FILENAME.APP=3920MS00.MWM

Radiotelemetry

Haiping Tsou and Kar-Ming Cheung

Jet Propulsion Laboratory

California Institute of Technology

4800 Oak Grove Drive, 238-343

Pasadena, CA 91109

Radiotelemetry, by definition, is the science and technology of automatic measurement and transmission of data by radio from remote sources, as from space vehicles, to receiving stations for recording and analysis. It involves the preparation of data generated at remote sources, the transmission of data through radio channels, and the processing of 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 to be addressed in this field.

There are many ways to define a communication link, depending on where the link starts and ends. Customarily, a communication link refers to the transmission path where waveform information 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 will see in later discussion, the link performance is indeed 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 propagation effect of data compression in link design and analysis, since the error propagation property affects the integrity and throughput of the link. In the following sections, we will 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, wc would like to 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 subjected 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 frequency of errors in the received data, i.e., 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 of modulation and error-correction coding schemes.

Communication Link Budget Analysis

The link budget is a balance sheet of the gains and losses in decibel (dB) of various parameters in the communication path. Many of these parameters are either 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 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, can be conveniently expressed as [1]

$$\frac{P_T}{No} - \frac{\text{EIRP}(G/T)}{kL_sL_0} \tag{1}$$

where El RP is the effective isotropic radiated power of the spacecraft, G/7 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 and λ is the wavelength, and L_0 denotes all other losses and degradation factors not specifically addressed in the above equation. Note that all the gains and losses are statistical quantities. Next, 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 codeword to the number of coded bits in a codeword. The data rate or bit rate, denoted as R_b, is related to the symbol and code rates via $R_b = R_sR_c$. The bit SNR can be written as

$$\frac{E_b}{\text{No}} - \frac{{}'T\ /NO}{R_s\ R_c} \tag{2}$$

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 can be expressed in dB as follows:

$$\left(\frac{E_b}{N_0}\right)_{req} + M \le \frac{P_T}{N_0} - R_s - R_c \quad \text{(in dB)} \tag{3}$$

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 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 trade-off on quantity versus quality, depending on the requirements and objectives of the service the link supports. Also in an operation support 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 is accumulated.

Data Compression

In the midst of the Information Age when new data - texts, images, sensor data, and many other form of knowledge - are generated at a lightning pace, how these data can be efficiently communicated and/or archived becomes increasingly important. Data obtained directly from the 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 concerning the use of data compression is the trade-off between the efficiency of data handling and the complexity of signal processing to retrieve the information.

In English language, alphabetic characters like “a” and “e” are used more frequently than “y” and “z”, and words like “and” and “the” are much more frequent than “data” and “compression”. For images and sensor data, the differences between adjacent samples tend to be small rather than large. This non-uniform 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 most simplistic view of data compression.

Data compression can be broadly classified into two categories: lossless and lossy compression. Lossless compression denotes compression approaches in which the decompressed or reconstructed data exactly match the original. Lossy compression represents compression methods where some degradation of quality may be tolerable if a more compact but approximate representation can be achieved. Next, we will give brief surveys on text compression (lossless) and image compression (lossless and 10 SSY). However, the compression of audio, video, and facsimile is beyond the scope of this article. The interested readers are encouraged to consult specialized books and technical journals in these areas.

Text compression is probably the first area where people applied compression. In 1832 Samuel Morse developed a code using dots, dashes, and spaces to represent letters of the alphabet, numbers, and punctuation for telegraph transmission. Each dot or dash is delimited by a space. A dot and a space takes the same time. A dash is 3 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 into 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 as 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 can be 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 reduced 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 the previously transmitted value, and compares it against the current one. The error value is coded instead of the original pixel value. On the receiver side, since the errors and the predictive scheme are known, the receiver can recover 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 semi-conductor manufacturers.

Forward Error Correction Coding

In 1948, the landmark paper "A Mathematical Theory of Communication" [4] by Claude Shannon provided a mathematical framework to analyze communication - the process of sending information from one point to another through a noisy media - 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 fraction, which is a function of the signal and noise statistics. The process of introducing redundancy into the signal stream to combat noise, so that re-transmission is not needed, is called forward error correction coding. Shannon did not show how this should be done in his famous theorem. Later on, much research effort has yielded a variety of interesting theoretical and practical results in this area.

Since redundant signals are deliberately introduced into the signal stream as the error-correction coding, bandwidth expansion is expected. Also the encoding and decoding processes inevitably add complexity to the overall system. However, it can be shown that with a proper choice of coding scheme, the resultingE_b/N_0canbe made considerably lower compared to an uncoded system. Thus error-correction coding allows one to trade power with bandwidth and complexity. In this article, the applications 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 will not be covered here. 'I'he interested readers are encouraged to consult specialized books and technical journals in these fields.

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

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

In convolutional coding, k information bit sequences are encoded as continuous sequences using a finite-state shift register. n output sequences are produced, each of which is generated by a selected set of combinations of the 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. There exist well-known encoding and decoding algorithms that arc reliable in software and hardware. RS codes are flexible: there exist long powerful RS codes for demanding applications, and short RS codes for simple applications. Deep Space Communications use the (255, 223) RS code as the outer code of a 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 Deep Space Network antennas. 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 Disk (CD) recording, which is a mass-marketed consumer product. The coding system used is called cross-interleaved Reed-Solomon Code (CIRC), and 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 burst-like errors, which are 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 arc 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 decoder (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 issue of interaction between data compression and error control (containment/detection/correction) processes in a modern digital communication system. Data compression and error correction are two indispensable building blocks in a modern digital communication system. Data compression conserves transmission and storage bandwidth by removing redundancy in the source data. Error correction introduces redundancy to the data in a controlled fashion 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 other 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 do 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 using better fault-tolerance techniques on various system components. The second issue on how to contain the 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 the errors. Some simple ways are: adding a delimiter pattern at regular intervals, and forcing fixed block size transmission by appending filling O's, etc. These techniques reduce the compression performance. The second approach is to choose the data compression schemes which are less susceptible to error propagation. These schemes are usually the ones which transmitted the compressed data in a fixed block size, or those which inherently have delimiter patterns in their compressed data stream. 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 by which the signals are transformed into waveforms that are suitable for transmission through a certain type. of media or channel. For example, when transmitting electronic signals through free space with the use of an antenna which converts electronic signals into electromagnetic (EM) waves, the physical dimensions of the antenna aperture should be at least on the same order as the wavelength of the EM wave. Therefore, a baseband signal has to modulate a sinusoidal signal at radio frequency (RF), referred to as the carrier, before it can be transmitted by antennas of reasonable size.

A typical modulation procedure for digital signal transmission starts with a baseband pulse code modulation (PCM) that converts the analog signal into a binary data stream. Depending upon the subsequent RF modulation schemes, this binary data stream will be fed into different devices. For example, fixed-length blocks of the binary data can be used to determine the instantaneous frequencies of the RF carrier for a frequency -shift-keyed (FSK) system. On the other hand, the binary data can drive a pulse generator whose output modulates the phase of the subcarrier (if used) and the RF carrier in a typical phase-shift-keyed (PSK) system.

Various binary data formats (or pulse shapes) [5], e.g., the commonly used Non-Return-to-Zero (NRZ), hi-phase (also known as Manchester code), and raised cosine pulse waveform, etc., can be selected, The choice: of using a particular format is determined by several factors, e.g., the considerations of bandwidth efficiency, inherent synchronization features and noise immunity associated with each data format.

The PSK modulation has been widely adopted for deep space communications mainly for the reasons that it is a very efficient type of modulation in terms of its 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 a PSK modulation can be designed to have a multiple phase-shift-keyed (MPSK) signal for which the number of phasor states, denoted by M, can be any power of 2. However, the increase in bit error rate with M for MPSK signals prevents its use when M is large. Furthermore, because of the ease of being able to be decomposed into two orthogonal channels for further signal processing, the MPSK modulated signals commonly used for radiotelemetry are limited to the cases of $M = 4$ or lower, namely the binary phase-shift-keying (BPSK), the quadrature phase-shift-keying (QPSK) or its variations such as offset QPSK, unbalanced QPSK, etc. For some applications, quadrature amplitude-shift-keying (QAM) modulation is used to allow multi-level signals transmitted on either one or both of the mutually orthogonal channels.

For BPSK modulation, it is possible to include subcarrier(s) in the transmitted signal. The purpose of using subcarrier(s) is to separate data sidebands of different signals from the RF carrier in such a way that they will not interfere with each other. Whether to use subcarrier(s) or not depends upon the mission design. For example, the use of a subcarrier is preferred when a residual carrier component is preserved and used for 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 subcarrier, the sine wave and the square wave, are commonly used. 'l'he use of sine wave subcarrier along with a 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 the use of a square wave subcarrier for deep space missions because of the ease of generating a square wave onboard the spacecraft and the relatively less stringent bandwidth allocation for this type of missions.

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

$$S_T(t) = \sqrt{2P_T}\sin\left(\omega_c t + \sum_{i=1}^{N} m_i S_i(t)\right) \tag{5}$$

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

$$s,(t) = \begin{cases} \sum\limits_{k=-\infty}^{\infty} d_{k,i} P_i(t-kT), \text{ for PCM/PM (without subcarrier)} \\ \left[\sum\limits_{k=-\infty}^{\infty} d_{k,i} P_i(t-kT)\right] \sin(\omega_{sc} t), \quad \text{for PCM/PSK/PM with sine wave subcarrier} \\ \left[\sum\limits_{k=-\infty}^{\infty} d_{k,i} P_i(t-kT)\right] \mathrm{sgn}\left[\sin(\omega_{sc} t)\right], \text{ for PCM/PSK/PM with square wave subcarrier} \end{cases} \qquad (6)$$

represents either a normalized baseband waveform (with $d_{k,i} = \pm 1$) or a normalized BPSK modulated subcarrier waveform (at the frequency ω_{sc}). $P_i(t)$ is the pulse function of unit-power and T is the reciprocal of the symbol rate.

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

$$S_T(t) = \sqrt{2P_T}\sin\left\{\omega_c t + m_1\sin(\omega_1 t) + m_2 d(t)\operatorname{sgn}\left[\sin(\omega_{sc} t + \theta_{sc})\right] + \theta_c\right\}$$

$$= \sqrt{2P_T}\left|\begin{aligned} & J_0(m_1)\cos(m_2)\sin(\omega_c t + \theta_c) + d(t)J_0(m_1)\sin(m_2)\operatorname{sgn}\left[\sin(\omega_{sc} t + \theta_{sc})\right]\cos(\omega_c t + \theta_c) \\ & + \cos(m_2)\left[\sum_{n=1}^{\infty} 2J_{2n}(m_1)\cos(2n\omega_1 t)\right]\sin(\omega_c t + \theta_c) \\ & + \cos(m_2)\left[\sum_{n=0}^{\infty} 2J_{2n+1}(m_1)\sin((2n+1)\omega_1 t)\right]\cos(\omega_c t + \theta_c) \\ & + d(t)\sin(m_2)\left[\sum_{n=1}^{\infty} 2J_{2n}(m_1)\cos(2n\omega_1 t)\right]\operatorname{sgn}\left[\sin(\omega_{sc} t + \theta_{sc})\right]\cos(\omega_c t + \theta_c) \\ & - \mathrm{din}\left[\sum_{n=0}^{\infty} 2J_{2n+1}(m_1)\sin((2n+1)\omega_1 t)\operatorname{sgn}\right]\left[\sin(\omega_{sc} t + \theta_{sc})\right]\sin(\omega_c t + \theta_c) \end{aligned}\right| \quad (7)$$

where $J_n(\cdot)$ is the nth order Bessel function and θ_c and θ_{sc} are random carrier and subcarrier phases respectively, each being uniformly distributed over O to 2π. The first term in this expression is the residual carrier component which can be 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 which can be extracted separately, and the fifth and sixth terms contain both the data and ranging modulations. On one hand, the ranging modulation index, m_1, is typically chosen to be small (around 1/2 or smaller) so that the power consumption by this ranging signal is relatively small. On the other hand, the data modulation index is selected to be large (close to its upper limit z/2) to ensure that only sufficient power goes to the residual carrier tone and the rest of the power is solely allocated for telemetry data transmission.

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

$$S_{,}(t) = \sqrt{\alpha P_T}\,\sin\left(\omega_c t + \sum_{i=1}^{M} m_{c,i}S_{c,i}(t)\right) + \sqrt{(1-\alpha)P_T}\,\cos\left(\omega_c t + \sum_{i.1}^{N} m_{s,i}S_{s,i}(t)\right) \qquad (8)$$

where *a* is the percentage of transmitted power in one of the channels. For the case that only one binary signal of NRZ format is transmitted on each channel, i.e., $M = N = 1$, with modulation indices $m_{s,1} = m_{c,1} = z/2$, the QPSK signal can be rewritten as

$$S_T(t) = \sqrt{\alpha P_T}\,S_{c,i}(t)\cos(\omega_c t) - \sqrt{(1-\alpha)\,P_T}\,S_{s,i}(t)\sin(\omega_c t) \qquad (9)$$

Clearly, it is a combination of two BPSK signals on two orthogonal basis functions. Several variants of QPSK modulation, including offset QPSK (OQPSK) and minimum shift keying (MSK), are also commonly used in near- Earth space missions. The detailed description of these modulation schemes will not be covered here.

Demodulation

Demodulation is a process by which the received waveforms are transformed back into their original state by reversing the modulation procedure. After traveling through various types of media or channels, the received waveform can be corrupted in many ways. For example, it is likely to be corrupted by internally generated noise which may be modeled as an additive white Gaussian noise (AWGN) or shot noise, or externally introduced interference such as multipath, fading, etc. 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 can be given by

$$r(t) = \sqrt{2P_T}\sin\left(\omega_c(t+\tau)+\left(\frac{\pi}{2}\right)\sum_{k=1}^{N} d_k P(t+\tau-kT)+\theta\right)+n(t) \qquad \text{(1o)}$$

where τ is the random propagation delay, θ is a uniformly distributed (over O to 2π) carrier phase and $n(t)$ is a noise modeled as an AWGN with a two-sided power spectral density level NO/2 Watt/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 can be rewritten as

$$r(t) = \sqrt{2P_T}\left(\sum_{k=1}^{N} d_k P(t+\tau-kT)\right)\cos(\omega_c t + \theta_c) + n(t) \qquad (11)$$

where $\theta_c = (\theta + \omega_c \tau)_{\mathrm{mod}\, 2\pi}$ is the total carrier phase. In order 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

$$r'(t) = \sqrt{P_T}\left[\sum_{k=1}^{N} d_k P(t + \tau - kT)\right]\cos(\phi_c) + n'(t) \tag{12}$$

where $\phi_c = \theta_c - \dot{\theta}_c$ 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 degradation, which will inevitably be translated into a degradation in the bit error performance. In order to make a decision on each of the transmitted bits, say the *i*th bit d,, the resulting signal $r'(t)$ needs to be 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 correct timing estimate, $\hat{\tau}$, is very important in this matched filter operation, since integrating across two bits with opposite polarities will reduce the detected symbol (for coded system) or bit (for uncoded system) energy and result in 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 will be 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, the incoming signal is referred to as synchronization.

As indicated previously, coherent reception and demodulation requires phase information of the carrier and subcarrier (if used) and also the symbol timing information. This information needs to be provided and updated for coherent receivers continuously since they are usually time-varying because of the changing characteristics of the channel. Therefore, individual tracking loops which continuously update their estimates of specific parameters are used 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 can achieve lock without the help from the others, except for the residual carrier tracking loop in which a carrier tone can be separately tracked. However, in practice, each loop's performance is usually analyzed independent of the others to keep the problem manageable.

It is also important to know that all the tracking loops discussed below are motivated by maximum *a posteriori* (MAP) estimation theory which only suggests the open-loop structure of a one-shot estimator. The closed-loop structure whose error signal is derived by differentiating the likelihood function and equating the resulting function to zero is only motivated by 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 the 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 referred to as the S-curve of the loop. A stable lock point exists at ϕ_c = O, 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, can be found as a Tikhonov distributed random variable and its probability density function is [5]

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

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

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

where B_L is the loop bandwidth. ‘I’he detailed description of a PLL is discussed in another article in this encyclopedia and will not be repeated here,

For the suppressed carrier case 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), has to be tracked by a Costas loop. ‘l’he Costas loop is a phase tracking loop whose functionality is very similar to that of a PLL. Except for the same feedback path comprised of the loop filter and the 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 of them. The incoming signal is first mixed with each of the two locally generated reference signals 90-dcg apart, i.e., $\sqrt{2}\sin(\omega_c t + \hat{\theta}_c$ and$)\sqrt{2}\cos(\omega_c t + \hat{\theta}_c)$, at thc corrcsponding phase detector and then passed through the arm filters. Although the low-pass arm filter can be implemented as either a passive RC-typc filter or an active filter, the matched filter (i.e., an active filter) is the optimal design. The output of the two arm filters are multiplied, which effectively removes the data modulation before it is fed into the loop filter. Because of this multiplication, a Costas loop is actually tracking twice the phase error. Accordingly, the Costas loop has two equally stable lock points at $\phi_c = O$ 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 points inevitably introduce a phase ambiguity, i.e., the demodulated data will have an inverted polarity if the loop locks at $\phi_c = \pi$. This 180-deg phase ambiguity can be resolved in several ways. For example, a known sequence pattern can be inserted in the transmitted symbol stream from time to time so that the receiver can detect the inverted polarity by comparing its received sequence pattern with the known one. However, the most efficient method is to incorporate 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 can be applied to extract the relative phase (or, the transmitted information) after each symbol decision. A small penalty in terms of the error performance exists for this differential encoding/decoding scheme because one incorrect symbol decision will render two consecutive errors in the relative phase.

The phase error for the first-order Costas loop with I&D arm filters can be similarly found as a Tikhonov distributed random variable and its probability density function is

$$p(\phi_c) = \frac{\exp\left(\frac{\rho_{\phi_c}}{4}\cos(2\phi_c)\right)}{\pi I_0\left(\frac{\rho_{\phi_c}}{4}\right)} \qquad |\phi_c| \leq \frac{\pi}{2} \tag{15}$$

and the associated loop SNR is

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

where E_s /NO = $P_d T/N_0$ is the symbol SNR. Note that the term in parentheses is usually referred to as the squaring loss, which results from the various signal-and-noise products in the error signal. At low symbol SNR, the squaring loss can be significant.

As previously discussed, the transmitted power is allocated to the residual carrier component and data-bearing component through the choice of modulation index for telemetry data. It has been proved that fully suppressed carrier is the best way to maximize data throughput [8]. However, if a residual carrier component is desired for purposes other than communication, it's always a dilemma to set this modulation index because, on one hand, sufficient power has to be given to the residual carrier so that it can be 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 the data throughput. Since the residual carrier and data sidebands are coherently related, a hybrid loop [9] which consists of the phase-locked loop and Costas loop structures can be used to exploit this coherence and thereby improve carrier phase tracking in this scenario, 'l'his 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 the 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 error signal. As a result, there are usually dual lock points existing for the hybrid loop, i.e. $\phi_c = O$ and $\phi_c = \pi$ similar to those of a Costas loop, yet, these two lock points in general are not equiprobable. It can be shown that [10] the lock point at $\phi_c = \pi$ vanishes when the modulation index is smaller than a threshold as a function of symbol SNR.

With a given modulation index, an optimal relative weight between the PLL and Costas loop portion can be derived to minimize the hybrid loop tracking jitter. Since the relative tracking performance between a 1'12]J 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 both modulation index and symbol SNR.

QPSK

The carrier tracking of a QPSK signal is usually done by a generalized Costas loop known as the cross-over Costas loop. There are basically two variants of this loop: the polarity-type for high SNR scenarios and the non-polarity-type for low SNR scenarios [7]. In the polarity-type loop structure, 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 polarity-type cross-over Costas loop is a Tikhonov distributed random variable with probability density function

$$p(\phi_c) = \frac{2\exp\left(\frac{\rho_{\phi_c}}{16}\cos(4\phi_c)\right)}{\pi \mathrm{I}_0\left(\frac{\rho_{\phi}\rho_{\phi_c}}{166}\right)} \qquad |\phi_c| \leq \frac{\pi}{4} \tag{17}$$

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

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

Note that, similar to eq. (16) of the Costas loop, the term in parentheses found here is the squaring loss the polarity-type cross-over Costas loop suffers. The non-polarity-type, i.e., the one without hard-limit operations in it, has a different squaring loss which is smaller than that of its polarity-type sibling in low SNR region. More details can be found in [1 1]. 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 the suppressed carrier tracking of BPSK signals since there is no residual tone left for the BPSK subcarrier. The Costas loop is used here to wipe off the data modulation and a squaring loss associated with this process is applied. However, depending on the use of a sine wave or square wave subcarrier, the tracking performance can be 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 can be realized by using a time-domain windowing function on the quadrature arm [12]. In this case, the windowing function implemented around the mid-phase transition of the Q-arm reference signal can be 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 the 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 (i.e., defined as the reciprocal of the tracking jitter $\sigma_{\phi_{sc}}^2$ in radian^2) can be greatly improved by shrinking the window size. The first-order loop SNR can be found as

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

where W_{sc} is the quadrature window size (between O 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 1/4 or 1/8 is usually used to provide 6 to 9 dB improvement in loop SNR.

No such improvement from quadrature windowing can be realized for the sine wave Costas loop. This is expected from the MAP criterion because the non-windowed loop already meets the theory but the windowed sine function cannot be approximated as the derivative of a cosine function,

Symbol Timing Tracking

Symbol synchronization has a direct impact on the data detection process since inaccurate symbol timing reduces the probability of making correct decisions. Although a separate channel can be used for sending signals for synchronization purposes, to extract the synchronization information directly from the data-bearing signal has the advantage of requiring no additional power and frequency spectrum.

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) has been widely used for symbol synchronization. Similar to the Costas loop, the DTTL has a double-arm loop structure with a hard decision followed by a transition detector on its in-phase arm and a delay on its quadrature arm to keep signals on both arms properly aligned. It is important to note that the term "in-phase" refers to a symbol timing which is synchronous to the received one 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 O and 1) relative to the symbol interval and is centered at each symbol epoch, causing the mid-point of the Q-arm integration offset by a half-symbol from its I-arm counterpart. Similar to the 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 ϕ_{sym} = O. The phase error for the first-order DTTL is a Tikhonov distributed random variable and its probability density function is

$$p(\phi_{sym}) = \frac{\exp(\rho_{\phi_{sym}} \cos(\phi_{sym}))}{2\pi I_0(\rho_{\phi_{sym}})} \qquad |\phi_{sym}| \le \pi \tag{20}$$

with the corresponding loop SNR given as

$$\rho_{\phi_{sym}} = \frac{1}{(2\pi)^2} \frac{P_d}{N_0 B_L} \left(\frac{1}{W_{sym}} \right) \left| \frac{2\left[\operatorname{erf}\left(\sqrt{\frac{E_s}{N_0}} \right) - \frac{W_{sym}}{2} \sqrt{\frac{1}{\pi}} \left(\frac{E_s}{N_0} \right) \exp\left(-\frac{E_s}{N_0} \right) \right]}{1 + \frac{W_{sym}}{2} \left(\frac{E_s}{N_0} \right) - \frac{W_{sym}}{2} \left[\sqrt{\frac{1}{\pi}} \exp\left(-\frac{E_s}{N_0} \right) + \sqrt{\frac{E_s}{N_0}} \operatorname{erf}\left(\sqrt{\frac{E_s}{N_0}} \right) \right|} \right. \Bigg) \tag{21}$$

Symbol SNR Degradation

The symbol SNR degradation is the direct cause of poor bit error performance and can be translated into the telemetry system loss as seen in the next section when the receiver performs a hard-decision on each demodulated symbol. For the case that no hard-decision is performed by the receiver, the symbol SNR degradation directly affects the decoder performance since the demodulated symbols, which are referred to as the soft symbols, are fed to the decoder without going through a hard-decision device.

Because of the difficulty in analyzing the coupled carrier, subcarrier and symbol tracking loops, the SNR degradation of the demodulated symbol (for coded system) or bit (for uncoded system) due to imperfectly synchronized references is usually approximated as a product of degradation factors of the individual loops, each factor being derived based upon the assumption of perfect tracking in the other loops. The overall degradation, conditioned on the corresponding phase errors, can be found for the telemetry signal given in eq. (7) as

$$D_{SNR}(\phi_c,\phi_{sc},\phi_{sym}) = D_c(\phi_c)D_{sc}(\phi_{sc})D_{sym}(\phi_{sym}) = [\cos(\phi_c)]^2\left[1-\frac{2}{\pi}|\phi_{sc}|\right]^2\left(-\frac{|\phi_{sym}|}{\pi}+\frac{\phi_{sym}^2}{2\pi^2}\right) \tag{22}$$

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

$$\left(\overline{D_{SNR}}\right)_{dB} = \left(\overline{D_c}\right)_{dB} + \left(\overline{D_{sc}}\right)_{dB} + \left(\overline{D_{sym}}\right)_{dB} \tag{23}$$

where $\overline{D_c}$, $\overline{D_{sc}}$ and D_{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 subcarrier and symbol waveform distortion due to the bandlimited channel.

Bit Error Performance (Uncoded System) and Telemetry System Loss

The 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,

$$P_b = \int_{-\pi}^{\pi} \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}} \cos(\phi_c) \right) p(\phi_c) d\phi_c \tag{24}$$

where $p(\phi_c)$ is the probability density function (p. d. f.) of the carrier phase error given in eq. (13), when the carrier is tracked by a PLL such that no phase ambiguity exists. However, it can be shown that, with a fixed loop SNR, there exists an irreducible error probability in this case no matter how large the bit SNR can be. This irreducible error probability is solely characterized by the carrier tracking loop SNR and, for a given loop bandwidth, can only be reduced by allocating more power to the residual carrier component which serves no purpose in the transmission of telemetry except for the PLL tracking function.

For suppressed carrier tracking of a BPSK signal by the Costas loop, the phase ambiguity, as mentioned previously, does exist and needs to be resolved. The bit error probability for the special case of perfect phase ambiguity resolution (say, by other means such as a periodically-inserted known sync pattern) can be found as

$$P_b = \int_{-\pi/2}^{\pi/2} \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}} \cos(\phi_c) \right) p(\phi_c) d\phi_c \tag{25}$$

where $p(\phi_c)$ is the p.d.f. given in eq. (1 5).

If a differentially coding scheme is utilized to resolve the phase ambiguity, the bit error probability becomes ['7]

$$P_b = \int_{-\pi/2}^{\pi/2} \mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\cos(\phi_c)\right)\left(1-\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}}\cos(\phi_c)\right)\right)p(\phi_c)d\phi_c \qquad (26)$$

where $p(\phi_c)$ is the p.d.f. given in eq. (15). No irreducible error probability exists in suppressed carrier tracking since, for a fixed loop bandwidth and bit duration product, the tracking loop SNR increases directly proportional to 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, i.e., $\mathrm{Cos}^2(\phi_c)$, repeatedly appears 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 will be a three-fold integration involving the overall symbol SNR degradation given in eq. (22). For example,

$$P_b = \int_{-\pi}^{\pi}\int_{-\pi/2}^{\pi/2}\int_{-\pi/2}^{\pi/2}\frac{1}{2}\mathrm{erfc}\left(\sqrt{\frac{E_b}{N_0}D_{SNR}(\phi_c,\phi_{sc},\phi_{sym})}\right)p(\phi_c)p(\phi_{sc})p(\phi_{sym})d\phi_c d\phi_{sc} d\phi_{sym} \qquad (27)$$

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

For QPSK signals, it can be shown that [7] the bit error probability is

$$P_b = \int_{-\pi/4}^{\pi/4} \left\{ \frac{1}{4} \operatorname{erfc}\left[\sqrt{\frac{E_b}{N_0}}\left(\cos(\phi_c) - \sin(\phi_c)\right)\right] + \frac{1}{4} \operatorname{erfc}\left[\sqrt{\frac{E_b}{N_0}}\left(\cos(\phi_c) + \sin(\phi_c)\right)\right]\right\} p(\phi_c) d\phi_c \qquad (28)$$

where $p(\phi_c)$ is the p.d.f. given 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 perfect synchronized system. For example, the bit error probability for a perfectly synchronized BPSK or QPSK system is

$$P_b = \frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{E_b}{N_0}}\right) \qquad (29)$$

and, therefore, the required bit SNR for this ideal system at a given bit error probability P_b^* is

$$\left(\frac{E_b}{N_0}\right)^* = \left[\operatorname{erfc}^{-1}\left(2P_b^*\right)\right]^2 \qquad (30)$$

The lossy system with a system loss L will need L times as much power, namely $E_b/N_0 = L\left(E_b/N_0\right)^*$, to achieve the same bit error probability or, in other words, to compensate the loss incurred.

Advanced Topics

With recent space missions moving towards high data rate and low transmitting power operations, to combine 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 have been pushed to their limits.

Three arraying techniques [13] are briefly discussed here, including the symbol stream combining, the baseband combining and the full spectrum combining, each combining signals at a different stage in the signal flow.

For the symbol stream combining scheme, each participating antenna performs the carrier, subcarrier, and symbol synchronization individually. The symbols at the output of each receiver are then combined, with the appropriate weights, to form the final symbols for detection or decoding. This scheme has the advantage of a small combining loss. Also, it can be easily handled even when a real-time combining from inter-continental sites is required because the combining is performed at the symbol rate. The disadvantage is that each antenna needs its full set of receiver hardware and needs to be able to lock on the signal individually.

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

In the full spectrum combining, the signals are combined at intermediate frequency (IF). Before they can be combined, the relative time delay and phase difference have to be estimated and compensated properly so that signals can be added coherently. The combined IF signal is then directed to a single receiver for further synchronization and demodulation. The advantage of this scheme is that only one of the participating antennas needs to be able to lock on the combined signal, which allows the inclusion of smaller antennas in this arraying scheme even though they are unable to lock on the signal. The disadvantage is the very large transmission or recording bandwidth required to carrier the IF signals through the networked antenna sites for combination.

Besides the above-mentioned arraying techniques, a scheme referred to as the carrier arraying which employs coupled carrier tracking devices from participating antennas should be addressed here, too. This scheme by itself does not combine the signals and, thus, needs to bc 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 Jet Propulsion Laboratory to support Galileo S-Band Mission. Many advanced technologies have been designed 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 will be 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.

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

The FSR downconverts the RF signal to IF for digitization, and then further open-loop downconverts each data sideband to baseband individually and coherently before it is sampled and recorded. This renders a significant reduction of the required bandwidth for transmission through the inter-continental antenna network since the processing rate is linked to the symbol rate, instead of the much higher subcarrierfrequency.

The recorded signals (residual carrier and data sidebands centered at the first four harmonics of the square wave subcarrier are kept in the Galileo S-Band Mission) from arrayed antennas are combined by the I? SC, 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 the acquisition, synchronization and demodulation, as well as miscellaneous monitoring functions through its carrier, subcarrier, symbol tracking loops and associated lock indicators [14]. In BTD, the individually combined data sidebands arc processed in a coherent fashion 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 but not hard-limited symbol stream (usually referred to as the soft symbol since no hard decision is attempted to determine its polarity), which is written to a file and transferred to the FCD upon request for decoding and decompression.

Since the FSR/FSC combined data are recorded on tape, BTD can actually work on any segment of data off-line in either direction, namely forward or backward in time. In fact, with the availability of multiple-CPU workstations, simultaneous BTD sessions can be initiated on different segments of data. For example, one session can be dedicated to process real-time samples forward (in time) while the others can reprocess other recorded segments as needed. The soft symbol streams from these simultaneous sessions can be merged into a single one as long as each of them is properly time tagged. Because of the flexibility in software implementation, many non-causal signal processing techniques can be performed to process or re-process the data to further enhance the quality of telemetry. One important feature of the BTD is the so-called gap closure processing [15] which can greatly reduce the possible data loss due to receiver acquisition, re-synchronization, anti loss-of-lock.

The need to reprocess a segment of sampled data arises either from the failure to maintain the in-lock status in any of the loops in BTD or the failure to properly decode the soft symbols by FCD. A segment of sampled data on which the telemetry can not be extracted reliably is referred to as a gap, and the processing of a gap to extract any valid information not available when that segment of samples was first processed is referred to as gap closure processing. Gaps caused by acquisition can be found in the beginning of each pass or at the instants where receiver drops out of lock, while gaps generated by cycle-slips in one of the loops can occur randomly in a pass. Along its demodulation efforts, BTD keeps track of its internal states, including the lock indicators, symbol SNR estimates and the state variables inside loop filter and NCO for all three loops. These state variables are recorded at fixed intervals as check points and, with them, a software receiver can be easily restored to its state at a check point immediately prior to or after a gap. By estimating the parameters of a phase process in a region near a restored check point where the phase tracking was successfully carried out, one can start the gap closure processing from this check point and move into the gap. Two configurations, one for closed loop and the other one for open loop, can be used here. The closed loop configuration needs to initiate the loop filter with phase parameter estimates in a particular way that, when the loop is closed and starts to track at the check point, the loop can virtually start with steady state tracking immediately. For a relatively stable phase process and a gap of small size, an open loop configuration can be applied by using an estimated phase profile as the reference without resorting to a loop operation. Both configurations can be applied to gap closure processing in either direction, namely forward or backward in time, since the buffered data can be processed in either order. This is especially useful when a gap occurs at the beginning of a track such that all the available check point 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/7' figure as the elevation angle of an antenna changes in a pass. With higher elevation angle, an antenna has higher G/T figure and can support higher symbol rate when the symbol SNR is fixed. The software implementation of BTD can handle symbol rate changes without dropping lock on symbol timing as long as the rate changes occur in a predictable manner.

The FCD is a subsystem that performs error-correction decoding and data decompression in the DGT. Implemented in software on a multi-processor 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 multi-pass decoding which is able to achieve a final bit error rate of 10^{-7} at a 0.65 dB bit SNR. The architecture and the detail operations of the FCD are described in the next section.

Advanced Source and Channel Coding for Space Applications

In this section, we will 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 [1 6]. First, the use of the integer cosine transform (ICT) scheme for lossy image compression will be briefly explained. Then, an advanced error-correction coding scheme used to protect the heavily edited and compressed data will be discussed, followed by the discussion on the issue of interaction between data compression and error control (contain ment/detection/correction) processes.

Galileo's Image Compression Scheme

The Galileo image compression scheme is a block-based lossy image compression algorithm that uses an 8 x 8 ICT. The ICT was first proposed in [17], and was streamlined and generalized in [1 8] [19]. It can be viewed as an integer approximation of the popular discrete cosine transform (DCT) scheme, 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 that resemble those of the DCT matrix. Besides, the rows of the ICT matrix are orthogonal. The integer property eliminates real multiplication and real addition operations, thus greatly reducing the 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 can be made orthonormal by multiplying it by an appropriate diagonal matrix. This operation can be incorporated in the quantization (dequantization) stage of the compression (decompression) scheme, 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

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

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

Figure 1 shows the rate-distortion performance of the ICT scheme compared to 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 [20] [21]. 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 bit signal-to-noise ratio of 0.65 dB at a bit error rate of 10^{-7}. Figure 2 shows the schematic of the FCD architecture. In this article, only the implementation and operation aspects of the FCD task is discussed. The FCD novel node/frame synchronization scheme is discussed in [22] and its code selection and performance analysis are discussed in detail in [23].

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

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

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

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

(i.e. $p(a)$ = O).

In the encoding/decoding process, each power of a is represented as a distinct non-zero 8-bit pattern. The zero byte is the zero element in GF(256). The basis for GF(256) is descending powers of a. Note that this is the conventional representation, not Berlekamp's dual basis [24]. The RS generator polynomial is defined as

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

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

'l'he Galileo mission utilizes four distinct RS codes. We define RS(n,k) to be an RS code which accepts as input k data bytes and produces as a code word n bytes, where $n > k$. An RS(n,k) code can correct 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 Figure 2. The RS decoders use a time-domain Euclid algorithm to correct both errors and erasures. The details of the decoding algorithm is discussed in [25].

The (1 4,1/4) Convolutional Code and Its Parallel Viterbi Decoder:

The (1 4,1/4) convolutional code used for the Galileo mission is the concatenation of a software (1 1, 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 (’l’MU), 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. ‘l’he generator polynomials of the equivalent (14,1/4) code are (26042, 36575, 25715, 16723).

‘I’he 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 due to the slow downlink 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 inter-processor synchronization and communication, and this greatly reduces the decoding speed. The second approach requires much less synchronization and communication since 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. ‘I’he details of the FCD software Viterbi decoder implementation are described in [26].

Rededcoding:

Rededcoding, as shown in Figure 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. The Viterbi decoder is thus constrained to follow only paths consistent with known symbols from previously decoded RS codewords. 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 4 feedback passes. In the first pass, only the first code word RS(255,161) is decoded. In the second pass, the fifth codeword RS(255,195) is decoded. In the third pass, the third and seventh codewords 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 the Viterbi decoding of the inner code in a progressively refined manner. The details of the FCD redecoding analysis are described in [23].

Interaction Between Data Compression and Error Control Processes

Packet loss and other uncorrectable errors in a compressed data stream cause error propagation, and how the error propagates depends on the compression scheme being used. 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 non-image) 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 (SS1) 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 can search for the next available synchronization marker and continue to decompress the rest of the data. In this case, the interval is chosen to be 8 lines of uncompressed data. The error containment strategy guarantees that error propagation will 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 Huffrnan code can be used to contain errors, but it is difficult to implement. A packetizing scheme with varying packet sizes can be used to contain errors (by matching packet boundaries and the compressed data block boundary), but the packet headers introduce excessive downlink overhead in the case of SS1 data.

The SS1 ICT error containment scheme works as follows. On the compression side, every 8 lines of data is 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 arc 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] O-s. The decompression scheme consists of two program modules: the SS1 ICT decompression module and the error detection/sync module. The SS1 ICT decompression module reconstructs the data from the compressed data stream, and the error detection/sync module checks the prefix condition of the Huff man 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 arc realigned using the modulo counter. The corrupted portion of the data is flagged and reported.

The downlink overhead of the SS1 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:

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

For example, an 800x 800 SS1 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. 'I'his dedication requires an efficient scheduling of the resources on the ground, including hardware, software and personnel. With more and more concurrent missions, a need of 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 from international space agencies will place more than a dozen spacecraft, including orbitors, landers as well as 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 beamwidth 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 savings over the use of multiple antennas. In a multiple spacecraft support scenario, a telecornmand 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 [27]. 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 re-design the spacecraft transponder so that the coherent turn around 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 since their TARs are distinct. Currently, a new digital transponder, known as the tiny deep space transponder developed by Jet Propulsion Laboratory, has such a feature built in.

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

FILENAME.APP=3920BI00.MWM

Bibliography

[1] B. Sklar, *Digital Communications: Fundamentals and Applications*, PTR Englewood Cliffs: Prentice Hall, 1988.

[2] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," IEEE Trans. on Information Theory, Vol. IT-23, May 1977.

[3] J. Ziv and A. Lempel, "Compression of individual sequences by variable rate coding," IEEE Trans. on Information Theory, Vol. IT-24, Sep. 1978.

[4] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Tech. J.27(3).

[5] W. C. Lindsey and M. K. Simon, *Telecommunicat ion System Engineering,* Englewood Cliffs: 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, 1989.

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

[8] M. K. Simon and S. Million, "Residual Versus Suppressed Carrier Coherent Communications," TDA Progress Report 42-127: July - September 1996, Jet Propulsion Laboratory, Pasadena, November 15, 1996.

[9] H. Tsou, M. K. Simon and S. M. Hinedi, "Closed Loop Carrier Phase Synchronization Techniques Motivated by Likelihood Functions," 1994 IEEE International Conference on Communications Conference Record, Vol. 2, pp. 934-939, May 1994.

[10] M. K. Simon, H. Tsou, S. M. Hinedi and K. Hamdan, "The Performance of a Coherent Residual Carrier PSK System Using Hybrid Carrier Phase Synchronization," 1996 IEEE International Conference on Communications Conference Record, Vol. 2, pp. 1275-1280, June 1996.

[11] J. K. Holmes, *Coherent Spread Spectrum Systems, New* York: John Wiley and Sons, 1982.

[12] W. J. Hurd and S. Aquirre, "A Method to Dramatically Improve Subcarrier Tracking," IEEE Trans. on Communications, Vol. COM-36, No. 2, pp. 238-243, February 1988.

[13] A. Mileant and S. M. Hinedi, "Overview of Arraying Techniques in the Deep Space Network," TDA Progress Report 42-104: October - December 1990, Jet Propulsion Laboratory, Pasadena, February 15, 1991.

[14] H. Tsou, B. Shah, R. Lee and S. M. Hinedi, "A Functional Description of the Buffered Telemetry Demodulator for the Galileo Mission to Jupiter," 1994

IEEE International Conference on Communications Conference Record, Vol. 2, pp. 923-928, May 1994.

[15] H. Tsou, A. Mileant, R. Lee and S. M. Hinedi, "The Recovery of Buffered Telemetry Data for Future Low Cost Space Missions," 1995 IEEE International Conference on Communications Conference Record, Vol. 2, pp. 919-923, June 1995.

[16] K. Cheung et. al., "Enhancing the Galileo Data Return Using Advanced Source and Channel Coding," presented in the NASA Technology 2004 Conference, Washington D. C., September 1994.

[17] W. Chain, "Development of Integer Cosine Transform by the Principle of Dyadic Symmetry," IEE Proceedings, Vol. 136, Pt. I, No. 4, August 1989.

[18] K. Cheung, F. Pollara, and M. Shahshahani, "Integer Cosine Transform for Image Compression," TDA Progress Report 42-105: January - March 1991, Jet Propulsion Laboratory, Pasadena, May 15, 1991.

[19] K. Cheung and K. Tong, "Proposed Data Compression Schemes for the Galileo S-Band Contingency Mission," Proceedings of the NASA Space and Earth Science Data Compression Workshop, April 1993, Snowbird, Utah.

[20] E. Paaske, "Improved Decoding for a Concatenated Coding System Recommended by CCSDS," IEEE Transaction of Communications, Vol. COM-38, August 1990.

[21] O. Collins and M. Hizlan, "I)ctcrn~inate-State Convolutional Codes," TDA Progress Report 42-107: July - September 1991, Jet Propulsion Laboratory, Pasadena, November 15, 1991.

[22] J. Statman, K. Cheung, T. Chauvin, J. Rabkin, and M. Belongie, "Decoder Synchronization for Deep Space Missions," TDA Progress Report 42-116: October - December 1993, Jet Propulsion Laboratory, Pasadena, February 15, 1994.

[23] S. Dolinar and M. Belongie, "Enhanced Decoding for the Galileo Low-Gain Antenna Mission," Proceedings 1994 IEEE International Symposium on Information Theory, Trondheim, Norway, June 27- July 1, 1994.

[24] E. Berlekamp, "Bit-Serial Reed-Solonlon Encoder," IEEE Transaction of Information Theory, Vol. 28, 1982.

[25] R. McEliece, "The Decoding of Reed-Solomon Codes," TDA Progress Report 42-95: July - September 1988, Jet Propulsion Laboratory, Pasadena, November 15, 1988.

[26] T. Chauvin and K. Cheung, "A Parallel Viterbi Decoder for Shared Memory Architecture", presented in the SIAM Conference on Parallel Signal/Image Processing on Multiprocessor Systems, Seattle, Washington, August 1993.

[27] H. Tsou, S. Million, S. M. Hinedi, 'l'. M. Nguyen, S. Kayalar, M. K. Simon, R. 1,. Horttor and W. Moore, "Description of Communication System Options for Single-Aperture Multiple-Link (SAM I,) Mission Support," TDA Progress

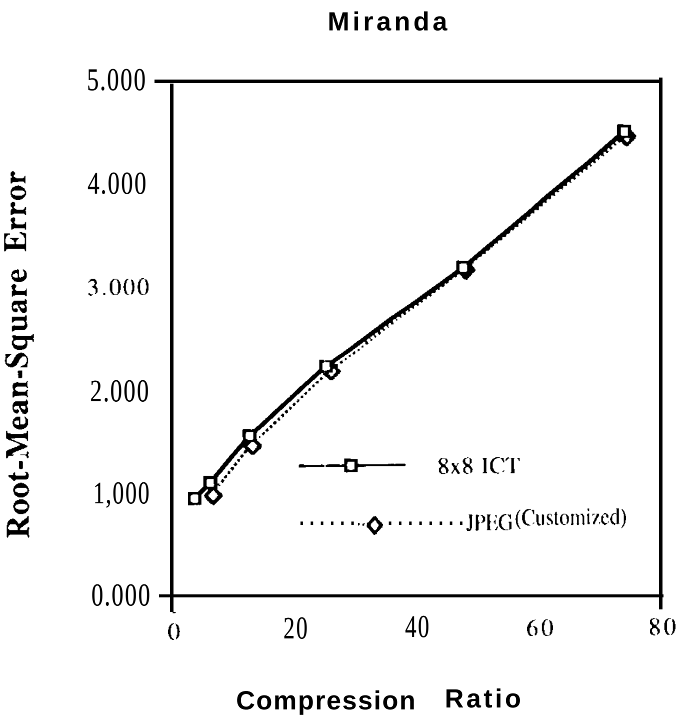
Report ***42-127:*** July - September 1996, Jet Propulsion Laboratory, Pasadena, November 15, 1996.

FILENAME.APP=3920 FL00.MWM

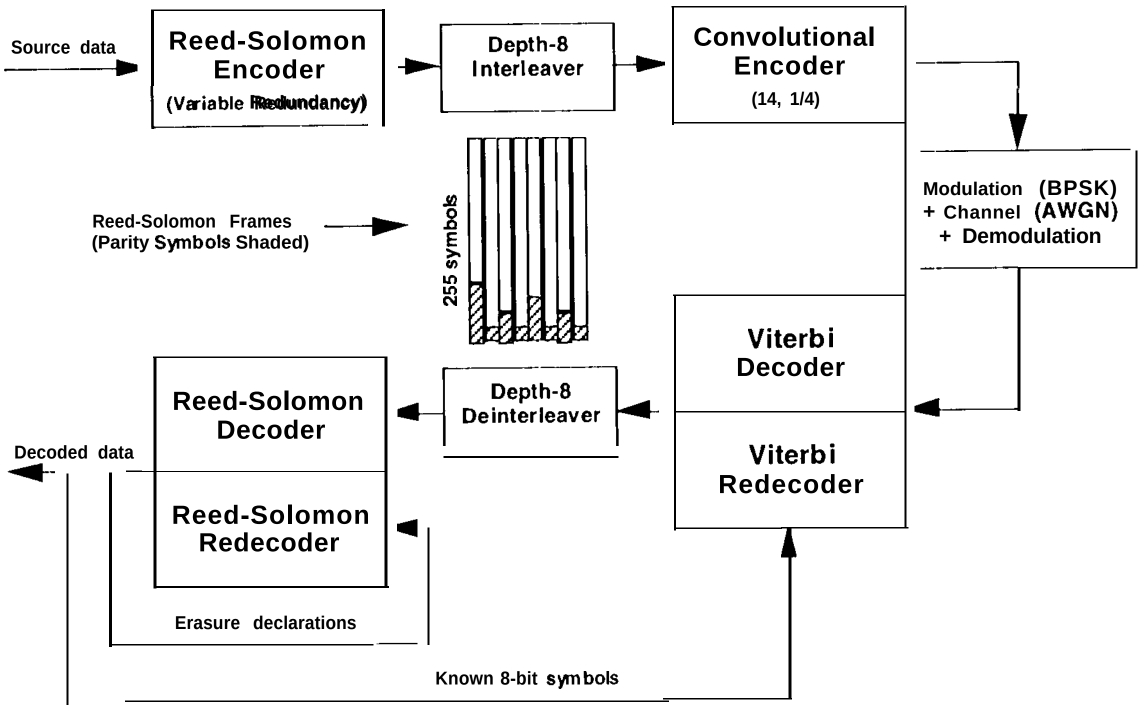
Figure 1. Rate-Distortion Performance of ICT

Figure 2. Schematic of the FCD

FILENAME.APP=3920FG01.MWM



FILENAME.APP=3920FG02.M WM



Source data
Reed-Solomon Encoder
(Variable Redundancy)
Depth-8 Interleaver
Convolutional Encoder
(14, 1/4)
Modulation (BPSK)
+ Channel (AWGN)
+ Demodulation
Reed-Solomon Frames
(Parity Symbols Shaded)
255 symbols
Viterbi Decoder
Viterbi Redecoder
Depth-8 Deinterleaver
Reed-Solomon Decoder
Reed-Solomon Redecoder
Decoded data
Erasure declarations
Known 8-bit symbols