(5)

MEMORANDUM REPORT BRL-MR-3721

BRL

1938 – Serving the Army for Fifty Years – 1988

SOFT DECISION DECODING OF BLOCK CODES

A. BRINTON COOPER III

DECEMBER 1988

U.S. ARMY LABORATORY COMMAND

BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

DESTRUCTION NOTICE

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) BRL-MR-3721	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION USA Ballistic Research Lab 6b. OFFICE SYMBOL (If applicable) SLCBR-SE-C	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Aberdeen Proving Ground, MD 21005-5066	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO 161602-AH43 / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
SOFT DECISION DECODING OF BLOCK CODES

12. PERSONAL AUTHOR(S)
A. BRINTON COOPER III

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM 1987 TO 1988		34

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
09	04	
17	02	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Soft Decision Decoding, Block Codes, Iterated Codes, Improved Iterated Codes, Generalized minimum distance, Erasures, Optimal Decoding.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
An open problem in communications is the construction and decoding of asymptotically good block codes - those which drive the probability of error to zero while bounding the code rate from below as the block length becomes infinite. The author's Improved Iterated Codes have these properties, but larger rate bounds and simpler decoding would be useful. Soft decision decoding promises some of each. This report reviews soft decision techniques from simple erasure reconstruction through generalized minimum distance decoding and its extensions. Those which seem most suited for Improved Iterated Codes are identified.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL A. BRINTON COOPER III	22b TELEPHONE (Include Area Code) 301-278-6883 22c. OFFICE SYMBOL SLCBR-SE-C

DD Form 1473, JUN 86 *Previous editions are obsolete.*

Contents

Chapter 1

Historical Note

In 1978 [1], a construction was published of Improved Iterated Codes (IIC) which were inspired by Elias [2] and constructed from primitive BCH codes [3]. IIC provided one of the few examples at the time of codes which provide both:

$$\lim_{n\to\infty} Pr(error) = 0 \tag{1}$$

and

$$\lim_{n\to\infty} R > 0 \tag{2}$$

where R is the code rate in information bits per transmitted symbol. Codes with both properties have been euphemistically labeled "good" codes.

Basically, the IIC consists of a row code and a set of column codes, one column for each information position in the row code. Each column code is an iteration of additional BCH codes with an Elias code. The IIC is best described by its decoding algorithm:

1. Decode the row code. Compute the resulting bit error probability. Set $i = 1$.

2. Decode the i^{th} column code. The error probability has now been reduced to an arbitrarily small value, and the first information position in the row code is assumed correct. Subtract that information position from the row code, thus "shortening" the row code (see Peterson [3]),*i.e.*, reducing

by one the number of degrees of freedom or bits of information that the remaining part of the codeword represents.

3. Decode the shortened row code. The resulting bit error probability is no greater than it was following Step 1, and it may be lower because the minimum distance of the code is unchanged while the block length is smaller.

4. Increment i and go to Step 2.

On the binary symmetric channel, the ratio of channel capacity to rate for $p = 0.1$ was found to be approximately 2.4 [1]. However, these iterated codes are extremely long and, therefore, mostly of theoretical interest. If, however, the decoding of the row code can be made to produce a lower error probability, then the columns can be constructed from fewer constituent codes. The resulting iterated code will be shorter and may have a larger rate.[1]

To this end, at a NATO Advanced Study Institute devoted to communications issues, Farrell [4] suggested that the rates of these codes could be improved by incorporating soft decision techniques into the decoder. In particular, soft decision decoding of the row code promises to give better estimates of the decoding error probability at the first step and, perhaps, to permit a shorter column code of higher rate to be iterated with it to achieve the same, arbitrarily low error probability as yielded by the original IIC but at a larger value of code rate.

In what follows, soft decision decoding is explained, and techniques from the literature are studied with an eye to selecting those which may provide lower error probabilities when decoding the rows of an IIC.

[1] The rate of an interated code is the product of the rates of its constituents. Each of the consituent rates is a positive number less than unity.

Chapter 2

Background

Traditionally, coding and decoding for error control are designed for discrete channels which communicate symbols from finite sets only. Usually, such a finite set is a finite field [5] or an algebraic extension of a finite field. Channel noise is represented by the probabilistic transition of one transmitted symbol to another received symbol.

This discrete approach is attractive because it permits strong focus on the design and implementation of good codes and efficient decoders using powerful tools from such disciplines as algebra, combinatorics, and digital design. Also this approach is useful because many continuous channels can be modeled as discrete channels by incorporating the modulator, demodulator, and threshold device into the channel and by modeling the effects of noise as the probability of an output symbol conditioned on the symbol transmitted. The most common example of such a channel model is the discrete memoryless channel [6] which has, in addition to the foregoing characteristics, the property that successive symbol transitions are statistically independent.

However, the use of this abstraction is not without its penalties. While the channel input symbols are discrete and mutually distinct, the channel noise is a continuous waveform which is combined (usually added) with the input signal so that the channel output is always a continuous function of time. A discrete channel output representation is realized by quantizing the

continuous output into two or more levels. In the binary case, whenever the channel output exceeds a preset threshold, the output is said to be "1." Whenever it does not, the output value is taken to be "0." Thus, the output signal is 1 regardless of whether it just barely exceeds the threshold or exceeds the threshold by a large value. For many noise waveforms, the value of the output symbol is more uncertain as the channel output approaches the threshold that separates 1 from 0. If the error control decoder has information about the relative likelihoods (probabilities) of received symbol errors, it can in many cases correct more errors in that word than if all symbols were assumed equally likely to be correct. That is, all output symbols having the same value do not necessarily inspire equal "confidence" in their values. It has been shown [7] that without this useful likelihood information (actually, it is channel state information) approximately 2.0 dB more transmitter power is needed to achieve same the decoded error probability that can be achieved with that information.

Decoding techniques which use estimates of the actual values of the channel output waveform are called *soft decision* decoding. Those which use estimates of only the discrete transmitted symbols are called *hard decision* decoding.

What follows is an introduction and survey of the development of soft decision techniques. First, we examine decoders which erase (with meaning to be made more precise) symbols having low measures of likelihood, substituting various combinations of permitted symbol values until predetermined decoding criteria are met. From these we move to decoders which make more direct use of the channel output values and finally to techniques built upon the examination of each received symbol.

To provide focus and continuity, we deal with block codes only. This does not imply judgement on the merits of convolutional codes and Viterbi decoding [8] but tries to examine thoroughly one aspect of soft decision decoding. An examination of the References reveals at least a theoretical interest in soft decision decoding techniques from the earliest days of error control coding theory. Like so many things, soft decision is experiencing a resurgence of interest that may be due, in part, to the escalating capabilities afforded by semiconductor technology, making possible today techniques that once were only theoretical ideas.

A standard set of notation has been attempted in this report. The following symbols retain their respective meanings throughout. Where additional entities are needed, they are defined locally.

$\lfloor a \rfloor$	=	the largest integer $\leq a$
C	=	a code taking symbols from $\{\pm 1\}$
$\mathbf{c}$	=	a member of C
C^*	=	code C with symbols from $\{0, 1\}$
$\mathbf{c}^*$	=	a member of C^*
d	=	the minimum Hamming distance of C [3]
d_E	=	Euclidean distance
n	=	length (number of symbols) of a codeword
$\mathbf{r}$	=	a vector or word (N-tuple received from the channel
t	=	$\lfloor \frac{d-1}{2} \rfloor$, the guaranteed error correction capability of a block code
$\mathbf{x}$	=	the received n-tuple after hard decision processing
α	=	a vector of confidence values
φ	=	the vector of bit log likelihood ratios.

Chapter 3

Simple Erasure Schemes

When a codeword is transmitted through a noisy channel, the set of symbols or waveform values produced at the receiver output is called the *received word* or *received vector*.

An *erasure* is a symbol of unknown value at a known location within the received word. By contrast, an *error* is a symbol of unknown value at an unknown location. Thus, the decoder knows the number of erasures and their locations but knows nothing of the number or locations of errors.

3.1 Forcing Erasures

Almost from its beginning, algebraic coding theory provided methods for computing the values of symbols that were erased by a noisy binary channel provided the number of such erasures is bounded [3]. A principal advantage of introducing erasures is that many algebraic decoding algorithms can be modified to handle erasures efficiently [9]. These techniques simply erase one or more of the received symbols and apply erasure correction techniques [10]. Since a block code with minimum distance d can correct up to $t = \lfloor (d-1)/2 \rfloor$ errors or $d - 1$ erasures [3], one would like to deal with erased symbols in known locations rather than with errors in unknown locations. If an

appropriate measure of confidence or reliability is chosen (*e.g.*, a monotone function of the signal to noise ratio at the receiver output), then the received symbols with lowest reliability are most likely[1] to occupy the error locations and can be considered erased.

In this section we consider soft decision decoders that use both the output of a hard decision demodulator and a set of reliability estimates produced by the receiver. Typically, one or more of the lowest reliability positions in a received word are erased. The decoder then tries various patterns of channel input symbols in the erased locations to determine if a hard decision decoder can produce one or more codewords from this "trial vector." Rules are given to choose among the multiple codewords which may arise.

3.2 Wagner Decoding

Wagner Decoding[2] is applied to codes constructed by appending a single parity check to a block of m information digits. Based upon the received vector $\mathbf{r}$ having real components, the receiver computes $p(1|\mathbf{r})$ and $p(0|\mathbf{r})$ and makes a hard decision on each symbol based on which of these probabilities is larger. (The decoder is assumed to know the channel noise probability distribution.) If parity checks, the hard decision block is accepted as it is. If parity fails, the position with the lowest value of $\Delta p = p(1|\mathbf{r}) - p(0|\mathbf{r})$ is inverted to force parity to check. This is perhaps the simplest example of an "erase and substitute" technique. Analysis [10] shows that, for moderately noisy channels having values of bit error probability around 0.01. Wagner decoding produces lower values of word error probability than does the Hamming code. For example, for $p = 0.01$, the Wagner decoder produces a word error rate of 0.001 while a Hamming code yields approximately 0.003 [10] .

Subsequently, Balser and Silverman [11] introduced multiple error correction to this scheme by adding additional check digits to the transmitted

[1] At the channel output, the symbol error probability is a monotone decreasing function of the signal to noise ratio. See, for example, [7].

[2] Wagner decoding was named by Balser and Silverman [10] for C.A. Wagner of MIT who, in 1954, "suggested the basic idea."

block and, if a double error is detected, inverting the two least reliable digits in the received word. In fact, they used the Hamming code [12] to provide the additional parity check structure since it has minimum distance of four and can, therefore correct one error per block or detect the presence of two (or any even number).

The original Wagner algorithm is found in the final Step of Chase's *rank decoding* algorithm [13] which is discussed later.

3.3 Forced-Erasure Decoding

One difficulty with applying erasure reconstruction techniques is the decoder's lack of knowing exactly how many positions should be erased—that is, the likely existence of undetected errors in the unerased positions [14]. With erasure reconstruction only (no error correction), there are no guarantees that all symbols containing errors will be erased, even when the total number of errors is $\lfloor \frac{d-1}{2} \rfloor$ or fewer. In forced erasure decoding (FED), the received symbols within a word are ordered according by increasing value of "confidence." Confidence is proportional to the *a posteriori* probability of a transmitted symbol value conditioned upon the received symbol. If the n symbols in a block have confidence values $\{\alpha_1, \alpha_2, ..., \alpha_n\}$, reordering gives $\alpha_{j_1} \leq \alpha_{j_2} \leq \cdots \leq \alpha_{j_n}$. Reconstruction of erased symbols is performed by trying a set of allowed symbol values in the erased position and testing the resulting vector for code membership. The algorithm is:

1. Set $i = 1$.

2. Erase the first i symbols ordered by increasing confidence. Attempt reconstruction.

3. If a resulting vector belongs to the code, accept it as the decoded word and stop.[3] If $i > 2t$, there may be more than one reconstruction which produces a codeword. Depending upon the type of channel noise assumed, a rational basis for choosing among these is needed. Choosing the solution

[3] Reason: If the received vector is within d of a codeword, it will be within d of exactly *one* codeword. Therefore, when the first codeword is found, it is unnecessary to look further.

which represents a minimum weight error pattern is intuitively useful for the Gaussian channel. No choice can be shown to be foolproof, however.

4. If no reconstructed codeword is found, increment i by 1. If $i \leq n-k$ go to 2. If $i > n-k$, stop, and declare a decoding failure.

3.4 Channel Measurement Decoding

The foregoing methods of forcing erasures and substituting known patterns of channel input symbols assume that finding the codeword at minimum Hamming distance[4] from the initial hard-decision vector produces the codeword most likely transmitted. However, this need not follow. Suppose codeword $\mathbf{c}_t$ is transmitted and, after hard decision, vector $\mathbf{x}$ is received. A binary decoder can produce codeword $\mathbf{c}_a$ from $\mathbf{x}$, by the criterion of *minimum Hamming distance* decoding. On the other hand, *minimum Euclidean distance* decoding prior to hard decision might produce codeword $\mathbf{c_b} \neq \mathbf{c_a}$.

Why is this? Some of the positions that were changed by the minimum d decoder may have been more reliable than some which were not. For example, if the first three positions of vector $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5)$ have high values of some reliability measure while the last two have low values, then the most likely transmitted codeword might be one that would agree with $\mathbf{r}$ in the first three positions. The decoder, then, selects values for the last two positions that produce a *waveform* closest to that received. To minimize this "analog distance" between waveforms, a set of real reliability numbers, as described below, is used to define an analog or real measure of distance in the decoding procedure. A consequence is that a different set of positions may be "corrected" than when Hamming distance is minimized.

$$\begin{array}{lccccc} & \textit{channel} & & \textit{HDD} & & \textit{min } d \\ \mathbf{c}_t & \longrightarrow & & \mathbf{r} \longrightarrow \mathbf{x} & \longrightarrow & \mathbf{c}_a \\ & \downarrow \textit{ min } d_E & & & & \\ & \mathbf{c}_b & & & & \end{array}$$

So, if there is a codeword $\mathbf{c}_a$ differing from the binary received vector $\mathbf{x}$ in

[4]The Hamming distance between two vectors is simply the count of the number of places in which they differ.

$2t$ or fewer positions, a binary decoder will identify it. We often say that the binary decoder determines the error pattern $\mathbf{e}$ having minimum Hamming weight such that $\mathbf{e} = \mathbf{x} \oplus \mathbf{c}_a$ where $\oplus$ represents addition modulo 2. Channel Measurement Decoding (CMD) [15] expands upon this by trying to find a small set of likely error patterns and choosing from this set the one having the smallest "analog weight." CMD uses a set of real numbers $\{\alpha_i, i = 1, 2, \cdots, n\}$ to represent the channel state. They have the property that $\alpha_i > \alpha_j \Rightarrow z_i$ is more correct than z_j and are called *channel measurement information.* To use this,

- From the received binary vector $\mathbf{x}$, obtain another vector $\mathbf{x}'$ by inverting some of the digits.
- Consider $\mathbf{x}'$ as a received vector and use a binary decoder which implements bounded distance decoding[5] to try and find its error pattern $\mathbf{e}'$.
- For all the new vectors $\{\mathbf{x}'\}$ obtained by inverting selected sets of digits, find the corresponding error patterns $\mathbf{e}'$ and select the one having minimum analog weight $w_\alpha(\mathbf{e}')$ given below.

$$w_\alpha(\mathbf{e}_i') = \sum_{j=1}^{n} \alpha_j e_{ij} \tag{3}$$

As the channel signal to noise ratio increases, the performance of CMD can approach that of maximum likelihood decoding.

It is desirable to select the digits to be inverted in CMD so that the number of error patterns $\mathbf{e}$ is small and so that the vectors $\mathbf{x}'$ lie "near" the received word. Chase [15] offered several algorithms for selecting the sets of digits to be inverted in CMD to meet these criteria.

[5] A bounded distance decoder can correct no more than $t \leq \lfloor (d-1)/2 \rfloor$ errors, irrespective of the actual capability of the code.

Chapter 4

Soft Decision Decoding – The Paradigm

A hard decision decoder measures the distance between the received word and various candidate code words by using Hamming distance (See Chapter 3.). This chapter discusses decoding techniques which use estimates of the real value of each received symbol in order to establish relative confidence in the decoded symbols. This permits one to measure distances between words in terms of real values rather than counts of differences.

4.1 Errors and Erasures Decoding

Let $\mathbf{c}_m$ be a codeword with symbols from $\{\pm 1\}$. If $\mathbf{x}$ is the received vector after hard decision processing (but not decoding) and also has values from $\{\pm 1\}$, then "conditional" Hamming distance decoding (decode only when there is a codeword within minimum distance of the received, hard decision word) is permitted by this theorem: [1]

[1] Proofs of the theorems in this section and the next can be found in [16] and will not be repeated here.

Theorem 1 *There exists at most one codeword* $\mathbf{c}_m$ *such that*

$$\mathbf{x} \cdot \mathbf{c}_m > n - d \tag{4}$$

where the left side is computed as the scalar product of two real vectors.

This theorem, and others in this chapter having a similar form can be thought of as generalizations of this well-known result for hard decision decoding and binary codes:

> There is at most one codeword $\mathbf{c}$ at Hamming distance d from the received binary vector $\mathbf{x}$ [3].

In these generalizations, the minimum distance of the code is "shared" in a discrete or "continuous" fashion (depending upon the nature of the received vector under consideration) over the values of the received symbols as necessary to achieve minimum d_E decoding.

As indicated earlier, a natural generalization of Wagner decoding is to erase several unreliable received symbols having reliability values less than a preset value [17].

In fact, one can define a *null zone* of amplitudes of the received waveform which represent the most unreliable values because they are closest to the decision threshold between 1 and 0. Let $\mathbf{r} = (r_1, r_2, \cdots, r_n)$ be a received vector having real, continuous components taking values on the closed interval $[-1, 1]$. Define

$$\alpha_i = \begin{cases} +1 & \text{if } r_i \geq T \\ 0 & \text{if } -T < r_i < T \\ -1 & \text{if } r_i \leq -T. \end{cases} \tag{5}$$

Values falling within the null zone defined by $-T < r_i < T$ are erased, thus presenting to the decoder a word constructed from an alphabet of three symbols, $\{-1, 0, 1\}$ and possibly containing erasures in known positions and errors in unknown positions. This suggests errors and erasures decoding (EED), which is based on the following theorem [16]:

Theorem 2 *There is at most one code word* $\mathbf{c}_m$ *from a code of length* n *and minimum distance* d *such that*

$$\mathbf{r} \cdot \mathbf{c}_m = n - 2t_m - s > n - d \tag{6}$$

where t_m *and* s *are the numbers of errors and erasures, respectively, that the code is guaranteed to correct.*

This exploits the fact [3] that a block code with minimum distance d can correct any pattern of t errors and s erasures so long as t and s satisfy

$$2t + s < d. \tag{7}$$

A three step procedure can be used for EED of a binary code when decoding for s erasures and an unknown number of errors in the received word. Let t be the number of errors per word guaranteed correctable by the minimum distance of the code:

$$d = 2t + 1. \tag{8}$$

Then:

- Set all s erased bits to 0 and allow the decoder to correct up to t errors.
- Set all s erased bits to 1 and decode up to t errors.
- If each of these produces a codeword, the decoder accepts the one which would have experienced the smaller number of errors in transmission. In this case, it has been shown [17] that the decoding which requires the fewer number of changes produces the actual transmitted word.

4.2 Generalized Minimum Distance Decoding

Generalized Minimum Distance (GMD) decoding was developed to provide a systematic way of minimizing the *Euclidean* distance between the

received vector and the code rather than by making hard decisions and minimizing the Hamming distance.

The receiver is assumed to produce a vector $\mathbf{r} = (r_1, \cdots, r_n)$ such that $-1 \leq r_j \leq 1,\ j = 1, 2, ..., n$. The Euclidean distance between the received vector and the m^{th} codeword is

$$\begin{aligned} d_E{}^2(\mathbf{r}, \mathbf{c}_m) &= |\ \mathbf{r} - \mathbf{c}_m\ |^2 \\ &= |\ \mathbf{r}\ |^2 + |\ \mathbf{c}_m\ |^2 - 2 \cdot \mathbf{r} \cdot \mathbf{c}_m. \end{aligned} \tag{9}$$

Formally, the problem is to minimize d_E over the message number m. This is achieved by maximizing $\mathbf{r} \cdot \mathbf{c}_m$ in the foregoing.

We have this important theorem for generalized minimum distance decoding [16]:

Theorem 3 *There exists at most one* $\mathbf{c}_m$ *such that*

$$\mathbf{r} \cdot \mathbf{c}_m > n - d. \tag{10}$$

Any decoding which meets this criterion is called *generalized minimum distance* (GMD) decoding.[2]

To use GMD decoding, order the indices on the received symbols so that:

$$|\ r_{i_1}\ | \leq |\ r_{i_2}\ | \leq \cdots \leq |\ r_{i_n}|. \tag{11}$$

Then define:

$$s(r_j) = \begin{cases} +1 & \text{if } r_j > 0 \\ -1 & \text{if } r_j < 0 \end{cases} \tag{12}$$

and

$$q_k(r_{i_j}) = \begin{cases} 0, & 1 \leq j \leq k,\ 0 \leq k \leq n \\ s(r_{i_j}), & k+1 \leq j \leq n. \end{cases} \tag{13}$$

[2]We emphasize that, in this Theorem, the components of $\mathbf{r}$ take continuous values whereas, in Theorem 1, the received vector takes values from $\{\pm 1\}$.

That is, $\mathbf{q}_k = (q_k(r_1), q_k(r_2), ..., q_k(r_n))$ is zero in the k weakest components of $\mathbf{r}$ and has values from $\{\pm 1\}$ in the remaining $n - k$ positions.

To adapt EED to GMD decoding, we need the next theorem.

Theorem 4 *If* $\mathbf{r} \cdot \mathbf{c}_m > n - d$, *then for some* k*:*

$$\mathbf{q}_k \cdot \mathbf{r}_m > n - d. \tag{14}$$

The point of Theorem 4 is to permit the use of any errors and erasures decoder on $\mathbf{q}_k$ according to the following algorithm.

- Order the indices of the components of the received vector as in (11).
- Set i = 0.
- From the received word, $\mathbf{r}$, determine $\mathbf{q}_i$ as above.
- If an errors and erasures decoder can find a codeword $\mathbf{c_m}$ corresponding to $\mathbf{q}_i$, then decode $\mathbf{r}$ as $\mathbf{c}_m$ and exit.
- Else $i = i + 1$. If $i < d$, repeat previous step. If $i \geq d$ quit.

The foregoing illustrates but one use of GMD decoding. Beyond this, many investigators have extended it to other applications. Examples include decoding on Q-ary output channels [18], burst error decoding on Q-ary output channels [19], [20], and majority logic decoding [21]. A few of these, important to the application cited in the first chapter, will now be discussed.

4.3 Decoding the Channel with Quantized Output

As earlier seen, errors and erasures decoding provides an elementary quantization of the continuous channel. Going beyond the definition of a simple null zone centered on the 1/0 threshold, investigators [22,23] sought to

determine how far from the threshold the received signal amplitude lay. This would provide a measure of confidence in the initial hard decision estimate.

In this paradigm, the continuous channel output is quantized into Q levels $(L_0, L_1, \cdots, L_{Q-1})$ where Q is typically, but not necessarily, a power of two. With each level is associated a w-weight w_i which represents the distance from the threshold to L_i or to L_{Q-1-i}. By assigning $w_0 = 0, w_{Q-1} = 1$, and defining the $w-$distance between two levels as

$$d_w(L_i, L_j) = \mid w_i - w_j \mid \tag{15}$$

we can define the w-distance between any two Q-ary n-tuples as

$$d_w(\mathbf{x}_i, \mathbf{x}_j) = \sum_{p=1}^{n} d_w(x_{ip}, x_{jp}). \tag{16}$$

w-distance has been shown to be a true metric [23].

Minimum w-distance decoding of code C becomes simply: given received Q-ary vector $\mathbf{r}$, find the codeword $\mathbf{c}_i$ that produces:

$$\min_i d_w(\mathbf{r}, \mathbf{c}_i), \ \mathbf{c}_i \in C \tag{17}$$

Then, the w-weight e_s of the error in the s^{th} digit is the w-distance between the s^{th} transmitted and received symbols, and

Theorem 5 *An (n,k) code with minimum Hamming distance d can correct any error pattern with w-weight satisfying*

$$\sum_{s=1}^{n} e_s = \frac{d}{2}. \tag{18}$$

This is proved in [23]. In the absence of a systematic decoding algorithm, one could simply compute the w-distance from the received vector to each code word until one with value less than $d/2$ is found or until all are computed and the smallest selected.

Fortunately, it is not necessary to search exhaustively. Algorithms exist which use the w-distance metric to augment the performance of any binary

decoder, thus making soft decision decoding improvements available for use with a wide variety of block codes. Such techniques are called *weighted erasure decoding* (WED) [23] although, in the strictest sense, nothing is really erased. An example of a WED algorithm follows.

Expand each w-weight on a set of $r \leq \frac{Q-1}{2}$ positive real numbers using binary coefficients.

$$w_i = \sum_{s=1}^{r} A_{is} r_s, \; A_{is} \in \{0, 1\} \tag{19}$$

Express each received symbol in this representation, and write each received word as the resulting array of r binary n-tuples. (It will soon become obvious that the complexity of this decoder is sensitive to the value of r which, therefore, should be kept as small as possible.) Using your favorite binary decoder, decode each n-tuple in this array. The following quantities are of interest:

F_s = the number of positions changed in the s^{th} row by the binary decoder
E_s = the actual number of errors that occured in the s^{th} row
E = $\sum e_r = \sum_{s=1}^{n} v_f E_f$ = total $w-$weight of the error pattern
R_s = $max(0, 2d - 2F_s)$

In any column define S_0 and S_1 to be the index sets of positions where the value is 0 and 1 respectively. Then, decide that an information digit has value 0 if

$$\sum_{S_0} R_f v_f > \sum_{S_1} R_f v_f \tag{20}$$

and 1 otherwise.

NOTE: This algorithm does not attempt to output a codeword from the decoder but rather to produce a set of "best" estimates of the information digits. It is, therefore, best suited to applications that are bit oriented rather than block or word oriented.

For channels having continuous outputs, Reddy [24] extended WED to two cases which cover many important applications.

Reddy's Type I channel is a binary input, channel with continuous output on $[0, 1]$. The Generalized Hamming Weight (GHW) of a channel output

is defined as $GHW(e_i) = \mid e_i \mid$. The GHW of a received vector is, as before, the sum of the GHW's of its components. Reddy [24] proved:

Theorem 6 *On a Type I channel, all error patterns having* $GHW < \frac{d}{2}$ *are correctable.*

It is instructive to outline Reddy's algorithm and to compare it with Yu's restriction of GMD to Q-ary channels. (See the next section.)

Reddy allows, the number of w-weights to be infinite, and represents every value $x_j \in [0,1]$ by an expansion of the w-weights, $x_j = \sum_{i=1}^{\infty} b_{ij} W_i$ where the b_{ij} are binary coefficients. Define:

$$V_j = (b_{11}, b_{12}, \cdots, b_{1n}) \tag{21}$$

$$L2D = \lfloor log_2(d+1) \rfloor \tag{22}$$

$$w_i = 2^{-i}. \tag{23}$$

The algorithm is:

- From the received word, derive $V_1, V_2, \cdots V_{L2D}$.
- Using the binary decoder, decode $\hat{V}_1, \hat{V}_2, \cdots, \hat{V}_{L2D}$.
- Select $\hat{V}_i$ such that $GHD(\hat{V}_i, Y) < \frac{d}{2}$.

If d is odd, this will correct all error patterns having $GHD < \frac{d}{2}$.

Reddy's Type II channel takes input from arbitrary alphabet A and presents as output pairs (y, α) where $y \in A$ and $\alpha \in [0,1]$. Defining GHD between (y_1, α_1) and (y_2, α_2) as $\frac{1}{2} \mid \alpha_1 - \alpha_2 \mid$, he showed that all error patterns having $GHD < \frac{d}{2}$ are correctable. Note that the obvious application of this case is to the soft decision demodulator output which presents both a tentative hard decision estimate and a confidence value for each demodulated symbol.

4.4 Improved GMD Algorithms for Q-ary Channels

Yu and Costello [18] extended GMD (which was developed for the arbitrary channel) to the Q-ary output channel. Relaxing the constraint on Theorem 4 by dividing each of the components of $\mathbf{r}$ by the absolute value of the largest and making arguments on the Q-ary channel, produces a received vector $\beta = (\beta_1, \cdots, \beta_n), -1 \leq \beta_i \leq 1$, that satisfies the theorem:

Theorem 7 *For any received β, there is at most one codeword $\mathbf{c}$ such that*

$$\beta \cdot \mathbf{c} > n - d. \tag{24}$$

That this is stronger than Theorem 4 can be seen by noting that, taken together, the two theorems imply that

$$\mathbf{r} \cdot \mathbf{c} > \mid r_M \mid (n - d) \tag{25}$$

where $\mid r_M \mid \leq 1$ is the magnitude of the largest component of $\mathbf{r}$. This result will be used in the next algorithm.

Now, the continuous value of each channel output symbol is quantized to the nearest level in the set $\{2(h-1)/(Q-1) - 1, h = 1, 2, \cdots, Q\}$. The channel output vector is thus replaced by its quantized version γ which obeys the following familiar looking theorem.

Theorem 8 *For any quantized received word γ, There exists at most one codeword $\mathbf{c}$ such that*

$$\gamma \cdot \mathbf{c} > n - d. \tag{26}$$

When Q can be written as q^l, the algorithm for finding such a codeword is quite similar to that of Reddy [24]. The quantized received vector γ generates l q-ary words, each of which is decoded by a q-ary decoder. Together, these produce a codeword, $\mathbf{c}_j$. If $\mathbf{c}_j$ satisfies Theorem 8, output it as the decoded word and stop. Otherwise, try another of the q-ary outputs until all are exhausted. If no codeword satisfies the theorem for the γ at hand, the

one with the largest value of the dot product is output as the best estimate because it is the "closest" to the received vector in the sense of Theorem 8. However, it has been shown that, if Theorem 8 is satisfied, then at least one of the q-ary outputs satisfies a similar condition.

4.5 Rank decoding

Finally, we present a decoding technique that is in rather a different vein from those preceeding, all of which are related in some way to GMD. It is included because, for its simplicity, it provides a useful amount of error correction on noisy channels.

Chase [13] developed *rank decoding* to be used with the product of two $(3, 2)$ parity check codes. In principle, it can be applied [25] to any binary code that is one-step orthogonalizable[3] [26]. The received symbols within a word are ordered according the reliability measure accompanying the hard decision estimate. Those estimates which satisfy all checks or which satisfy one check and are sufficiently reliable are accepted. Others are flagged. When a row or column in the product array contains only one undecoded position, that position is decoded with a value that will force the parity of the row or column to check, as in Wagner decoding. Rank decoding gives coding gains of 2 to 4 dB for the codes originally used by Chase. Coding gain decreases as longer single parity check equations are used; this is expected since the rate increases and the ratio of minimum distance to length decreases with block length [25].

[3] A code is orthogonalizable if, for every information position, a set of $d - 1$ check equations involving that position can be written so that no other position appears in more than one.

Chapter 5

Optimal Methods

Perhaps less practical than techniques of the preceeding section, so-called optimal methods demonstrate the limits of what is possible and give a measure of the difficulty in achieving these limits.

Several criteria of optimality can be defined for decoders. These include:

minimum error probability,
maximum-likelihood, and
minimum cost.

We shall not consider minimum cost because, in a communications context, all types of errors are generally of equal cost.

The *minimum error probability* criterion is to decode the received vector $\mathbf{r}$ into the codeword $\mathbf{c}_m$ which is most likely to have been transmitted, given that $\mathbf{r}$ was received. That is, choose $\mathbf{c}'_m$ so that

$$Pr(\mathbf{c}_{m'} \mid \mathbf{r}) \geq Pr(\mathbf{c}_m \mid \mathbf{r}),\ m \neq m'. \tag{27}$$

The *maximum likelihood* criterion is to decode $\mathbf{r}$ into the codeword $\mathbf{c}_{m'}$ for which $\mathbf{r}$ is most likely to have been received, given that $\mathbf{c}_{m'}$ was transmitted.

Or, choose $\mathbf{c}'_m$ so that

$$Pr(\mathbf{r} \mid \mathbf{c}_{m'}) \geq Pr(\mathbf{r} \mid \mathbf{c}_m),\ m \neq m'. \tag{28}$$

Bayes' rule makes (27) equivalent to

$$Pr(\mathbf{c}_{m'})Pr(\mathbf{r} \mid \mathbf{c}_{m'}) \geq Pr(\mathbf{c}_m)Pr(\mathbf{r} \mid \mathbf{c}_m). \tag{29}$$

Hence, when codewords are equally likely to be transmitted, $Pr(\mathbf{c}_{m'}) = Pr(\mathbf{c}_m)$ and we have [27]:

Lemma 1 *If all codewords are equally likely to be transmitted, then any maximum-likelihood decoder performs minimum error probability decoding.*

The decoding error can be minimized over a word or over each symbol within a word.

5.1 Optimal Word Decoding

Optimal word decoding will mean maximum likelihood or minimum probability of error decoding since our concern is with equiprobable code words. Hwang [28] has presented an algorithm which, like correlation decoding, is guaranteed to be optimal and which presents avenues for reducing its complexity by exploiting the algebraic structure of the linear block code.

Unlike most of the soft decision decoders heretofore presented, this one has no associated binary decoder. In the spirit of reducing complexity, it does not require multiplication of real numbers.

The received vector is $\mathbf{r} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c}$ is the $\{\pm 1\}$ representation of binary code vector and C is the set of code vectors over $\{\pm 1\}^n$. The error vector $\mathbf{e}$ is an n-tuple of real numbers. The bit log likelihood ratios are

$$\phi_i = ln\left[\frac{Pr(r_i \mid 1)}{Pr(r_i \mid -1)}\right],\ i = 1, 2, \cdots, n \tag{30}$$

and are the components of the so-called *channel measurement information vector* of $\mathbf{r}$: $\phi = (\phi_1, \phi_2, \cdots, \phi_n)$.

The following definition is needed.

For n-tuples **a** and **b**,

$$\mathbf{a} \times \mathbf{b} = (a_1 b_1, a_2 b_2, \cdots, a_n b_n) \tag{31}$$

and is a vector.

If C^* is the binary code, define a subset C_A^* of code vectors thusly.

$$C_A^* = \{c_a^* \in C^* \mid \sum_{i=1}^{k} c_{ai}^* < \sum_{j=k+1}^{n} c_{aj}^*\} \tag{32}$$

That is. C_A^* is the set of codewords in C^* which have more 1's in the last $n-k$ positions than in the first k positions.

Hwang's algorithm, which follows, exemplifies the types of computation needed to do maximum likelihood information and provides a vehicle for discussing the complexity of such decoders. An algorithm for finding $\mathbf{c}_m$, the maximum likelihood codeword, is:

- Calculate ϕ. Find $\mathbf{c}_1$ such that $\phi_i \cdot \mathbf{c}_1 > 0$, $i = 1, 2, \cdots, k$. Set $\mathbf{x} = \phi \times \mathbf{c}_1$.
- Check values of $\mathbf{c}_a^* \cdot \mathbf{x}$ for all $c_a^* \in C_A^*$.
- If $\mathbf{c}_{a_m}^* \cdot \mathbf{x} < 0$ and $\mathbf{c}_{a_m}^*$ minimizes $\mathbf{c}_a^* \cdot \mathbf{x}$ then set $\mathbf{c}_m = \mathbf{c}_1 \times \mathbf{c}_{a_m}$. Else set $\mathbf{c}_m = \mathbf{c}_1$.

Note how the complexity of the algorithm depends directly on the size of C_A^*.

5.2 Optimal Symbol Decoding

To minimize the probability of a *symbol* error, select the symbol $\mathbf{c}_m = a_i$ to maximize

$$P_r(\mathbf{c}_m = a_i|\mathbf{r}) = \sum_{\mathbf{c} \in S_m(a_i)} P_r(\mathbf{c}|\mathbf{r}) \tag{33}$$

where $a_i \in$ the source alphabet and $S_m(a_i)$ is the set of all codewords where the m-th symbol has value a_i. That is, select that value of the m-th codeword symbol that minimizes the probability of symbol error conditioned on the received vector.

An implementation of the minimum probability of symbol error for the binary symmetric channel and for any *linear* code [29] results in the decoding rule:

Set $\hat{c}_m = 0$ if

$$\sum_{j=1}^{2^{n-k}} \prod_{l=0}^{n-1} \left(\frac{1-\phi_l}{1+\phi_l} \right)^{c'_{jl} \oplus \delta_{ml}} > 0 \tag{34}$$

and $\hat{c}_m = 1$ otherwise. $\hat{c}'_{jl}$ is the $l-th$ component of the $j-th$ member of the dual code C', δ_{ml} is the Kronecker delta, $\oplus$ represents addition modulo 2, and ϕ_l is the *bit likelihood ratio* given by

$$\phi_l = \frac{P_r(r_m \mid 1)}{P_r(r_m \mid 0)}. \tag{35}$$

In general, a decoders for such symbol decoding rules will produce an output vector $\hat{\mathbf{c}}$ which may not be a codeword. To extract the actual information symbols (which is, after all, the point of the algorithm) when the code is not *systematic*, a set of k linear equations must be solved after decoding. [1]

To select the *most likely* transmitted m-th *symbol*, select $\mathbf{c}_m = a_i$ which maximizes

$$\sum_{\mathbf{c} \in S_m(a_i)} P_r(\mathbf{r}|\mathbf{c}). \tag{36}$$

Bit by bit algorithms are quite complex and will not be considered further in this work.

[1] In a systematic code, the k encoded information positions appear unaltered, usually in the first or last k positions of the codeword. In a non-systematic code, every one of the n encoded symbols is a linear combination of the k information symbols. That is, $c_j = \sum_{i=0}^{k-1} g_{i,j} a_i$, $i = 0, 1, \cdots, n-1$.

Chapter 6

Conclusions and Directions

The purpose of this overview is to consider candidate soft decision techniques for decoding IIC's. In the original IIC decoding algorithm, the first decoding of the row code from the channel reduces the error probability from p to p_1. Decoding the first column reduces p_1 to ϵ, an arbitrarily small value, thus giving estimates for the first position in each row with very high confidence. After the decoded first position is subtracted from the row code, the entire process is repeated, now possibly requiring a shorter and higher rate code for the second column. These steps are iterated until each information position in the row code, and hence in the entire code, is decoded.

Soft decision techniques promise improvements to this process, *viz*:

1. smaller values of $p_1, p_2, \cdots$, thus requiring shorter and higher rate column codes,

2. the creation of erasures in the leftmost column to be reconstructed by errors and erasures decoding of the column code with a possible improvement in code rate, and

3. a redesign of Improved Iterated Codes using codes more easily decoded than BCH codes, counting on the stronger soft decision decoding to provide comparable or even improved performance.

Soft decision decoders which should be examined for use in a decoder for IIC include:

- channel measurement decoders
- errors and erasures decoding with GMD
- GMD on a binary-input discrete memoryless channel.

Finally, certain techniques have been tailored specifically for iterated codes and. for that reason, should be tried. These include

- a form of WED [23]
- an extension of GMD to iterated codes [20]
- a recent scheme bult on Elias's codes [30].

References

[1] Cooper, A.B. & W.C. Gore, "Iterated Codes with Improved Performance," *IEEE Trans. Inform. Theory* IT-24 (Jan. 1978), 116–118.

[2] Elias, P., "Error-free Coding," *IRE Trans. Inform. Theory* PGIT-4, 29–37.

[3] Peterson, W.W. & E.J. Weldon, *Error-Correcting Codes*, MIT Press, Cambridge, 1972.

[4] Farrell, P.G., "Code Structure and Decoding Complexity," in *NATO Advanced Study Institute on The Impact of Processing Techniques on Communications (1983)*, J.K. Skwirzynski, ed., Dordrecht, The Netherlands, 1985.

[5] Herstein, I.N., *Topics in Algebra*, Blaisdell, Waltham, MA, 1964.

[6] McEliece, R.J., *The Theory of Information and Coding*, Cambridge University Press, Cambridge, 1977.

[7] Wozencraft, J.M. & I.M. Jacobs, *Principles of Communications Engineering*, Wiley, New York, 1965.

[8] Viterbi, A.J. & J.K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, New York, 1979.

[9] Forney, G.D., "Generalized Minimum Distance Decoding," *IEEE Trans. Inform. Theory* IT-12 (Apr. 1966), 125–31.

[10] Balser, M. & R.A. Silverman,, "Coding for Constant-Data-Rate Systems– Part I, A New Error-Correcting Code," *Proc. IRE*, September, 1954.

[11] Balser, M. & R.A. Silverman,, "Coding for Constant-Data-Rate Systems– Part II, Multiple-Error-Correcting Codes," *Proc. IRE*, June, 1955.

[12] Hamming, R.W., "Error Detecting and Error Correcting Codes," *B.S.T.J* 29 (April, 1950), 147–160.

[13] Chase, D., "A Combined Coding and Modulation Approach for Communication over Dispersive Channels," *IEEE T-COM* COM-21 (March, 1973), 159–174.

[14] Heller, R.M., "Forced-Erasure Decoding and the Erasure Reconstruction Spectra of Group Codes," *IEEE Trans. Commun.* COM-15 (June 1967).

[15] Chase, D., "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inform. Theory* IT-18 (Jan. 72), 170–82.

[16] Forney, G.D., *Concatenated Codes*, MIT Press, Cambridge , 1969.

[17] Michelson, A.M. & A.H. Levesque, *Error-Control Techniques for Digital Communication*, Wiley, New York, 1985.

[18] Yu, C.C.H. & D.J. Costello Jr., "Generalized Minimum Distance Decoding Algorithms for Q-ary Output Channels," *IEEE Trans. Inform. Theory* IT-26 (Mar. 1980), 238–243.

[19] Wainberg, S. & J.K. Wolf, "Burst decoding of binary block codes on Q-ary output channels," *IEEE Trans. Inform. Theory* IT-18 (Sep 1972), 684–686.

[20] Reddy, S.M. & J.P. Robinson, "Random Error and Burst Correction by Iterated Codes," *IEEE Trans. Inform. Theory* IT-18 (Jan 1977), 182–185.

[21] Sundberg, C-E, "One step Majority-logic Decoding with Symbol Reliability Information," *IEEE Trans. Inform. Theory* IT-21 (Mar. 1975), 236–242.

[22] Farrell, P.G., "Soft-Decision Detection Techniques," in *Algebraic Coding Theory and Applications*, G. Longo, ed., Springer-Verlag, New York, 1979.

[23] Weldon, E.F., "Decoding Binary Codes on Q-ary Output Channels," *IEEE Trans. Inform. Theory* IT-17 (Nov 1971), pp 713–8.

[24] Reddy, S.M., "Further Results on Decoders for Q-ary Output Channels," *IEEE Trans. Inform. Theory* IT-20 (July 1974), 552–4.

[25] Cooper, A.B., III, "Coding Gains for Rank Decoding," Ballistic Research Laboratory, BRL IMR 902, Aberdeen Proving Ground, 24 May 1988.

[26] Lin, S. & D.J. Costello Jr., *Error Control Coding*, Prentice-Hall, Englewood Cliffs, 1983.

[27] Gallager, R.G., *Information Theory and Reliable Communication*, Wiley, New York, 1968.

[28] Hwang, T-Y., "Efficient Optimal Decoding of Linear Block Codes," *IEEE Trans. Inform. Theory* IT-26 (Sep. 1980), 603–6.

[29] Hartmann, C.R.P & L.D. Rudolph, "Optimum Symbol-by-Symbol Decoding Rule for Linear Codes," *IEEE Trans. Inform. Theory* IT-22 (Sep. 1976), 514–7.

[30] Battail, G., J.C. Belfiore & R. Sfez, "Decoding as a Means of Reestimating a Probability Distribution, Application to an Iterated Product of Parity Check Codes," IEEE International Symposium on Information Theory, Kobe, Japan, Jun 1988.

No. Copies	Organization
12	Administrator Defense Technical Information Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145
1	HQDA (SARD-TR) Washington, DC 20310
1	Commander US Army Material Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
1	Commander US Army Laboratory Command ATTN: AMSLC-TD Adelphi, MD 20783-1145
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-MSI Picatinny Arsenal, NJ 07806-5000
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TDC Picatinny Arsenal, NJ 07806-5000
1	Director Benet Weapons Laboratory Armament R&D Center ATTN: SMCAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Armament, Munitions and Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299
1	Commander US Army Aviation Systems Command ATTN: AMSAV-DACL 4300 Goodfellow Blvd St Louis, MO 63120
1	Director US Army Aviation Research and Technology Activity Ames Research Center Moffett Field, CA 94035-1099
3	Commander US Army Communications Electronics Command ATTN: AMSEL-ED AMSEL-RD-COM-D AMSEL-RD-COM-T Fort Monmouth, NJ 07703
1	Commandant US Army Infantry School ATTN: ATSH-CD-CS-OR Fort Benning, GA 31905
1	Commander US Army Missile Command ATTN: AMSMI-RD Redstone Arsenal, AL 35898-5000

No. Copies	*Organization*
1	Commander US Army Tank Automotive Command ATTN: AMSTA-DI Warren, MI 48397-5000
1	Director US Army TRADOC Analysis Center ATTN: ATAA-SL White Sands Missile Range, NM 88002-5502
2	Director US Army Research Office ATTN: Dr William Sander Dr Arthur Wouk PO Box 12211 Research Triangle Park, NC 27709-2211
1	Director Naval Research laboratory ATTN: Code 7521 (Dr J E Wieselthier) Washington, DC 20375-5000
1	Dr Stephen Wolff, Director Division of Networking and Communications National Science Foundation 1800 G St, NW Washington, DC 20550
1	Dr William W Wu INTELSAT 3400 International Drive, NW Washington, DC 20008
1	Prof Brian Hughes Department of Electrical and Computer Engineering GWC Whiting School of Engineering The Johns Hopkins University Baltimore, MD 21218
1	Prof Willis C Gore Department of Electrical and Computer Engineering GWC Whiting School of Engineering The Johns Hopkins University Baltimore, MD 21218
1	Prof Rod Goodman Department of Electrical Engineering California Institute of Technology Pasadena, CA 91125
1	Prof Stephen Wicker School of Electrical Engineering Georgia Institute of Technology Atlanta, GA 30332
1	Prof P.G. Farrell Department of Electrical Engineering University of Manchester Manchester ML3 9PL UK
1	Prof Shigeichi Hirasawa Department of Industrial Engineering Waseda University 3-4-1 Ohkubo, Shinjuku-ku Tokyo 160, Japan

No. Copies	*Organization*	*No. Copies*	*Organization*

1 Dr Toshihisa Nishijima
Department of Industrial Engineering
Waseda University
3-4-1 Ohkubo, Shinjuku-ku
Tokyo 160, Japan

Aberdeen Proving Ground

Director, USAMSAA
ATTN: AMXSY-D
AMXSY-MP (Mr H Cohen)
AMXSY-CT (Mr F Fox)

Commander, UASTECOM
ATTN: AMSTE-TO-F

Cdr, CRDC, AMCCOM
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number_____ Date of Report_____

2. Date Report Received_____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)_____

_____

_____

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.)_____

_____

_____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate._____

_____

_____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

_____

_____

_____

CURRENT ADDRESS

_____
Name

_____
Organization

_____
Address

_____
City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

_____
Name

_____
Organization

_____
Address

_____
City, State, Zip

(Remove this sheet, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC
POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE