# Iterative Viterbi Decoding, Trellis Shaping, and Multilevel Structure for High-Rate Parity-Concatenated TCM

Qi Wang, Lei Wei, *Senior Member, IEEE*, and Rodney A. Kennedy, *Member, IEEE*

***Abstract*—We define and apply a new algorithm called the iterative Viterbi decoding algorithm (IVA) to decode a high-rate parity-concatenated TCM system in which a trellis code is used as the inner code and a simple parity-check code is used as the outer code. With trellis shaping, the IVA can achieve a performance 1.25 dB away from the Shannon limit at a BER of 3 $\times 10^{-5}$ with low complexity. By augmenting the system with a binary BCH code, the error floor can be reduced to $10^{-9}$ with very little additional cost.**

***Index Terms*—Iterative decoding, multilevel coding, serial concatenated coding, trellis shaping, trellis-coded modulation, viterbi algorithm.**

## I. Introduction

TURBO CODE has achieved remarkable error performance close to the Shannon capacity limit [5]. Following the principles of turbo code, researchers have designed several varieties of compound codes which are composed of interacting constituent codes [4], [7]. Attention has also been focussed on variants called turbo TCM [13] and multilevel codes [14], [15]. For such codes, many high performance iterative decoding algorithms have been developed.

In [16], Cabral, Costello, and Chevillat noted significant similarity between the turbo decoding method and the bootstrap iterative decoding method developed in the 1970's by Jelinek and Cocke [17], [18]. Bootstrap decoding is a method which imposes algebraic constraints on streams of convolutionally encoded information sequences so as to gather *extrinsic* information from other streams when one stream is decoded. In [19] Wei extended the results of [16], [17] to near optimally decode the bootstrap case using long convolutional codes. One of the simplified bootstrap algorithms, which only uses the Viterbi algorithm, was given in [19] and named as BIVA in [20]—it is now called iterative Viterbi algorithm (IVA) in this paper. The open problem of how to apply the IVA to trellis codes is addressed in this paper.

In [25]–[27], several shaping techniques were developed to achieve shaping gains by using nonequiprobable signaling. In this paper we show that the parity-concatenated TCM system can be modified to accommodate the trellis shaping scheme and full shaping gains can be achieved. In addition, to mitigate the errors dominated by the uncoded bits, we show that a multilevel structure can be constructed by combining the concatenated TCM with a simple binary BCH block code to significantly reduce the error floor level.

The paper is organized as follows. In Section II, the principle of encoding a parity-concatenated trellis code is described. Two variants of parity-concatenated structure are also proposed. We then present the IVA in Section III. A simple example is given to illustrate the operation of the IVA. In Section IV, we show that trellis shaping and a multilevel structure utilizing a BCH code can be combined with the parity-concatenated TCM to further reduce the bit error rate. An analysis of the error floor is also presented. In Section V, simulation results of the IVA are reported, and the conclusions can be found in Section VI.

## II. Construction of Parity-Concatenated Trellis Codes

In this section, we first introduce the parity-concatenated trellis code and then describe two variants of the parity-concatenated structure.

The parity-concatenated trellis code is a serial-concatenated code, in which a simple parity-check code is concatenated with an Ungerboeck trellis code. As illustrated in Fig. 1, $(m-1)\times l$ information symbols $U_{j,i}$ $(j=1,2,\cdots,m-1;\ i=1,2,\cdots,l)$ form the first $(m-1)$ packets with length $l$. Each symbol $U_{j,i}$ is represented by $k$ bits $u_{j,i}^1 u_{j,i}^2 \cdots u_{j,i}^k$. Of these $k$ bits, $\tilde{k}$ bits, $u_{j,i}^1 \cdots u_{j,i}^{\tilde{k}}$, will be encoded by a $\tilde{k}/(\tilde{k}+1)$ convolutional encoder, and the remaining $(k-\tilde{k})$ bits, $u_{j,i}^{\tilde{k}+1}\cdots u_{j,i}^k$, will be left intact as uncoded bits, where $\tilde{k}\le k$. We then define the $m^{th}$ packet in terms of the information symbol $U_{m,i}$ $(i=1,\cdots,l)$ bits through

$$u_{m,i}^b = u_{1,i}^b \oplus u_{2,i}^b \oplus \cdots \oplus u_{m-1,i}^b, \quad b=1,2,\cdots,k \qquad (1)$$

where $\oplus$ denotes modulo-2 addition. It is clear that $m$ bits in a column follow a parity-check (PC) constraint, and hence $m$ symbols in a column also follow the PC constraint. Therefore, the symbols $U_{m,i}$ are referred to as *parity-check symbols* and

Paper approved by P. Hoeher, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received August 15, 1999; revised October 22, 2000 and March 7, 2001.

Q. Wang was with the Department of Engineering, FEIT, The Australian National University, ACT, Canberra, Australia. He is now with the 3G Mobile R&D Division, NEC Australia, Melbourne, Vic. 3170, Australia (e-mail: qiw@icpdd.neca.nec.com.au).

L. Wei was with the Department of Engineering, FEIT, The Australian National University, ACT, Australia. He is now with the Institute of Telecommunications Research, University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: l.wei@elec.uow.edu.au).

R. A. Kennedy is with the Department of Telecommunications Engineering, Research School of Information Sciences and Engineering, The Australian National University, Canberra ACT 0200, Australia (e-mail: rodney.kennedy@anu.edu.au).

Publisher Item Identifier S 0090-6778(01)00515-3.

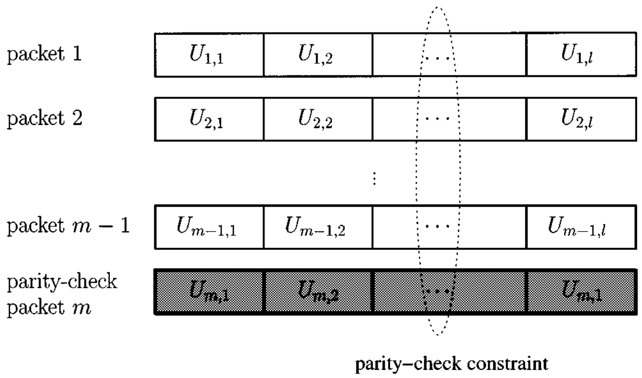


Fig. 1. Parity-check structure for one block symbols.

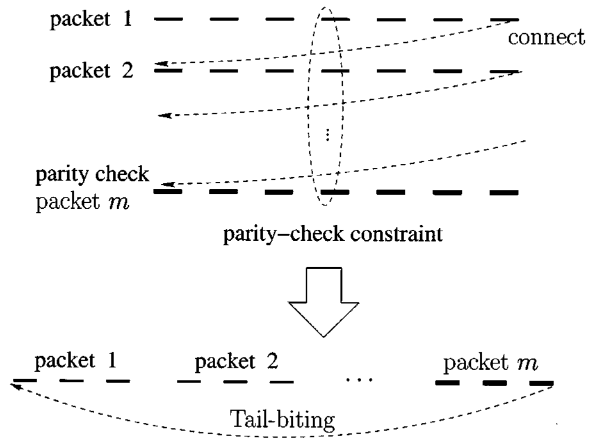


Fig. 2. Modified single-parity-check structure by connecting all the row packets.

the $m^{th}$ packet is called the *parity-check packet*, as shown in Fig. 1.

Since every bit in an information symbol is related to a PC constraint, we refer to it as a *full-parity-check* code. However, if only the coded bits of the $m$ symbols in a column satisfy the PC constraint, i.e, for $b = 1, 2, \cdots, \tilde{k}$, $\bigoplus_{j=1}^{m} u_{j,i}^{b} = 0$, then we refer to it as a *partial-parity-check* code, since only part of symbols are protected by the parity-check bits. For most of the Ungerboeck codes, a partial-parity-check constraint should be employed for achieving better performance in terms of approaching the Shannon limit [23].

We then feed the packets (or the rows), one at a time, into a rate $\tilde{k}/(\tilde{k}+1)$, $2^{\nu}$-state trellis encoder, where $\nu$ is the number of shift registers in convolutional encoder. Let $V_{j,i}$ denote the output symbol corresponding to the input symbol $U_{j,i}$. Each $V_{j,i}$ symbol consists of $(k+1)$ bits, $v_{j,i}^{1} \cdots v_{j,i}^{\tilde{k}+1} v_{j,i}^{\tilde{k}+2} \cdots v_{j,i}^{k+1}$, in which the first $(\tilde{k}+1)$ bits, $v_{j,i}^{1} \cdots v_{j,i}^{\tilde{k}+1}$, are the output bits from the convolutional encoder, and the remaining $(k-\tilde{k})$ bits, $v_{j,i}^{\tilde{k}+2} \cdots v_{j,i}^{k+1}$, are the uncoded bits. Finally, each output symbol $V_{j,i}$ is mapped (according to Ungerboeck's set-partitioning rule) into a signal constellation to produce a modulated signal. Only rectangular QAM constellations are considered in this paper.

Due to the linearity of convolutional encoding[1], we then have

$$v_{1,i}^{b} \oplus v_{2,i}^{b} \oplus \cdots \oplus v_{m,i}^{b} = 0, \quad i = 1, 2, \cdots, l \tag{2}$$

where $b = 1, 2, \cdots, k+1$ for full-parity-check code, and $b = 1, 2, \cdots, \tilde{k}+1$ for partial-parity-check code. The PC constraint in (2) is very important for constructing the iterative Viterbi decoding algorithm in Section III.

Based on the code structure given in Fig. 1, two variants of parity-concatenated code structure can be built up as follows.

### *A. Modified Single-Parity-Check Structure*

In Fig. 1, we have described a parity-concatenated trellis code, in which a single-bit parity-check code is applied. This structure is then referred to as a single-parity-check structure. To reduce the rate loss, we may apply a tail-biting technique [12] to each packet. Developing the idea further, we propose that a modified single-parity-check structure can be built up by connecting all the row packets into one super packet. In other words, instead of terminating each row packet, the symbols of all row packets will continue to be fed into the encoder until the last symbol. Then the trellis at the last symbol in the last row packet is tail-bitten to the first symbol in the first row packet. Fig. 2 illustrates this modification.

We know that the PC constraint is essential for decoding the parity-concatenated trellis code. After connecting all the packets into one super packet, it can be shown that the PC constraint is still valid provided a feed-forward encoder is used, see [21].

### *B. Double-Parity-Check Structure*

We can reapply the parity-check constraint on the modified single-parity-check structure to build up a double-parity-check structure. The main motivation is to provide a more powerful block code decodable by the IVA. The procedure for constructing a double-parity-check structure is described as follows.

At first, $(q-1)$ blocks of $V$ symbols are stored by row in the first $(q-1)$ rows of a $q$-by-$(m \times l)$-symbol memory device, as shown in Fig. 3. Each row is partially protected and encoded by a tail-biting trellis encoder as described in the previous section. The $q$th row is constructed as follows. Each trellis-coded bit of the $V$ symbol in the $q$th row is the even parity-check bit of coded bits of the first $(q-1)$ rows in the corresponding column. Each uncoded bit of the $V$ symbol in the $q$th row carries information in analogy to the uncoded bits in the single-parity-check structure. Last, the symbols are read out by row and mapped into a constellation for transmission.

In the double-parity-check structure, there are two types of parity-check constraints[2] used in the IVA decoder: (a) a horizontal PC constraint provides the parity check given in Fig. 2 for each super packet; and (b) a vertical PC constraint provides the parity check for each column in the structure given in Fig. 3.

[1]All the convolutional (or trellis) encoders considered in this paper are assumed to be linear which guarantees that the parity-check constraints among the symbols of packets also exist for coded symbols after the encoding process.

[2]A third type of parity-check constraint can be contemplated for the double-parity-check structure which corresponds to the combination of row and column check sums. However, it is found that this posed computational and technical difficulties, and did not significantly enhance the performance.

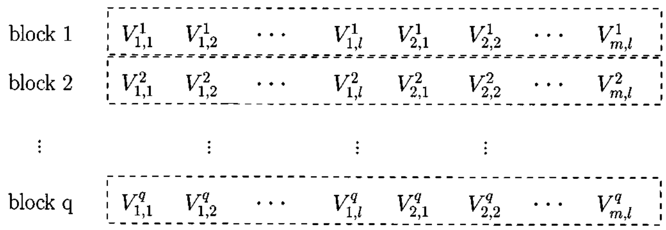


Fig. 3. Double-parity-check structure.

## III. Iterative Viterbi Algorithm for Parity-Concatenated Trellis Code

The IVA is a simplified min-sum algorithm [23]. In [19], [20], Wei has presented the IVA for decoding the concatenated convolutional codes, and performance close to the Shannon limit has been achieved. However, we notice that the IVA developed for convolutional codes cannot be simply extended to trellis codes. The difficulty is due to the computational complexity of the likelihood $\lambda_{ij,parity}$ [19] which is an essential parameter in the IVA. In this section, we will present the IVA for parity-concatenated trellis codes. For simplicity, we will only focus on the case of a single-parity-check structure.

The key concept of the IVA is using the hard decisions of other rows together with the parity-check constraints at the current iteration to update the branch metric for the next iteration of VA decoding. Thus, the operation of the IVA is exactly same as that of the standard VA except in the details of updating the branch metrics which is considered next.

Let $R_{j,i}$ $(j = 1, 2 \cdots, m;\ i = 1, 2, \cdots, l)$ denote the $i^{th}$ received signal in the $j^{th}$ packet at the receiver, and $V_{j,i} \equiv (V_{j,i}^{(p)}; V_{j,i}^{(np)}) = (v_{j,i}^1 \cdots v_{j,i}^{\tilde{k}+1}; v_{j,i}^{\tilde{k}+2} \cdots v_{j,i}^{k+1})$ denote the $i^{th}$ coded symbol of the $j^{th}$ packet, where $V_{j,i}^{(p)} = (v_{j,i}^1 \cdots v_{j,i}^{\tilde{k}+1})$ is the output bit vector from the convolutional encoder, and $V_{j,i}^{(np)} = (v_{j,i}^{\tilde{k}+2} \cdots v_{j,i}^{k+1})$ is the uncoded part. If only the coded bits are protected, then only $V_{j,i}^{(p)}$ satisfies the PC constraint in the coded symbol. Suppose that the $1^{st}, 2^{nd}, \cdots, (m-1)^{th}, m^{th}$ packets have been decoded by the standard VA in the first iteration, and estimated values $\widehat{V}_{j,i}$ obtained. Then in the next iteration we can decode $m$ packets using the updated branch metrics.

Consider the update of the branch metric of the $i$th symbol in the 1st packet. Using the PC constraint among the received signals $R_{1,i}, \cdots, R_{m-1,i}$ and $R_{m,i}$, we can get the likelihood function for this symbol as follows:

$$\lambda'_{1,i} = -\log\left[P\left(R_{1,i}, \cdots, R_{m-1,i}, R_{m,i} | V_{1,i}^{(p)}; V_{1,i}^{(np)}\right)\right] \quad (3)$$

where $V_{1,i}^{(p)}$ and $V_{1,i}^{(np)}$ are the coded and uncoded parts of symbol $V_{1,i}$, respectively. Assuming $R_{1,i}, \cdots, R_{m-1,i}$ and $R_{m,i}$ are independent of each other[3], we then have

$$\begin{aligned}\lambda'_{1,i} \approx & -\log\left[P\left(R_{1,i}|V_{1,i}^{(p)}; V_{1,i}^{(np)}\right)\right] \\ & -\log\left[P\left(R_{2,i}, \ldots, R_{m-1,i}, R_{m,i}|V_{1,i}^{(p)}; V_{1,i}^{(np)}\right)\right] \\ = & \lambda_{1,i}^{(v)} + \lambda_{1,i}^{(p)} \end{aligned} \quad (4)$$

[3]Actually, $R_{1,i}, \cdots, R_{m,i}$ are weakly dependent due to the parity-check constraint among them.

where $\lambda_{1,i}^{(v)}$ denotes the branch metric value which is identical to the metric used in the standard VA, and $\lambda_{1,i}^{(p)}$ denotes the *extrinsic* metric value introduced by the PC constraint from the other packets. It is very difficult to precisely evaluate $\lambda_{1,i}^{(p)}$ for high rate TCM, but, if we use the decisions of the previous iteration, the computation can be significantly simplified.

It can be noticed that the original PC constraint on the $i^{th}$ symbols in one block ($m$ packets) is

$$V_{1,i}^{(p)} \oplus V_{2,i}^{(p)} \oplus \cdots \oplus V_{m,i}^{(p)} = 0, \quad i = 1, 2, \cdots, l. \quad (5)$$

Let $W_{3,i} = V_{3,i}^{(p)} \oplus \cdots \oplus V_{m-1,i}^{(p)} \oplus V_{m,i}^{(p)}$. The receiver replaces the $j^{th}$ $(j = 3, 4, \ldots, m)$ received packets by the $j^{th}$ estimated packets. So we can get the estimated PC constraint

$$\widehat{W}_{3,i} = \widehat{V}_{3,i}^{(p)} \oplus \cdots \oplus \widehat{V}_{m-1,i}^{(p)} \oplus \widehat{V}_{m,i}^{(p)} \quad (6)$$

where $\widehat{V}_{j,i}^{(p)}$ is the decision of $V_{j,i}^{(p)}$. If $\widehat{W}_{3,i} = W_{3,i}$, then we have

$$V_{1,i}^{(p)} \oplus V_{2,i}^{(p)} \oplus \widehat{W}_{3,i} = 0. \quad (7)$$

Therefore, $V_{2,i}^{(p)}$, the coded part of the $i^{th}$ symbol in the second packet, can be obtained through the PC constraint and the estimates of $(m-2)$ symbols, given the hypothesis $V_{1,i}^{(p)}$. However, there is no PC constraint among the uncoded part $V_{j,i}^{(np)}$ of the symbols. So the question is how to determine the uncoded part $V_{2,i}^{(np)}$ of the $i^{th}$ symbol in the second packet.

We know that the coded part $V_{2,i}^{(p)}$ decides which subset will be selected in the constellation, and the parallel transition error in the subset can be ignored at high SNRs. Therefore, after $V_{2,i}^{(p)}$ is determined, $V_{2,i}^{(np)}$ can also be decided by selecting one point $(V_{2,i}^{(p)}; V_{2,i}^{(np)})$ which is the closest to the received signal $R_{2,i}$ in this subset. Therefore, we have

$$\begin{aligned}\lambda_{1,i}^{(p)} &= -\log\left[P\left(R_{2,i}, \cdots, R_{m,i}|V_{1,i}^{(p)}; V_{1,i}^{(np)}\right)\right] \\ &\approx -\log\left[P\left(R_{2,i}, \cdots, R_{m,i}|V_{2,i}^{(p)} \oplus \widehat{W}_{3,i}; \widehat{V}_{2,i}^{(np)}\right)\right]. \end{aligned} \quad (8)$$

Here $V_{1,i}^{(np)}$ has been replaced by $\widehat{V}_{2,i}^{(np)}$. Exact calculation of (8) is complicated and practically impossible when $m$ is large. Also, by considering $R_{3,i}, \cdots, R_{m,i}$ are just weakly dependent on $R_{2,i}$, we can approximate (8) as

$$\lambda_{1,i}^{(p)} \approx -\log\left[P\left(R_{2,i}|V_{2,i}^{(p)} \oplus \widehat{W}_{3,i}; \widehat{V}_{2,i}^{(np)}\right)\right]. \quad (9)$$

Now the metric function $\lambda_{1,i}^{(p)}$ is approximately equal to the VA branch metric of the second packet, but "selected" by a parameter $\widehat{W}_{3,i}$. Thus the computation of (4) is simply the sum of the VA branch metric functions of the 1st and 2nd packets. It is worth mentioning here that $\lambda_{1,i}^{(p)}$ will not be the correct metric and the error propagation will result if $\widehat{W}_{3,i} \neq W_{3,i}$. The effect of error propagation can be reduced by scaling down the value

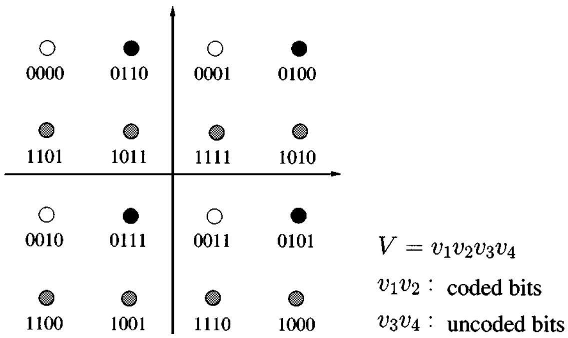


Fig. 4. Set partitioning and mapping on a 16-QAM constellation.

of $\lambda_{1,i}^{(p)}$ with a positive positive scaling factor $\alpha$ less than unity, i.e.,

$$\lambda'_{1,i} = \lambda_{1,i}^{(v)} + \alpha \cdot \lambda_{1,i}^{(p)}. \tag{10}$$

So far we have got the updated branch metrics for the symbols of the first packet. For the symbols of the $2^{nd}$, $3^{rd}$, $\cdots$, $m^{th}$ packet, the updated branch metrics can be obtained by similar means. After $\lambda'_{j,i}$ $(j = 1, \cdots, m; i = 1, \cdots, l)$ are obtained, the remaining procedures are exactly the same as those of the standard VA.

For a better understanding, let us go through an example to illustrate how to update the branch metrics for a symbol in the IVA. Consider a parity-concatenated trellis code using the single-parity-check structure with $k = 3$, $\tilde{k} = 1$ and $m = 4$. Assume that a linear rate 1/2 convolutional code is used and the output symbols are mapped into a 16-QAM constellation, which has been divided into four subsets denoted as 00, 01, 10, and 11, respectively (see Fig. 4). Also, only the coded bits $v_1 v_2$ follow the PC constraint. Now let us focus on updating the branch metrics of $V_{1,1}$, i.e., the first symbol of the first packet.

After the standard VA decoding in the first iteration, suppose that the estimates of $V_{1,1}$, $V_{2,1}$, $V_{3,1}$, and $V_{4,1}$ are $\widehat{V}_{1,1} = 0101$, $\widehat{V}_{2,1} = 0011$, $\widehat{V}_{3,1} = 1000$ and $\widehat{V}_{4,1} = 1110$, respectively. Also assume that the VA branch metrics of symbols $V_{1,1}$, $V_{2,1}$, $V_{3,1}$ and $V_{4,1}$ are $\lambda_{1,1}^{(v)} = \{2.5, 0.5, 2.5, 4.5\}$, $\lambda_{2,1}^{(v)} = \{0.25, 4.25, 10.25, 6.25\}$, $\lambda_{3,1}^{(v)} = \{4.5, 2.5, 0.5, 2.5\}$, $\lambda_{4,1}^{(v)} = \{2.5,\ 4.5, 2.5, 0.5\}$ for four subsets 00, 01, 10, and 11, respectively. Suppose the VA branch metrics of the 2nd packet will be viewed as *extrinsic* metrics, then the calculation of $\lambda_{1,1}^{(p)}$ follows two stages:

*Stage (a)* Calculate the estimated PC constraint $\widehat{W}_{3,1} = \widehat{V}_{3,1}^{(p)} \oplus \widehat{V}_{4,1}^{(p)} = \hat{v}_1 \hat{v}_2(\text{of } \widehat{V}_{3,1}) \oplus \hat{v}_1 \hat{v}_2(\text{of } \widehat{V}_{4,1}) = 10 \oplus 11 = 01$;

*Stage (b)* Then, $\lambda_{1,1}^{(p)}$ equals the branch metrics of the symbol $V_{2,1}$, i.e., 0.25, 4.25, 10.25 and 6.25, but reordered according to $V_{2,1}^{(p)} \oplus \widehat{W}_{3,1}$, i.e., these metrics now correspond to the "selected" subsets 01, 00, 11, and 10, respectively.

It is clear that the computation of $\lambda_{1,1}^{(p)}$ involves only $(m - 3 + 2^{\tilde{k}+1})$ modulo-2 additions provided the VA branch metrics $\lambda_{j,i}^{(v)}$ have already been stored during the first iteration. After $\lambda_{1,1}^{(p)}$ is obtained, the calculation of the updated branch metrics of the symbol $V_{1,1}$ will be straightforward. Let $\alpha$ equal 0.25 in this case, then $\lambda'_{1,1} = \lambda_{1,1}^{(v)} + 0.25\lambda_{1,1}^{(p)} = \{2.5,0.5,2.5,4.5\} + 0.25\{4.25,0.25,6.25,10.25\} = \{3.5625,0.5625,4.0625,7.0625\}$, for subsets 00, 01, 10 and 11, respectively.

So far, we have considered the branch metric function on a single-parity-check structure. In the double-parity-check structure, there are two types of PC constraints which produce two sorts of *extrinsic* metrics. The horizontal PC constraint provides the *extrinsic* metric given in (9) for each super packet, and the vertical PC constraint provides the *extrinsic* metric for each column in the structure given in Fig. 3. These two *extrinsic* metrics are almost independent each other. Therefore, for a double-parity-check structure, the updated branch metrics of one symbol will be approximately the sum of its original VA branch metric values and two *extrinsic* metric values. For instance, given the double-parity-check structure in Fig. 3, the updated branch metrics of the first symbol in block 1 will be

$$\lambda_{1,i}^{1'} = \lambda_{1,i}^{1(v)} + \alpha \cdot \left( \lambda_{1,i}^{1(p)(h)} + \lambda_{1,i}^{1(p)(v)} \right) \tag{11}$$

where $\lambda_{1,i}^{1(v)}$ is the original VA branch metric, $\lambda_{1,i}^{1(p)(h)}$ and $\lambda_{1,i}^{1(p)(v)}$ are the *extrinsic* metric values introduced by the horizontal and vertical PC constraints, respectively, and $\alpha$ is a scalar in analogy to (10).

Finally, we summarize the IVA for decoding the parity-concatenated trellis codes as follows.

- Step a) In the first iteration, calculate the VA branch metrics for each symbol, and then decode all the packets in the single- or double-parity-check structure using the standard VA without consideration of the PC constraints among those packets.
- Step b) In the next iteration, update the estimated PC constraint $\widehat{W}$ for each symbol based on the decisions of coded symbols from the previous iteration.
- Step c) Update the branch metrics for each symbol, which are the sum of original VA branch metrics and *extrinsic* branch metrics "selected" by $\widehat{W}$.
- Step d) Decode all the packets using the VA except that the updated branch metrics given in step (c) instead of original ones are applied.
- Step e) Repeat steps (b), (c), and (d) for several iterations until a stop criterion (which will be discussed next) is satisfied or a pre-set maximum number of iterations is reached.

There are some relevant issues on the IVA worth further discussion.

### A. Updating the Parity-Check Constraint $\widehat{W}$

In the derivation of updated branch metric used in the IVA, when the $j^{th}$ packet is decoded, the VA branch metric of the $(j+1)$th packet, $\lambda_{j+1,i}^{(v)}$, is supposed to be "combined" (added) with $\lambda_{j,i}^{(v)}$ of the $j$th packet. Through simulations, however, we find that randomly selecting a packet to be "combined" with the packet which is being decoded can achieve more than 0.1–0.2-dB gain, and also improves the convergence speed of the IVA.

### B. Stop Criteria in the IVA

The stop criteria in the IVA is given as follows. If the decisions of all coded symbols satisfy $\sum_{j=1}^{m} \widehat{V}_{j,i}^{(p)} = 0$ $(i = 1, \cdots, l)$ (in the single-parity-check structure) or $\sum_{y=1}^{q} \sum_{j=1}^{m} \widehat{V}_{y,j,i}^{(p)} = 0$ (in the double-parity-check structure), then the iterative decoding process will be terminated.

## IV. Trellis Shaping and Multilevel Code With Parity-Concatenated TCM

### A. Combining Trellis Shaping With Parity-Concatenated TCM

It has been recognized that shaping and coding are two separable and complementary components of TCM systems. In [25]–[27], it has been shown that a shaping gain can be achieved by using nonuniform, Gaussian-like signaling. One approach, called trellis shaping, was proposed by Forney [25]. It was shown that a simple 4-state shaping code can achieve about 1.0-dB shaping gain. In this paper, we find that trellis shaping can be combined with the parity-concatenated TCM to yield full shaping gain.

The means of combining trellis shaping with parity-concatenated TCM is the same as with conventional TCM.

### B. Combining Multilevel Code With Parity-Concatenated TCM

When applying partial parity checking to protect the coded bits, we may find that in many cases errors among the uncoded bits become dominant, which results in an error floor. Here, we show that the multilevel coding technique based on the set partitioning rule [30] can be utilized to reduce the uncoded bit errors.

Following the concepts of the multilevel scheme proposed by Imai and Hirakawa [30], the subsets formed by Ungerboeck's set partitioning rule is further binary partitioned, with one bit associated with such a partition. Then, a binary BCH block code $(n_b, k_b, q_b)$ can be chosen to protect this bit, where $n_b$ is the BCH codeword length, $k_b$ is the number of information bits and $q_b$ is the number of error bits that the code can correct. Specifically, $k_b$ bits which are taken from $k_b$ symbols (one bit per symbol) are encoded as a codeword with length $n_b$. Apparently, $(n_b - k_b)$ redundant bits are produced for every $k_b$ trellis symbols due to the introduction of a BCH code.

In the receiver, the received signals are decoded using multistage decoding [30]. First, the IVA decoder in each symbol interval decides upon the correct subsets of the constellation. Then, based on this decision, the BCH decoder in every $k_b$ symbol intervals finds the correct subsets formed by the binary partitioning. Finally, the uncoded bits are determined based on the results of the BCH decoder.

### C. Performance Analysis of Error Floor

In this subsection, we give upper bounds on the error floors of the parity-concatenated TCM system with and without trellis shaping. The premise is to suppose that the coded bits of trellis codes have been successfully decoded. Based on the calculated upper bound, the proper BCH code is then selected.

The upper bound for the error floor of the parity-concatenated TCM without shaping can be obtained through the union bound technique since the subsets normally are nonrectangular constellations. Assume a signal point $\mathbf{a}$ in the constellation $\Lambda_0$ is transmitted and decoded as $\mathbf{a}'$ in the same subset as $\mathbf{a}$. Suppose $\mathbf{a}'$ is one of the closest points to $\mathbf{a}$ and the squared distance between $\mathbf{a}$ and $\mathbf{a}'$ is $d^2_{\mathbf{a},\mathbf{a}'}$. Then the average bit error probability is upper-bounded as [31]

$$P_b(e) \le \frac{N_{\rm av}}{n} \cdot 4Q\left(\sqrt{\frac{d^2_{\mathbf{a},\mathbf{a}'}}{E_{av}} \frac{n \cdot E_b}{2N_0}}\right) \tag{12}$$

where $N_{\rm av}$ is the average number of error bits per symbol over all subsets, $E_{\rm av}$ is the average energy of the constellation, $n$ is the number of bits per symbol, and $E_b/N_0$ is the average SNR per bit.

If trellis shaping is combined with the concatenated TCM, then the bit error rate (BER) will be slightly effected by the trellis shaping decoder, which can be chosen to be feedback-free syndrome-former [25] and, therefore, only limited error propagation will be caused for the shaping coded bits. So after the trellis shaping decoding, the error bit rate is rectified as

$$P'_b(e) = P_b(e) \cdot L_{\rm av} \tag{13}$$

where $L_{\rm av}$ is the average length caused by the error propagation. It is also worth mentioning here that the effect of trellis shaping should be considered when calculating $E_{av}$ in (13).

According to the calculated upper bound, we can then derive the error probability when a BCH code is considered. For a BCH code with hard-decision decoding, the probability of a code word error is upper-bounded by the expression [31]

$$P_M \le P_M^{(max)} = \sum_{i=q+1}^{n_b} \binom{n_b}{i} p^i (1-p)^{n_b - i} \tag{14}$$

where $p$ is the error probability of binary digit protected by the BCH code.

## V. Numerical Results

In this section, we present simulation results for parity-concatenated trellis codes. The trellis codes are typical Ungerboeck codes and the notation, following that of [1], is in octal form. For all cases the rate loss due to the parity redundancy has been deducted from the $E_b/N_0$ computation. Each simulation trial was terminated if 100 block errors were obtained, or if the total number of bits processed reached $6 \times 10^8$.

Fig. 5 shows the performance of the IVA for the parity concatenated trellis codes using a double-parity-check structure with partial protection. The inner code is either a 16-state $(\nu = 4, (h^0, h^1, h^2) = (23, 04, 16))$ or a 256-state $(\nu = 8, (h^0, h^1, h^2) = (401, 056, 304))$ TCM code in a feed-forward form. Without the parity-check code, the spectral efficiency is 6 bits/*T*. With the parity-check code using $m = 20$, $l = 50$, and $q = 20$, the spectral efficiency is 5.805 b/*T*. The Shannon limit for this rate is 9.76 dB. Each block has a total of 20 000 symbols. A 16-state shaping code (assuming that the baseline constellation is a 128-point cross constellation) is applied in the simulation. The peak iteration number is set

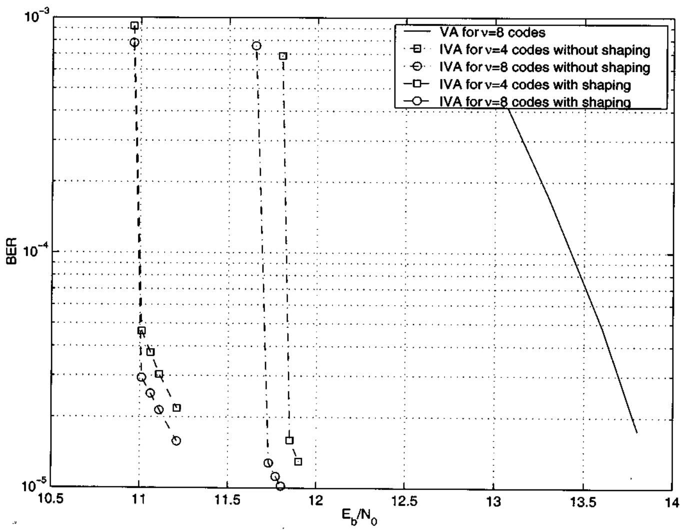


Fig. 5. BER performance of the 16-state and 256-state trellis codes using the IVA based on a double-parity-check structure at a spectral efficiency of 5.805 b/*T* with partial protection.

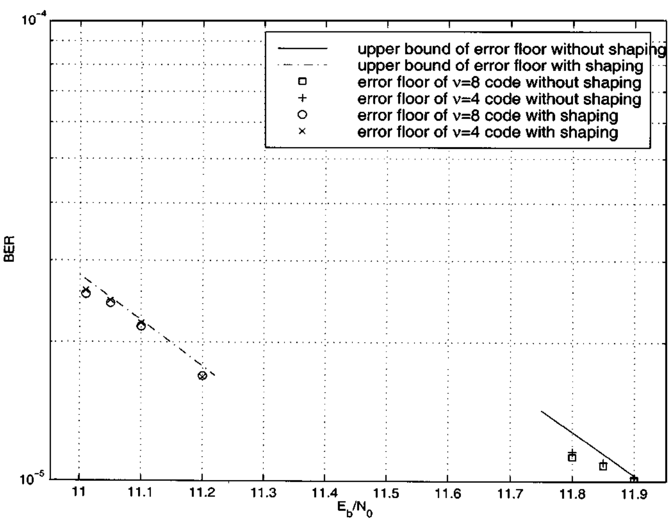


Fig. 6. Error floors and their upper bounds corresponded to the codes in Fig. 5.

to 50. For comparison, the performance of a 256-state TCM scheme using the standard VA is also reported in Fig. 5. The results show that for the $\nu = 8$ codes about 2.0 dB gross gain can be achieved by the IVA beyond the VA without shaping and about 2.7-dB gross gain with shaping. Note that the shaping gain is only about 0.7 dB in this case (a similar example can be found in [34]). Clearly, a performance 1.25 dB away from the Shannon limit at a BER of $3 \times 10^{-5}$ is achieved by the parity-concatenated trellis code at a spectral efficiency of 5.805 b/*T*.

In addition, we can see the appearance of error floors in Fig. 5. These error floors are mainly dominated by the parallel transition errors. In our simulation, we find that over 95% of the errors belong to the parallel transition bits (i.e., uncoded bits) in the $\nu = 8$ codes when the SNR equals 11.1 dB. The error floors without counting the errors caused by coded bit errors are shown in Fig. 6. We also show the upper bounds for the error floors in terms of (12) and (13). If trellis shaping is applied, in our case

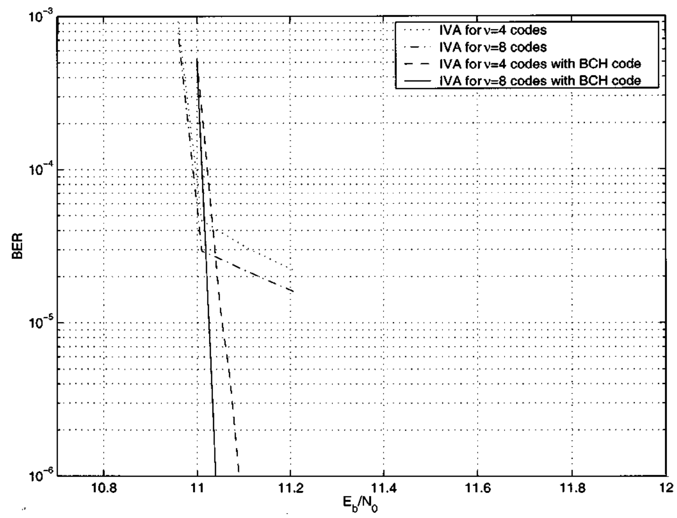


Fig. 7. BER performance of the parity-concatenated 16-state and 256-state trellis codes (the case of Fig. 5 combined with a (511,493,2) BCH code.

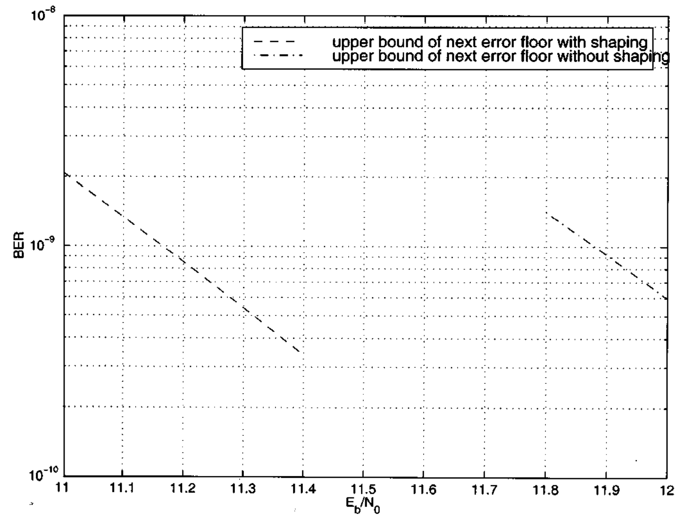


Fig. 8. Upper bounds of the new error floors after introduction of a BCH code.

$E_{\text{av}}$ equals 65.8 through simulation, and $L_{\text{av}}$ is 2.3 based on the simulation and the specific shaping technique (constellation mapping) employed.

According to the upper bounds, an appropriate BCH code can be selected to decrease the error floor to a specified lower level. Fig. 7 shows the performance of the parity-concatenated 16-state and 256-state trellis codes combined with the (511, 493, 2) binary BCH code.

We can compute the error floors in terms of (12) and (13) which, as shown in Fig. 8, have been reduced to a BER of $10^{-9}$.

The parity concatenated code with the BCH code has a spectral efficiency of 5.769 bits/*T*. Therefore, in this case a performance 1.35 dB away from the Shannon limit at a BER of $10^{-9}$ can be achieved.

Finally, we point out that the average number of iterations varies according to the different SNRs and different codes. Generally, less iterations are required at higher SNRs or by longer code memory length. In our simulation, for the $\nu = 8$ code, the average numbers of iterations are about 28, 9 and 6 for SNR of 1.01, 11.1, and 11.2 dB, respectively. It is shown that the peak and average complexity can be significantly reduced if we increase the SNR by 0.1 dB.

## VI. Conclusion

An IVA has been developed for decoding parity-concatenated trellis codes. Significant gains using the IVA can be obtained with low computational complexity. For packet transmission, single- and double-parity-check structures have been designed. Furthermore, a trellis shaping technique can be employed in the concatenated TCM systems and full shaping gains can be achieved. To decrease the effect of parallel bit errors and subsequently lower the error floor, a multilevel structure combining concatenated TCM with a BCH code has been constructed. A performance 1.35 dB away from the Shannon limit at BER of $10^{-9}$ can be achieved by a 256-state concatenated TCM system with trellis shaping and a high rate binary BCH code.

Several other important advantages of the IVA include: (a) erroneous packets can be determined by the decoder with high accuracy; (b) noise variance estimation is not required; and (c) partial block data recovery is possible. The evident disadvantages of the IVA include: (a) occasional high peak complexity for some cases; and (b) lack of soft output.

## Acknowledgment

The authors would like to thank Prof. D. J. Costello, Jr. for introducing us to this area, and also three anonymous reviewers for their constructive advice, which significantly improved the quality of this paper.

## References

[1] G. Ungerboeck, "Trellis coded modulation with redundant signal sets, Part I: Introduction and Part II: State of the art," *IEEE Commun. Mag.*, vol. 25, pp. 5–22, Feb 1987.

[2] E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-coded Modulation with Applications*. New York: Macmillan, 1991.

[3] F. Q. Wang and D. J. Costello Jr, "Probabilistic construction of large constraint length trellis codes for sequential decoding," *IEEE Trans. Commun.*, vol. 43, pp. 2439–2447, Sept 1995.

[4] S. Benedetto and G. Montorsi, "Serial concatenation of block and convolutional codes," *Electron. Lett.*, vol. 32, pp. 887–888, 1996.

[5] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. IEEE ICC'93*, pp. 1064–1070.

[6] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA: MIT Press, 1963.

[7] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[8] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 219–230, Feb 1998.

[9] G. D. Forney Jr, "On iterative decoding and the two-way algorithm," in *Proc. Int. Symp. Turbo Codes and Related Topics*, Brest, France, Sept 1997.

[10] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sept 1981.

[11] G. D. Forney Jr, F. R. Kschischang, and B. Marcus, "Iterative decoding of tail-biting trellises," in *Proc. Inform. Theory Workshop*, San Diego, CA, Feb 1998.

[12] G. Solomon and H. C. A. van Tilborg, "A connection between block and convolutional codes," *SIMA J. Appl. Math*, vol. 37, no. 2, pp. 358–369, Oct 1979.

[13] P. Robertson and T. Wörz, "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 206–218, Feb 1998.

[14] G. D. Forney Jr, M. D. Trott, and S.-Y. Chung, "Approaching AWGN channel capacity with coset codes and multilevel coset codes," *IEEE Trans. Inform. Theory*, vol. 46, pp. 820–850, May 2000.

[15] U. Wachsmann and J. Huber, "Power and bandwidth efficient digital communication using turbo codes in multilevel codes," *Euro. Trans. Telecomm.*, vol. 6, pp. 557–567, Sept 1995.

[16] H. A. Cabral, D. J. Costello Jr, and P. R. Chevillat, "Bootstrap hybrid decoding using the multiple stack algorithm," in *Proc. IEEE ISIT'97*, Ulm, Germany, July 4, p. 494.

[17] F. Jelinek and J. Cocke, "Bootstrap hybrid decoding for symmetrical binary input channels," *Inform. Contr.*, vol. 18, pp. 261–298, Apr. 1971.

[18] F. Jelinek, "Bootstrap trellis decoding," *IEEE Trans. Inform. Theory*, vol. 21, pp. 318–325, May 1975.

[19] L. Wei and H. Qi, "Near Optimal Limited Search Decoding on ISI/CDMA channels and decoding of long convolutional codes," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1459–1482, July 2000.

[20] L. Wei, "On bootstrap iterative Viterbi algorithm," in *Proc. IEEE ICC'99*, Vancouver, Canada, June 6–9, 1999, pp. 1187–1192.

[21] ——, "Near Shannon limit iterative Viterbi algorithm for conventional serial concatenated codes," IEEE Trans. Inform. Theory, 1999, submitted for publication.

[22] Q. Wang, L. Wei, and R. A. Kennedy, "Near optimal decoding for trellis-coded modulation using the BIVA and trellis shaping," in *AAECC-13*, Honolulu, HI, Nov. 1999, pp. 191–200.

[23] Q. Wang and L. Wei, "Graph-based iterative decoding algorithms for parity-concatenated trellis codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1062–1074, Mar. 2001.

[24] ——, "Iterative Viterbi algorithm for concatenated multi-dimensional TCM," IEEE Trans. Commun., to be published.

[25] G. D. Forney Jr, "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. 38, pp. 281–300, Mar. 1992.

[26] A. R. Khandani and P. Kabal, "Shaping multidimensional signal spaces—Part I: Optimum shaping, shell mapping and part II: Shell-addressed constellations," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1799–1819, Nov. 1993.

[27] A. R. Calderbank and L. H. Ozarow, "Nonequiprobable signaling on the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. 36, pp. 726–740, July 1990.

[28] J. E. Porath, "Algorithm for converting convolutional codes from feedback to feedforward form and vice versa," *Electron. Lett.*, vol. 25, no. 15, pp. 1008–1010, July 1989.

[29] G. D. Forney Jr, "Final report on a coding system design for advance solar mission," NASA Ames Res. Ctr, Moffett Field, CA, Rep. NAS2–3637, Contract NASA CR73167, 1967.

[30] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 371–377, May 1977.

[31] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[32] G. D. Forney Jr, *Concatenated Codes*. Cambridge, Mass: MIT Press, 1963.

[33] B. J. Frey, R. Koetter, and A. Vardy, "Skewness and pseudocodewords in iterative decoding," in *ISIT'98*, Cambridge, MA, 1998, p. 148.

[34] S. Couturier, D. J. Costello Jr, and F. Q. Wang, "Sequential decoding with trellis shaping," *IEEE Trans. Inform. Theory*, vol. 41, pp. 2037–2040, Nov. 1995.

**Qi Wang** received the B.E. and M.E. degrees from the Nanjing University of Posts and Telecommunications (NUPT), China, in 1994 and 1997, respectively, and the Ph.D. degree from the Australian National University in 2001, all in telecommunications.

From 1997 to 1998, he was a research engineer from Shenzhen Huawei Technology Pty. Ltd., Guangdong, China. In 2000, he joined the Institute for Telecommunications Research, University of South Australia as a lecturer. Since July 2001, he has been with the 3G Mobile R&D Division, NEC Australia Pty. Ltd., Melbourne, Australia. His research interests include the area of WCDMA communication systems.

**Lei Wei** (S'93–M'95–SM'99) received the B.E. degree from the Nanjing University of Posts and Telecommunications (NUTP), China, in 1986, the M.E. degree from the University of New South Wales, Australia, in 1993, and the Ph.D. degree from the University of South Australia, in 1995, all in electrical engineering.

In 1995, he was with the Telecommunications Engineering Group, RSISE, the Australian National University (ANU). From 1996 to 2000, he was with the Department of Engineering, Faculty of Engineering and Information Technology, the ANU, Canberra, Australia, as a senior lecturer. He is currently with the Telecommunications and Information Technology Research (TITR) Institute, University of Wollongong, Australia, as an associate professor. His current research interests include iterative decoding, digital modulation, multiuser CDMA systems and spread-spectrum multiple access with applications to mobile communications.

Dr. Wei was the founder and chair of the IEEE Australian Information Theory Chapter in 1999. He is Co-Chair of the IEEE Informatio Theory Workshop, Cairns, Australia, held in September 2001. He is currently an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS in the area of wireless communications.

**Rodney A. Kennedy** (S'86–M'88) was born in Sydney, Australia, in 1960. He received the B.E. (hons.) degree in electrical engineering from the University of New South Wales, Australia, in 1982, the M.E. degree in degital control theory from the University of Newcastle, Australia, in 1986, and the Ph.D. degree in 1988 from the Department of Systems Engineering, Australian National University, Canberra.

Since 1994, he has been head of the Department of Telecommunication Engineering, Research School of Information Sciences and Engineering, Australian National University. His research interests are in the fields of digital communications, digital signal processing, and acoustical signal procesing.

Professor Kennedy is currently an Editor for Data Communications for the IEEE TRANSACTIONS OF COMMUNICATIONS.