TRELLIS-CODED MODULATION

Any communication in nature suffers from impairments such as noise, which corrupts the data transmitted from the transmitter to the receiver. In this article, we consider the principles behind trellis-coded modulation (TCM), which is an established method to combat the aforementioned impairments. TCM is one of the main components of the modern modulator-demodulator (modem) systems for data transmission over telephone lines.

HISTORICAL REMARKS

Trellis diagrams (or state transition diagrams) were originally introduced in communications by Forney (1) to describe maximum likelihood sequence detection of convolutional codes. They were employed to soft decode convolutional codes using a dynamic programming algorithm (also known as the Viterbi algorithm).

The concept of trellis was later extended by Bahl et al. (2) to linear block codes where they were used as a natural framework to implement the maximum a posteriori probability (MAP) algorithm. Later, Forney unveiled the trellis structure of Euclidean Codes and Lattices.

Trellis-coded modulation is perhaps the most frequently applied branch of trellis theory. Such an implementation combines channel coding and modulation for transmission over band-limited channels. Specifically, trellis-coded modulation integrates the trellis of convolutional codes with M-ary linear modulation schemes such as, for example, M-phase-shift keying. Generally, modulation schemes containing larger Euclidean distances between their signal sets provide more robustness against noise over Gaussian channels. On the other hand, traditionally channel codes were designed so that distinct codewords have large Hamming distances (3). These two criteria are not equivalent unless 2-amplitude modulation or 4-phase-shift keying (4-PSK) modulation is used. Combining channel coding and modulation makes it possible to use a distance measure in coding which is equivalent to Euclidean distance in modulation. When the noise is additive white Gaussian, trellis-coded modulation provides 3–6 dB improvements over uncoded modulation schemes for the same bandwidth efficiency. Although Massey had proposed the idea of combining channel coding and modulation in 1974 (4), the first trellis-coded modulation scheme was introduced by Ungerboeck and Csjaka in 1976 (5,6).

OVERVIEW

Figure 1 shows a block diagram of a communication system in which binary data are transmitted over a noisy channel. Since the signal transmitted over the physical channel is a continuous electrical waveform, the modulation scheme converts its binary (discrete) input to continuous signals which are suitable for transmission over band-limited channels. If the effects of noise on the transmitted signal can be modeled by adding uncorrelated Gaussian noise samples, the channel is called an *additive Gaussian noise* channel. The ratio of the transmitted power to the noise power, signal-to-noise ratio (SNR), is an important parameter which affects the performance of the modulation scheme. For a given SNR and band-

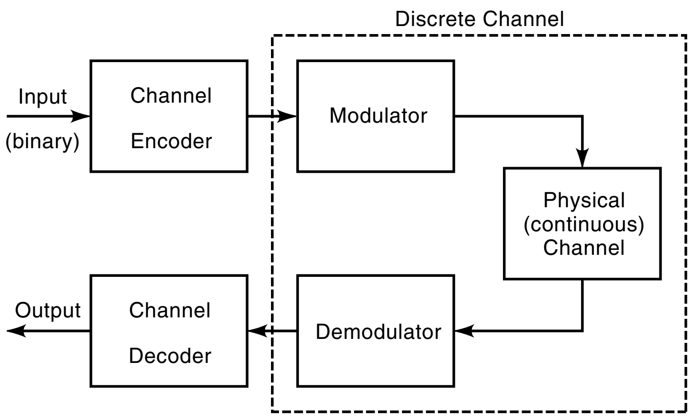

Figure 1. Block diagram of a communication system.

width, there is a theoretical limit for the maximum bit rate which can be reliably transferred over a continuous channel (Shannon capacity) (7). If the bit rate is less than the Shannon capacity, the objective of a modulation scheme is to minimize the bit error rate for a given SNR and a given bandwidth.

The combination of modulation, continuous channel, and demodulation can be considered as a discrete channel. Because of the hard-decision at the demodulator, the input and output of the discrete channel are binary. The effects of noise in the physical channel translates into bit errors in the discrete channel. The job of channel coding is to correct errors by adding some redundancy to the bit stream. In other words, error correcting codes systematically add new bits to the bit stream such that the decoder can correct some of the bit errors by using the structure of the redundancy. Of course, the adding redundancy reduces the effective bit rate per transmission bandwidth.

Before the seminal work of Ungerboeck and Csjaka, channel codes and modulation schemes were designed separately. Error correcting codes were designed to have codewords with large Hamming distance from each other. Modulation schemes utilize signal sets with maximum Euclidean distance. Since Hamming distance and Euclidean distance are not equivalent for most modulation schemes, designing modulation and coding scheme separately results in about 2 dB loss in SNR. In contrast, trellis-coded modulation is designed to maximize Euclidean distance between the channel signal sets by combining channel codes and modulation (Fig. 2). For a given rate and bandwidth, trellis-coded modulation uses a redundant signal set at the modulator and a maximum likelihood soft decoder at the demodulator. In trellis-coded modulation, the necessary redundancy of coding comes from expanding the signal sets not bandwidth expansion, as will be discussed in the next section. Designing a good coded modulation scheme is possible by maximizing the free Euclidean distance for the code. In fact, Ungerboeck and Csjaka's point of departure from traditional coding is that the free distance of a trellis-coded modulation can be significantly more than that of the corresponding uncoded modulation scheme.

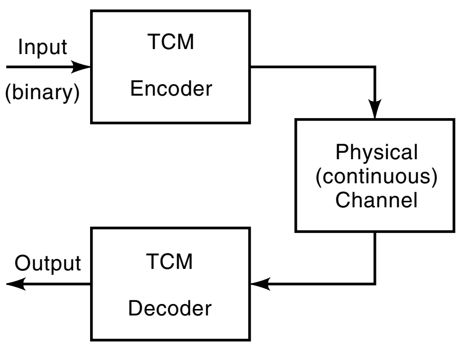

Figure 2. Using trellis-coded modulation to combine channel coding and modulation.

A trellis (state-transition diagram) can be used to describe trellis-coded modulation. This trellis is similar to that of convolutional codes. However, the trellis branches in trellis-coded modulation consist of modulation signals instead of binary codes. Since the invention of trellis-coded modulation, it has been used in many practical applications. The use of trellis-coded modulation in modulator–demodulators (modems) for data transmission over telephone lines has resulted in tremendous increased in the bit rate. International Telegraph and Telephone Consultative Committee (CCITT) and its successor International Telecommunication Union (ITU) have widely utilized trellis-coded modulation in high-speed modems for data transmission over telephone lines (8–10).

TRELLISES AS FINITE-STATE MACHINES

Much of the existing literature (11–13) uses *set partitioning* and trellis structure of convolutional codes to describe trellis-coded modulation. This may be attributed to the fact that this approach was taken by Ungerboeck and Csjaka in their seminal paper where the foundation of coded modulation was laid. In this exposition, the goal is to present the results with the required background kept as small as possible. In this light, we pursue a different line of thought and approach the topic using finite-state machines.

Finite-State Machines

A *finite-state machine* can be thought of as a three-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{T}, \mathcal{L})$, where $\mathcal{S}$, $\mathcal{T}$, and $\mathcal{L}$, respectively are referred to as the *set of states,* the *set of transitions,* and the *defining alphabet* of $\mathcal{M}$. Each element of the set $\mathcal{T}$ is a *transition* (s_i, s_e, l) with $s_i, s_e \in \mathcal{S}$ and $l \in \mathcal{L}$. Such a transition is said to start in s_i, end in s_e, and is labelled with l. All transitions starting from the same state, s_i, and ending at the same state, s_e, are called *parallel transitions.* For each state s, the number of transitions starting (respectively ending) in s is called the out-degree (respectively the in-degree) of s.

The finite-state machine $\mathcal{M}$ is said to be regular if the in-degrees and out-degrees of all the states of $\mathcal{S}$ are the same. The machine $\mathcal{M}$ is binary if it is regular and if the out-degrees and in-degrees of elements of $\mathcal{S}$ as well as the number of states of $\mathcal{S}$ are powers of 2. In this article, we are only interested in binary machines.

The Trellis of a Binary Finite-State Machine

Every finite-state machine $\mathcal{M}$ has a trellis diagram $T(\mathcal{M})$ which is a graphical way to represent the evolution path of $\mathcal{M}$. Let $\mathcal{M}$ denote a binary finite-state machine having 2^n states. A trellis diagram $T(\mathcal{M})$ of $\mathcal{M}$ is defined as a labelled directed graph having levels 0, 1, 2, 3, ... Each level of $\mathcal{M}$

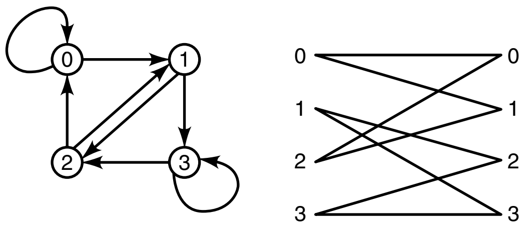

Figure 3. A four-state finite-state machine and the corresponding trellis. Graphical equivalence between trellises and finite state machines is clearly visible.

has 2^n states labelled $0, 1, \ldots, 2^n - 1$ corresponding to, respectively, $s_0, s_1, \ldots, s_{2^n-1}$ elements of $\mathscr{S}$. There is an edge labelled with l between state i of level k and j of level $k + 1$ if and only if $(s_i, s_j, l) \in \mathscr{T}$ where $i, j = 1, 2, \ldots, 2^n$, $k = 1, 2, \ldots$ and $l \in \mathscr{L}$.

Figure 3 shows an example of a finite-state machine $\mathscr{M}$ containing four states and the corresponding trellis diagram $T = T(\mathscr{M})$. In Fig. 3, we only show the transitions between different states (not the labels). One can use different labels on the transitions to construct different codes. This is the subject of the next section. It is clear that given a trellis diagram T as defined, one can construct a finite-state machine $\mathscr{M}$ such that $T = T(\mathscr{M})$ and vice versa.

Trellis Codes

A *trellis code* is the trellis of a binary finite-state machine where the alphabet $\mathscr{L}$ comes from a signal constellation having unit average energy (we use unit average energy for all signal constellations in this article). Practical signal modulation includes but is not restricted to the 4-PSK, 8-PSK, and 16-quadrature amplitude (16-QAM) constellations. In this light, we only consider these signal constellations here.

Let $\mathscr{M}$ denote a trellis code with 2^n states such that the in-degree and out-degree of each state is 2^R. Let $T(\mathscr{M})$ denote the trellis of $\mathscr{M}$ and assume that at time zero the machine is at state zero. The trellis code $\mathscr{M}$ can be used to encode R bits of information at each time instance. At each time $t = 0, 1, 2, \ldots$ a block of R bits of data denoted by $B(t)$ arrives at the encoder. Depending on the 2^R possible values of this block of data and the state $s_i(t)$ of the machine at time t, a transition beginning in that state such as $(s_i(t), s_e(t + 1), l(t))$ is chosen. The trellis code then moves to the state $s_e(t + 1)$ and outputs $l(t)$ the label of the transition. Thus, $B(0)B(1)B(2) \ldots$ is mapped to the codeword $l(0)l(1)l(2) \ldots$. We let $C(\mathscr{M})$ denote the set of all possible output sequences and also refer to it as the *code* of $\mathscr{M}$ when there is no ambiguity.

The alert reader notices that such an encoder may be completely useless. Indeed, if all the transitions are labelled with the same signal constellation symbol, all bit sequences will be mapped to the same codeword. Thus, it is important to design the trellis code so that such a scenario is avoided.

The assignment of labels to transition in particular is what determines the performance of a code over a transmission media. Thus, a performance criterion is needed before designing a trellis code for real applications. In most of the situations, an exact performance criterion is intractable for design and a tractable approximate criterion is used instead. Tractable approximate design criteria are known for the Gaussian channel, rapidly fading channel, slowly fading channel, and numerous other cases. A good general reference for trellis codes is (14).

Trellis Codes for the Gaussian Channel

The design criterion (albeit an approximate one) for the Gaussian channel is well established in the literature. In general a code $\mathscr{C}$ is expected to perform well over a Gaussian channel if the codewords are as far from each other (in terms of Euclidean distance) as possible. The computation of Euclidean distance of two codewords of a code is not that difficult and hence this criterion is tractable for design. To remove any ambiguity, we mathematically define the distance between two paths of $T(\mathscr{M})$ with the same starting and ending states. Without loss of generality, let us assume that the two paths emerge at time $t = 0$ and remerge at time $t = t'$. Suppose that the branches are labelled c_t^1 and c_t^2, $t = 0, 1, \ldots, t'$, for the first and second path, respectively. Then, the distance between the two paths is defined by $\sum_{t=0}^{t'} |c_t^1 - c_t^2|^2$.

For the design of a trellis code $\mathscr{M}$, the minimum of distances between any two paths of $T(\mathscr{M})$ that emerge from some state at some time and remerge at another state of the trellis at a later time dominates the performance of the code. This quantity is called the *free distance* of the trellis code. Thus trellis codes that are useful for Gaussian channel must have large free distances.

However, in pursuing such a design, we should take the bandwidth requirements into account. Fixing the symbol duration (time to transmit a constellation symbols), the dimensionality of the signal constellation directly relates to the bandwidth requirement for the channel. This is a fundamental result known as the Landau–Pollak–Slepian Theorem (15,16). The consequence of this result is that a comparison between the free distances of two trellis codes is justified only if they use signal constellations of same dimensionality.

An Ungerboek–Csjaka Idea

Suppose that we would like to design a trellis code for the transmission of R bits per channel use. One way of transmission is using a trellis code $\mathscr{M}$ that has one state and use a signal constellation $\mathscr{SC}$ having 2^R elements. The 2^R edges between the state of level t with that of $t + 1$ in $T(\mathscr{M})$ are labelled with the different signal constellation symbols. This trellis code is called the *uncoded* signal constellation $\mathscr{SC}$. The uncoded binary phase-shift keying (BPSK) constellation is given in Fig. 4. Clearly the free distance of the uncoded signal constellation $\mathscr{SC}$ is the minimum distance between the points of $\mathscr{SC}$.

One way of obtaining larger free distances is to use a signal constellation having more than 2^R elements for transmission of R bits per channel use. In practice, it is good to double the constellation size while designing over the Gaussian channels. As the dimensionality of the signal constellation is fixed and the number of signals in the constellation is doubled, we can expect a reduction in minimum distance of the new constellation.

Figure 4. An uncoded BPSK constellation. Each point represents a signal to be transmitted over the channel.

As an example to transmit 1 bit per channel use we will use a 4-PSK (Fig. 5) instead of BPSK constellation. The minimum distance of the 4-PSK constellation is $\sqrt{2}$ while the minimum distance of the BPSK constellation is 2 (both have unit average energy). Thus, there is a loss in minimum distance by doubling the size of constellation. A Trellis code on 4-PSK alphabet can only be useful as compared to the uncoded case if it can compensate this loss by having a larger free distance than 2.

Ungerboeck and Csjaka demonstrated that there exist trellis codes that can outperform the uncoded signal constellations. They also proposed mapping by set partitioning as the machinery to construct these trellis codes.

MAPPING BY SET-PARTITIONING

Let $\mathcal{SC}$ be a signal set. Let $\mathcal{SC}_1 \subseteq \mathcal{SC}$ such that $|\mathcal{SC}|$ the number of elements of $\mathcal{SC}$ be a multiple of $|\mathcal{SC}_1|$. A *partitioning* of $\mathcal{SC}$ based on $\mathcal{SC}_1$ is a collection Σ_1 of disjoint subsets of $\mathcal{SC}$ such that Σ_1 contains $\mathcal{SC}_1$ and $\cup_{X\in\Sigma_1} X = \mathcal{SC}$. Elements of Σ_1 are called the *cosets* of $\mathcal{SC}_1$ in $\mathcal{SC}$. The concept of partitioning can be extended to the nested chains of subsets of $\mathcal{SC}$.

Specifically, consider a decreasing chain of subsets of a signal constellation $\mathcal{SC}$

$$\mathcal{SC} = \mathcal{SC}_0 \supseteq \mathcal{SC}_1 \supseteq \mathcal{SC}_2 \supseteq \ldots \supseteq \mathcal{SC}_J$$

such that $|\mathcal{SC}_i|$ is a multiple of $|\mathcal{SC}_{i+1}|$ for $i = 0, 1, \ldots, J - 1$. Such a decreasing chain induces partitioning in each level. First, $\mathcal{SC}$ is partitioned into a set Σ_1 of cosets of $\mathcal{SC}_1$ in $\mathcal{SC}$ which in particular contains $\mathcal{SC}_1$. Each element of Σ_1 contains $|\mathcal{SC}_1|$ elements of $\mathcal{SC}$. In a similar way, $\mathcal{SC}_1$ can be partitioned into cosets of $\mathcal{SC}_2$ in $\mathcal{SC}_1$ and the other elements of Σ_1 can be partitioned into sets of cardinality $|\mathcal{SC}_2|$. The result is Σ_2, the collection of all the cosets of $\mathcal{SC}_2$ in $\mathcal{SC}$ which in particular includes $\mathcal{SC}_2$. The process is then repeated for J times and all the cosets of $\mathcal{SC}_i$ in $\mathcal{SC}_j$ for $1 \le j \le i \le J$ are derived. In this article, we are only interested in partitions based on *binary* chains corresponding to the case when $|\mathcal{SC}_i|$, $i = 1, 2, \ldots, J$, are powers of two.

The central theme of the Ungerboeck–Csjaka paper (5) is that given a binary set partitioning based on a decreasing chain of subsets of $\mathcal{SC}$ as described, the minimum distance of cosets of $\mathcal{SC}_i$ in $\mathcal{SC}$ is a nondecreasing function of i. Indeed, if the partitioning is done in a clever way, the distances can substantially increase. Examples of such a set partitioning for the 4-PSK, 8-PSK, and 16-QAM are given in Figs. 6, 7, and 8, respectively. The notations

$$A_k = \cos(2\pi k/4) + \sin(2\pi k/4)\mathbf{j}, k = 0, 1, 2, 3$$
$$B_k = \cos(2\pi k/8) + \sin(2\pi k/8)\mathbf{j}, k = 0, 1, 2, \ldots, 7$$
$$Q_{k_1,k_2} = ((2k_1 - 3) + (2k_2 - 3)\mathbf{j})/\sqrt{10},$$
$$k_1 = 0, 1, 2, 3, k_2 = 0, 1, 2, 3$$

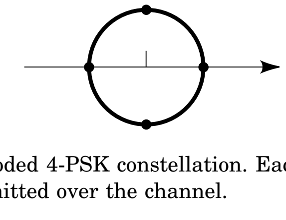

Figure 5. An uncoded 4-PSK constellation. Each point represents a signal to be transmitted over the channel.

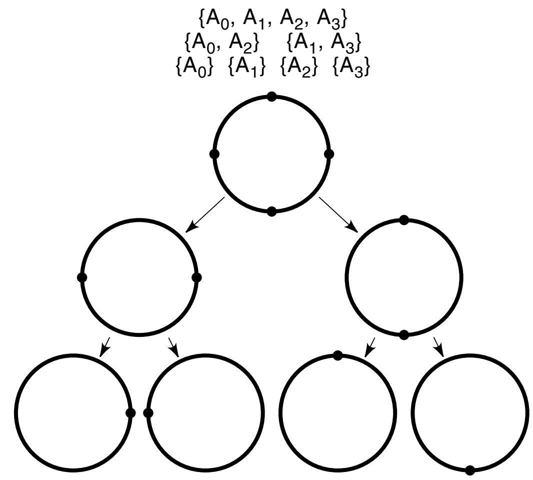

Figure 6. Set partitioning for 4-PSK constellation. The partitioning increases the minimum distance in each level.

(where $\mathbf{j} = \sqrt{-1}$) are used to represent the 4-PSK, 8-PSK, and 16-QAM constellations throughout this article.

As can be seen from Fig. 8, the minimum distances of the partitions in the 16-QAM case increase by a factor of $\sqrt{2}$ for each level. By choosing appropriate signals from each partition level as the labels of transitions of a finite-state machine, we could achieve very high free distances. This is the heart of Ungerboeck–Csjaka design and is called *mapping by set partitioning.*

The general heuristic rules established for design by Ungerboeck–Csjaka are

- Parallel transitions (those starting from and ending in the same states) are assigned to signal points with maximum Euclidean distance.
- The signal points should occur with the same frequency.
- Transitions originating from and merging into any state are assigned from elements of different cosets.

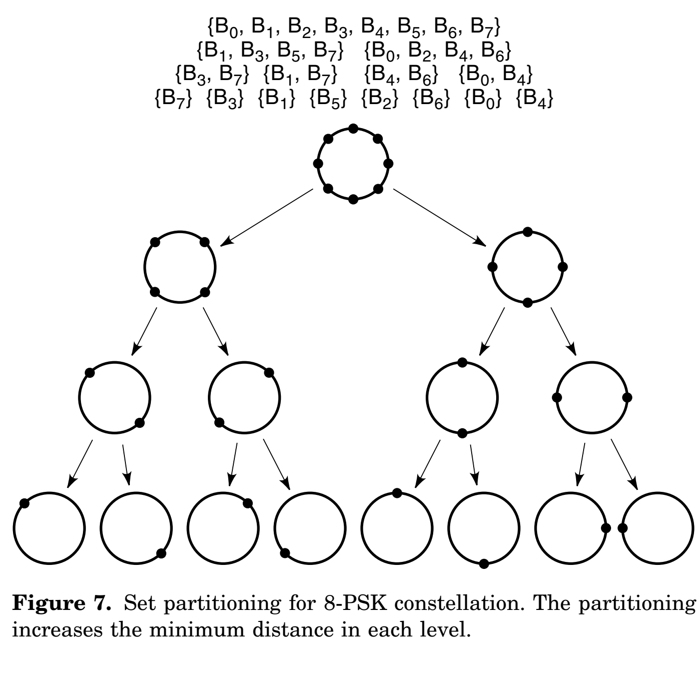

Figure 7. Set partitioning for 8-PSK constellation. The partitioning increases the minimum distance in each level.

$$\{Q_{0,3}, Q_{2,1}, Q_{0,1}, Q_{2,3}, Q_{1,2}, Q_{3,0}, Q_{1,0}, Q_{3,2}, Q_{0,0}, Q_{2,2}, Q_{0,2}, Q_{2,0}, Q_{1,1}, Q_{3,3}, Q_{1,3}, Q_{3,1}\}$$

$$\{Q_{0,3}, Q_{2,1}, Q_{0,1}, Q_{2,3}, Q_{1,2}, Q_{3,0}, Q_{1,0}, Q_{3,2}\} \quad \{Q_{0,0}, Q_{2,2}, Q_{0,2}, Q_{2,0}, Q_{1,1}, Q_{3,3}, Q_{1,3}, Q_{3,1}\}$$

$$\{Q_{0,3}, Q_{2,1}, Q_{0,1}, Q_{2,3}\} \quad \{Q_{1,2}, Q_{3,0}, Q_{1,0}, Q_{3,2}\} \quad \{Q_{0,0}, Q_{2,2}, Q_{0,2}, Q_{2,0}\} \quad \{Q_{1,1}, Q_{3,3}, Q_{1,3}, Q_{3,1}\}$$

$$\{Q_{0,3}, Q_{2,1}\} \ \{Q_{0,1}, Q_{2,3}\} \ \{Q_{1,2}, Q_{3,0}\} \ \{Q_{1,0}, Q_{3,2}\} \ \{Q_{0,0}, Q_{2,2}\} \ \{Q_{0,2}, Q_{2,0}\} \ \{Q_{1,1}, Q_{3,3}\} \ \{Q_{1,3}, Q_{3,1}\}$$

$$\{Q_{0,3}\} \{Q_{2,1}\} \{Q_{0,1}\} \{Q_{2,3}\} \{Q_{1,2}\} \{Q_{3,0}\} \{Q_{1,0}\} \{Q_{3,2}\} \{Q_{0,0}\} \{Q_{2,2}\} \{Q_{0,2}\} \{Q_{2,0}\} \{Q_{1,1}\} \{Q_{3,3}\} \{Q_{1,3}\} \{Q_{3,1}\}$$

Figure 8. Set partitioning for 16-QAM constellation. The partitioning increases the minimum distance in each level.

These rules follow the intuition that good codes should have symmetry and large free distances. Examples of 4-PSK, 8-PSK, and 16-QAM codes are given in Tables 1–5.

From these tables, it is clear that by increasing the number of states in the trellis, the free distance (and hence the performance) can be improved. However, we will see that this has a penalty in terms of decoding complexity.

Let us now consider an example. Consider the set partitioning of the 8-PSK and the four-state trellis code given in Table 3 based on the previous partitioning. As can be seen from the table, the labels of the transitions originating from each state of the trellis belong to the same coset while those of distinct states belong to different cosets. The design has a lot of symmetries as it is expected that good codes should demonstrate a lot of symmetries. It can be easily shown that free distance of the previous trellis code is $\sqrt{2}$ times the minimum distance of a 4-PSK constellation. This translates into 3-dB asymptotic gain (in SNR). In general the asymptotic gain of a trellis code with rate R bits per channel use (2^{R+1} elements in the constellation) over an uncoded constellation with the same rate is defined by 10 log $d^2_{\text{free}}/d^2_{\min}$ where d_{free} is the minimum free distance of the code and $d_{\min}$ is the minimum distance between the uncoded constellation elements.

Figures 9 and 10 give information about the coding gain versus the number of states of best 8-PSK and 16-QAM trellis codes known for transmission of 2 and 3 bits/channel use, respectively.

DECODING TRELLIS CODES: THE DYNAMIC PROGRAMMING ALGORITHM

Decoding trellis codes is usually done through the dynamic programming algorithm also known as the Viterbi algorithm. The Viterbi algorithm is in some sense an infinite algorithm that decides on the path taken by the encoder. This was proved to be optimum for sequence estimation by Forney. However, in practice one has to implement a finite version of the algorithm. Naturally, only practice is of interest here.

Table 1. A 4-State 4-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$
$s_i = 0$	A_0	A_2		
$s_i = 1$			A_1	A_3
$s_i = 2$	A_2	A_0		
$s_i = 3$			A_3	A_1

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding transition label is given in the table. Blank entries represent transitions that are not allowed.

To understand the implementation of the decoder, we first define the *constraint length* $\nu(C)$ of a trellis code $C(\mathcal{M})$ to be the minimum t such that there exists two paths of time length t starting at the same state and remerging at another state. Practically, we choose a multiple of $\nu(C)$ depending on the decoding delay allowed in the application and refer to it as the *decoding depth* $\theta(C)$. We then proceed to execute the finite decoding depth Viterbi algorithm. At each stage of the algorithm, for every possible state s of the encoder, a *survivor path* $P_t(s)$ of length $\theta(C)$ and an accumulated metric $m_t(s)$ is preserved. We denote the possible states of the encoder by s_i, $i = 0, 1, \ldots, 2^n - 1$, and the received signal at time t by r_t. We always follow the convention that the encoder is in the zero state at time zero.

The decoder starts by setting $m_0(s_0) = 0$ and $m_0(s_i) = \infty$ for all $i = 1, 2, \ldots, 2^n - 1$. In practice, one can choose a large number instead of ∞. Further, at the beginning of the decoding process, the decoder sets the survivor paths $P_t(s_i)$, $i = 0, 1, 2, \ldots, 2^n - 1$, to be the void string. In other words, at the beginning of the decoding nothing is saved as the survivor paths of each state.

The decoder then starts decoding by computing the branch metrics of each branch at time $t = 0, 1, 2, 3, \ldots$. Suppose that a branch at time t is labelled with c_t, then the metric of this branch is $|r_t - c_t|^2$. The decoder computes for each state s_i, the sum of the accumulated metric $m_t(s_j)$ and the branch metric of any state s_j with any branch starting at state s_j at time t and ending in state s_i at time $t + 1$. The decoder then computes the minimum of all these possible sums and sets $m_{t+1}(s_j)$ to be this minimum. If this minimum is given by the state i at time t and some branch b_t, the survivor path $P_{t+1}(s_j)$ is given by the path $P_t(s_i)$ continued by the branch b_t. This process is then repeated at each time.

The decoder starts outputting decision bits after time $t \geq \theta(C)$, where $\theta(C)$ denotes the decoding depth. At each time $t \geq \theta(C)$, the decoder looks at the survivor path of the state with the lowest accumulated metric. The decoder outputs the sequence of bits corresponding to the branch of path at time $t - \theta(C)$. In this way, a decoding delay of $\theta(C)$ must be tolerated.

MULTIDIMENSIONAL TRELLIS CODES

The trellis codes constructed in the previous section use an element of a two-dimensional constellation for labels. It is neither necessary to have a two-dimensional constellation nor only one symbol of the constellation per label of transitions. This gives rise to *multidimensional trellis codes* or M-TCM codes.

Table 2. An 8-State 4-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$	$s_e = 4$	$s_e = 5$	$s_e = 6$	$s_e = 7$
$s_i = 0$	A_0	A_2						
$s_i = 1$			A_1	A_3				
$s_i = 2$					A_2	A_0		
$s_i = 3$							A_3	A_1
$s_i = 4$	A_0	A_2						
$s_i = 5$			A_1	A_3				
$s_i = 6$					A_2	A_0		
$s_i = 7$							A_3	A_1

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding transition label is given in the table. Blank entries represent transitions that are not allowed.

Table 3. A 4-State 8-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$
$s_i = 0$	B_0, B_4	B_2, B_6		
$s_i = 1$			B_1, B_5	B_3, B_7
$s_i = 2$	B_2, B_6	B_0, B_4		
$s_i = 3$			B_3, B_7	B_1, B_5

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding possible transition labels are given in the table. Blank entries represent transitions that are not allowed.

Table 4. An 8-State 8-PSK Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$	$s_e = 4$	$s_e = 5$	$s_e = 6$	$s_e = 7$
$s_i = 0$	B_0	B_4	B_2	B_6				
$s_i = 1$					B_1	B_5	B_3	B_7
$s_i = 2$	B_4	B_0	B_6	B_2				
$s_i = 3$					B_5	B_1	B_7	B_3
$s_i = 4$	B_2	B_6	B_0	B_4				
$s_i = 5$					B_3	B_7	B_1	B_5
$s_i = 6$	B_6	B_2	B_4	B_0				
$s_i = 7$					B_7	B_3	B_5	B_1

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding possible transition labels are given in the table. Blank entries represent transitions that are not allowed.

Table 5. A 4-State 16-QAM Trellis Code

	$s_e = 0$	$s_e = 1$	$s_e = 2$	$s_e = 3$
$s_i = 0$	$Q_{1,3}, Q_{3,3}, Q_{1,1}, Q_{3,1}$	$Q_{0,0}, Q_{0,2}, Q_{2,0}, Q_{2,2}$		
$s_i = 1$			$Q_{0,1}, Q_{0,3}, Q_{2,1}, Q_{2,3}$	$Q_{1,0}, Q_{1,2}, Q_{3,0}, Q_{3,2}$
$s_i = 2$	$Q_{0,0}, Q_{0,2}, Q_{2,0}, Q_{2,2}$	$Q_{1,3}, Q_{3,3}, Q_{1,1}, Q_{3,1}$		
$s_i = 3$			$Q_{1,0}, Q_{1,2}, Q_{3,0}, Q_{3,2}$	$Q_{0,1}, Q_{0,3}, Q_{2,1}, Q_{2,3}$

Note: The states s_i and s_e are, respectively, the beginning and ending states. The corresponding possible transition labels are given in the table. Blank entries represent transitions that are not allowed.

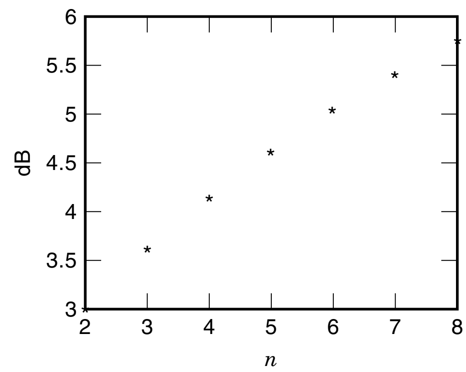

Figure 9. Asymptotic coding gain of coded 8-PSK over uncoded 4-PSK (number of states = 2^n). Coding gain represents the improvement in the performance of the coded system over that of the uncoded system.

An example of an M-TCM code is given in Ref. 16, which is a four-dimensional trellis code known as the Wei code (17).

RESEARCH ACTIVITIES

An active area of theoretical research is studying the tradeoff between the complexity and coding gain of trellis codes. In essence, we would like to see trellis codes with lower complexity of decoding and higher coding gain. Much effort has been put into finding solutions to this problem, but only meager improvements have been observed over the codes constructed in the original paper of Ungerboeck–Csjaka.

A second active area is to find suboptimal algorithms for decoding trellis codes which give performance close to that of the optimum Viterbi algorithm. Numerous papers have been written on this topic proposing reduced complexity algorithms including the sequential decoding algorithm and the *M*-algorithm (18). These decoding algorithms perform close to optimal but do not seem promising due to other implementation problems including the problem with buffer overflow.

Another research area is to combine mathematical objects called *lattices* with trellis codes (14). These theoretically achieve higher coding gains but have other implementation problems including the design of slicer and increased decoding complexity.

Trellis ideas were also applied to quantization giving rise to *trellis-coded quantization* which can be used to quantize various sources (19,20).

In general, we believe that a fruitful area of research may be the study of implementation issues of trellis codes over channels with ISI and non-Gaussian channels in the presence of various impairments due to practical situations. There is a well-established body of literature on this topic (21,22) but we believe that there is a lot more to be done.

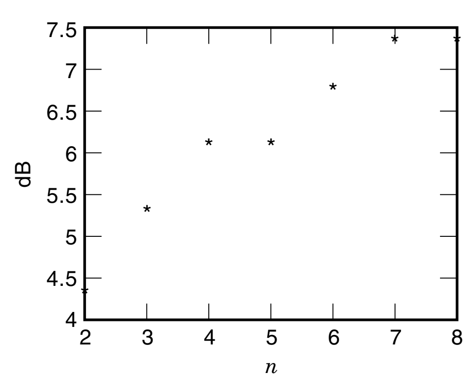

Figure 10. Asymptotic coding gain of coded 16-QAM over uncoded 8-PSK (number of states = 2^n). Coding gain represents the improvement in the performance of the coded system over that of the uncoded system.

BIBLIOGRAPHY

1. G. D. Forney, Trellises old and new, in *Communications and Cryptography: Two Sides of One Tapestry,* R. E. Blahut et al. (eds.). Dordrecht, The Netherlands: Kluwer, 1994.
2. L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimum decoding of linear codes for minimizing symbol error rates, *IEEE Trans. Inf. Theory,* **IT-20**: 284–287, 1974.
3. F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes,* New York: Elsevier, 1996.
4. J. L. Massey, Coding and modulation in digital communications, *Proc. 1974 Int. Zürich Seminar on Digital Commun.,* E2(1)–(4), Mar. 1974.
5. G. Ungerboeck and I. Csajka, On improving data-link performance by increasing the channel alphabet and introducing sequence coding, *Int. Symp. Inform. Theory,* Ronneby, Sweden, June 1976.
6. G. Ungerboeck, Channel coding with multilevel/phase signals, *IEEE Trans. Inf. Theory,* **IT-28**: 55–67, 1982.
7. C. E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.,* **27**: 379–423, 1948.
8. CCITT, A family of 2-wire, duplex modems operating on the general switched telephone network and on leased telephone-type circuits, Recommendation V.32, 1984.
9. CCITT, 14400 bits per second modem standardized for use on point-to-point 4-wire leased telephone-type circuits, Recommendation V.33, 1988.
10. ITU-T, A modem operating at data signaling rates of up to 33600 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits, Recommendation V.34, 1996.
11. G. Ungerboeck, Trellis-coded modulation with redundant signal sets part II: State of the art, *IEEE Commun. Magazine,* **25**: 12–21, 1987.
12. E. A. Lee and D. G. Messerschmitt, *Digital Communication,* Boston: Kluwer, 1988.
13. J. G. Proakis, *Digital Communications,* New York: McGraw-Hill, Inc. 1989.
14. E. Biglieri et al., *Introduction to Trellis-Coded Modulation with Applications,* New York: Macmillan, 1991.
15. H. O. Pollak and H. J. Landau, Prolate spheroidal wave functions, Fourier Analysis and uncertainty II, *Bell Syst. Tech. J.,* **40**: 65–84, 1961.
16. H. J. Landau and H. O. Pollak, Prolate spheroidal wave functions, fourier analysis and uncertainity III: The dimension of the space of essentially time and band-limited signals, *Bell Syst. Tech. J.,* **41**: 1295–1366, 1962.
17. L. F. Wei, Trellis-coded modulation with multi-dimensional constellations, *IEEE Trans. Inf. Theory,* **IT-33**: 483–501, 1987.

18. J. M. Wozencraft and B. Reiffen, *Sequential Decoding,* Cambridge, MA: MIT Press, 1961.
19. M. W. Marcellin and T. R. Fischer, Trellis-coded quantization of memoryless and Gauss-Markov sources, *IEEE Trans. Commun.,* **38**: 82–93, 1990.
20. M. Wang and T. R. Fischer, Trellis-coded quantization designed for noisy channels, *IEEE Trans. Inf. Theory,* **40**: 1792–1802, 1994.
21. D. Divsalar and M. K. Simon, The design of trellis-coded MPSK for fading channel: Performance criteria, *IEEE Trans. Comm.,* **36**: 1004–1012, 1988.
22. D. Divsalar and M. K. Simon, The design of trellis-coded MPSK for fading channel: Set partitioning for optimum code design, *IEEE Trans. Commun.* **36**: 1013–1021, 1988.

HAMID JAFARKHANI
VAHID TAROKH
AT&T Labs

TRELLIS CODES. See TRELLIS-CODED MODULATION.

TRENDS IN SYSTEMS ENGINEERING. See SYSTEMS ENGINEERING TRENDS.

TRENDS, SYSTEMS ENGINEERING. See SYSTEMS ENGINEERING TRENDS.

TRIANGLE WAVE GENERATION. See RAMP GENERATOR.