future holographic metrology of newly built DSN BWG antennas, time for the following measurements be provided:

(1) Low resolution at Ku-band (12 GHz) from f1 rigging angle (≈45.0 deg) to set the subreflector position.

(2) High resolution at Ku-band from f1 rigging angle to set the panels.

(3) Low resolution at Ku-band from f1 at approximately 37-deg elevation to set the subreflector.

(4) Low resolution at Ku-band from f1 low elevation (≈12 deg) to set the subreflector.

(5) High resolution at Ku-band from f1 low elevation to image the surface and derive high-resolution gravity deformation maps.

(6) Medium resolution at Ku-band from the f3 rigging angle to diagnose misalignments in the BWG mirrors and characterize the BWG effects.

(7) Medium resolution at Ku-band from the f3 low elevation angle to diagnose misalignments in the BWG mirrors and their effect on performance.

(8) Medium resolution at X-band (7.7 GHz) to diagnose misalignments in the BWG mirrors and detect any problems (moding) in the feed.

Acknowledgments

The authors wish to thank Mr. Dan Bathker for his tireless advocacy of this project and his vision of the next generation of Ka-band DSN antennas. We are also grateful to Mr. Boris Seidel for his invaluable contributions to development of the JPL MAHST. Thanks are also due to Mr. Jeff Pardo from GTE-SPACENET and Mr. Ken Sweigart from COMSAT Corp. for supplying the ephemeris data for GSTAR-1 and INTELSAT-V (307) satellites, respectively, and to Mike Britcliffe, Manual Franco, and DSS-24 personnel for supporting the panel-setting activity.

References

[1] D. J. Rochblatt, "Microwave Antenna Holography System," *Proceedings of Technology 2004 Conference, Test and Measurements Part 1*, Washington, DC, November 8–10, 1994.

[2] D. J. Rochblatt, "Microwave Holography of DSN Reflector Antennas," *Proceedings of the 1994 AMTA Workshop, Modern Imaging and Diagnostic Technique for RCS and Antennas*, University of Washington, Seattle, Washington, June 24, 1994.

[3] D. J. Rochblatt, "Microwave Holography Helps Improve Performance of Large Antennas," *LASER Tech Briefs*, Publication of NASA Tech Briefs, vol. 1, no. 1, p. 81, September 1993.

[4] D. J. Rochblatt and B. L. Seidel, "Microwave Antenna Holography," *IEEE Trans. Microwave Theory and Techniques, Special Issue on Microwaves in Space*, vol. 40, no. 6, pp. 1294–1300, June 1992.

[5] D. J. Rochblatt, "A Microwave Holography Methodology for Diagnostics and Performance Improvement for Large Reflector Antennas," *The Telecommunications and Data Acquisition Progress Report 42-108, October–December 1991*, Jet Propulsion Laboratory, Pasadena, California, pp. 235–252, February 15, 1992.

[6] J. Ruze, "Antenna Tolerance Theory—A Review," *Proc. IEEE*, vol. 54, pp. 663–640, April 1966.

[7] V. Galindo-Israel and R. Mittra, "A New Series Representation for the Radiation Integral with Application to Reflector Antennas," *IEEE Trans. AP*, vol. AP-25, pp. 631–635, September 1977 (Correction, *IEEE Trans. AP*, vol. AP-26, p. 628, July 1978).

[8] D. J. Rochblatt and Y. Rahmat-Samii, "Effects of Measurement Errors on Microwave Antenna Holography," *IEEE Trans. Antennas Propagat.*, vol. 39, no. 7, pp. 933–942, July 1991.

Appendix A
Theory

The mathematical relationship between an antenna far-field pattern (T) and the antenna-induced surface current distribution (J) is given by the exact radiation integral relationship (Fig. A-1):[5]

$$\vec{T}(u,v) = \int\int_{s} \tilde{J}(x',y')\,\exp^{jkz'}\left[\exp^{-jkz'(1-\cos\theta)}\right]\exp^{jk(ux'+vy')}\,dx'dy' \qquad \text{(A-1)}$$

where $Z'(x',y')$ defines the surface $S, (u,v)$ is the direction cosine space, and θ is the observation angle. For a small angular extent of the far-field pattern, this expression reduces to

$$\vec{T}(u,v) = \int\int_{s} \tilde{J}(x',y')\,\exp^{jkz'}\,\exp^{-jk(ux'+vy')}\,dx'dy' \qquad \text{(A-2)}$$

[5] D. J. Rochblatt, op cit.

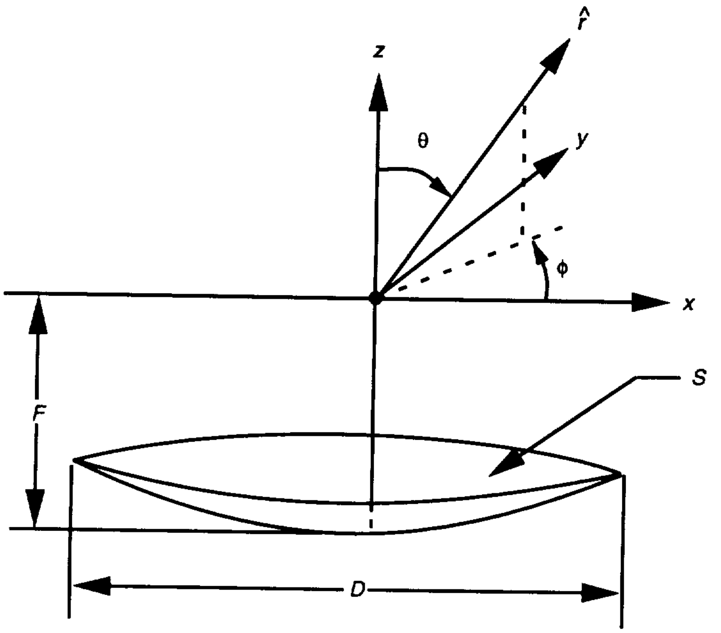


Fig. A-1. Antenna geometry.

Equation (A-2) is an exact Fourier transform of the induced surface current. To derive the residual surface error, geometrical optics ray tracing is used to relate the normal error ε to the axial error and phase in a main reflector paraboloid geometry (Fig. A-2).

$$\frac{1}{2}\Delta PL = \frac{1}{2}\left[P'P + PQ\right] = \frac{1}{2}\left[\frac{\varepsilon}{\cos\varphi} + \frac{\varepsilon\cos 2\varphi}{\cos\varphi}\right] = \varepsilon\cos\varphi \tag{A-3}$$

$$\text{Phase}(\Delta PL) = \frac{4\pi}{\lambda}\varepsilon\cos\varphi \tag{A-4}$$

and for a paraboloid,

$$\cos\varphi = \frac{1}{\sqrt{1 + \dfrac{x^2+y^2}{4F^2}}} \tag{A-5}$$

where F is the antenna focal length.

Allowing for the removal of a constant phase term and substituting Eq. (A-4) into Eq. (A-2),

$$\vec{T}(u,v) = \exp^{-j2kF}\int\!\!\int_{s}\left[\left|\tilde{J}(x',y')\right|\exp^{j4\pi\frac{\epsilon}{\lambda}\cos\varphi}\right]\exp^{jk(ux'+vy')}\,dx'dy' \tag{A-6}$$

For processing sampled data, the associated discrete Fourier transform (DFT) is utilized:

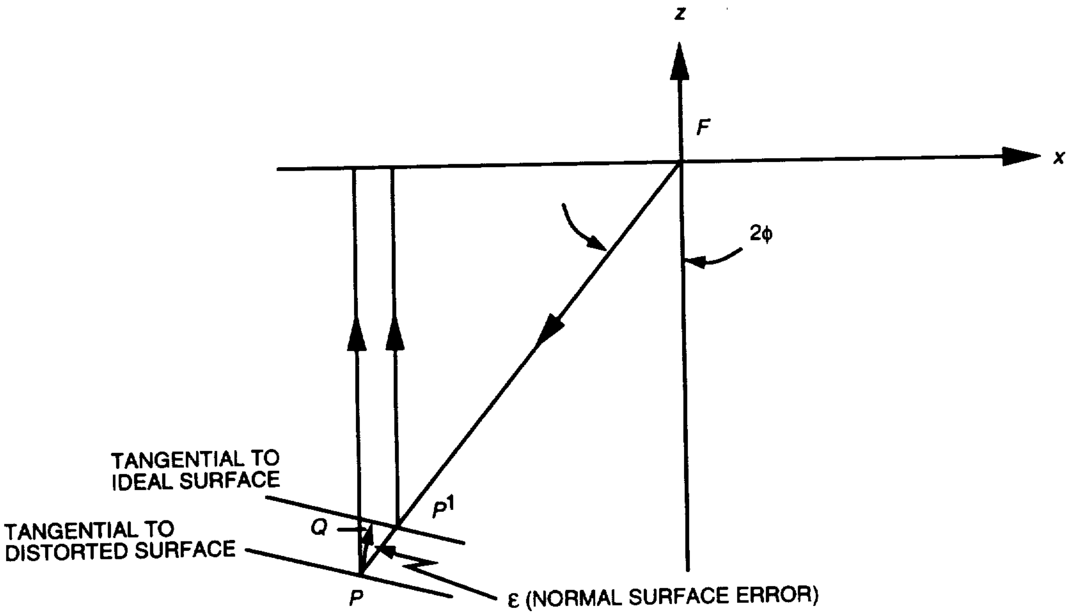


Fig. A-2. Surface distortion geometry.

$$T(p\Delta u,\ q\Delta v) = sxsy \sum_{n=-N1/2}^{N1/2-1} \sum_{m=-N2/2}^{N2/2-1} J(nsx, msy)\exp^{j2\pi((np/N1)+(mq/N2))} \tag{A-7}$$

where $N1 \times N2$ is the measured data array size; sx and sy are the sampling intervals in the aperture coordinates; n, m, p, and q are the integers indexing the discrete samples; and Δu and Δv are the sampling intervals in u, v far-field space. Since the magnitude of the far-field pattern is essentially bounded, the fast Fourier transform (FFT) is usually used for computation. The solution for the antenna residual surface error in the normal direction is, therefore,

$$\varepsilon(x,y) = \frac{\lambda}{4\pi}\sqrt{1+\frac{x^2+y^2}{4F^2}}\text{Phase}\left[\exp^{j2kF}(FFT)^{-1}[T(u,v)]\right] \tag{A-8}$$

The spatial resolution in the final holographic map δ is defined here at the −3-dB width of the convolving function [4]:

$$\delta = \frac{D}{kN} \tag{A-9}$$

where D is the main reflector diameter, N is the square root of the total number of data points, and k is the sampling factor, usually $0.5 < k < 1.0$. The lateral resolution is inversely proportional to the number of sidelobes measured. For a 34-m-diameter antenna, for example, a resolution of 0.337 m in the final holographic map can be achieved with a data array size of 127 × 127 (16,129) and sampling factor of 0.794. For a 34-m antenna constructed of 348 panels, this measurement will enable imaging of each panel by 33 resolution cells. In Figs. 15 and 16, the far-field amplitude and phase are measured on rectangular coordinates of 127 × 127 with sampling intervals of 34.0 mdeg (the sampling factor is 0.80).

C-4.

Figures 17 and 10 show the aperture amplitude and surface error function, respectively, with a lateral resolution of 0.337 m.

The accuracy in each resolution cell of the final holographic map is [8]

$$\sigma = 0.082 \frac{\lambda D}{\delta SNR} \tag{A-10}$$

where λ is the wavelength, SNR is the beam peak voltage signal-to-noise ratio, and σ is the standard deviation (accuracy) in recovering the mean position of a resolution cell. The accuracy across holographic maps varies with the antenna aperture amplitude taper illumination. Results are better at the center of the dish and gradually become worse toward the edge of the dish. For a uniformly illuminated dish, accuracy stays relatively constant through most of the dish and becomes quickly worse just at the edge where the illumination falls off rapidly. Note in Eq. (A-10) that the accuracy is inversely proportional to the spatial resolution of Eq. (A-9) due to the larger averaging area available at the larger resolution cell. For a holographic measurement receiver incorporating a multiplier integrator or a divider integrator (for example, HP8530A), the effective signal-to-noise ratio SNR_e can be expressed as [8]

$$SNR_e \left[\sqrt{\frac{1}{SNR_t^2} + \frac{1}{SNR_r^2} + \frac{1}{SNR_t^2 SNR_r^2}} \right]^{-1} \tag{A-11}$$

where SNR_t is the test channel SNR and SNR_r is the reference channel SNR.

Phase errors introduced during the measurement due to pointing and subreflector position errors are removed via a best-fit paraboloid program. The best-fit paraboloid is found by least-squares fitting the data (residual surface error function), allowing 6 degrees of freedom in the model [3].[6] This algorithm ensures that the minimum adjustment (distance) is computed for the screw adjusters. The least-squares fit is computed by minimizing S, the sum of the squares of the residual path-length changes:

$$S = \sum_{i=1}^{N^2} \Gamma(\Delta PL_i)^2 A_i \tag{A-12}$$

where A_i is the amplitude weighing factor associated with the ith data point, Γ is the masking operation that is antenna-type dependent, and ΔPL_i is the path-length change at point (x_i, y_i, z_i). It is correct to apply the best-fit paraboloid algorithm to either the conventional Cassegrain paraboloid-hyperboloid or dual-shaped reflector systems, even though the latter does not use a paraboloid as the main reflector. Both of the reflector antenna designs are, overall, plane-wave-to-point source transformers, differing only in their intensity field distribution.

The resultant aperture function at the end of this process is defined here as the effective map,[7] since it includes all phase effects that are contributing to the antenna performance. These frequency-dependent effects include the subreflector scattered feed phase function and strut diffraction effects. Removal of the frequency-dependent effects results in a mechanical map.[8] By deriving panel adjustments based on the effective map, the surface shape will conjugate the phase errors, optimizing the performance of the antenna at a single frequency while degrading the performance of the antenna at all other frequencies.

[6] Ibid.

[7] Ibid.

[8] Ibid.

For antennas operating at a single frequency, this procedure is advantageous. However, many antennas operate at several different frequencies and require a wide bandwidth performance response. For these antennas, the mechanical map must be used to derive panel-setting information.

From the mechanical map, surface tolerance efficiency can be computed at frequencies other than the measured frequency by scaling the residual aperture phase errors (which are now due only to surface deviations) to other frequencies [5]:

$$(K)_{surface} = 20 \times \log_{10}$$

$$\times \frac{\sqrt{\left[\sum_{i=1}^{N^2} 10^{ampdb_i/20} \cos\left(\phi_{m_i}\left(\frac{\lambda_m}{\lambda_k}\right)\right)^2\right] + \left[\sum_{i=1}^{N^2} 10^{ampdb_i/20} \sin\left(\phi_{m_i}\left(\frac{\lambda_m}{\lambda_k}\right)\right)\right]^2}}{\sum_{i=1}^{N^2} 10^{ampdb_i/20}} \tag{A-13}$$

In this computation, it is assumed that the aperture amplitude illumination is frequency independent. The error introduced in this assumption is thus negligible.

To simplify the discussion on panel settings, the normal component of the residual surface error (E_n) is comprised of two parts in this model. One is due to panel misalignment or rigid body motion, and the second is due to surface error resulting from panel bending:[9]

$$E_n = E_b + E_p \tag{A-14}$$

where E_n is the total surface normal error, E_b is the normal error due to panel bending, and E_p is the normal error due to panel misalignment.

To improve the antenna surface error due to panel misalignment, panels are allowed to move as rigid bodies, with 3 degrees of freedom. The panel position correction is computed by least-squares fit. The derived motion of the panel is then used to compute the needed adjustment at the exact location of each screw on the panel. Only the pixels (resolution-cell data) projected on the panel are considered in the computation, with the center of the pixel taken as the criterion of its location. This criterion provides some averaging near the panel edges, flaring it somewhat with its neighbors. In the panel rigid motion algorithms, 3 degrees of freedom are allowed: a translation (Eq. (A-5)) at a reference point and two rotations (tilts) about the radial and circumferential axis (α and β). Screw adjustments at point qi are computed via

$$E_{p_{q_i}} = -(S + d_i \times \tan(\alpha) - (e_i/\cos(\gamma)) \times \tan(\beta)) \tag{A-15}$$

where

$$\gamma = \arctan\left(\frac{R_{Q_k}}{2}F\right) \tag{A-16}$$

and F is the focal length of the best-fit paraboloid and R_{Q_k} is the radial distance from dish center to panel coordinate center.

[9] Ibid.

Appendix B
Subreflector Position Correction Via Holography

Subreflector position correction is derived from the low-order phase distortion in the antenna aperture function. The antenna aperture function in holography is derived from the measured far-field complex (amplitude and phase) function. Zernike or modified Jacobi polynomial and global parameter fit can all be applied.[10] The global best-fit paraboloid is found by permitting 6 degrees of freedom in the model: three vertex translations (X_0, Y_0, Z_0), two rotations (α, β), and a focal length (F) change (K).

The least-squares fit problem is solved by minimizing the sum squares of the residual path-length error:

$$S = \sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} (\Delta PL_i)^2 A_i \tag{B-1}$$

where $\Gamma_{(DSS\ 24)}$ is the masking operator for DSS 24, ΔPL_i is the path-length change, and A_i is the amplitude weighing. The minimum for S is found by solving the six partial differential equations simultaneously:

$$\left.\begin{aligned}
\frac{\partial S}{\partial X_0} &= 2\sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} \frac{\partial \Delta PL_i}{\partial X_0} \Delta PL_i A_i = 0 \\
\frac{\partial S}{\partial Y_0} &= 2\sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} \frac{\partial \Delta PL_i}{\partial Y_0} \Delta PL_i A_i = 0 \\
\frac{\partial S}{\partial Z_0} &= 2\sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} \frac{\partial \Delta PL_i}{\partial Z_0} \Delta PL_i A_i = 0 \\
\frac{\partial S}{\partial \alpha} &= 2\sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} \frac{\partial \Delta PL_i}{\partial \alpha} \Delta PL_i A_i = 0 \\
\frac{\partial S}{\partial \beta} &= 2\sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} \frac{\partial \Delta PL_i}{\partial \beta} \Delta PL_i A_i = 0 \\
\frac{\partial S}{\partial K} &= 2\sum_{i=1}^{N^2} \Gamma_{(DSS\ 24)} \frac{\partial \Delta PL_i}{\partial K} \Delta PL_i A_i = 0 \\
K &= \frac{1}{4}\left(\frac{1}{F} - \frac{1}{F'}\right)
\end{aligned}\right\} \tag{B-2}$$

[10] Ibid.

After removing systematic pointing errors, the parameters are used to compute the subreflector position error:

$$\left.\begin{aligned}
\Delta X &= X_0 - F\sin(\beta) \\
\Delta Y &= Y_0 - F\sin(\alpha) \\
\Delta Z &= [Z_0 + F\{\cos(\alpha) + \cos(\beta)\} - 2F]
\end{aligned}\right\} \qquad \text{(B-3)}$$

TDA Progress Report 42-121

May 15, 1995

Digital Signal Processing in the Radio Science Stability Analyzer

C. A. Greenhall
Communications Systems Research Section

The Telecommunications Division has built a stability analyzer for testing Deep Space Network installations during flight radio science experiments. The low-frequency part of the analyzer operates by digitizing sine wave signals with bandwidths between 80 Hz and 45 kHz. Processed outputs include spectra of signal, phase, amplitude, and differential phase; time series of the same quantities; and Allan deviation of phase and differential phase. This article documents the digital signal-processing methods programmed into the analyzer.

I. Introduction

The recently developed radio science stability analyzer (RSA) is an instrument for real-time testing and certification of Deep Space Network (DSN) equipment to be used during gravity wave and planetary occultation experiments [1]. Two sets of equipment can be tested: (1) the radio science open-loop receiver and (2) the 100-MHz frequency standards and distribution network of the DSN frequency and timing system (FTS). Signals from either of these two sources are downconverted to low-frequency band-limited sine wave signals. The last stage of the open-loop receiver, called radio science intermediate frequency to video (RIV), produces sine wave signals with frequencies and bandwidths ranging from 150 Hz in an 82-Hz band to 275 kHz in a 45-kHz band; these depend on the choice of RIV filter. RIV signals are processed directly by the low-frequency RSA circuitry. Pairs of 100-MHz FTS signals are processed in a portion of the RSA called the 100-MHz interface assembly (100 MHz IA), which resides near the frequency standards. The 100 MHz IA mixes the two signals at 10 GHz and downconverts the mixer output to a 100-kHz sine wave signal in a 30-kHz bandwidth, which is sent over a fiber-optic cable to the low-frequency RSA circuitry.

The low-frequency circuitry has two methods for converting a band-limited sine wave signal to digital information. First, the signal can be sampled with a 16-bit analog-to-digital (A–D) converter clocked by a synthesizer. In this mode, two signal channels can be accommodated with the aim of extracting their differential phase. The maximum total data rate is about 230 kilosamples per second. Second, if the carrier frequency is known within approximately 0.1 Hz, it can be mixed with the output of another synthesizer set to this frequency minus 1 Hz. The 1-Hz mixer output is filtered and hard limited by a zero-crossing detector, and the up-crossing times of the resulting sequence of pulses are captured by a time-interval counter according to the "picket fence" method [4].

The principal aim of processing the A–D data is to reduce their bandwidth by a user-selected factor, and to extract the amplitude and phase modulations that constitute the sidebands of the sine wave signal.

The phase of two channels can be combined into differential phase. Three output types can be generated: spectrum of the signal and its modulations, time series of the modulations, and Allan deviation of phase. As described below, the digital signal processing operates in three alternate modes, called full band, medium band, and narrow band. The choice among these depends on the desired bandwidth reduction factor. The 1-Hz zero-crossing data are processed in the same way as sequences of phase residuals produced by narrow-band processing.

The digital signal processing (DSP) methods are designed to take advantage of the architecture of a floating-point vector processor based on the 40-MHz Intel I860. Most of the heavy lifting is done by manufacturer-supplied vector library routines, which include fast Fourier transform (FFT) and finite impulse response (FIR) filtering routines. Throughputs of approximately 25 million floating-point operations per second were achieved.

The remainder of this article explains the DSP methods in some detail.

II. Signal Properties

A. Radio Frequencies

In any test setup, there are two radio frequencies of interest. Let $f_{\rm mix}$ be the frequency at which the primary comparative mixing takes place, and let $f_{\rm ref}$ be the reference frequency for phase noise and Allan deviation. For a RIV test, $f_{\rm mix} = f_{\rm ref} = 2295$ MHz (S-band) or 8415 MHz (X-band). For an FTS test, $f_{\rm mix} = 9.9$ GHz, $f_{\rm ref} = 100$ MHz. This is because the phase of the 100-kHz output of the 100 MHz IA is approximately 99 times the difference between the phases of the two 100-MHz inputs. Phase results are scaled by $f_{\rm ref}/f_{\rm mix}$.

B. Analog Sine Wave Signal

The downconverted signal is assumed to lie in an analog frequency band with the center at $f_{\rm ofst}$ and width $W_{\rm vid} < f_{\rm ofst}$, which are parameters of the RIV filter or the 100-MHz IA. The frequency $f_{\rm ofst}$ can be positive or negative; see the discussion of polarity below. Somewhere in this band is the carrier. Except in full-band processing, it is assumed that the signal consists of a carrier with weak sidebands; the total carrier-to-noise ratio should be at least about 30 dB. (This instrument is a stability analyzer, not a receiver.)

C. Digitized Sine Wave Signal

The analog signal is sampled by a 16-bit A–D converter at the sample rate f_s, which has to be chosen so that the analog frequency band is aliased into the Nyquist band $(0, f_s/2)$ or $(-f_s/2, 0)$. In this way, both sidebands of the carrier are preserved. Each RIV filter is designed for a certain f_s. In any case, an acceptable f_s can be obtained from the formulas

$$m = \mathrm{int}\left(\frac{|f_{\rm ofst}|}{W_{\rm vid}} - 0.5\right), \qquad f_s = \frac{4\,|f_{\rm ofst}|}{2m+1}$$

where int (x) is the integer part of x. This choice of f_s centers the aliased signal band in the Nyquist band. If the actual carrier frequency is close to $f_{\rm ofst}$, however, then distortion in the analog signal or A–D converter may cause spurious harmonics to appear near the carrier. To push the images of the lowest harmonics away from the carrier, one can offset the sample rate slightly, according to the formulas

$$a = 0.944272, \qquad m = \mathrm{int}\left(a\left(\frac{|f_{\rm ofst}|}{W_{\rm vid}} - 0.5\right)\right), \qquad f_s = \frac{4\,|f_{\rm ofst}|}{2m+a}$$

The number a is related to the golden ratio $\left(\sqrt{5}-1\right)/2$.

D. Polarity

In the radio science receiver, the 2.3-GHz or 8.4-GHz signal is downconverted and filtered three times until the carrier is at 10 MHz $+f_{\text{ofst}}$, where f_{ofst} can be positive or negative. At this point, the spectrum or phase polarity of the signal is positive, i.e., the same as the radio frequency (RF) signal. The fourth downconversion by the 10-MHz local oscillator and subsequent filtering, therefore, yield a signal whose polarity equals the sign of f_{ofst}. Moreover, the sampling can flip the polarity again. To make better sense of this, it is good to think about the two-sided representation of the signal. One side of the signal has the right polarity (positive), and the other side has the wrong polarity. If we let

$$n_{\text{base}} = \text{nint}\left(\frac{f_{\text{ofst}}}{f_s}\right), \qquad s_{\text{pol}} = \text{sign}\left(f_{\text{ofst}} - n_{\text{base}} f_s\right)$$

where nint (x) is the nearest integer to x, then s_{pol} is the polarity of the digitized signal, the side of the analog signal with the right polarity lies between $n_{\text{base}} f_s$ and $\left(n_{\text{base}} + s_{\text{pol}}/2\right) f_s$, and the side of the digitized signal with the right polarity lies between 0 and $s_{\text{pol}} f_s/2$. The user has the responsibility of entering f_{ofst} with the correct sign.

III. Full-Band Processing

This mode allows the user to see a snapshot of the signal in the time and frequency domains before proceeding to a closer view. The user selects an FFT size N (2048 or 4096). A frame of A–D data $x[0], \cdots, x[N-1]$ is collected. These can be plotted against elapsed time in the frame, after scaling them back to volts at the A–D input (10 V = 32,768). A spectral estimate of the frame is computed by scaling the frame so that $\sum x[n]^2 = 1$ and calculating

$$S_x[k] = \frac{2}{f_s N} \left| \sum_{n=0}^{N-1} x[n] u_0[n; N, 5] \exp\left(-i2\pi nk/N\right) \right|^2, \qquad k = 0, \cdots, N/2 \tag{1}$$

where $u_0[n; N, 5]$ is the 0th-order, N-point "trig prolate" data taper [5] with bandwidth parameter $w = 5$ (Appendix B), scaled so that $\sum u_0[n]^2 = N$. The sidelobes of this taper (Ω_{05} in Fig. 1) are low enough so that no leakage from the carrier should be visible in the sidebands. The array $10 \log_{10} S_x[k]$ (labeled dBc/Hz) is plotted against the frequency array

$$f[k] = f_s \left(n_{\text{base}} + s_{\text{pol}} k/N\right), \qquad k = 0, \cdots, N/2$$

which shows the side of the signal with the correct polarity. The user chooses how many of these frame spectra are averaged into a run spectrum. The frames do not have to be adjacent; it is all right to lose data while processing the previous frame.

The resolution bandwidth of the spectral estimate, given by

$$W_{\text{nb}} = \frac{f_s N}{\left(\sum u_0[n]\right)^2}$$

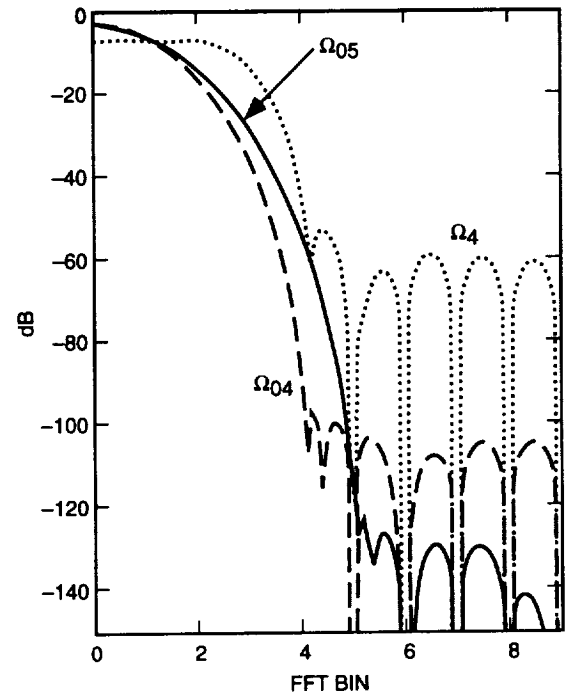


Fig. 1. Spectral windows: full band Ω_{05}, medium band Ω_{04}, and narrow band Ω_4.

has two purposes: (1) It gives the user a rough idea of the resolution of the spectral plot, and (2) it allows the user to estimate the power of a bright line (narrower than W_{nb}) in dBc by adding $10\log_{10}W_{\text{nb}}$ to the dBc/Hz reading at the peak of the line.

Because the main purpose of this function is a check on what sort of signal is actually in the Nyquist band, it might be preferable to scale the spectrum to dBm/Hz or dBV^2/Hz instead of scaling the frame to power 1 and claiming that we are seeing dBc/Hz. Then, for example, if no signal were present, the display would show the correct spectral density level of the noise.

IV. Medium-Band Processing

In this mode of processing, we assume that the sampled signal consists of a carrier with weak sidebands. The purpose of the processing is to reduce the bandwidth of the signal by a modest amount (up to 128 with current parameters), remove the carrier, and measure properties of the sidebands.

A. z-Frame Production

The user having selected an FFT size N_{fft} and a decimation factor r, both powers of 2, define the frame size $N_{\text{xf}} = rN_{\text{fft}}$. In order to limit memory usage, the frame is divided into n_{bf} adjacent batches of size N_{xb}, a divisor of N_{xf} that is not more than some maximum batch size (currently 8192). One batch at a time is processed. We use the first batch to measure the carrier frequency by a simple vector computation called "Pony, Part 1" (Appendix A). Let $\hat{o}$ be the measured frequency in radians per sample, the sign of $\hat{o}$ being s_{pol}, and let $u = \exp(-i\hat{o})$. Let $x[n]$, $n = 0, \cdots, N_{\text{xf}} - 1$, be the A–D x-frame. A complex z-frame $z[m]$ of size $N_{\text{zf}} < N_{\text{fft}}$ is computed by

$$z_1[n] = x[n]u^{-n}, \qquad n = 0, \cdots, N_{\text{xf}} - 1 \tag{2}$$

$$z[n] = \sum_{k=0}^{n_h-1} h_r[k] z_1[rn+k], \qquad n = 0, \cdots, N_{zf} - 1 \tag{3}$$

where h_r is a lowpass FIR filter designed for decimation by r (Appendix B). Its length n_h is assumed to be a multiple of r (currently $16r$), and it follows that we can take $N_{zf} = N_{fft} - n_h/r + 1$. The ripples of the frequency response of h_r above the decimated Nyquist frequency (Fig. 2) are low enough so that the aliased image of the wrong side of the carrier at $-\hat{o}$ barely appears above the 16-bit quantization noise in a spectrum output with simulated data.

The computation in Eqs. (2) and (3) is carried out batch by batch, the z-frame being built up in n_{bf} steps by an overlap-add operation. The result is a complex representation of the carrier (at zero frequency now) and sidebands within $f_s/(2r)$ of the carrier. Because frames are processed independently, it is all right to lose A–D data between frames while carrying out further processing on completed z-frames.

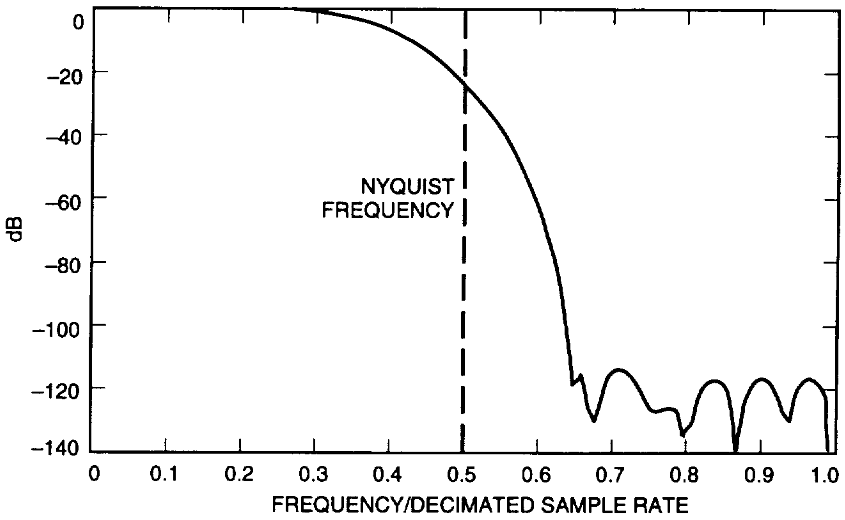


Fig. 2. Frequency response of the FIR filter for lowpass decimation.

B. Signal Spectrum

The signal spectrum is obtained as a two-sided spectrum of the z-frame. First, the z-frame is scaled to unit energy. Most of the energy is in the carrier, which is now at dc (zero frequency). To prevent this dc energy from leaking into the rest of the spectrum, we get rid of most of it by removing a linear fit from the frame. We call this kind of preconditioning operation a calibration. The specific example used here can be defined on a general array $y[0], \cdots, y[N-1]$ as follows: Let M be an integer approximately equal to $N/6$. Compute the centroid points

$$(t_0, y_0) = \frac{1}{M} \sum_{n=0}^{M-1} (n, y[n]), \qquad (t_1, y_1) = \frac{1}{M} \sum_{n=N-M}^{N-1} (n, y[n])$$

and pass a straight line $c_0 + c_1 n$ through them. The calibrated array is given by $y_0[n] = y[n] - c_0 - c_1 n$. If y itself is a straight line, then $y_0 = 0$.

The choice of this particular operation (especially the $N/6$) for spectral preconditioning is admittedly seat-of-the-pants engineering. Perhaps removing a conventional least-squares fit would do as well. To deal with time series modeled by processes that are possibly nonstationary but do have stationary first or second increments, it is desirable to subtract *some* linear fit, not just the mean. This makes all frames statistically identical, so that the average of the spectral estimates of J disjoint frames converges as $J \rightarrow \infty$, just as in the theory of stationary-process spectral estimates.

The spectrum and frequency arrays are now given in terms of the calibrated array z_0 by

$$\begin{aligned} S_z[k] &= \frac{r}{f_s N_{\mathrm{zf}} \left| H_r(f[k]) \right|^2} \left| \sum_{n=0}^{N_{\mathrm{zf}}-1} z_0[n] u_0[n; N_{\mathrm{zf}}, 4] \exp\left(\frac{-i2\pi nk}{N_{\mathrm{fft}}} \right) \right|^2 \\ f[k] &= \frac{f_s}{r N_{\mathrm{fft}}} k \end{aligned}$$

where $k = -N_{\mathrm{fft}}/2 + 1, \cdots, N_{\mathrm{fft}}/2$. The squared magnitude of

$$H_r(f) = \sum_{n=0}^{n_h-1} h_r[n] \exp\left(\frac{-i2\pi nf}{f_s} \right)$$

is used for equalizing the spectrum against the lowpass decimation filter. As before, a plot of $10 \log_{10} S_z[k]$ is labeled dBc/Hz. Points corresponding to frequencies with absolute value below $4f_s/(N_{\mathrm{zf}} r)$ or above 95 percent of the Nyquist frequency $0.5 f_s/r$ are not displayed. The low cutoff hides doubtful values near dc; the high cutoff hides a 3-dB rise at the Nyquist frequency caused by the combination of lowpass decimation, noise folding at the Nyquist frequency, and equalization. The user chooses how many of these frame spectra are averaged into a run spectrum. The resolution bandwidth is given by

$$W_{\mathrm{nb}} = \frac{f_s N_{\mathrm{zf}}}{r \left(\sum u_0[n] \right)^2} \tag{4}$$

C. Amplitude and Phase

Extraction of amplitude and phase residuals starts with a rectangular-to-polar operation on the z-frame. The result is a complex "amplitude-phase" frame ap$[n]$, $n = 0, \cdots, N_{\mathrm{zf}} - 1$, whose real part is the amplitude of $z[n]$ and whose imaginary part $\theta[n]$ is the phase of $z[n]$ wrapped into $[-\pi, \pi]$. The amplitudes are replaced by their fractional deviations from the mean. The phases are unwrapped into phase deviations $\phi[n]$ (replacing $\theta[n]$ in the ap array) by the following algorithm:

$$\phi[0] = 0, \qquad \phi[n] = \phi[n-1] + \text{ mods } (\theta[n] - \theta[n-1], 2\pi), \qquad n = 1, \cdots, N_{\mathrm{zf}} - 1$$

The symmetric residue function mods is defined by

$$\mathrm{mods}(x, a) = x - a \text{ nint } (x/a) \tag{5}$$

The correctness of this algorithm requires only that $|\Delta\phi[n]| < \pi$. The mods function also plays the central role in the unwrapping algorithm described in Appendix C.

The amplitude or phase residuals can be displayed as time series for the frame. Often, the phase residuals are dominated by a ramp (a frequency offset), so that it is desirable to subtract a linear fit to reveal the random fluctuations. This can be done with the calibration operation described above in connection with spectral preconditioning.

For better or worse, amplitude and phase spectra are computed together by a single complex FFT instead of two real FFTs. The real and imaginary parts of the ap$[n]$ array are calibrated as above, tapered by $u_0[n; N_{zf}, 4]$, and zero-padded to N_{fft} elements. Let AP$[k]$, $k = 0, \cdots, N_{\text{fft}} - 1$ be the complex Fourier transform of the resulting array. The transforms of the real amplitude and phase frames are given by

$$A[k] = \frac{1}{2}\left(\text{AP}[k] + \text{AP}[N_{\text{fft}} - k]^*\right), \qquad \Phi[k] = \frac{1}{2i}\left(\text{AP}[k] - \text{AP}[N_{\text{fft}} - k]^*\right)$$

for $k = 0, \cdots, N_{\text{fft}}/2$, where AP$[N_{\text{fft}}]$ is defined to be AP$[0]$. The one-sided amplitude and phase spectra for the frame are given by

$$S_a[k] = \frac{r}{f_s N_{zf} \left|H_r(f[k])\right|^2} \left|A[k]\right|^2, \qquad S_\phi[k] = \frac{r}{f_s N_{zf} \left|H_r(f[k])\right|^2} \left|\Phi[k]\right|^2 \tag{6}$$

with frequency array $f[k] = (f_s/(rN_{\text{fft}}))k$. We apply the same low- and high-frequency cutoffs as we did with the medium-band signal spectrum. The absence of a factor of 2 in the scaling factor of Eq. (6) [see Eq. (1)] gives a single-sideband presentation of the spectra, so that they can be labeled dBc/Hz when converted to dB. If a factor of 2 were present in the numerators, the unit for S_ϕ would have to be rad^2/Hz. As before, a number of frame spectra can be averaged into a run spectrum. The resolution bandwidth is given by Eq. (4).

V. Narrow-Band Processing

This processing mode also assumes that the signal consists of a carrier with weak sidebands. Its purpose is to achieve an arbitrarily large reduction in data rate, limited only by the user's patience. The stream of A–D data is reduced to a sequence of average amplitude and phase residuals, the averaging time being chosen by the user. The phase residuals from two channels can be combined into a differential phase. These streams of band-reduced data can be processed into time series, spectra, or Allan deviations (phase or differential phase only).

A. Amplitude and Phase Extraction

The stream of A–D data is divided into batches of size N_{xb}, which must be adjacent for the entire run. There is a minimum and maximum batch size (now 200 and 8192). A frame consists of n_{bf} batches, or $N_{\text{xf}} = n_{\text{bf}} N_{\text{xb}}$ A–D data, where n_{bf} can be any positive integer. Each batch is reduced to one sample of average amplitude and phase, and n_{bf} batch samples are averaged to produce a frame sample. The user has to choose N_{xb} and n_{bf} (with the bounds on N_{xb} enforced by the user interface) to achieve the desired reduced sample rate f_s/N_{xf}. Unless there are phase tracking problems (see below), the results for a fixed frame size should depend little on the number of batches per frame.

Let us represent the digitized signal by

$$x(t) = A(t) \cos \Phi(t)$$

where $\Phi(t)$ is the total phase, which one can think of as $\omega t + \theta + \phi(t)$, where $\phi(t)$ is a phase residual. The point is that $\Phi(t)$ is an intrinsic part of the signal (except for an unknowable additive constant

$2\pi n_0$), while ω and $\phi(t)$ trade off with each other. We assume that $A(t)$ and $\Phi(t)$ satisfy two imprecisely given conditions, called here the assumptions of small local variations: (1) Over a batch, the fractional variations of $A(t)$ from its mean are much less than 1, and (2) over at least two batches, the total phase differs by much less than one radian from a first-degree polynomial fit (constant phase offset plus constant frequency). Over longer time spans, the phase might deviate from a straight-line fit by many radians.

Let the batches of a run be indexed by k, $k = 0, 1, \cdots$. Batch k starts at time $t_k = kN_{\text{xb}}/f_s$. For the moment, let t run over the sequence of times $t_k + n/f_s, n = 0, \cdots, N_{\text{xb}} - 1$, in batch k. The Pony computation (Parts 1 and 2) of Appendix A is used to estimate the local frequency, amplitude, and phase of the batch. It gives $\hat{o}_k$ (radians per cycle), $\hat{A}_k$, and $\hat{\theta}_k$ such that

$$x(t) \approx \hat{A}_k \cos\left(\hat{o}_k f_s (t - t_k) + \hat{\theta}_k\right) \tag{7}$$

(The sign of $\hat{o}_k$ is taken to be the same as the polarity s_{pol}.) Write $\hat{\omega}_k = \hat{o}_k f_s$. With the assumptions of small local variations, it turns out that, to first order in these variations,

$$\hat{A}_k \approx \bar{A}_k \tag{8}$$

$$\psi_k := \hat{\omega}_k(\bar{t}_k - t_k) + \hat{\theta}_k \approx \bar{\Phi}_k \pmod{2\pi} \tag{9}$$

where $\bar{t}_k$, $\bar{A}_k$, and $\bar{\Phi}_k$ are the averages of t, $A(t)$, and $\Phi(t)$ over batch k. It is important to note that the approximation [Eq. (9)] of ψ_k to $\bar{\Phi}_k$ (mod 2π) is better than the approximation of the phase on the right side of Eq. (7) to $\Phi(t)$ because the errors in $\hat{o}_k$ and $\hat{\theta}_k$ tend to compensate each other in just the right way.

The average amplitude residual for batch k is computed by $a_k = \hat{A}_k/\hat{A}_0 - 1$. The computation of phase residuals is more delicate. According to Eq. (9), ψ_k, to first order, is the average total phase of the signal in batch k, modulo 2π. There are two problems. First, there is the 2π ambiguity. Second, we would like to have a phase *residual* instead of the large total phase. Let us use the initial measured frequency $\hat{\omega}_0$ and phase $\hat{\theta}_0$ to calibrate the total phase to a phase residual

$$\hat{\phi}(t) = \Phi(t) - \hat{\omega}_0\,(t - t_0) - \hat{\theta}_0 \tag{10}$$

where t now runs over all time beyond the starting time t_0 of the run. Note that $\hat{\phi}(t)$ depends on the calibration parameters $\hat{\omega}_0$, $\hat{\theta}_0$, so it is not intrinsic. Its average over batch k is

$$\hat{\phi}_k = \bar{\Phi}_k - \hat{\omega}_0\,(\bar{t}_k - t_0) - \hat{\theta}_0 \tag{11}$$

These are the batch phase residuals that we would like to compute. From Eq. (9) it follows that, to first order,

$$\left.\begin{aligned} \hat{\phi}_k &\approx \psi_k - \hat{\omega}_0\,(\bar{t}_k - t_0) - \hat{\theta}_0 \pmod{2\pi} \\ \hat{\phi}_0 &\approx 0 \pmod{2\pi} \end{aligned}\right\} \tag{12}$$

To a good approximation, then, we know the $\hat{\phi}_k$, modulo 2π. Because of the assumption of small local variations, we also can predict, with an error $< \pi$, how many radians the average total phase advances from one batch to the next, given its previous behavior. With this information, and with the measured $\hat{\phi}_0$ assumed to be 0, the 2π ambiguity can be removed sequentially from all the $\hat{\phi}_k$ by means of a second-order unwrapping algorithm given in Appendix C. It is the same algorithm, with different parameters, that is used for unwrapping the picket fence time-interval measurements that capture the 1-Hz zero crossings.

The algorithm also produces a sequence of prediction errors z_k that satisfies $|z_k| \leq \pi$. It measures how much the current phase differs from what we think it should be, based on the behavior of the previous batches. If any $|z_k|$ exceeds a certain threshold, now set at $\pi/2$, a caution is issued to the user. Perhaps the frequency is changing so fast that the assumption of small variations fails for the batch length N_{xb}. In effect, the analyzer may be losing phase lock, like a phase-locked loop whose bandwidth is too small. If this happens, the user can try decreasing N_{xb}. As mentioned above, the amplitude and phase residual averages for a frame are obtained simply by averaging n_{bf} batch values. Thus, if the user has to decrease N_{xb} to keep the analyzer in lock, he can maintain his chosen averaging time by increasing n_{bf}.

B. Differential Phase

By differential phase we mean some method of subtracting the phases of two channels that are being sampled simultaneously at the same rate. There are two flavors of differential phase processing. In S–S or X–X differential phase, it is assumed that both channels (1 and 2) originate at the same RF band and are downconverted to the same frequency. In this case, the total phases should not be too far apart, and so it makes sense to compute the batch averages

$$\delta\Phi_k = \Phi_k(1) - \Phi_k(2) - 2\pi n_0$$

where (1) and (2) identify the two channels and n_0 is the integer that makes $-\pi < \delta\Phi_0 \leq \pi$. Applying Eq. (11) to both channels, we obtain

$$\delta\Phi_k = \hat{\phi}_k(1) - \hat{\phi}_k(2) + (\hat{\omega}_0(1) - \hat{\omega}_0(2))\,(\bar{t}_k - t_0) + \hat{\theta}_0(1) - \hat{\theta}_0(2) - 2\pi n_0 \tag{13}$$

which gives the intrinsic quantity $\delta\Phi_k$ in terms of measured quantities.

The original design of the analyzer included a sample-and-hold unit so that channels 1 and 2 could be sampled simultaneously. This is no longer the case and, hence, the channel samples have to be interleaved at total rate $2f_s$ through the A–D converter: (1), (2), (1), (2), $\cdots$, where a channel 1 sample is paired with the *following* channel 2 sample. To deal with this situation, we use current batch frequency estimates to adjust the total phases of the two channels as if they were sampled halfway between the channel 1 sample time and the channel 2 sample time. The phase advance of channel 1 over a delay $1/(4f_s)$ is estimated as $\pi f_{\text{vid}}(1)/(2f_s)$, where $f_{\text{vid}}(1)$, the current estimate of the analog carrier frequency of channel 1, is computed by $f_{\text{vid}}(1) = f_s(n_{\text{base}}(1) + \hat{o}_k(1)/(2\pi))$. A similar correction of opposite sign is applied to the channel 2 total phase. Consequently, a correction

$$\frac{\pi}{2}\,(n_{\text{base}}(1) + n_{\text{base}}(2)) + \frac{1}{4}\,(\hat{o}_k(1) + \hat{o}_k(2))$$

has to be added to $\delta\Phi_k$.

In S–X differential phase, channel 1 is downconverted from 2295 MHz (S-band), channel 2 from 8415 MHz (X-band), or the reverse, and we are required to produce some version of

$$\text{S band phase} - \frac{3}{11}\,(\text{X band phase})$$

In a preliminary design, the analyzer simply computed the nonintrinsic quantity $\hat{\phi}_k(1) - (3/11)\hat{\phi}_k(2)$, which depends on the initial measured frequencies $\hat{\omega}_0(i)$, $i = 1, \cdots, 2$, and which, if a linear fit is not removed, has a random ramp component that depends on these measured frequencies. The current design uses a more objective method in which the measured frequencies are replaced by a priori known design frequencies $\omega_0(i) = f_s o_0(i)$. These are computed from the user-provided analog offset frequencies $f_{\text{ofst}}(i)$ by $\omega_0(i) = 2\pi(f_{\text{ofst}}(i) - f_s n_{\text{base}}(i))$. One can then produce phase residuals $\phi_k(i) = \hat{\phi}_k(i) + (\hat{\omega}_0(i) - \omega_0(i))(t_k - t_0)$ that start at zero but show ramps if the actual channel frequencies differ from the design frequencies. S–X differential phase is now just $\phi_k(1) - (3/11)\phi_k(2)$, which shows a ramp if the frequencies of the S- and X-channels are not related in exactly the right way. In contrast with the S–S or X–X situations, the first sample of this differential phase is zero; we are calibrating for frequency only and not attempting to measure the absolute synchronization of the two channels.

As with amplitude and phase, the batch averages of differential phase are combined into frame averages.

C. Time Series

The stream of narrow-band samples (frame average amplitude residuals, phase residuals, or differential phases) can be collected into a buffer and plotted against time. In the present software, we use a buffer management scheme that automatically subsamples the buffer by a factor of 2 when it fills up, crunches it to half its size, and begins to accept data at half the previous rate. At any time during the run, the buffer contains a record of the entire data stream, subsampled by some power of 2. Because phase residuals and differential phases are likely to be dominated by a straight line, we normally apply the calibration operation described in Section V.B before plotting them so that random fluctuations can be seen.

D. Spectrum

Any of the streams of narrow-band samples can be subjected to the same spectral estimation process. Because it takes longer to collect the data arrays, there is incentive to use the narrow-band data more efficiently than the medium-band data. In compensation, there is more processor time available per A–D sample for expensive postprocessing. We use an unweighted Thomson multitaper spectral estimator [10,7 (Chapter 7)] with orthogonal data tapers (trig prolates) computed by the author [5] (Appendix B). The user chooses a FFT size N_{fft}, a power of 2. At the start of the test, we compute an array of K orthogonal data tapers $u_k[n; N_{\text{fft}}, w], n = 0, \cdots, N_{\text{fft}} - 1, k = 0, \cdots, K - 1$. The value of K depends on w and on the sidelobe level we wish to tolerate in the frequency responses of the u_k. In the present design, $w = 4$, $K = 4$. An array of samples $x[0], \cdots, x[N_{\text{fft}} - 1]$, called a "narrow-band frame" (nbframe), is preconditioned by the calibration operation of Section V.B. Then K distinct "eigenspectra" $S_0, \cdots, S_{K-1}$ are computed by applying the tapers and a real FFT, giving

$$S_k[m] = \frac{N_{\text{xf}}}{N_{\text{fft}} f_s} \left| \sum_{n=0}^{N_{\text{fft}}-1} x[n] u_k[n; N_{\text{fft}}, w] \exp\left(-i2\pi nm/N_{\text{fft}}\right) \right|^2 \tag{14}$$

with frequency array $f[m] = (f_s/(N_{\text{xf}} N_{\text{fft}}))m$, $m = 0, \cdots, N_{\text{fft}}/2$. The spectrum of the nbframe is computed by averaging the eigenspectra:

$$S[m] = \frac{1}{K} \sum_{k=0}^{K-1} S_k[m] \tag{15}$$

and the overall run spectrum is computed by accumulating and averaging all the nbframe spectra. One advantage of this method is that, over smooth regions of the true spectrum, the variance of $S[m]$ is about K times smaller than the variance of each $S_k[m]$. With a single-taper method, variance could be reduced by using shorter nbframes or averaging the spectrum over frequency. Either of these methods increases the resolution bandwidth.

To prepare the spectrum for display, we cut off frequencies below $(f_s/(N_{\text{xf}}N_{\text{fft}}))w$ and do the usual conversion to dBc/Hz. The resolution bandwidth W_{nb} is given by

$$\frac{1}{W_{\text{nb}}} = \frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{W_{\text{nb},k}}$$

where

$$W_{\text{nb},k} = \frac{f_s N_{\text{fft}}}{N_{\text{xf}}\left(\sum u_k[n]\right)^2}$$

is the resolution bandwidth of S_k. Although it is not apparent, W_{nb} is proportional to $1/N_{\text{fft}}$; one can use N_{fft} to trade off resolution against run length.

The user should be aware that the spectral window of this method is not bell shaped but approximately rectangular with ripples across the top. If the spectrum has a bright line whose width is of the order of one FFT bin or less, the image of the line may appear to have four small peaks at the top. These are artifacts of the method and do not indicate a splitting of the line. (See Appendix B and Fig. 3.)

In the current version of narrow-band processing, we have achieved bandwidth reduction by unweighted averaging: The batch samples of amplitude and phase are, to first order, unweighted averages of these quantities, and frame samples are unweighted averages of batch samples. Consequently, a calculated

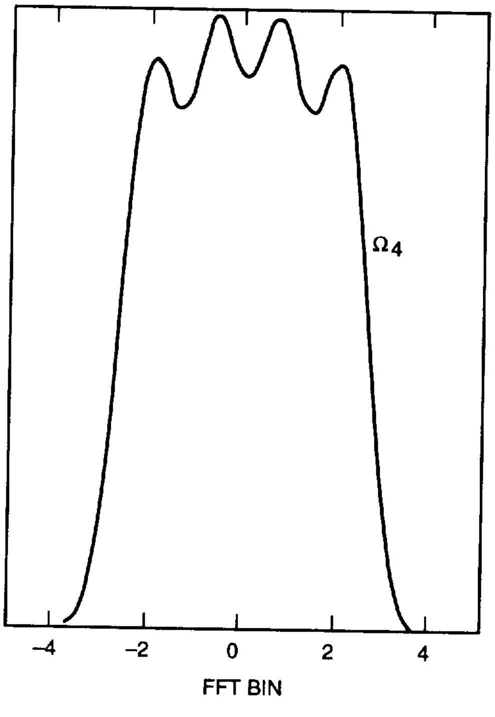


Fig. 3. Shape of a bright line for narrow-band spectrum.

spectrum for frequencies between 0 and the Nyquist frequency $f_s/(2N_{\mathrm{xf}})$ is not, strictly speaking, an estimate of the spectrum of the quantity in that frequency range, but rather an estimate of the spectrum of the averages of the quantity over an averaging time $\tau_0 = N_{\mathrm{xf}}/f_s$, sampled at rate $1/\tau_0$. This spectrum suffers from both aliasing and distortion. The Pony method of extracting batch samples of amplitude and phase leads inherently to this situation for frames consisting of one batch. The main decision was how to deal with further bandwidth reduction: whether to use a lowpass decimation filter, a bank of such filters, or simply to extend the situation with unweighted averaging. The advantages of the chosen design are simplicity, consistency, and flexibility in the choice of decimation factor (frame length), which can be large enough to exhaust the patience of any user.

E. Allan Deviation

The stability analyzer can compute the Allan deviation of frame samples of phase or differential phase for an array of averaging times τ that are powers of 2 times the frame duration τ_0. It was required to remove an estimate of linear frequency drift from the results. For a drift estimator, we use the simple three-point estimator suggested by Weiss [11]. Although the basic method is covered in [2] and [3], we run through the computations for a particular value of $\tau = n\tau_0$. Let the stream of phase samples be ϕ_0, $\phi_1, \cdots$. At a given point in the run, we have accumulated sums of the first and second powers of m second differences of ϕ_i with stride n, namely,

$$s_p = \sum_{j=2}^{m+1} \left(\Delta_n^2 \phi_{nj}\right)^p, \qquad p = 1,\ 2$$

where $m \geq 4$. (The author realizes that the sum for $p = 1$ telescopes.) We have also collected a subsampled version

$$\phi_0, \phi_d, \phi_{2d}, \cdots, \phi_{Id}$$

of the whole run so far by the same buffer mechanism used for time series above. The calculations proceed as follows:

$$n_c = \left\lfloor \frac{I}{2} \right\rfloor d$$

$$D_c = \phi_{2n_c} - 2\phi_{n_c} + \phi_0 \qquad \text{(unscaled drift estimate)}$$

$$V = \frac{s_2}{m} - \left(\frac{s_1}{m}\right)^2 \qquad \text{(sample variance)}$$

$$V = V + \left(\frac{s_1}{m} - D_c\left(\frac{n}{n_c}\right)^2\right)^2 \qquad \text{(drift correction)}$$

$$\nu = (m-1)\left(0.8776 + 0.0643e^{-(1/2)(m-4)}\right) \qquad \text{(degrees of freedom)}$$

$$V^{\pm} = V\left(1 \pm \sqrt{\frac{2}{\nu}}\right)$$

$$\sigma_y(\tau) = \frac{\sqrt{V}}{\sqrt{2}\, 2\pi f_{\text{ref}}\tau}, \qquad \sigma_y^{\pm}(\tau) = \frac{\sqrt{V^{\pm}}}{\sqrt{2}\, 2\pi f_{\text{ref}}\tau} \qquad \text{(Allan deviation with error bar)}$$

The formula for degrees of freedom is an empirical formula fitted to the author's numerical results for the random-walk-of-frequency model of phase deviations (f^{-4} noise). The error bars, which are really the square roots of "one-sigma" error bars for $\sigma_y^2(\tau)$, should be conservative for f^{β} noise, $\beta > -4$.

VI. Zero-Crossing Processing

To capture the up-crossing times of the 1-Hz square wave, a preliminary measurement of the nominal period p of the square wave is taken with the interval timer, which is then set to measure the time intervals between each subsequent up-crossing and the next pulse of a 10-Hz train of reference pulses, the "picket fence." These readings are unwrapped into a sequence of time residuals, as described in [4]. The algorithm, which is really the same as the one used for unwrapping the narrow-band phase deviations (Appendix C), need not be reproduced here. The time deviations produced by this algorithm are multiplied by the scale factor

$$\frac{2\pi f_{\text{ref}}}{f_{\text{mix}} p}$$

to give phase deviations that can be used like the batch averages of phase deviation that come from the narrow-band process. For time series and Allan deviation, we allow only one batch per frame, as the 1-second period is natural for the user. For spectrum, an arbitrary number of batches per frame is allowed so that users can shrink the Nyquist frequency below 0.5 Hz as much as they want.

References

[1] J. C. Breidenthal, C. A. Greenhall, R. L. Hamell, and P. F. Kuhnle, "The Deep Space Network Stability Analyzer," *Proceedings of the 26th Annual Precise Time and Time Interval Applications and Planning Meeting*, Reston, Virginia, December 5–8, 1994, in press.

[2] C. Greenhall, "Removal of Drift From Frequency Stability Measurements," *The Telecommunications and Data Acquisition Progress Report 42-65, July–August 1981*, Jet Propulsion Laboratory, Pasadena, California, pp. 127–131, October 15, 1981.

[3] C. Greenhall, "Frequency Stability Review," *The Telecommunications and Data Acquisition Progress Report 42-88, October–December 1986*, Jet Propulsion Laboratory, Pasadena, California, pp. 200–212, February 15, 1987.

[4] C. Greenhall, "A Method for Using a Time Interval Counter to Measure Frequency Stability," *IEEE Trans. Ultrason. Ferroelec. Freq. Control*, vol. 36, pp. 478–480, 1989.

[5] C. Greenhall, "Orthogonal Sets of Data Windows Constructed From Trigonometric Polynomials," *IEEE Trans. Acoust. Speech. Sig. Proc.*, vol. 38, pp. 870–872, 1990.

[6] S. Marple, *Digital Spectral Analysis With Applications*, Englewood Cliffs, New Jersey: Prentice Hall, 1987.

[7] D. Percival and A. Walden, *Spectral Analysis for Physical Applications*, Cambridge, United Kingdom: Cambridge University Press, 1993.

[8] T. Pham, J. Breidenthal, and T. Peng, "Stability Measurements of the Radio Science System at the 34-m High-Efficiency Antennas," *The Telecommunications and Data Acquisition Progress Report 42-114, April–June 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 112–139, August 15, 1993.

[9] D. Slepian, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty—V: The Discrete Case," *Bell System Tech. J.*, vol. 57, pp. 1371–1430, 1978.

[10] D. Thomson, "Spectrum Estimation and Harmonic Analysis," *Proc. IEEE*, vol. 70, pp. 1055–1096, 1982.

[11] M. Weiss and C. Hackman, "Confidence on the Three-Point Estimator of Frequency Drift," *Proceedings of the 24th Annual Precise Time and Time Interval Applications and Planning Meeting*, McLean, Virginia, pp. 451–455, 1992.

Appendix A

The Pony Calculation

This is a batch method for computing the frequency, amplitude, and phase of a sampled sine wave. It comes from a method of harmonic analysis called Prony's method [6 (Chapter 11)], which analyzes a waveform into the sum of n sine waves. The calculation we call "Pony" is simply a modification of Prony's method for $n = 1$.

I. Part 1: Frequency

Let the data array be $x[0], \cdots, x[N-1]$. If $x[n]$ were exactly of form $A \cos\ (on + \theta)$, then we would have

$$x[n+1] + x[n-1] = (2 \cos o) x[n], \qquad n = 1, \cdots, N-2 \tag{A-1}$$

On the other hand, if $x[n]$ is a noisy cosine wave, then let us estimate $\cos o$ by projecting the vector $x[n+1] + x[n-1]$ orthogonally onto the vector $x[n]$. The computation is

$$c = \frac{(1/2)\left(x[0]x[1] + x[N-2]x[N-1]\right) + \sum_{n=1}^{N-3} x[n]x[n+1]}{\sum_{n=1}^{N-2} x[n]^2}$$

$$o = \arccos\ (c) \quad \text{in } [0, \pi]$$

if $|c| \leq 1$, else o goes to the nearest port in the storm, 0 or π. One may also change the sign of o according to polarity considerations.

For use in Part 2 and elsewhere, a single-precision complex array of powers u^n, $n = 0, \cdots, N-1$, where $u = \exp(-io)$, is generated by a vectorized algorithm that we illustrate for the case $N = 16$. Compute the dyadic powers u^2, u^4, u^8 in double precision and convert them to single precision. Lay down the powers u^0, u^8 in the array, multiply them by u^4, and lay down the products to give u^0, u^4, u^8, u^{12}. Multiply by u^2 to give $u^0, u^2, \cdots, u^{14}$. Multiply by u to give the desired array. For large N, the successive steps get more and more efficient for a vector processor.

II. Part 2: Amplitude and Phase

Having estimated the frequency, we use it to estimate amplitude and phase. Let $a = A\cos\theta$, $b = A\sin\theta$, and solve the least-squares problem

$$x[n] \approx a\cos\ on\ - b\sin\ on, \qquad n = 0, \cdots, N-1$$

for the parameters a and b. The coefficients of the normal matrix can easily be expressed in closed form, and the solution computed as follows:

$$x_c = \sum_{n=0}^{N-1} x[n]\cos\ on, \qquad x_s = -\sum_{n=0}^{N-1} x[n]\sin\ on \tag{A-2}$$

$$\mathrm{cc} = \frac{1}{2}\left[N + \cos\ (o(N-1))\frac{\sin\ oN}{\sin\ o}\right], \qquad \mathrm{ss} = \frac{1}{2}\left[N - \cos\ (o(N-1))\frac{\sin\ oN}{\sin\ o}\right]$$

$$\mathrm{cs} = \frac{1}{2}\sin\ (o(N-1))\frac{\sin\ oN}{\sin\ o}$$

$$D = \mathrm{cc}\cdot\mathrm{ss} - \mathrm{cs}^2$$

$$a = \frac{\mathrm{ss}\cdot x_c + \mathrm{cs}\cdot x_s}{D}, \qquad b = \frac{\mathrm{cs}\cdot x_c + \mathrm{cc}\cdot x_s}{D}$$

$$A = \sqrt{a^2 + b^2}, \qquad \theta = \mathrm{angle}(a + ib)$$

Most of the work is in the in-phase and quadrature mixing operation [Eq. (A-2)], which uses the array u^n whose generation is described in Part 1.

The calculation given here can be regarded as an improvement on the approximations

$$a \approx \frac{2}{N}x_c, \qquad b \approx \frac{2}{N}x_s$$

which are exact if oN is an integer. It has been observed [8] that these approximations are inadequate if oN is not an integer, because the double-frequency terms have not entirely been eliminated by the mixing–filtering operation, Eq. (A-2).

Appendix B
Windows and Filter

The data tapers and lowpass decimation filter are based upon the author's "trig prolate" approximations [5] to the discrete prolate spheroidal sequences of Slepian [9]. The notation $u_k[n; N, w]$ is used here in place of the notation $u_k[n; N, w/N, w]$ in [5].

Figure 1 shows the frequency responses (spectral windows) of the data tapers used for spectral estimation. The Ω_{05} curve applies to full-band spectrum, Ω_{04} to medium-band spectra, and Ω_4 to narrow-band spectra. Note that Ω_4 is the average of the windows of the four eigenspectra, Eq. (14), that are averaged into the total spectrum, Eq. (15). The expectation of a spectral estimate is the convolution of the true spectrum with the spectral window. The Ω_{0w} windows are bell shaped. Figure 3 plots Ω_4 on a linear scale against a two-sided frequency axis to show how a narrow bright line would appear in the spectral estimate if it were plotted on a linear scale. The ripples at the top will not be so prominent on a typical dB scale.

The N-point FIR lowpass filter used in medium-band processing before decimation by r is built in a conventional way from the trig prolate window $u_0[n; N, w]$. The formula for it is

$$h_r[n] = u_0[n; N, w] \text{ sinc}\left(2\pi f_h \left(n - \frac{N-1}{2}\right)\right), \qquad n = 0, \cdots, N-1$$

normalized so that $\sum h_r[n] = 1$, where

$$w = 4, \qquad N = 16r, \qquad f_h = \frac{0.4}{r}, \qquad \text{sinc } x = \frac{\sin x}{x}$$

Figure 2 shows the frequency response of this filter for $r = 2$. The response is essentially the same for all r if frequency is scaled according to the x-axis of Fig. 2. Only one table is needed to represent the frequency response for the purpose of equalizing the medium-band spectra.

Appendix C

Phase Unwrapping Algorithm

This algorithm produces the narrow-band phase residuals $\hat{\phi}_k$ from the carrier frequency and phase estimates extracted from each batch by the Pony calculation. It is assumed that the batches all have length N_{xb} and are adjacent. Recall the definition, Eq. (5), of the mods function. Let the damping constant λ be a number between 0 and 1. In the following algorithm, ψ'_k is related to ψ_k of the main text by $\psi'_k = \psi_k - \hat{o}_0(N_{\text{xb}} - 1)/2$.

```
$\psi'_0 = \hat{\theta}_0$
$z_0 = 0, \hat{\phi}_0 = 0, q_0 = 0$
For $k =$ $1, 2, \cdots$

    Obtain the batch frequency and phase $\hat{o}_k$, $\hat{\theta}_k$.
    $\psi'_k = (\hat{o}_k - \hat{o}_0)(N_{\text{xb}} - 1)/2 + \hat{\theta}_k$    ! $\psi_k$ is total phase $\Phi_k$ mod $2\pi$.
    $z_k =$ mods $(\psi'_k - \psi'_{k-1} - \hat{o}_0 N_{\text{xb}} - q_{k-1},\ 2\pi)$    ! prediction error.
    If $|z_k| > \pi/2$ (say), then issue caution "losing lock" to user.
    $\hat{\phi}_k = \hat{\phi}_{k-1} + q_{k-1} + z_k$    ! output phase residual.
    $q_k = q_{k-1} + \lambda z_k$    ! low pass-filtered $\Delta\hat{\phi}_k$.

Next $k$.
```

Note that q_k, z_k satisfy

$$q_k = (1 - \lambda) q_{k-1} + \lambda \Delta\hat{\phi}_k, \quad z_k = \Delta\hat{\phi}_k - q_{k-1}$$

This says that q_k is a lowpass-filtered version of $\Delta\hat{\phi}_k$, and z_k is a prediction error for $\Delta\hat{\phi}_k$. The basis of the algorithm is (1) the assumption that $|z_k| < \pi$ and (2) the knowledge of z_k modulo 2π, namely,

$$z_k = \Delta\Phi_k - \hat{\omega}_0 \Delta\bar{t}_k - q_{k-1} \cong \Delta\psi_k - \hat{o}_0 N_{\text{xb}} - q_{k-1} \qquad (\text{mod } 2\pi)$$

Any value for λ in [0, 1] is meaningful. If $0 < \lambda < 1$, then, in effect, a weighted average of previous phase advances, with weights $(1 - \lambda)^n$, is used to judge what the current phase advance should be. In the script files that drive the software, λ has been set to $1/10$. This provides some stability against large errors while maintaining the ability of the algorithm to follow frequency drifts.

Errata

In "Adaptive Line Enhancers for Fast Acquisition" by H.-G. Yeh and T. M. Nguyen, which appeared in *The Telecommunications and Data Acquisition Progress Report 42-119, July–September 1994*, November 15, 1994, the plot in Fig. 14 was incorrectly situated. The correct figure is provided below.

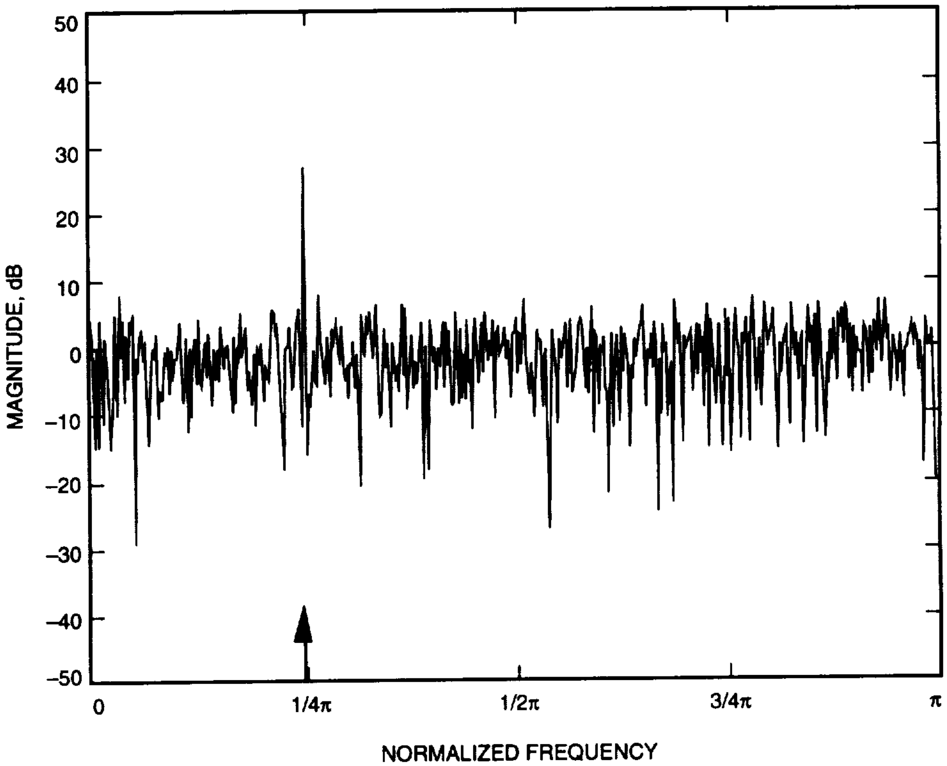


Fig. 14. Magnitude of the input data to the ALE, ALEDF, AND ALECA.