RADIOTELEMETRY

Radiotelemetry, by definition, is the science and technology of automatically measuring and transmitting data by radio from remote sources, as from space vehicles, to receiving stations for recording and analysis. It involves preparing of data generated at remote sources, transmitting data through radio channels, and processing data at the receiving stations. Behind these three tasks, how to convey the information in a reliable and timely manner through a communication link is the most fundamental problem in this field.

There are many ways to define a communication link, depending on where the link starts and ends. Customarily, a communication link is the transmission path through which information-bearing waveform flows from one point to another. A generic link includes the modulator, the transmitter, the channel, the receiver, and the demodulator. Another popular view of a link is to include error-correction coding and encoding as part of the link because, as we see in a later discussion, link performance is a function of both modulation and coding. Yet, for a communication system that uses data compression, one might want to take an end-to-end view and include the error propagative effect of data compression in link design and analysis, because the error propagative property affects the integrity and throughput of the link. In the following sections, we briefly discuss some important elements of a communication link: data compression, forward error-correction coding, modulation, demodulation, and synchronization. However, before moving on to these individual topics, we first discuss a higher level consideration of the overall link reliability, namely, the link-budget analysis.

The main objective of link-budget analysis is to maximize the data return for a communication link subject to (1) the available communication resources and (2) the required data quality. Power and bandwidth are the two primary resources in communication, and data quality is usually expressed in terms of the frequency of errors in the received data, that is, the bit-error rate (BER). To design a communication system, the first step is to understand what the available resources are. Whether a system is power-limited (e.g., deep space communications) or bandwidth-limited (e.g., near-Earth satellite communications) determines the available options for modulation and error-correction coding schemes.

COMMUNICATION LINK-BUDGET ANALYSIS

The link budget is a balance sheet of the gains and losses in decibels (dB) of various parameters in the communication path. Many of these parameters are statistical or time-varying or both. The required BER is a direct function of the bit signal-to-noise ratio (SNR), denoted as E_b/N_0, which in turn is a function of the modulation and error-correction coding used. Because of the statistical nature (uncertainty) of these parameters, a safety margin M is built in to guarantee the transmission data quality at any given time.

To put link-budget analysis into perspective, more parameters need to be defined and they are given as follows. To begin with, the total received signal-power-to-noise-power-spectral-density ratio, P_T/N_0, which encompasses all the power gains and losses in the communication path, is conveniently expressed as (1)

$$\frac{P_T}{N_0} = \frac{\mathrm{EIRP}(G/T)}{kL_sL_0} \tag{1}$$

where EIRP is the effective isotropic radiated power of the spacecraft, G/T is the receiver sensitivity of the ground system, k is Boltzmann's constant, $L_s = (4\pi d/\lambda)^2$ is the space loss where d is the distance between the transmitter and receiver, λ is the wavelength, and L_0 denotes all other losses and degradation factors not specifically addressed in the above equation. Note that all gains and losses are statistical quantities. Then, let R_s be the symbol rate and R_c be the code rate expressed as the ratio of the number of information bits in a code word to the number of coded bits in a code word. The data rate or bit rate, denoted as R_b, is related to the symbol and code rates via $R_b = R_s R_c$. The bit SNR is written as

$$\frac{E_b}{N_0} = \frac{P_T/N_0}{R_s R_c} \tag{2}$$

Finally, let $(E_b/N_0)_{req}$ be the required E_b/N_0 to guarantee a certain BER. To ensure a reliable link, the constraining dependency is expressed in dB as follows:

$$\left(\frac{E_b}{N_0}\right)_{req} + M \leq \frac{P_T}{N_0} - R_s - R_c \tag{3}$$

The essence of link-budget analysis is to determine how much link margin M is required and this, in turn, is determined by how well the statistics of various link parameters are known. If M is too small, one might get a large data return but the data might have frequent errors. If M is too large, one might get continuous clean data but the throughput might be low. The analyst needs to make the right tradeoff on quantity versus quality, depending on the requirements and objectives of the service the link supports. Also, in an operational scenario, one usually starts out with a large M (a conservative approach) for a newly deployed system, and gradually reduces M (an aggressive approach) as more experience and confidence accumulates.

DATA COMPRESSION

In the midst of the Information Age, when new data—texts, images, sensor data, and many other forms of knowledge—are generated at a lightning pace, how these data are efficiently communicated and/or archived becomes increasingly important. Data obtained directly from sources usually contains much redundancy, and data compression is the process of applying “pressure” to remove the redundancy. The processes of compression and decompression add complexity to the overall system. The main issue in the use of data compression is the tradeoff between the efficiency of data handling and the complexity of signal processing to retrieve the information.

In the English language, letters like “a” and “e” are used more frequently than “y” and “z”, and words like “and” and “the” are much more often than “data” and “compression”. For images and sensor data, the differences between adjacent samples are small rather than large. This nonuniform frequency distribution of data associated with most information sources allows one to assign shorter bit patterns to represent the more frequent elements from the source and longer bit patterns to represent the less frequent elements from the source. This is the simplest view of data compression.

Data compression is broadly classified in two categories: lossless and lossy compression. Lossless compression denotes compression in which the decompressed or reconstructed data exactly match the original. Lossy compression represents compression methods where some degradation of quality is tolerable if a more compact but approximate representation is achieved. Next, we give brief surveys on text compression (lossless) and image compression (lossless and lossy). However, the compression of audio, video, and facsimile is beyond the scope of this article. Interested readers are encouraged to consult specialized books and technical journals in these areas.

Textual data is probably the first area where people applied compression. In 1832 Samuel Morse developed a code using dots, dashes, and spaces to represent letters, numbers, and punctuation for telegraph transmission. Each dot or dash is delimited by a space. A dot and a space take the same time. A dash is three times longer. Morse assigned fewer time units to more commonly occurring letters (e.g., “e” is a dot), and more time units to ones (e.g., “q” is dash-dash-dot-dash) that rarely occur. Today, most popular text compression schemes are different variations of the Lempel–Ziv (LZ) scheme developed in 1977 (2) and 1978 (3). LZ schemes use a dictionary approach. Dictionary coding operates by replacing groups of consecutive characters with indices in some dictionary. It exploits the frequent reoccurrence of certain exact patterns that are very typical of textual data. Other than compressing text, LZ schemes are reasonably good at compressing images and other sources. LZ schemes are generally considered universal compression algorithms. The UNIX command “compress” and the modem standard V42.bis are examples of applications using LZ schemes.

Image compression and other waveform compression is lossless or lossy. For scientific and medical applications that demand further processing and analysis, one might prefer lossless compression to ensure data accuracy. In other applications, such as image database, electronic photography, and desktop publishing, where transmission and storage bandwidths are limited, lossy compression is usually employed to reduce the data size at the expense of image quality.

Lossless image compression consists of two steps, modeling and coding. A predictive model predicts a pixel value based on a previously transmitted one, and compares it with the current value. The error value is coded instead of the original pixel value. On the receiver side, because the errors and the predictive scheme are known, the receiver recovers the value of the original pixel. An example of lossless image compression is the Rice code used in space and satellite communications.

Lossy image compression involves an irreversible quantization step that causes distortion between the original and the reconstructed image. Popular techniques in lossy image compression include differential pulse code modulation (DPCM), transform coding, subband coding, vector quantization, and some hybrid techniques. The Joint Photographic Expert Group (JPEG) established an international standard on still image compression that uses the discrete cosine transform along with scalar quantization and entropy coding. Currently, JPEG encoder and decoder chips are available from many semiconductor manufacturers.

FORWARD ERROR-CORRECTION CODING

In 1948, the landmark paper “A Mathematical Theory of Communication” (4) by Claude Shannon provided a mathematical framework for analyzing communication, the process of sending information from one point to another through a noisy media, by using the concept of entropy. He provided an existence proof which shows that communication systems can be made arbitrarily reliable as long as the fraction of redundant signals in the signal stream exceeds a certain value, which is a function of the signal and noise statistics. The process of introducing redundancy in the signal stream to combat noise, so that retransmission is not needed, is called forward error-correction coding. Shannon did not show how this could be done in his famous theorem. Later on, much research effort has yielded a variety of interesting theoretical and practical results in this area.

Because redundant signals are deliberately introduced in the signal stream as error-correction coding, bandwidth expansion is expected. Encoding and decoding processes inevitably also add complexity to the overall system. However, it can be shown that, with a proper choice of coding scheme, the resulting E_b/N_0 can be considerably lower compared with an uncoded system. Thus error-correction coding allows one to trade power for bandwidth and complexity. In this article, the application of two popular error-correction coding schemes, block codes and convolutional codes are explained, without providing the detailed performance analysis of these codes. Other more advanced coding schemes, like trellis codes and turbo codes, are not covered here. Interested readers are encouraged to consult specialized books and technical journals in this field.

In block coding, a block of k bits is encoded into a block of n bits, where $n > k$. There are 2^k code words in a code space of 2^n. The code rate R_c, which is the ratio k/n, is a measure of the amount of added redundancy. The minimum distance $d_{\min}$, which is the Hamming distance between two closest code words, is a measure of the error-correction capability of the code. It can be shown that $d_{\min} \le n - k + 1$ (Singleton bound). For bounded-distance decoding, the number of errors t that the code can correct is related to $d_{\min}$ by

$$2t + 1 \le d_{\min} \tag{4}$$

In convolutional coding, sequences of k information bits long are encoded as continuous sequences using a kK-stage shift register, where K is called the constraint length. n output sequences are produced, each of which is generated by a selected set of combinations of these K input sequences. The n output sequences are multiplexed into a single-bit stream for transmission. Similar to a block code, the ratio k/n is a measure of the redundancy of the convolutional code. The minimum distance d_{free} between two closest code sequences, is a measure of the error correction capability of the code.

The most popular block code used today is the class of Reed–Solomon (RS) codes. RS codes are maximum-distance-separable codes that achieve maximum error-correction capability in the bounded-distance decoding sense. Well-known encoding and decoding algorithms exist that are reliable in software and hardware. RS codes are flexible. Long, powerful RS codes exist for demanding applications, and short RS codes for simple applications. Deep space communications use the (255, 223) RS code as the outer code of the concatenated coding system. This code is block-interleaved and is used in conjunction with an inner convolutional code to provide reliable communications between the spacecraft and the ground stations. The concatenated coding system is designed to combat the additive white Gaussian noise of the deep space communication links. Simpler RS codes are used in compact disks (CD), a mass-marketed consumer product. The coding system used, called the cross-interleaved Reed–Solomon code (CIRC), is composed of two RS codes with (n,k) values (32, 28) and (28, 24). In between the two encoders are sets of delay lines which scramble the data stream. The CIRC coding system is designed to correct the burst-like errors, typical of a CD channel.

The most popular decoding scheme for convolutional codes is the Viterbi algorithm. The complexity of Viterbi decoding grows exponentially with the constraint length of the code, which is the number of memory elements in the encoder shift register. As mentioned earlier, convolutional codes are used in deep space communications. In the past, short constraint-length codes (7 or 9) are used because of the limitations in hardware technologies. Today, long constraint-length Viterbi decoders (up to 15) are designed and built. Convolutional codes are also popular in satellite communications and wireless communications. Presently, only short constraint-length convolutional codes are used in these areas.

INTERACTION BETWEEN DATA COMPRESSION AND ERROR CONTROL IN A COMMUNICATION SYSTEM

This section addresses the interaction between data compression and error control (containment/detection/correction) processes, two indispensable building blocks in modern digital communication systems. Data compression conserves transmission and storage bandwidth by removing redundancy in the source data. Error correction introduces controlled redundancy to the data to eliminate channel errors. A combination of both techniques ensures efficient and reliable transmission of information from one point to another. The famous separation principle indicates that data compression and error control processes can be completely separated, that is, the task of transmitting the output of a source through a channel can be separated without loss into the task of forming a binary representation of the source output and the task of sending a binary sequence through the channel. This is not quite true in practice, however. In evaluating an end-to-end communication system, we have to consider effects, like instrument failures, adverse weather conditions, synchronization loss, and other unpredictable phenomena in the communication system and the channel. The error-control process cannot handle all possible kinds of errors. Thus the effect of error propagation should be considered when evaluating a data compression scheme in a communication system.

There are two issues to be dealt with: how to prevent errors from occurring and, if errors occur, how to contain the errors. The former issue can be dealt with easily (at least in the conceptual sense) by using higher transmission power or more powerful error-correction coding techniques on the channel and better fault-tolerance techniques on various system components. The second issue of how to contain errors is more intricate and has commonly been overlooked. There are mainly two approaches (or a combination of both) to deal with the problem. The first approach is to add extra redundancy to the compressed data to detect and confine errors. Some simple ways are adding a delimiter pattern at regular intervals and forcing fixed block-size transmission by appending filling 0's, etc. These techniques reduce compression performance. The second approach is to choose data compression schemes which are less susceptible to error propagation. These schemes usually transmit the compressed data in fixed block size or inherently have delimiter patterns in their compressed data streams. We illustrate the interaction between data compression and error-correction coding with an example in a later section.

MODULATION AND DEMODULATION

Modulation

Modulation is the process of transforming signals into waveforms suitable for transmission through a certain type of medium or channel. For example, when transmitting electronic signals through free space with an antenna which converts electronic signals into electromagnetic (EM) waves, the physical dimensions of the antenna aperture are at least of the same order of the wavelength of the EM wave. Therefore, a baseband signal has to modulate a sinusoidal signal at radio frequency (RF), called the carrier, before transmitting it by antennas of reasonable size.

The typical modulation procedure for digital signal transmission starts with a baseband pulse code modulation (PCM) which converts the analog signal into a binary data stream. Depending on the subsequent RF modulation schemes, this binary data stream is fed into different devices. For example, fixed-length blocks of the binary data are used to determine the instantaneous frequencies of the carrier for a frequency-shift-keyed (FSK) system. On the other hand, the binary data drive a pulse generator whose output modulates the subcarrier (if used) and the RF carrier in a typical phase-shift-keyed (PSK) system. There are several different binary data formats (5). For example, non-return-to-zero (NRZ) and bi-phase (also known as the Manchester code) are the ones more commonly used. In some applications, pulse shapes other than the square pulse are selected to represent a binary digit in the data stream. The choice of a particular data format and pulse shape is determined by several factors, for example, considerations of bandwidth efficiency, inherent synchronization features, and the noise immunity of each data format.

PSK modulation has been widely adopted for deep space communications mainly because it is a very efficient type of modulation in bit-error performance and the resulting signal has a constant envelope which allows the power amplifier to achieve the maximum efficiency by operating at the saturation point. In general, a communication system using the PSK modulation is designed to have a multiple phase-shift-keyed (MPSK) signal of which the number of phasor states, denoted by M, is any power of 2. The increased BER with M for MPSK signals, however, prevents its use when M is large. Furthermore, because of the ease of being decomposed into two orthogonal channels for further signal processing, the MPSK modulated signals commonly used for radiotelemetry are limited to the cases where $M = 4$ or lower, namely, the binary phase-shift-keying (BPSK), the quadrature phase-shift-keying (QPSK) or its variations, such as offset QPSK, unbalanced QPSK, and pulse-shaped QPSK, etc. For some applications, the quadrature amplitude-shift-keying (QAM) modulation is used to allow multilevel signals transmitted on either one or both of the mutually orthogonal channels.

It is possible to include subcarrier(s) for BPSK modulation. The purpose of using subcarrier(s) is to separate data sidebands of different signals and the carrier so that they do not interfere with each other. Whether or not to use subcarrier(s) depends on the mission design. For example, a subcarrier is preferred when a residual carrier component is preserved for spacecraft navigation or certain radio science experiments. On the contrary, it may not be a good idea to keep a subcarrier when transmission power is weak or the bandwidth efficiency becomes a major concern. Two types of subcarriers, the sine wave and the square wave, are commonly used. The sine-wave subcarrier along with the phase-modulated carrier produces fast-decaying data sidebands and, therefore, is recommended for near-Earth space missions where the interference caused by the power spillover to the adjacent frequency bands is the major concern (6). The same recommendation also suggests using the square-wave subcarrier for deep space missions because of the ease of generating a square wave onboard spacecraft and the relatively less stringent bandwidth allocation for this type of mission.

A phase-modulated (PM) telemetry signal is represented mathematically by

$$S_{\mathrm{T}}(t) = \sqrt{2P_{\mathrm{T}}} \sin\left(\omega_{\mathrm{c}} t + \sum_{i=1}^{N} m_i S_i(t)\right) \tag{5}$$

where P_{T} is the total power of the received signal, ω_{c} is the carrier frequency, m_i is the modulation index associated with the ith data source, and

$$S_i(t) = \begin{cases} \displaystyle\sum_{k=-\infty}^{\infty} d_{k,i} P_i(t-kT), \\ \text{for PCM/PM (without subcarrier)} \\ \left[\displaystyle\sum_{k=-\infty}^{\infty} d_{k,i} P_i(t-kT)\right] \sin(\omega_{\mathrm{sc}_i} t), \\ \text{for PCM/PSK/PM with a sine-wave subcarrier} \\ \left[\displaystyle\sum_{k=-\infty}^{\infty} d_{k,i} P_i(t-kT)\right] \operatorname{sgn}[\sin(\omega_{\mathrm{sc}_i} t)], \\ \text{for PCM/PSK/PM with a square-wave subcarrier} \end{cases} \tag{6}$$

represents either a normalized baseband waveform (with $d_{k,i} = \pm 1$) or a normalized BPSK modulated subcarrier waveform (at the frequency ω_{sc_i}). $P_i(t)$ is the pulse function of unit power and T is the reciprocal of the symbol rate. Note that, when more than one data sources are included (that is, $N > 1$ in Eq. (5)), only one data source output is allowed to directly modulate the carrier, denoted by its processing order as PCM/PM, and the rest must be put on subcarrier(s) before modulating the carrier, similarly denoted as PCM/PSK/PM in Eq. (6). An exception of this single PCM/PM rule is the code division multiplexing, in which multiple data sources are put directly on the carrier and remain distinguishable because of the unique code sequence (for example, pseudo-random codes or Walsh-Hadamard codes) used by each of them.

Typically, a deep space downlink signal (i.e., a signal sent from spacecraft to Earth) consists of two or more data sources of which, besides the telemetry signal, a ranging signal is included for spacecraft navigational purpose (7). For example, a mathematical expression of a downlink signal consisting of a sinusoidal ranging signal at the frequency ω_1 and a binary telemetry signal of NRZ format, $d(t)$, which is modulated onto a square-wave subcarrier, is given by

$$
\begin{aligned}
&S_{\mathrm{T}}(t)\\
&= \sqrt{2P_{\mathrm{T}}}\sin\{\omega_{\mathrm{c}}t + m_1\sin(\omega_1 t)\\
&\quad + m_2 d(t)\mathrm{sgn}[\sin(\omega_{\mathrm{sc}}t+\theta_{\mathrm{sc}})] + \theta_{\mathrm{c}}\}\\
&= \sqrt{2P_{\mathrm{T}}}\\
&\left\{\begin{aligned}
&\mathrm{J}_0(m_1)\cos(m_2)\sin(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}})\\
&\quad + d(t)\mathrm{J}_0(m_1)\sin(m_2)\mathrm{sgn}[\sin(\omega_{\mathrm{sc}}t+\theta_{\mathrm{sc}})]\cos(\omega_{\mathrm{c}}t+\omega_{\mathrm{c}})\\
&\quad + \cos(m_2)\left[\sum_{n=1}^{\infty}2\mathrm{J}_{2n}(m_1)\cos(2n\omega_1 t)\right]\sin(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}})\\
&\quad + \cos(m_2)\left\{\sum_{n=0}^{\infty}2\mathrm{J}_{2n+1}(m_1)\sin[(2n+1)\omega_1 t]\right\}\cos(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}})\\
&\quad + d(t)\sin(m_2)\left[\sum_{n=1}^{\infty}2\mathrm{J}_{2n}(m_1)\cos(2n\omega_1 t)\right]\\
&\quad \mathrm{sgn}[\sin(\omega_{\mathrm{sc}}t+\theta_{\mathrm{sc}})]\cos(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}})\\
&\quad - d(t)\sin(m_2)\left\{\sum_{n=0}^{\infty}2\mathrm{J}_{2n+1}(m_1)\sin[(2n+1)\omega_1 t]\right\}\\
&\quad \mathrm{sgn}[\sin(\omega_{\mathrm{sc}}t+\theta_{\mathrm{sc}})]\sin(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}})
\end{aligned}\right\}
\end{aligned}
\tag{7}
$$

where $J_n(\cdot)$ is the nth order Bessel function and θ_c and θ_{sc} are random carrier and subcarrier phases, respectively, each uniformly distributed over 0 to 2π. The first term in this expression is the residual carrier component which is fully suppressed if the data modulation index m_2 equals $\pi/2$. The second term is the desired data-bearing component containing the telemetry information which needs to be demodulated. The third and the fourth terms contain the ranging information to be extracted separately, and the rest are from intermodulation of telemetry and ranging signals. Typically, on one hand, the ranging modulation index m_1 chosen is small (around $\frac{1}{2}$ or smaller) so that the power consumption by this ranging signal is relatively small. On the other hand, the data modulation index selected is large (close to its upper limit $\pi/2$) to ensure that only sufficient power goes to the residual carrier component and the rest of power is allocated solely for telemetry data transmission. The strategy of optimizing modulation indices for efficient power allocation is discussed in (8).

A QPSK phase-modulated telemetry signal is usually treated as a combination of two orthogonal BPSK signals. Mathematically, a general QPSK (or, more specifically, an unbalanced QPSK) signal takes the form

$$
\begin{aligned}
S_{\mathrm{T}}(t) &= \sqrt{\alpha P_{\mathrm{T}}}\sin\left(\omega_{\mathrm{c}}t + \sum_{i=1}^{M}m_{\mathrm{c},i}S_{\mathrm{c},i}(t)\right)\\
&\quad + \sqrt{(1-\alpha)P_{\mathrm{T}}}\cos\left(\omega_{\mathrm{c}}t + \sum_{i=1}^{N}m_{\mathrm{s},i}S_{\mathrm{s},}i(t)\right)
\end{aligned}
\tag{8}
$$

where α is the percentage of transmitted power in one of the two channels. When only one binary signal of NRZ format is transmitted on each channel, that is, $M = N = 1$, with the modulation indices $m_{s,1} = m_{c,1} = \pi/2$, the QPSK signal is rewritten as

$$
S_{\mathrm{T}}(t) = \sqrt{\alpha P_{\mathrm{T}}}S_{\mathrm{c},i}(t)\cos(\omega_{\mathrm{c}}t) - \sqrt{(1-\alpha)P_{\mathrm{T}}}S_{\mathrm{s},i}(t)\sin(\omega_{\mathrm{c}}t) \tag{9}
$$

a combination of two BPSK signals on two orthogonal basis functions. As long as the orthogonality is maintained, these two BPSK signals do not interfere with each other and the bandwidth efficiency, measured as how many bits of information transmitted over a unit bandwidth, of a QPSK signal is twice of a BPSK signal (5). Several variants of QPSK modulation, including the offset QPSK (OQPSK) and the minimum-shift-keying (MSK), are also commonly used in near-Earth space missions. A detailed description of these modulation schemes is not covered here.

Demodulation

Demodulation is the process of transforming received waveforms back into their original state by reversing the modulation procedure. After traveling through various types of media or channels, the received waveform is corrupted in many ways. For example, it is corrupted by receiver's internally generated noise which is typically modeled as an additive white Gaussian noise (AWGN), or externally introduced interference, such as multipath, fading. Hence, for demodulation, it is important to correctly estimate the vital parameters in the transmitted signal from the corrupted waveform and apply the locally generated reference signals to remove the modulation. The following simple example illustrates how an AWGN corrupted BPSK signal is demodulated. The received signal is given by

$$
\begin{aligned}
r(t) &= \sqrt{2P_{\mathrm{T}}}\sin\left(\omega_{\mathrm{c}}(t+\tau) + \left(\frac{\pi}{2}\right)\sum_{k=1}^{\infty}d_k P(t+\tau-kT) + \theta\right)\\
&\quad + n(t)
\end{aligned}
\tag{10}
$$

where τ is the random propagation delay, θ is a uniformly distributed (over 0 to 2π) carrier phase, and $n(t)$ is a noise modeled as an AWGN with a two-sided power spectral density level at $N_0/2$ W/Hz. For the signal of NRZ format, the phase-modulated signal is equivalent to the product of the carrier and the baseband binary data waveforms, which is rewritten as

$$
r(t) = \sqrt{2P_{\mathrm{T}}}\left[\sum_{k=1}^{\infty}d_k P(t+\tau-kT)\right]\cos(\omega_{\mathrm{c}}t+\theta_{\mathrm{c}}) + n(t) \tag{11}
$$

where $\theta_c = (\theta + \omega_c\tau)_{\mathrm{mod}\,2\pi}$ is the total carrier phase. To demodulate the carrier, the receiver needs to generate a local reference, say $\sqrt{2}\cos(\omega_c t + \hat{\theta}_c)$, where $\hat{\theta}_c$ is an estimate of θ_c. A low-pass filtered version of the product of the local carrier reference and the received signal becomes

$$
r'(t) = \sqrt{P_{\mathrm{T}}}\left[\sum_{k=1}^{\infty}d_k P(t+\tau-kT)\right]\cos(\phi_{\mathrm{c}}) + n'(t) \tag{12}
$$

where $\phi_c = \theta_c - \hat{\theta}_c$ is the phase error between the actual and the estimated carrier phases. For constant or at least slowly varying ϕ_c, the factor $\cos(\phi_c)$ represents a signal amplitude attenuation, which is inevitably translated into degradation in bit-error performance. To make a decision on each of the transmitted bits, say, the ith bit d_i, the resulting signal $r'(t)$ is sent to a matched filter whose operation is mainly to form a product of the input signal and a local replica of the pulse function followed by an integrate-and-dump (I&D) operation. The accurate timing estimate $\hat{\tau}$ is very important in this matched filter operation because an error renders integrating across two bits, which reduces the detected symbol (for coded system) or bit (for uncoded system) energy when adjacent bits are of opposite polarities and results in a higher probability of decision error.

Additional signal power degradation due to imperfect subcarrier synchronization, similar to the carrier case given here, is expected when a subcarrier is used. The power degradation resulting from each of the carrier, subcarrier, and symbol tracking operations is discussed later.

SYNCHRONIZATION

The process of estimating the phase and timing parameters from the incoming noise-corrupted signal and using this information to keep the locally generated reference signal aligned with these estimates and, therefore, with the incoming signal is called synchronization.

As indicated previously, coherent reception and demodulation require phase information about the carrier and subcarrier (if used) and also symbol-timing information. This information must be provided and updated for coherent receivers all the time because they are usually time-varying with the changing characteristics of the channel. Therefore, individual tracking loops which continuously update their estimates of specific parameters are required to track and provide the needed information for a coherent receiver.

Although the tracking of carrier, subcarrier, and symbol timing are individually discussed in the following, one should keep in mind that, strictly speaking, all these loops are effectively coupled together in the sense that no one achieves lock without help from others, except for residual carrier tracking in which a carrier tone is separately tracked. However, in practice, each loop's performance is usually analyzed independently to keep the problem manageable.

It is also important to know that all the tracking loops discussed later are motivated by the maximum *a posteriori* (MAP) estimation which suggests only the open-loop structure of a one-shot estimator. The closed-loop structure derived by differentiating the likelihood function and equating the resulting loop feedback signal (also known as the error signal) to zero is only motivated by the MAP estimation (8).

Carrier Tracking

BPSK. The most commonly used device to track the phase of a sinusoidal signal, for example, the residual carrier component in Eq. (7), is the phase-locked loop (PLL). The PLL is composed of a phase detector, a loop filter, and a voltage-controlled oscillator (VCO) or, in a digital PLL design, a numerically controlled oscillator (NCO). The low-pass component of the phase detector output is a periodic function (of period 2π) of the phase error ϕ_c, which is called the S-curve of the loop. A stable lock point exists at $\phi_c = 0$, one of the zero-crossing points where the S-curve has a positive slope. The phase error for the first-order PLL, which has its loop filter implemented as a constant gain, is a Tikhonov distributed random variable and its probability density function (pdf) is given by (5)

$$p(\phi_c) = \frac{\exp[\rho_{\phi_c}\cos(\phi_c)]}{2\pi \mathrm{I}_0(\rho_{\phi_c})} \quad |\phi_c| \le \pi \tag{13}$$

where $\mathrm{I}_k(\cdot)$ denotes the modified Bessel function of order k and $\rho_{\phi_c} = (\sigma^2_{\phi_c})^{-1}$ is the loop SNR defined as the reciprocal of the phase error variance in radian^2. The loop SNR for the first-order PLL is expressed by

$$\rho_{\phi_c} = \frac{P_c}{N_0 B_L} \tag{14}$$

where B_L is the loop bandwidth. The detailed description of a PLL is discussed in another article in this encyclopedia and is not to be repeated here.

For the suppressed carrier in which no discrete carrier component appears in its spectrum, the carrier phase, embedded in the data-bearing component as the second term of Eq. (7), must be tracked by the Costas loop. The Costas loop is a phase-tracking loop whose functionality is similar to that of a PLL. Except for the same feedback path comprised of a loop filter and an NCO, a Costas loop has a double-arm loop structure, denoted, respectively, as the in-phase (I) and quadrature (Q) arms, with a phase detector and a low-pass arm filter in each. The incoming signal is first mixed with each of the two locally generated reference signals 90° apart, that is, $\sqrt{2}\sin(\omega_c t + \hat{\theta}_c)$ and $\sqrt{2}\cos(\omega_c t + \hat{\theta}_c)$, at the corresponding phase detector and then passed through the arm filter. Although the low-pass arm filter is either a passive RC-type filter or an active filter, it turns out that a matched filter (that is, an active filter) is the optimal design. The output of the two arm filters are multiplied, which effectively removes the data modulation, to form the loop feedback signal before it is fed into the loop filter. Because of this multiplication, a Costas loop is actually tracking twice the error phase. Accordingly, the Costas loop has two equally stable lock points at $\phi_c = 0$ and $\phi_c = \pi$, each corresponding to a zero-crossing point in the S-curve (of period π) where the slope is positive. These dual lock point inevitably introduce phase ambiguity such that the demodulated data has inverted polarity if the loop locks at $\phi_c = \pi$. This 180° phase ambiguity is resolved in several ways. For example, a known sequence pattern is inserted in the transmitted symbol stream from time to time so that the inverted polarity is detected by examining the received sequence pattern. However, the most efficient method is employing a differential encoding scheme in the transmitted data so that the information is kept in the relative phase between adjacent symbols instead of in the absolute phase of each symbol (5). On the receiver side, a corresponding differential decoding scheme is applied to extract the relative phase (or the transmitted information) after the symbol decision. A small penalty in terms of error performance exists for this differential encoding/decoding scheme because one incorrect symbol decision creates two consecutive errors in the relative phase.

The phase error for the first-order Costas loop with I&D arm fitters is similarly found as a Tikhonov distributed random variable and its pdf is given by

$$p(\phi_c) = \frac{\exp\left[\dfrac{\rho_{\phi_c}}{4}\cos(2\phi_c)\right]}{\pi \mathrm{I}_0\left(\dfrac{\rho_{\phi_c}}{4}\right)} \quad |\phi_c| \le \frac{\pi}{2} \tag{15}$$

and the associated loop SNR is given by

$$\rho_{\phi_c} = \frac{P_d}{N_0 B_L}\left(1 + \frac{1}{2E_s/N_0}\right)^{-1} \tag{16}$$

where $E_s/N_0 = P_d T/N_0$ is the symbol SNR. Note that the term in the parentheses is usually called the squaring loss, which results from the signal-noise product in the loop feedback signal. At low symbol SNR, the squaring loss is significant.

As discussed previously, transmitted power is allocated to the residual carrier component and data-bearing component by the choice of the modulation index for the telemetry data. It has been proved that a fully suppressed carrier is the best way to maximize data throughput (9). However, if a residual carrier component is desired for purposes other than communications, it is always a dilemma to set this modulation index because, on one hand, sufficient power must be given to the residual carrier so that it is successfully tracked by a PLL, and, on the other hand, the power allocated for data transmission should be kept as high as possible to maximize data throughput. Because the residual carrier and data sidebands are coherently related, a hybrid loop (10) which consists of the phase-locked loop and Costas loop structures is used to exploit this coherence and thereby improve carrier phase tracking in this scenario. This technique is also known as sideband aiding because it utilizes the power in the data-bearing component as the second term of Eq. (7) to help residual carrier tracking.

In the hybrid loop, both error signals from the single-arm PLL structure and the double-arm Costas loop structure are weighted and added together to form an effective loop feedback signal. As a result, there are usually dual lock points for the hybrid loop, that is, $\phi_c = 0$ and $\phi_c = \pi$ and similar to those of a Costas loop. Yet, these two lock points generally are not equiprobable. It can be shown that (11) the lock point at $\phi_c = \pi$ vanishes when the modulation index is smaller than a threshold as a function of the symbol SNR.

With a given modulation index, an optimal relative weight between the PLL and Costas loop portion is derived to minimize the hybrid-loop tracking jitter. Because the relative tracking performance between a PLL and Costas loop is determined by the relative power allocation and the additional squaring loss incurred in the Costas loop, it is not surprising to find that the optimal weight is a function of the modulation index and the symbol SNR.

QPSK. The carrier tracking of a QPSK signal is usually done by a generalized Costas loop motivated by MAP estimation theory. There are basically two variants of this generalized Costas loop: the polarity-type known as the crossover Costas loop for high SNR scenarios and the squaring-type for low SNR scenarios (7). In the crossover loop, two products are formed by multiplying the hard-limited version of one arm-filter output with the other arm-filter output before they are combined as the loop error feedback. The phase error for the first-order crossover Costas loop is a Tikhonov distributed random variable with pdf given by

$$p(\phi_c) = \frac{2\exp\left[\frac{\rho_{\phi_c}}{16}\cos(4\phi_c)\right]}{\pi \mathrm{I}_0\left(\frac{\rho_{\phi_c}}{16}\right)} \quad |\phi_c| \leq \frac{\pi}{4} \tag{17}$$

The associated loop SNR for this first-order cross-over loop is

$$\rho_{\phi_c} = \frac{P_d}{N_0 B_L} \left\{ \frac{\left[\mathrm{erf}\left(\sqrt{\frac{E_s}{2N_0}}\right) - \sqrt{\frac{2}{\pi}\left(\frac{E_s}{N_0}\right)}\exp\left(-\frac{E_s}{2N_0}\right)\right]^2}{1 + \frac{E_s}{N_0} - \left[\sqrt{\frac{2}{\pi}}\exp\left(-\frac{E_s}{2N_0}\right) + \sqrt{\frac{E_s}{N_0}}\mathrm{erf}\left(\sqrt{\frac{E_s}{2N_0}}\right)\right]^2} \right\} \tag{18}$$

where the error function erf(·) and its complimentary erfc(·) are defined as

$$\mathrm{erf}(x) = 1 - \mathrm{erfc}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-\alpha^2}\, d\alpha$$

Note that, similar to Eq. (16) of the Costas loop, the term in the braces in Eq. (18) is the squaring loss of the crossover Costas loop. The squaring-type loop has a different squaring loss which is smaller than that of its polarity-type sibling in low SNR region. More details about the crossover Costas loop can be found in (13). Another alternative in tracking a QPSK signal is to use the demod-remod quadriphase tracking loop which can be viewed as a fourth-power loop with a multiplication done at the IF level (7).

Subcarrier Tracking

Subcarrier tracking is almost identical to suppressed carrier tracking for BPSK signals because there is no residual tone left for the binary-phase-shift-keyed subcarrier. The Costas loop is used here to remove the data modulation and a squaring loss associated with this process is inevitable. However, depending on the use of a sine-wave or square-wave subcarrier, tracking performance is quite different. For the sine-wave subcarrier, there is no difference between its tracking and that of a suppressed carrier. On the contrary, additional improvement in the square-wave subcarrier tracking is realized by using a time-domain windowing function on the quadrature arm (13). In this case, the windowing function around the midphase transition of the Q-arm reference signal is treated as an approximation of the time-domain derivative of its I-arm counterpart. According to the derivation of the MAP estimation, which implies the existence of an optimal open-loop structure when one of the I-arm and Q-arm reference signals is the derivative of the other, the resulting loop SNR is greatly improved by shrinking the window size. The first-order loop SNR is given by

$$\rho_{\phi_{sc}} = \left(\frac{2}{\pi}\right)^2 \frac{P_d}{N_0 B_L}\left(\frac{1}{W_{sc}}\right)\left(1 + \frac{1}{2E_s/N_0}\right)^{-1} \tag{19}$$

where W_{sc} is the quadrature window size (between 0 and 1) relative to a subcarrier cycle. It is clear that the loop SNR is inversely proportional to the window size. However, using a small window inevitably reduces the loop's pull-in range and raises the issue of loop stability. A reasonable window size of one-quarter or one-eighth is usually used to provide a 6 to 9 dB improvement in loop SNR.

No such improvement from quadrature windowing is realized for a sine-wave Costas loop of which the I-arm and Q-arm reference signals are two sine functions separated by 90° and, therefore, have the derivative relationship between them as suggested by the MAP estimation. Applying a quadrature window in this case actually destroys the derivative relationship and renders inferior loop performance.

Symbol-Timing Tracking

Symbol synchronization has a direct impact on the data detection process because inaccurate symbol timing reduces the probability of making a correct decision. Although a separate channel may be used to send timing signals for synchronization, to extract the synchronization information directly from the data-bearing signal has the advantage that no additional power and frequency spectrum are required. Of course, to successfully extract symbol timing information from the transmitted symbol stream relies on the presence of adequate symbol transitions (zero-crossings).

The data transition tracking loop (DTTL) is widely used for symbol synchronization. Similar to the Costas loop, DTTL has a double-arm loop structure with a hard decision followed by a transition detector in its in-phase arm and a delay in its quadrature arm to keep signals on both arms properly aligned. It is important to note that the term "in-phase" refers to an operation synchronous with the timing of the received symbols and, therefore, the I-arm phase detector becomes a matched filter integrating from one symbol epoch to the next. The Q-arm phase detector performs another integration within a window, which is of a size W_{sym} (between 0 and 1) relative to the symbol interval and centered at the symbol epoch, causing the midpoint of the Q-arm integration interval offset by a half-symbol from its I-arm counterpart. Similar to square-wave subcarrier tracking, the time-domain windowing function on the quadrature arm improves the tracking performance but inevitably raises the issue of loop stability at the same time (5).

The DTTL has a single stable lock point at $\phi_{\mathrm{sym}} = 0$. The phase error for the first-order DTTL is a Tikhonov distributed random variable and its pdf is given by

$$p(\phi_{\mathrm{sym}}) = \frac{\exp(\rho_{\phi_{\mathrm{sym}}}\cos(\phi_{\mathrm{sym}})]}{2\pi \mathrm{I}_0(\rho_{\phi_{\mathrm{sym}}})} \qquad |\phi_{\mathrm{sym}}| \le \pi \tag{20}$$

with the corresponding loop SNR given as

$$\rho_{\phi_{\mathrm{sym}}} = \frac{1}{(2\pi)^2}\frac{P_{\mathrm{d}}}{N_0 B_{\mathrm{L}}}\left(\frac{1}{W_{\mathrm{sym}}}\right) \left\{ \frac{2\left[\mathrm{erf}\left(\sqrt{\frac{E_{\mathrm{s}}}{N_0}}\right) - \frac{W_{\mathrm{sym}}}{2}\sqrt{\frac{1}{\pi}\left(\frac{E_{\mathrm{s}}}{N_0}\right)}\exp\left(-\frac{E_{\mathrm{s}}}{N_0}\right)\right]^2}{1 + \frac{W_{\mathrm{sym}}}{2}\left(\frac{E_{\mathrm{s}}}{N_0}\right) - \frac{W_{\mathrm{sym}}}{2}\left[\sqrt{\frac{1}{\pi}}\exp\left(-\frac{E_{\mathrm{s}}}{N_0}\right) + \sqrt{\frac{E_{\mathrm{s}}}{N_0}}\mathrm{erf}\left(\sqrt{\frac{E_{\mathrm{s}}}{N_0}}\right)\right]^2} \right\} \tag{21}$$

SYMBOL SNR DEGRADATION

Symbol SNR degradation is the direct cause of poor bit-error performance and is translated into the telemetry system loss as seen in the next section when the receiver performs a hard decision on each demodulated symbol. When no hard decision is performed by the receiver, symbol SNR degradation directly affects decoder performance because the demodulated symbols, called soft symbols, are fed to the decoder without going through a hard-decision device.

Because of the difficulty of analyzing the coupled-carrier, subcarrier, and symbol-tracking loops, SNR degradation of the demodulated symbol (for a coded system) or bit (for an uncoded system) caused by imperfectly synchronized references is usually approximated as a product of degradation factors of the individual loops, each factor being derived on the basis of assuming perfect tracking in other loops. The overall degradation, conditioned on the corresponding phase errors, for the telemetry signal given in Eq. (7) is expressed by

$$\begin{aligned} D_{\mathrm{SNR}}(\phi_{\mathrm{c}}, \phi_{\mathrm{sc}}, \phi_{\mathrm{sym}}) &= D_{\mathrm{c}}(\phi_{\mathrm{c}}) D_{\mathrm{sc}}(\phi_{\mathrm{sc}}) D_{\mathrm{sym}}(\phi_{\mathrm{sym}}) \\ &= [\cos(\phi_{\mathrm{c}})]^2 \left[1 - \frac{2}{\pi}|\phi_{\mathrm{sc}}|\right]^2 \left[1 - \frac{|\phi_{\mathrm{sym}}|}{\pi} + \frac{\phi_{\mathrm{sym}}^2}{2\pi^2}\right] \end{aligned} \tag{22}$$

where $D_{\mathrm{c}}(\phi_{\mathrm{c}})$, $D_{\mathrm{sc}}(\phi_{\mathrm{sc}})$ and $D_{\mathrm{sym}}(\phi_{\mathrm{sym}})$ are the degradation factors for the imperfect carrier, subcarrier, and symbol (or bit) synchronization, respectively. Hence, the averaged symbol (or bit) SNR degradation due to imperfect synchronization becomes

$$(\overline{D_{\mathrm{SNR}}})_{\mathrm{dB}} = (\overline{D_{\mathrm{c}}})_{\mathrm{dB}} + (\overline{D_{\mathrm{sc}}})_{\mathrm{dB}} + (\overline{D_{\mathrm{sym}}})_{\mathrm{dB}} \tag{23}$$

where $\overline{D_{\mathrm{c}}}$, $\overline{D_{\mathrm{sc}}}$ and $\overline{D_{\mathrm{sym}}}$ are the averaged power degradation factors obtained by averaging over the corresponding Tikhonov distributed phase errors.

In addition to the degradation caused by imperfect synchronization, it is also important to know that there are other sources of SNR degradation, for example, the intermodulation terms and possible interference from the ranging signal found in Eq. (7), and the subcarrier and symbol waveform distortion introduced by the bandlimited channel.

BIT-ERROR PERFORMANCE (UNCODED SYSTEM) AND TELEMETRY SYSTEM LOSS

Telemetry information is extracted from the demodulated data stream by a symbol decision process. For binary signals, it is typically a hard-limiting decision on an AWGN corrupted, antipodal, random variable. For an uncoded system, the bit- (or symbol) error probability of a BPSK signal is well known as

$$P_{\mathrm{b}} = \int_{-\pi}^{\pi} \frac{1}{2}\mathrm{erfc}\left[\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\cos(\phi_{\mathrm{c}})\right] p(\phi_{\mathrm{c}})\, d\phi_{\mathrm{c}} \tag{24}$$

where $p(\phi_{\mathrm{c}})$, is the pdf of the carrier phase error given in Eq. (13), when the carrier is tracked by a PLL so that no phase ambiguity exists. However, with a fixed-loop SNR, an irreducible error probability exists no matter how large the bit SNR. This irreducible error probability is characterized solely by the carrier tracking loop SNR and, for a given loop bandwidth, is reduced only by allocating more power to the residual carrier component which serves no purpose in transmitting telemetry except being tracked by PLL.

For suppressed carrier tracking of a BPSK signal by the Costas loop, the phase ambiguity exists and must be resolved. The bit-error probability for a special case of perfect phase ambiguity resolution (say, by other means, such as a periodically inserted known sync pattern) is given by

$$P_{\mathrm{b}} = \int_{-\pi/2}^{\pi/2} \frac{1}{2}\mathrm{erfc}\left[\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\cos(\phi_{\mathrm{c}})\right] p(\phi_{\mathrm{c}})\,d\phi_{\mathrm{c}} \tag{25}$$

where $p(\phi_{\mathrm{c}})$ is the pdf in Eq. (15).

If a differential coding scheme is utilized to resolve the phase ambiguity, the bit-error probability becomes (7)

$$P_{\mathrm{b}} = \int_{-\pi/2}^{\pi/2} \mathrm{erfc}\left[\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\cos(\phi_{\mathrm{c}})\right] \left\{1 - \frac{1}{2}\mathrm{erfc}\left[\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\cos(\phi_{\mathrm{c}})\right]\right\} p(\phi_{\mathrm{c}})\,d\phi_{\mathrm{c}} \tag{26}$$

where $p(\phi_{\mathrm{c}})$ is the pdf in Eq. (15). No irreducible error probability exists in the suppressed carrier tracking because the tracking loop SNR for a fixed loop bandwidth and bit duration product increases with the bit SNR.

So far, only the impact of bit-error probability from carrier tracking has been discussed and one can find the SNR degradation from imperfect carrier tracking, that is, $\cos^2(\phi_{\mathrm{c}})$ appears repeatedly in Eqs. (24)–(26). When the overall impact of bit-error performance from all levels of imperfect tracking, including carrier, subcarrier, and symbol, is considered, the bit-error probability becomes a threefold integration involving the overall symbol SNR degradation given in Eq. (22). For example,

$$P_{\mathrm{b}} = \int_{-\pi}^{\pi}\int_{-\pi/2}^{\pi/2}\int_{-\pi/2}^{\pi/2} \frac{1}{2}\,\mathrm{erfc}\left[\sqrt{\frac{E_b}{N_0}D_{\mathrm{SNR}}(\phi_{\mathrm{c}}, \phi_{\mathrm{sc}}, \phi_{\mathrm{sym}})}\right] p(\phi_{\mathrm{c}})p(\phi_{\mathrm{sc}})p(\phi_{\mathrm{sym}})\,d\phi_{\mathrm{c}}\,d\phi_{\mathrm{sc}}\,d\phi_{\mathrm{sym}} \tag{27}$$

where $p(\phi_{\mathrm{sc}})$ and $p(\phi_{\mathrm{sym}})$ are Tikhonov distributed pdfs of subcarrier and symbol phase errors, respectively. In fact, $p(\phi_{\mathrm{sc}})$ takes the form of Eq. (15) of the Costas loop and $p(\phi_{\mathrm{sym}})$ is given by Eq. (20). Note that the product of pdfs of individual phase errors is used in lieu of the hard-to-establish joint pdf from the coupled loops.

For QPSK signals, the bit-error probability is given by (7)

$$P_{\mathrm{b}} = \int_{-\pi/4}^{\pi/4} \left(\frac{1}{4}\mathrm{erfc}\left\{\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\left[\cos(\phi_{\mathrm{c}}) - \sin(\phi_{\mathrm{c}})\right]\right\} + \frac{1}{4}\mathrm{erfc}\left\{\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\left[\cos(\phi_{\mathrm{c}}) + \sin(\phi_{\mathrm{c}})\right]\right\}\right) p(\phi_{\mathrm{c}})\,d\phi_{\mathrm{c}} \tag{28}$$

where $p(\phi_{\mathrm{c}})$ is the pdf in Eq. (17).

The telemetry system loss is defined as a loss factor $L \geq 0$ dB, which represents the amount of additional bit SNR required for a lossy system to meet the same bit-error performance of a perfectly synchronized system. For example, the bit-error probability for a perfectly synchronized BPSK or QPSK system is given by

$$P_{\mathrm{b}} = \frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_{\mathrm{b}}}{N_0}}\right) \tag{29}$$

and, therefore, the required bit SNR for this ideal system at a given bit-error probability p_{b}^* is expressed by

$$\left(\frac{E_{\mathrm{b}}}{N_0}\right)^* = [\mathrm{erfc}^{-1}(2P_{\mathrm{b}}^*)]^2 \tag{30}$$

The lossy system with a system loss L needs L times as much bit energy, namely, $E_{\mathrm{b}}/N_0 = L(E_{\mathrm{b}}/N_0)^*$, to achieve the same bit-error probability or, in other words, to compensate for the loss incurred within.

ADVANCED TOPICS

Antenna Arraying

With recent space missions moving toward high data rate and low transmitting power operations, combining signals from several antennas to improve the effective SNR becomes the only viable option when the existing technologies of building larger single-aperture antenna and lowering the system noise temperature are pushed to their limits.

Three arraying techniques (14) are briefly discussed here: symbol-stream combining, baseband combining and full spectrum combining, each combining signals at a different stage of signal processing.

For symbol-stream combining, each participating antenna performs carrier, subcarrier, and symbol synchronization individually. Then the symbols at each receiver output are combined, with the appropriate weights, to form the final symbols for detection or decoding. This scheme has the advantage of a small combining loss. It is also suitable for real-time combining from intercontinental antenna sites because combining is performed at a relatively low processing rate, that is, the symbol rate. The disadvantage is that each antenna needs a full set of receiver hardware and must lock on the signal individually.

In baseband combining, each antenna needs to lock on and remove the (residual) carrier by itself. Then the resulting baseband signals, including the data-modulated subcarrier, are combined for further synchronization and demodulation. The advantage of this scheme is that less hardware is required because only a single set of subcarrier and symbol tracking devices is needed to process the combined signal. The disadvantage is that each antenna still must lock on, at least, the carrier, individually.

In full spectrum combining, signals are combined at an intermediate frequency (IF). Before they are combined, the relative time delay and phase difference must be properly estimated and compensated for so that signals are coherently combined. Then the resulting IF signal is directed to a single receiver for further synchronization and demodulation. The advantage of this scheme is that only one of the participating antennas must lock on the combined IF signal, which allows including smaller antennas in this arraying scheme even though they cannot lock on the signal. The disadvantage is the very large transmission or recording bandwidth required to carry the IF signals through the networked antenna sites for combination.

Besides the above-discussed arraying techniques, a scheme called the carrier arraying, which employs coupled carrier tracking devices from participating antennas should be mentioned here. This scheme by itself does not combine the signals and, thus, must be operated with symbol-stream or baseband combining to array the telemetry. In a carrier array scenario, a large master antenna generally locks on the signal by itself and then helps other smaller antennas to track by estimating and removing the signal dynamics in their input.

Buffered Telemetry Processing

The Deep Space Communications Complex (DSCC) Galileo Telemetry (DGT) is developed and implemented by the Jet Propulsion Laboratory to support the Galileo S-Band Mission. Many advanced technologies have been developed for this mission to cope with the failure to fully deploy the high-gain antenna of the Galileo spacecraft, making itself a showcase of future signal processing technologies in the radiotelemetry field. In the following, selected key features of the DGT and the technologies behind these features are briefly described to illustrate the concept of buffered telemetry processing in which telemetry is recorded, processed, and re-processed to minimize data loss in space missions operated with low link margins.

DGT is composed of four major subsystems, the full spectrum recorder (FSR), the full spectrum combiner (FSC), the buffered telemetry demodulator (BTD), the feedback concatenated decoder (FCD), and other control functions to coordinate the operations of these subsystems. Except for the FSR, the rest of the DGT is implemented in software and can be run on general-purpose workstations, which allows greater flexibility of signal processing without expensive custom-made hardware.

The FSR downconverts the RF signal to IF for digitization and then further open-loop downconverts each data sideband to the baseband, individually and coherently, before it is sampled and recorded. This significantly reduces the required bandwidth for transmission through the intercontinental antenna network because the processing rate is linked to the symbol rate, instead of the much higher subcarrier frequency.

The recorded signals (residual carrier and data sidebands centered at the first four harmonics of the square-wave subcarrier are kept for the Galileo S-Band Mission) from arrayed antennas are combined by the FSC, which estimates and adjusts the time delay and phase for each recorded sideband coherently to a reference point chosen as the center of the Earth, and then combines the time- and phase-aligned signals from arrayed antennas to form an enhanced signal. The combined telemetry is archived and transferred to the BTD upon request for synchronization and demodulation.

The BTD, known as the software receiver, is the signal processing core of DGT, which provides acquisition, synchronization, demodulation, and miscellaneous monitoring functions through its carrier, subcarrier, symbol-tracking loops and associated lock indicators (15). In the BTD, the individually combined data sidebands are processed coherently and then are synthesized to form an equivalent signal as if it were a single signal processed by a regular receiver. The end product of the BTD is a demodulated soft symbol stream which is written to a file and transferred to the FCD upon request for decoding and decompression.

Because the FSR/FSC combined data are recorded on tape, the BTD is actually designed to be able to work on any segment of data off-line in either direction, forward or backward in time. In fact, with the availability of multiple CPU workstations, simultaneous BTD sessions are initiated on different segments of data. For example, one session is dedicated to process real-time samples forward (in time) while the others reprocess other recorded segments as needed. Then the soft symbol streams from these simultaneous sessions are merged into a single stream because each of them is properly time tagged. By taking advantage of the flexibility in software implementation, many noncausal signal processing techniques can be performed to process or reprocess the data to further enhance the quality of the telemetry. One important feature of the BTD is the so-called gap-closure processing (16) which greatly reduces possible data loss due to receiver acquisition, resynchronization, and loss of lock.

The need to reprocess a segment of sampled data arises from the failure of the BTD to maintain the in-lock status in any of its tracking loops or the failure of the FCD to properly decode the soft symbols. A segment of sampled data on which the telemetry cannot be extracted reliably is called a gap, and the processing of a gap to extract any valid information not available when that segment of samples was first processed is called gap-closure processing. Gaps caused by acquisition are found in the beginning of each pass or at instants when the receiver drops out of lock, whereas gaps generated by cycle slips in one of the loops occur randomly in a pass. Along with its demodulation efforts, the BTD tracks its internal states, including the lock indicators, the symbol SNR estimates, and the state variables inside the loop filter and the NCO for all three loops. These state variables are recorded at fixed intervals as checkpoints and, with them, a software receiver is easily restored to its state at a checkpoint immediately before or after a gap. By estimating the parameters of a phase process in a region near a restored checkpoint where the phase tracking was successfully carried out, gap closure processing can start from this checkpoint and move into the gap. Two configurations, one for closed-loop and the other for open-loop, are used here. The closed-loop configuration needs to initiate the loop filter with phase parameter estimates in a particular way, so that, when the loop is closed and starts to track at the checkpoint, the loop virtually starts immediately with steady state tracking. For a relatively stable phase process and a gap of small size, an open-loop configuration is applied by using an estimated phase profile as the reference without resorting to a loop operation. Both configurations are applied to gap-closure processing in either direction, forward or backward in time, because the buffered data can be processed in either order. This is especially useful when a gap occurs at the beginning of a track so that all of the available checkpoint information is from the region behind this gap.

Another useful feature of BTD is its capability of seamless tracking through symbol rate changes. The reason for changing the symbol rate during a pass is to take advantage of the changing G/T figure as the elevation angle of an antenna changes in a pass. With a higher elevation angle, an antenna has a higher G/T figure and supports a higher symbol rate when the symbol SNR is fixed. The software implementation of BTD handles symbol rate changes without dropping a lock on symbol timing, as long as the rate changes follow a set of specific rules and their schedule is roughly known in advance.

The FCD is a subsystem that performs error-correction decoding and data decompression in the DGT. Implemented in software on a multiprocessor workstation, it employs a feedback mechanism that passes intermediate decoding information from the outer code of the concatenated code to the inner code to facilitate multipass decoding which achieves a final bit error rate of 10^{-7} at a 0.65 dB bit SNR. The architecture and the detailed operations of the FCD are described in the next section.

Advanced Source and Channel Coding for Space Applications

In this section, we use the Galileo S-Band Mission again as an example to illustrate the application of advanced source and channel coding schemes to enhance telemetry return (17). First, using the integer cosine transform (ICT) for lossy image compression is briefly explained. Then, an advanced error-correction coding scheme used to protect the heavily edited and compressed data is discussed, followed by a discussion of the interaction between data compression and error-control (containment/detection/correction) processes.

Galileo's Image-Compression Scheme. Galileo image compression is a block-based lossy image-compression algorithm that uses an 8 × 8 ICT. The ICT was first proposed in (18), and was streamlined and generalized in (19,20). It is an integral approximation of the popular discrete cosine transform (DCT), which is regarded as one of the best transform techniques in image coding. Its independence from the source data and the availability of fast transform algorithms make the DCT an attractive candidate for many practical image processing applications. In fact, the ISO/CCITT standards for image processing in both still-image and video transmissions include the two-dimensional DCT as a standard processing component in many applications.

The elements in an ICT matrix are small integers with sign and magnitude patterns resembling those of the DCT matrix. Besides, the rows of the ICT matrix are orthogonal. The integral property eliminates real multiplication and real addition operations, thus greatly reducing computational complexity. The orthogonality property ensures that the inverse ICT has the same transform structure as the ICT. Notice that the ICT matrix is only required to be orthogonal, but not orthonormal. However, any orthogonal matrix may be made orthonormal by multiplying it by an appropriate diagonal matrix. This operation is incorporated in the quantization (dequantization) stage of the compression (decompression), thus sparing the ICT (inverse ICT) from floating-point operations and, at the same time, preserving the same transform structure as in the floating-point DCT (inverse DCT). The relationship between the ICT and DCT guarantees efficient energy packing and allows the use of fast DCT techniques for the ICT. The ICT matrix used in the Galileo mission is given as follows:

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 3 & 2 & 1 & -1 & -2 & -2 & -5 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 3 & -1 & -5 & -2 & 2 & 5 & 1 & -3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 2 & -5 & 1 & 3 & -3 & -1 & 5 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -5 & 5 & -3 & 2 & -1 \end{matrix}$$

Figure 1 shows the rate-distortion performance of the ICT scheme compared with the JPEG scheme. Simulation results indicate that the difference in performance between the use of floating-point DCT and the ICT is insignificant.

Galileo's Error-Correction Coding Scheme. The Galileo error-correction coding scheme uses a (255,k) variable redundancy RS code as the outer code and a (14,1/4) long constraint-length convolutional code as the inner code. The RS codewords are interleaved to depth 8 in a frame. The redundancy profile of the Reed–Solomon codes is (94, 10, 30, 10, 60, 10, 30, 10). The staggered redundancy profile was designed to facilitate the novel feedback concatenated decoding strategy (21,22). This strategy allows multiple passes of channel symbols through the decoder. During each pass, the decoder uses the decoding information from the RS outer code to facilitate the Viterbi decoding of the inner code in a progressively refined manner. The FCD is implemented in software on a multiprocessor workstation. The code is expected to operate at a bit signal-to-noise ratio of 0.65 dB at a bit error rate of 10^{-7}. Figure 2 shows a schematic of the FCD architecture. In this article, only the implementation and operational aspects of the FCD task are discussed. The FCD novel node/frame synchronization scheme is discussed in (23) and its code selection and performance analysis are discussed in detail in (24).

The (255,k) Variable-Redundancy Reed–Solomon Code. All RS codes for the Galileo mission use the same representation of the finite field GF(256). Precisely, GF(256) is the set of elements

$$\mathrm{GF}(256) = \{0, a^0, a^1, a^2, \cdots, a^{254}\} \tag{31}$$

where a, by definition, is a root of the primitive polynomial

$$p(x) = x^8 + x^7 + x^2 + x + 1 \tag{32}$$

[i.e., $p(a) = 0$].

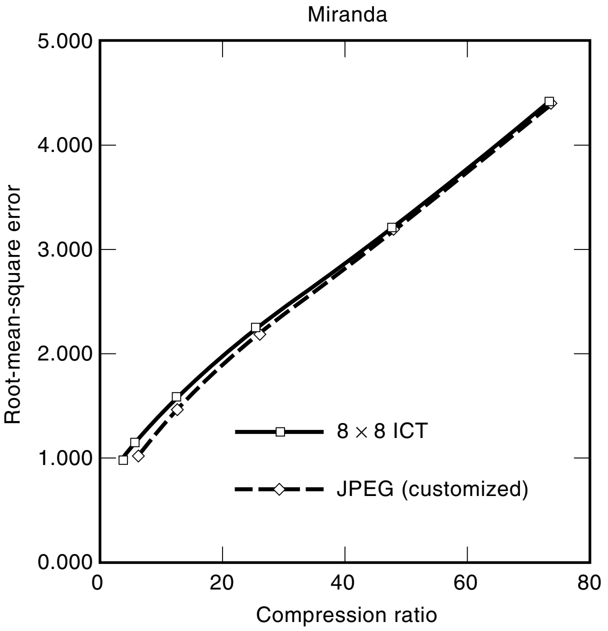

Figure 1. Rate-distortion performance of ICT.

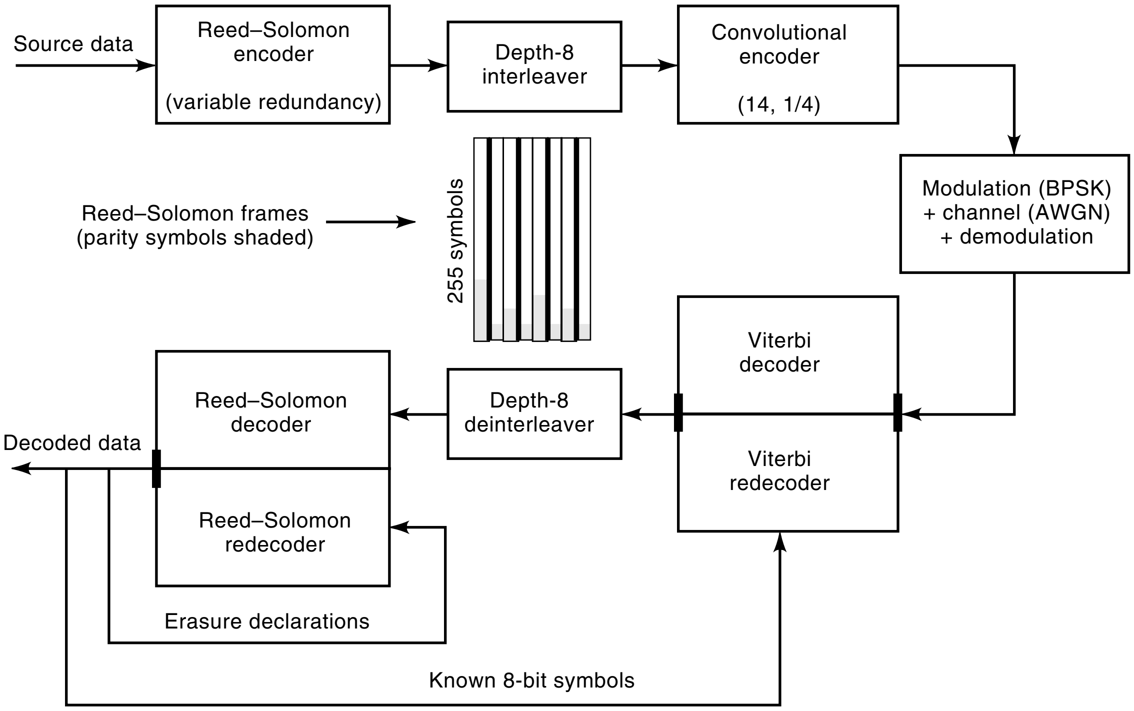

Figure 2. Schematic of the FCD.

In the encoding/decoding process, each power of α is represented as a distinct nonzero 8-bit pattern. The zero byte is the zero element in GF(256). The basis for GF(256) is descending powers of α. Note that this is the conventional representation, not Berlekamp's dual basis (25). The RS generator polynomial is defined as

$$g(x) = \prod_{i=0}^{n-k-1} (x - \alpha^{\beta(i+L)}) = \sum_{i=0}^{n-k} g_i x^i \tag{33}$$

where n denotes the codeword length in bytes, k denotes the number of information bytes, and α^b is a primitive element of GF(256). The parameter b is chosen in some applications to minimize the bit-serial encoding complexity. Because the Galileo RS encoders are implemented in software, there is little advantage in preferring a particular value of b. The parameter L is chosen so that the coefficients of $g(x)$ are symmetrical. This reduces the number of Galois field multiplications in encoding by nearly a factor of 2.

The Galileo mission utilizes four distinct RS codes. We define RS(n,k) as an RS code which accepts k data bytes as input and produces n bytes as a code word, where $n > k$. An RS(n,k) code corrects t errors and s erasures if $2t + s \leq n - k$. These codes are referred to as RS(255,161), RS(255,195), RS(255,225), RS(255,245). Specifically, the parameters b and L of these four codes are as follows:

• RS(255,161)	$b = 1, L = 81$
• RS(255,195)	$b = 1, L = 98$
• RS(255,225)	$b = 1, L = 113$
• RS(255,245)	$b = 1, L = 123$

These RS codes, being interleaved to depth 8, are arranged in a transfer frame as shown in Fig. 2. The RS decoders use a time-domain Euclid algorithm to correct errors and erasures. The details of the decoding algorithm are discussed in (26).

The (14,1/4) Convolutional Code and Its Parallel Viterbi Decoder. The (14,1/4) convolutional code used for the Galileo mission is the concatenation of a software (11,1/2) code and an existing hardware (7,1/2) code. The choice of convolution code is constrained by the existing (7,1/2) code, which is hardwired in the Galileo Telemetry Modulation Unit (TMU), and by the processing speed of the ground FCD. The generator polynomials of the (11,1/2) code and the (7,1/2) code in octal are (3403, 2423) and (133, 171), respectively. The generator polynomials of the equivalent (14,1/4) code are (26042, 36575, 25715, 16723).

The Viterbi decoder for the (14,1/4) code is implemented in software in a multiprocessor workstation with shared memory architecture. The use of a software decoder is possible because of the slow downlink data rate of the Galileo S-Band Mission. The advantages of a software-based decoder are that the development cost is low and it allows the flexibility to perform feedback concatenated decoding. We examined two different approaches to parallelize the Viterbi algorithm: (1) state-parallel decomposition in which each processor is equally loaded to compute the add-compare-select operations per bit and (2) round-robin frame decoding that exploits the multiple processors by running several complete but independent decoders for several frames in parallel. Our early prototypes indicate that the first approach requires a substantial amount of interprocessor synchronization and communication and this greatly reduces the decoding speed. The second approach requires much less synchronization and communication because each processor is now an entity independent of the others. The performance scaling is nearly perfect. We chose the round-robin approach for the FCD Viterbi decoder. The details of the FCD software Viterbi decoder implementation are described in (27).

Redecoding. Redecoding, as shown in Fig. 2, uses information fed back from code words successfully decoded by the RS decoder to improve the performance of Viterbi decoding. A correctly decoded RS bit forces the add-compare-select operation at each state to select the path that corresponds to the correct bit. Thus the Viterbi decoder is constrained to follow only paths consistent with known symbols from previously decoded RS code words. The Viterbi decoder is much less likely to choose a long erroneous path because any path under consideration is pinned to coincide with the correct path at the locations of the known symbols. Each RS frame is decoded with four feedback passes. In the first pass, only the first code word RS(255,161) is decoded. In the second pass, the fifth code word RS(255,195) is decoded. In the third pass, the third and seventh code words RS(255,225) are decoded, and finally, in the fourth pass, the second, fourth, sixth, and eighth code words RS(255,245) are decoded. During each pass, the decoder uses the decoding information from the Reed–Solomon outer code to facilitate Viterbi decoding of the inner code in a progressively refined manner. The details of the FCD redecoding analysis are given in (24).

Interaction Between Data Compression and Error Control Processes. Packet loss and other uncorrectable errors in a compressed data stream cause error propagation, and the way the error propagates depends on the compression scheme. To maximize the scientific objectives with the limited transmission power of the low-gain antenna used in the Galileo S-Band Mission, most of the data (image and nonimage) are expected to be heavily edited and compressed. These valuable compressed data must be safeguarded against catastrophic error propagation caused by packet loss and other unforeseeable errors.

The ICT scheme for solid-state imaging (SSI) data is equipped with a simple but effective error-containment strategy. The idea is to insert synchronization markers and counters at regular intervals to delimit uncompressed data into independent blocks so that, in case of packet loss and other anomalies, the decompressor searches for the next available synchronization marker and continues to decompress the rest of the data. In this case, the interval chosen is eight lines of uncompressed data. The error-containment strategy guarantees that error propagation does not go beyond the compressed code block where errors reside. Other options to prevent error propagation are also considered, but these options usually result in great onboard implementation complexity or excessive downlink overhead. For example, a self-synchronizing feature in Huffman code may be used to contain errors, but it is difficult to implement. A packetizing scheme with varying packet sizes may also be used to contain errors (by matching packet boundaries and the compressed data block boundary), but the packet headers introduce excessive downlink overhead in SSI data.

The SSI ICT error-containment scheme works as follows. On the compression side, every eight lines of data are compressed into a variable-length, compressed data block. The dc (steady-state bias) value is reset to zero at the start of each compressed data block, thus making every block independent of the others. A 25-bit synchronization marker and a 7-bit modulo counter are inserted at the beginning of every compressed data block. The sync marker is chosen to minimize the probability of false acquisition in a bursty channel environment. The 25-bit synchronization marker pattern is 024AAAB in hex. Simulation results indicate that this synchronization marker gives a probability of false acquisition of less than 10^{-8}. The decompression scheme consists of two program modules, the SSI ICT decompression module and the error detection/sync module. The SSI ICT decompression module reconstructs the data from the compressed data stream, and the error detection/sync module checks the prefix condition of the Huffman codes to detect any anomaly. When an anomaly is detected, a synchronization marker search is initiated to find the next available one. Decompression resumes from there on, and the reconstructed blocks are realigned by using the modulo counter. The corrupted portion of the data is flagged and reported.

The downlink overhead of the SSI ICT error-containment scheme is a function of compression ratio (CR) and image width W. It is measured by the percentage of sync data (sync marker and counter) compared to the compressed data and is given by the following equation:

$$\frac{4 \times \mathrm{CR}}{8 \times \mathrm{W}}$$

For example, an 800 × 800 SSI image has the following overhead as a function of the compression ratio:

Compression Ratio	Overhead
2	0.00125
4	0.00250
8	0.00500
16	0.01000

Multiple Spacecraft Support

Traditionally, every spacecraft is supported by one of the ground antennas for its uplink and downlink. This dedication requires efficient scheduling of the resources on the ground, including hardware, software, and personnel. With more and more concurrent missions, the need for multiple spacecraft support by a single ground antenna to alleviate the scheduling problem becomes evident. For example, several proposed future missions to Mars by various joint efforts of international space agencies will place more than a dozen spacecraft, including orbiters, landers, and rovers, on or around Mars in the next 10 years. For these missions, it is highly possible that more than one spacecraft will come within the same beam width of a single ground antenna, and it constitutes the opportunity to communicate with them by using this single antenna with a considerable amount of operational cost saving over multiple antennas. In multiple spacecraft support, a telecommand uplink from a single ground antenna will be shared by the supported spacecraft, and multiple telemetry downlinks originated from these spacecraft will also have to be established by a single ground antenna.

Several options have been studied to support this multiple spacecraft scenario (28). The most straightforward (and the most inefficient) option is to carefully assign different subcarrier frequencies to the supported spacecraft, allowing sufficient guard band to accommodate Doppler effects and tolerating some degree of spectrum overlapping in data sidebands in exchange for more simultaneous support. This method requires very tedious planning and is extremely inflexible when facing a dynamic scenario.

Another option is to redesign the spacecraft transponder so that the coherent turnaround ratio (TAR), which specifies the uplink to downlink carrier frequency ratio, is programmable. Each supported spacecraft receives its unique TAR from the uplink commands. As a result, different spacecraft will be instructed to use different downlink carrier frequencies because their TARs are distinct. Currently, a new digital transponder, known as the Small Transponder Modem developed by the Jet Propulsion Laboratory, has such a built-in feature.

The third option is to use code division multiple access (CDMA) techniques similar to those used in commercial mobile cellular systems. CDMA offers far more simultaneous support than that of the previous two options. However, to support various types of spacecraft, the power dissimilarity problem between weak rover and strong orbiter signals has to be properly solved to avoid severe performance degradation for weaker signals. This option may be the best choice when more and more multiple spacecraft support scenarios emerge in the future.

BIBLIOGRAPHY

1. B. Sklar, *Digital Communications: Fundamentals and Applications,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
2. J. Ziv and A. Lempel, A universal algorithm for sequential data compression, *IEEE Trans. Inf. Theory,* **IT-23**: 337–343, 1977.
3. J. Ziv and A. Lempel, Compression of individual sequences by variable rate coding, *IEEE Trans. Inf. Theory,* **IT-24**: 530–536, 1978.
4. C. E. Shannon, A mathematical theory of communication, *M. D. Comput.,* **14** (4): 306–317, 1997.
5. W. C. Lindsey and M. K. Simon, *Telecommunication System Engineering,* Englewood Cliffs, NJ: Prentice-Hall, 1973.
6. International Consultative Committee for Space Data Systems, CCSDS, *Recommendations for Space Data System Standards, Radio Frequency and Modulation Systems, part I, Earth Stations Spacecraft,* CCSDS 401.0-B, Blue Book, 1994.
7. J. H. Yuen, *Deep Space Telecommunications Systems Engineering,* New York: Plenum, 1983.
8. T. M. Nguyen, "Technique to select the optimum modulation indices for suppression of undesired signals for simultaneous range and data operations," *IEEE Trans. Electromagn. Compat.,* **32**: 7–19, 1990.
9. M. K. Simon and S. Million, *Residual versus suppressed carrier coherent communications,* TDA Progress Rep. 42-127: July–September 1996, Jet Propulsion Laboratory, Pasadena, November 15, 1996.
10. H. Tsou, M. K. Simon, and S. M. Hinedi, Closed loop carrier phase synchronization techniques motivated by likelihood functions, *1994 IEEE Int. Conf. Commun. Conf. Rec.,* **2**: 1994, pp. 934–939.
11. M. K. Simon et al., The performance of a coherent residual carrier PSK system using hybrid carrier phase synchronization, *1996 IEEE Int. Conf. Commun. Conf. Rec.,* **2**: 1996, pp. 1275–1280.
12. J. K. Holmes, *Coherent Spread Spectrum Systems,* New York: Wiley, 1982.
13. W. J. Hurd and S. Aquirre, A method to dramatically improve subcarrier tracking, *IEEE Trans. Commun.,* **COM-36**: 238–243, 1988.
14. A. Mileant and S. M. Hinedi, *Overview of Arraying Techniques in the Deep Space Network,* TDA Progress Rep. 42-104: October–December 1990. Jet Propulsion Laboratory, Pasadena, 1991.
15. H. Tsou et al., A functional description of the buffered telemetry demodulator for the Galileo mission to Jupiter, *1994 IEEE Int. Conf. Commun. Conf. Rec.,* **2**: 1994, pp. 923–928.
16. H. Tsou et al., The recovery of buffered telemetry data for future low cost space missions, *1995 IEEE Int. Conf. Commun. Conf. Rec.,* **2**: 1995, pp. 919–923.
17. K. Cheung et al., Enhancing the Galileo data return using advanced source and channel coding, *NASA Technol. 2004 Conf.,* Washington, D.C., September 1994.
18. W. Cham, Development of integer cosine transform by the principle of dyadic symmetry, *IEE Proc.,* part I **136**: 276, 282, 1989.
19. K. Cheung, F. Pollara, and M. Shahshahani, *Integer cosine transform for image compression,* TDA Progress Rep. 42-105: January–March 1991. Jet Propulsion Laboratory, Pasadena, 1991.
20. K. Cheung and K. Tong, Proposed data compression schemes for the Galileo S-band contingency mission, *Proc. NASA Space Earth Sci. Data Compression Workshop,* Snowbird, UT, 1993.
21. E. Paaske, Improved decoding for a concatenated coding system recommended by CCSDS, *IEEE Trans. Commun.,* **COM-38**: 1138–1144, 1990.
22. O. Collins and M. Hizlan, Determinate-state convolutional codes, TDA Progress Rep. 42-107: July–September 1991, Jet Propulsion Laboratory, Pasadena, 1991.
23. J. Statman et al., Decoder synchronizaton for deep space missions, TDA Progress Rep. 42-116: October–December 1993, Jet Propulsion Laboratory, Pasadena, 1994.
24. S. Dolinar and M. Belongie, Enhanced decoding for the Galileo low-gain antenna mission, *Proc. 1994 IEEE Int. Symp. Inf. Theory,* Trondheim, Norway, 1994.
25. E. Berlekamp, Bit-serial Reed–Solomon encoder, *IEEE Trans. Info. Theory,* **28**, 1982.
26. R. McEliece, The decoding of Reed–Solomon codes, TDA Progress Rep. 42-95: July–September 1988, Jet Propulsion Laboratory, Pasadena, 1988.
27. T. Chauvin and K. Cheung, A parallel Viterbi decoder for shared memory architecture, *SIAM Conf. Parallel Signal/Image Process. Multiprocess Syst.,* Seattle, WA, August 1993.
28. H. Tsou et al., Description of communication system options for single-aperture multiple-link (SAML) mission support, TDA Progress Rep. 42-127: July–September 1996, Jet Propulsion Laboratory, Pasadena, 1996.

HAIPING TSOU
KAR-MING CHEUNG
California Institute of Technology

RADIO TELESCOPES. See REFLECTOR ANTENNAS.

RADIOWAVE PROPAGATION. See PROPAGATION OF BROADCAST TRANSMISSIONS.