P^2 Codes: Pragmatic Trellis Codes Utilizing Punctured Convolutional Codes

A single convolutional code of fixed rate can be punctured to form a class of higher rate convolutional codes. The authors extend this pragmatic approach to the case where the core of the trellis decoder is a Viterbi Decoder for a punctured version of the de facto standard, rate 1/2 convolutional code.

Jack Keil Wolf and Ephraim Zehavi

Error correction codes come in two flavors: block codes and convolutional codes. The most popular method of combining modulation and coding has been termed *trellis coded modulation* [1] (or more simply trellis codes). *Pragmatic trellis codes* were introduced in a previous *IEEE Communications Magazine* article [2] whereby the encoders and decoders for trellis codes were obtained by making slight modifications in a readily available convolutional encoder and decoder. In particular, the article described how the Viterbi decoder for the de facto industry and government standard, rate 1/2, 64-state, *convolutional* code could be modified to become the decoder for trellis codes for 8-PSK and 16-PSK modulation. These pragmatic trellis codes performed almost as well as the best trellis codes of comparable complexity (for an AWGN channel). Indeed, a pragmatic trellis code on a chip has been available for several years for 8-PSK and 16-PSK modulation.

Pragmatic codes for PAM and QAM signal constellations were also described [2]. The approach suggested was to treat the QAM constellation as the product of two orthogonal PAM constellations and to code each of these separately. This approach, which was limited to rectangular or square QAM constellations, required quadrupling the number of signal points. Since decoding was accomplished by decomposing the two-dimensional QAM constellation into two PAM signals, two Viterbi decoders were required. Furthermore, the technique did not apply to the familiar cross constellation.

One characteristic of convolutional codes that has been exploited in commercially available products is that a single convolutional code of fixed rate (say rate 1/2) can be *punctured* to form a class of higher rate convolutional codes (say rate 2/3, 3/4, etc.). In this article, we extend this pragmatic approach to the case where the core of the trellis decoder is a Viterbi Decoder for a punctured version of the de facto standard, rate 1/2 convolutional code. We show that this approach leads to a wide class of high-rate pragmatic punctured (or P^2) trellis codes for both PSK and QAM modulation. For PSK modulation, we show that the pragmatic punctured trellis codes lead to more efficient codes than the unpunctured pragmatic trellis codes. For QAM modulation, we demonstrate trellis codes which require only twice the number of signal points, which use only one Viterbi decoder, and which apply to rectangular, square or cross constellations.

In a recent patent [3] a rate 4/5 pragmatic (but unpunctured) trellis code was described for the 32-QAM cross constellation, which required only twice the number of signal points and which used only one Viterbi decoder. Using our P^2 approach, one is able to obtain almost the same performance as this code but with a higher rate (a rate 9/10) code.

JACK KEIL WOLF is the Stephen O. Rice Professor in the Electrical and Computer Engineering Department at the University of California, San Diego and a part-time employee of QUALCOMM, Inc.

EPHRAIM ZEHAVI is assistant general manager and vice president of technology of QUALCOMM Israel.

Pragmatic Trellis Codes

Consider a binary, rate 1/2 convolutional encoder with 2^S states. Assume that the encoder is in the present state $\underline{S} = (X_1, X_2, \ldots, X_{S-1}, X_S), = (\underline{X}, X_S)$, where the X_i, $i=1,2,\ldots,s$, are binary digits. Assume that if the input to the encoder is the binary digit A, then the encoder produces two binary output digits, say B_1 and B_2, which are linear functions of the input A and the components of the current state vector $\underline{S}$, and the next encoder state is $\underline{S}' = (A, \underline{X})$.

The convolutional encoder is described by a code trellis having 2^S states. Each (current) state in the code trellis is connected by branches to two (next) states. The two (next) states connected to state $(\underline{X}, X_S)$ are $(0, \underline{X})$ and $(1, \underline{X})$ corresponding to the inputs $A = 0$ and $A = 1$, respectively. The branch between two states is labeled by the two encoder outputs B_1 and B_2, which correspond to that encoder state transition. The infinite sequences produced by the encoder, called code words, are the binary sequences composed of the branch labels produced by transversing paths in the code trellis.

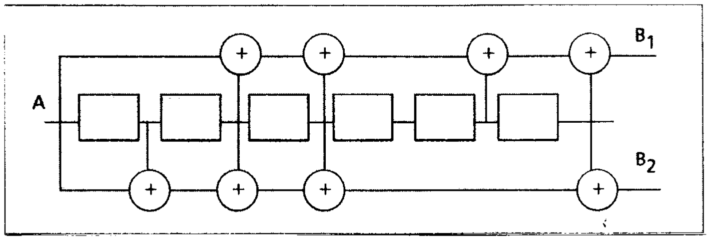
One useful measure of goodness of the convolutional code is its free Hamming distance, defined as the smallest number of binary digits by which two code words differ; these code words start in some common state and end in some common (but perhaps different) state. Sometimes the pair of encoder outputs, B_1 and B_2, along with some other uncoded bits are mapped into signal points in a 2-dimensional Euclidean space. In that case, the branches of the trellis are labeled by the corresponding points in the two-dimensional signal space and the measure of goodness of the code is defined as the minimum of the sum of the squared Euclidean distances between the corresponding signal points of two distinct code words that start in some common state and end in some common (but perhaps different) state. Although the above measure of goodness is sufficient for PSK modulation without normalization, for QAM modulation it is convenient to normalize the *free squared Euclidean* distance by dividing by the average energy of the signal points in the constellation.

The binary rate 1/2 convolutional code that has become an industry and government de facto standard has $2^6 = 64$ states. The outputs of the encoder, B_1 and B_2, for this code are given by the following equations:

$$B_1 = A + X_2 + X_3 + X_5 + X_6$$

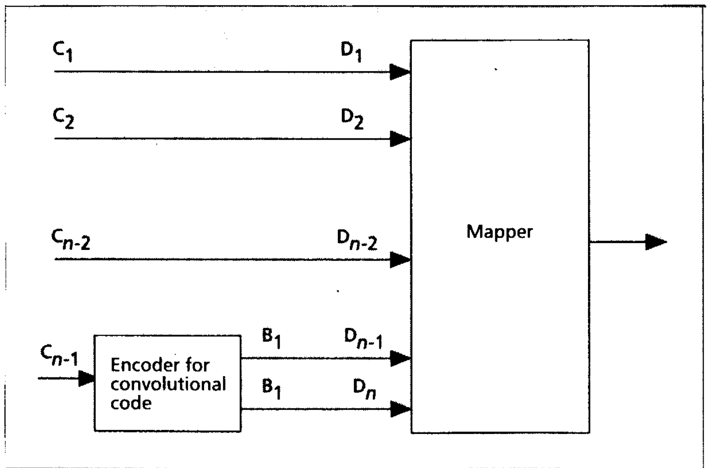
$$B_2 = A + X_1 + X_2 + X_3 + X_6.$$

The free Hamming distance of this de facto standard code is equal to 10. The encoder for this code is depicted in Fig. 1.



■ **Figure 1.** *Encoder for rate 1/2 de facto standard convolutional code.*

Assume that we desire a rate $(n-1)/n$ pragmatic trellis code for a signal constellation with 2^n points. We first take a binary $(n-1)$ vector, $C_1, C_2, \ldots, C_{n-1}$ and transform it into the binary n-vector, $D_1, D_2, \ldots, D_{n-1}, D_n$ in accordance with the rules: $D_i = C_i$ for $i = 1$ to $n-2$, and $(D_{n-1}, D_n) = (B_1, B_2)$, where B_1 and B_2 are the two outputs of the rate 1/2 encoder corresponding to the input $A = C_{n-1}$. This encoding rule describes a $(n-1)$-input, n-output binary encoder where the first $(n-2)$ inputs appear directly as outputs, and the last input, C_{n-1}, is encoded using the 2^S state rate 1/2 binary encoder to produce the remaining outputs. The n outputs are then mapped into one of the 2^n points in some signal constellation. The resultant pragmatic trellis encoder is shown in Fig. 2.



■ **Figure 2.** *Encoder for pragmatic trellis code.*

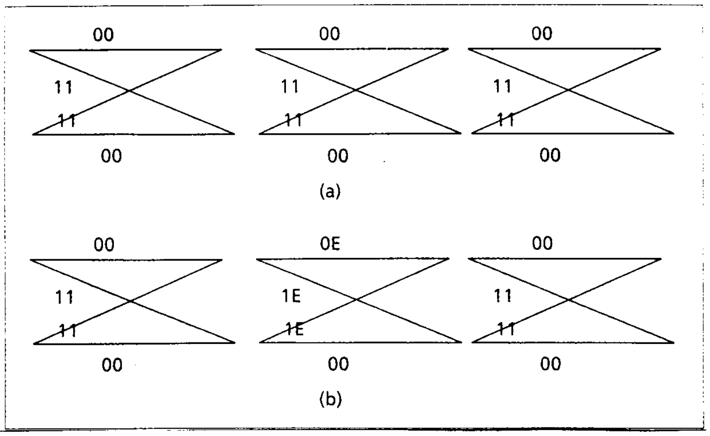
The code trellis for this pragmatic trellis encoder greatly resembles the code trellis for the convolutional code upon which it is based. It has the same number of states, (i.e., 2^S states) but has many more branches. Every branch in the convolutional code trellis is replaced by 2^{n-2} parallel branches, one for each of the possible values that $C_1, C_2, \ldots, C_{n-2}$ can take on. Each branch is labeled by its corresponding point in the signal constellation.

In finding the normalized free squared Euclidean distance of the code, two different cases need be considered. One case is that of a pair of code words that pass through the exact same sequence of states but differ in only one branch. That is, the two code words are identical except that one takes one of the parallel branches between two of the states while the other takes a different parallel branch between these same two states. The (unnormalized) squared Euclidean distance between such a pair of code words is equal to the squared Euclidean distance between the signal points on the two parallel branches in which the code words differ. For later use, we define $d^2_{\|}$ as the minimum squared Euclidean distance over all pairs of parallel branches.

The second case is a pair of code words that diverge from a common state and remerge at some common (but perhaps different) state, but at some point in their history pass through distinct states. Let us define d^2_{Tr} as the minimum squared Euclidean distance over all such pairs of code words. A reasonable design philosophy for the design of trellis codes is to balance the two quantities $d^2_{\|}$ and d^2_{Tr}.

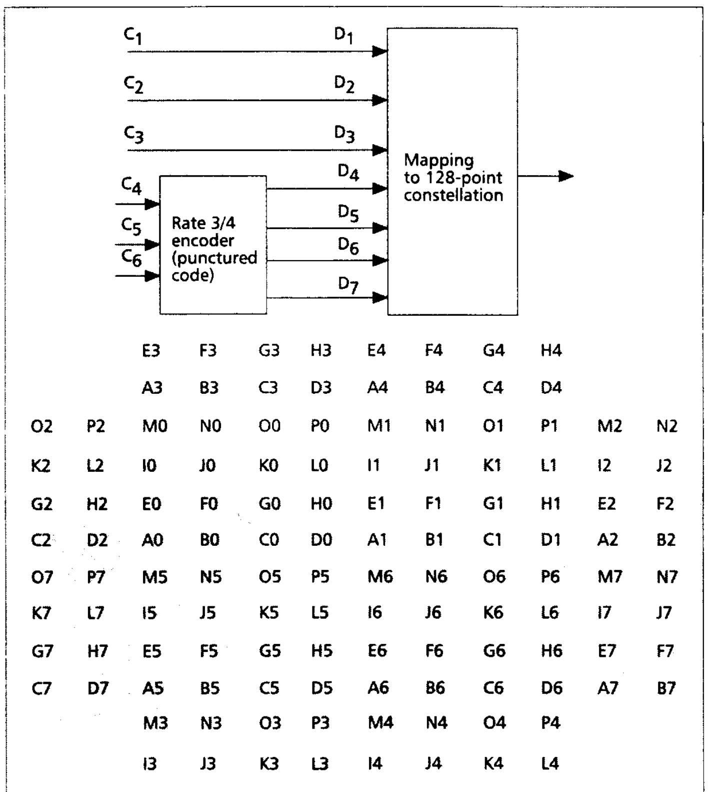
Punctured Convolutional Codes

Consider the de facto standard, rate 1/2 binary convolutional code as described in the previous section. Any two code words that are produced by the encoder when it starts in any particular state must differ in at least 10 binary digits. Now assume that every fourth code symbol is deleted from the code sequence. In particular, assume that the input to the encoder is thought of as the interleaving of odd and even streams of binary digits. When a binary digit from the odd stream is input into the encoder, the encoder produces the output pair B_1 and B_2. However, when a binary digit from the even stream is input into the encoder, the encoder only produces the single output B_1. The result is that the encoder produces three output (coded) binary digits for every pair of input (uncoded) binary digits. The resultant code has a rate equal to 2/3. The free Hamming distance of this rate 2/3 code is equal to 6. A portion of the trellis for the rate 1/2 convolutional code is shown in Fig. 3a and its corresponding rate 2/3 punctured code is shown in Fig. 3b.



■ **Figure 3.** *A portion of the trellis for the rate 1/2 convolutional code and its corresponding rate 2/3 punctured code.*

Higher rate codes with correspondingly smaller free Hamming distances can be obtained by omitting (or puncturing) more output coded binary digits [4-6]. For example, by properly puncturing the de facto standard, a rate 3/4 punctured code with free Hamming distance 5 can be obtained. The practical importance of this puncturing technique is that a single Viterbi decoder matched to the rate 1/2 convolutional code can be used to decode *all* higher rate codes obtained by puncturing this one code. Indeed many implementations of the encoder/decoder chip for the de facto standard code allows for encoding and decoding many punctured versions of this code.



■ **Figure 4.** *Encoder and signal constellation for rate 6/7 punctured pragmatic trellis code for 128-QAM. (The 128-points are subdivided into 16 groups, each group denoted by a letter {A,B, …,P}. Each group contains eight points identified by an 8 integer {0, 1, …, 7}.)*

Rate 6/7 P^2 Code for QAM with 128 Signal Points

Consider the trellis encoder and the 128 point QAM signal constellation shown in Fig. 4. Every T seconds, six information bits are encoded into seven coded binary digits, which then are mapped into one of the points in the 128-point QAM signal constellation. The 128 points are to be thought of as being composed of 16 groups, each group containing eight points. Of the seven encoded binary digits, the first three are simply the first three uncoded information bits and the last four are the four outputs from the rate 3/4 convolutional code obtained by puncturing the rate 1/2 de facto standard code. The last four specify to which group the signal point belongs and the first three specify the specific point within group (Table 1 and Fig. 4). Puncturing is accomplished by inputting three binary digits into the rate 1/2 encoder and first choosing two output binary digits, B_1 and B_2, then choosing only the one output binary digit, B_1, and finally choosing only the second output binary digit, B_2. The process then is repeated.

If the 128 signals are placed on the vertices of a regularly spaced square grid with the horizontal and vertical distance between grid lines equal to two, the average energy of these 128 points is equal to 82. The quantities $d^2_{\|}$ and d^2_{Tr} can be shown to satisfy the relations $d^2_{\|} = 64$, and $d^2_{Tr} \geq 20$, so that the normalized free squared Euclidean distance is at least $20/82 = 0.244$.

Since each transmission of a 128-ary signal carries 6 bits of information, we will compare the perfor-

4-tuple	Signal
0000	A
0001	B
0011	C
0010	D
0100	E
0101	F
0111	G
0110	H
1100	I
1101	J
1111	K
1110	L
1000	M
1001	N
1011	O
1010	P

■ **Table 1.** *Mapping between signals and coded 4-tuples.*

mance of our system with an uncoded QAM system with 64 points. Assuming that these 64 signals form an 8-by-8 array on this same grid (centered at the origin), the average energy of these equally probable 64 signals is equal to 42. The minimum squared Euclidean distance between any two of these 64 uncoded signal points is equal to 4. Thus the normalized free squared Euclidean distance of the uncoded system is 4/42=0.096.

To compare the P^2 trellis coded system with the uncoded system, we define the asymptotic coding gain (ACG) as the ratio of the normalized free squared Euclidean distances of the two systems (measured in db). Thus, for this case

$$ACG = 10 \log_{10} (0.244 / 0.096) = 4.05 \text{ db}.$$

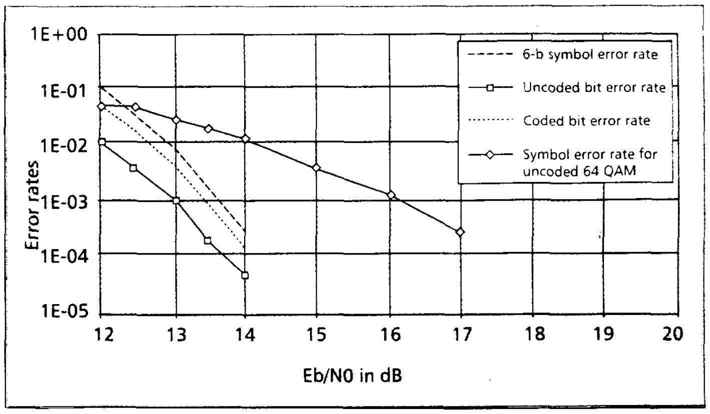
Simulation results for this code where the only channel impediment is additive white Gaussian noise is shown in Fig. 5. Four curves are presented. Two curves are plots of the 6-b symbol error probability, one for uncoded 64 QAM and the other for the rate 6/7 P^2 code for 128-QAM. Note that at a symbol error rate of 3×10^{-4}, the P^2 coded system has a coding gain of approximately 3 db. This is less than the promised 4.05 db ACG but note that the curves are still diverging at this high error rate. Two curves of bit error probability are also shown, one for the uncoded bits (the information bits that do not pass through the convolutional encoder) and the other for the coded bits (the information bits that are encoded by the convolutional encoder). Note the uncoded bits are more reliable than the coded bits. This is to be expected, since $d^2_{\parallel}$ is larger than d^2_{Tr} for this P^2 code. This difference could be exploted in some applications where some bits are more important than other bits.

Assume that the Viterbi decoder acting on the unpunctured rate 1/2 convolutional code can decode at a maximum information rate of 20 Mb/s. This same decoder chip, when used as the basic engine for the pragmatic punctured code, would be able to decode at a maximum information rate of 40 Mb/s.

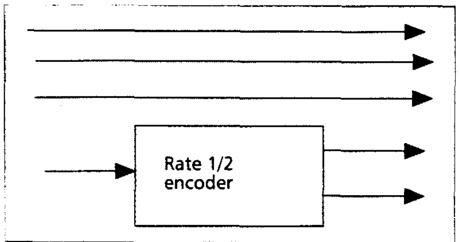
A P^2 Code for QAM with 32 Signal Points

Our second example is chosen to illustrate the advantage of using punctured rather than unpunctured pragmatic trellis codes. We begin with a description of a patented [3] rate 4/5 (unpunctured) pragmatic trellis code for 32-QAM modulation whose encoder is shown in Fig. 6. The details of the mapping are unimportant to us here except to state that $d^2_{\parallel} = 16$ and $d^2_{Tr} = 40$ for this code.

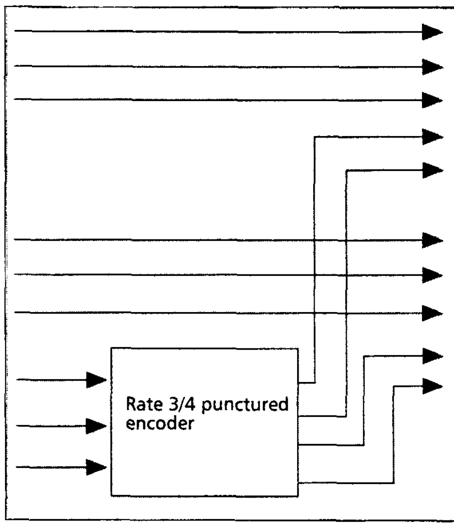
We next consider the design of a rate 9/10 P^2 trellis code for this same 32-point cross QAM constellation. In the P^2 code, 9 information bits are encoded into 10 coded binary digits (Fig. 7), which are then transmitted by sending two signals from the 32-QAM constellation. The top six input leads carry uncoded information bits while the bottom three input leads are input to the encoder of the de facto standard rate 1/2 convolutional code punctured to rate 3/4. The top five output lines are mapped into one 32-QAM signal while the bottom five output lines are mapped into a second 32-QAM signal. For each of the cluster of five output lines, we denote the three uncoded binary digits by an octal symbol from the set {0, 1, ..., 7} and the two coded binary digits by a symbol



■ **Figure 5.** *Performance of P^2 Code for 128-QAM.*



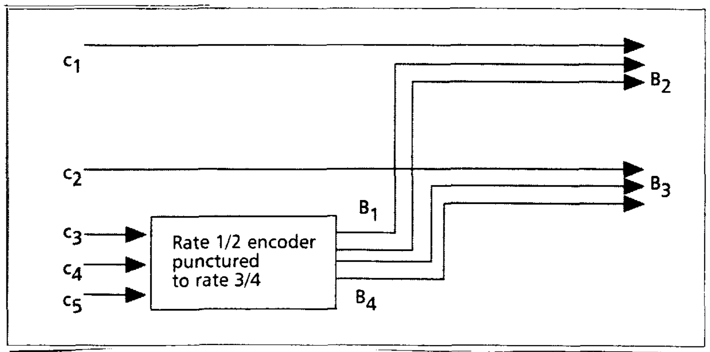
■ **Figure 6.** *Encoder for rate 4/5 trellis code for 32-QAM (no puncturing).*



■ **Figure 7.** *Encoder for rate 9/10 trellis code for 32-QAM (with puncturing).*

	D6	C7	D7	C6	
A0	B0	A1	B1	A2	B2
C0	D0	C1	D1	C2	D2
A3	B3	A4	B4	A5	B5
C3	D3	C4	D4	C5	D5
	B6	A7	B7	A6	

■ **Figure 8.** *Mapping for encoder of Fig. 7.*



■ **Figure 9.** *Rate 5/6 punctured trellis encoder for 8-PSK modulation.*

from the set {A,B,C,D}. Then the mapping by which these 5 lines choose one of the points from the 32-QAM cross constellation is shown in Fig. 8. Points with the same letter represent parallel paths in the trellis. It can be verified that $d^2_{\|} = 16$ and $d^2_{Tr} = 20$. Thus both this P^2 code and the rate 4/5 code have the same free squared Euclidean distance but the P^2 code is more bandwidth efficient.

A P^2 Code for 8-PSK

An example of a rate 5/6 P^2 code for 8-PSK is presented as follows. We assume the following specific mapping between binary digits and the phases of the PSK signal: 0° = 000, 45° = 001, 90° = 011, 135° = 010, 180° = 100, 225° = 101, 270° = 111, 315° = 110.

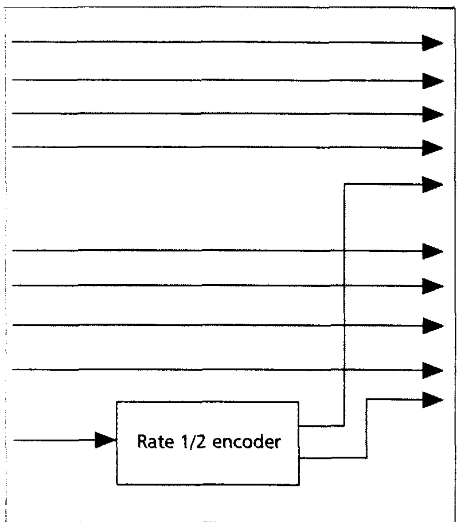
Consider the encoder shown in Fig. 9 with five input and six output lines. The 6 output lines are then mapped into two 8-PSK signals. Since uncoded 8-PSK can transmit information at 3 b/Hz, this trellis coded modulation will transmit information at 2.5 b/Hz.

Assuming that the 8-PSK points are uniformly placed on the unit circle, $d^2_{\|} = 4$. Since the rate 3/4 punctured code obtained from the de facto standard has free Hamming distance of 5, $d^2_{Tr} \geq 5(2*\sin(22.5°))^2 = 2.929$. Thus, the free squared Euclidean distance for this code is greater than or equal to 2.929. Note that uncoded QPSK, (which has an information rate of 2 b/Hz) has a minimum squared Euclidean distance of 2, so that the P^2 code has both a higher information rate and an ACG as compared to uncoded QPSK.

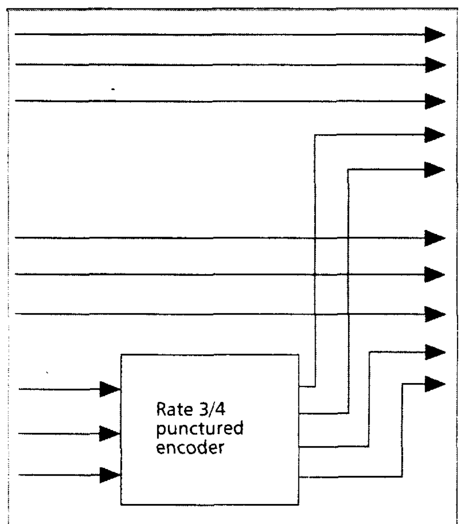
Comparison of Codes for PSK

Table 2 gives the minimum squared free Euclidean distance for a number of pragmatic trellis codes for PSK modulation. The table includes results for non-punctured as well as punctured codes. For PSK modulation with 2^k signals, the minimum squared Euclidean distance is given by the formula

$$d^2(k) = \min(d_p 4\sin^2(360°/2^{k+1}), 4\sin^2(360°/2^{p+1})).$$



■ **Figure 10.** *Encoder for rate 9/10 trellis code for 32-QAM (no puncturing).*



■ **Figure 12.** *Encoder for rate 9/10 trellis code for 32-QAM (with puncturing).*

	B0	A0	B1	A1	
B2	A2	B3	A3	B4	A4
A5	B5	A6	B6	A7	B7
B8	A8	B9	A9	Ba	Aa
Ab	Bb	Ac	Bc	Ad	Bd
	Ae	Be	Af	Bf	

■ **Figure 11.** *Mapping for encoder of Fig. 10.*

	D6	C7	D7	C6	
A0	B0	A1	B1	A2	B2
C0	D0	C1	D1	C2	D2
A3	B3	A4	B4	A5	B5
C3	D3	C4	D4	C5	D5
	B6	A7	B7	A6	

■ **Figure 13.** *Mapping for encoder of Fig. 12.*

Here p is the number of uncoded binary digits for each of the m 2^k-ary signals, and d_p is the free Hamming distance of the binary convolutional code punctured to a rate $((k\text{-}p)m\text{-}1)/(k\text{-}p)m$ code.

Higher Rate P² Codes

Higher rate codes can also be obtained by this P2 approach for both QAM and PSK modulation. As an example, we consider the design of a rate 9/10 code for a 32 point cross QAM constellation. To illustrate the advantage of the P^2 approach we first consider using a pragmatic approach with an unpunctured rate 1/2 code and then consider using a punctured version of the code.

For the unpunctured approach, consider the encoder shown in Fig. 10 where 9 information bits are first encoded into 10 coded binary digits and transmitted by sending two signals from the 32-QAM constellation. Denoting the four uncoded binary digits by a symbol from the set {0, 1, ..., 9, a, b, ..., f} and the coded bits by a symbol from the set {A,B}, the method by which each set of five lines is mapped to a QAM signal is shown in Fig. 11.

The minimum squared free Euclidean distance of this trellis code is governed by the minimum squared free Euclidean distance between parallel branches. Assuming that the points are placed on the vertices of a grid with horizontal and vertical distance between grid lines equal to 2, the minimum squared Euclidean distance of this code is equal to 8.

Now consider the encoder shown in Fig. 12, where the design is based upon a punctured convolutional code. For each bundle of five output lines, denoting the three uncoded binary digits by an octal symbol from the set {0, 1, ..., 7} and the two coded binary digits by a symbol from the set {A,B,C,D}, the method by which each set of five lines is mapped to a QAM signal is shown in Fig. 13. Assuming that the cross is on the same spaced grid, the minimum squared Euclidean distance between parallel branches, $d^2_{\parallel}$ is now 16. It is easily verified that d^2_{Tr} is equal to four times the minimum Hamming distance of the punctured trellis code. But for the rate 3/4 punctured code, this minimum Hamming distance is equal to 5, so that $d^2_{Tr} = 20$. Thus the minimum squared Euclidean distance of this code is now 16 which is a 3 db improvement over the unpunctured case.

Even though the code produced by the encoder of Fig. 10 does not have the distance of the code produced by the encoder of Fig. 12, it does have the advantage of a higher maximum decoding speed. Using a chip which can decode the rate 1/2 convolutional code at 20 Mb/s, the decoder for the code in Fig. 10 can decode at a maximum rate of 180 Mb/s (9 times 20 Mb/s) while the decoder for the code in Fig. 13 has a maximum rate of 60 Mb/s. The decoders for these two codes also use different branch metrics.

Summary

A brief introduction to the construction of efficient trellis codes for QAM, 8-PSK, and 16-PSK modulation is presented in this article. A number of subjects are not discussed, including the choice of the branch metrics in the decoder, how to achieve phase invariance, etc. A discussion of these subjects hopefully will appear in a subsequent article.

Modulation	Code Rate	Min. Squared Free Euclidean Distance (Not Punctured)	(Punctured)
QPSK	1 (uncoded)	2.000	n.a.
8-PSK	2/3 (Q1875)	4.000	n.a.
	5/6	2.000	2.929
	8/9	1.750	2.34
	1 (uncoded)	0.586	n.a.
16-PSK	3/4 (Q1875)	2.000	n.a.
	7/8	0.586	0.7612
	11/12	0.457	0.6088
	1 (uncoded)	0.152	n.a.
2^k-PSK	(km-1)/km	n.a.	$d^2(k)$

■ **Table 2.** *Minimum squared free Euclidean distance of trellis coded PSK.*

Acknowledgments

We gratefully acknowledge the intellectual stimulation of our coworkers at Qualcomm, Inc., who indirectly or directly contributed to this work. In particular, we acknowledge the contributions of Roberto Padovani and Andrew Viterbi, who were co-inventors of the pragmatic approach and Rajiv Vijayan, who carried out the simulations of the 128-QAM P^2 code.

References

[1] G. Ungerboeck, "Channel Coding With Multilevel/Phase Signals," *IEEE Trans. Info. Theory*, vol. IT-28, 1982, pp. 55-67.
[2] A. J. Viterbi, *et. al.*, "A Pragmatic Approach to Trellis-Coded Modulation," *IEEE Communications Magazine*, vol. 27, 1989, pp. 11-19.
[3] W. H. Paik, S. A. Lery, and C. Heegard, "Method and Apparatus for Communicating Digital Data Using Trellis Coded QAM," US. Patent No. 5,233,629, Aug. 3, 1993.
[4] LINKABIT Corporation, "Instruction Manual for LV7017MRB Rate 1/2 and 3/4 Convolutional Viterbi Decoder Circuit Board," San Diego, CA, 1972.
[5] J. B. Cain, G. C. Clark, and J. M. Geist, "Punctured Convolutional Codes of Rate (n-1)/n and Simplified Maximum Likelihood Decoding," *IEEE Trans. Info. Theory*, vol. IT-25, 1979, pp. 97-100.
[6] Y Yasuda, Y. Kashiki, and Y. Hirata, "High-Rate Punctured Convolutional Codes for Soft Decision Viterbi Decoding," *IEEE Trans. Commun.*, vol. COM-32, 1984, pp. 315-319.

Biographies

JACK KEIL WOLF [F '73] is the Stephen O. Rice Professor in the Electrical and Computer Engineering Department and the Center for Magnetic Recording Research at the University of California, San Diego. He also is a part-time employee of QUALCOMM, Inc., San Diego, California. He is a graduate of the University of Pennsylvania and Princeton University, where he received his Ph.D. in electrical engineering in 1960. Prior to joining UCSD, he held full-time faculty appointments at New York University, the Polytechnic Institute of Brooklyn, and the University of Massachusetts at Amherst. He was an officer in the USAF stationed at the Rome Air Development Center from 1960 to 1963. He has worked at RCA Laboratories and Bell Laboratories. His present research interests at UCSD are concerned with the application of modern signal processing to high density storage systems. At QUALCOMM, Inc., he is concerned with the design and development of wireless communication systems.

EPHRAIM ZEHAVI [M '87] received B.Sc. and M.Sc. degrees in electrical engineering from the Technion-Israel Institute of Technology, Haifa, in 1977 and 1981, respectively, and a Ph.D. in electrical engineering from the University of Massachusetts, Amherst, in 1986. From 1977 to 1983, he was an R & D engineer and group leader at the Department of Communication, Rafael, Armament Development Authority Haifa, Israel. From 1983 to 1985, he was a research assistant in the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. In 1985 he joined QUALCOMM Incorporated, as a senior engineer, where he was involved in the design and development of satellite communication systems, VLSI design of Viterbi decoder chips. From 1988 to 1992, he was on the faculty of the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa. In 1992 he rejoined QUALCOMM Incorporated, where he was involved in the design of PCS CDMA system. He is currently assistant general manager and vice president of technology of QUALCOMM Israel.