

2-1-1992

HYDRO-ACOUSTIC COMMUNICATION SYSTEM BASED ON A NEURAL NETWORK ERROR CORRECTOR

M. F. Tenorio

Purdue University School of Electrical Engineering

R. Z. Machado F.

Petrobras R&D Center, Subsea Engineering Group

J. R. M. Silva

Purdue University School of Electrical Engineering

Follow this and additional works at: <http://docs.lib.purdue.edu/ecetr>

Tenorio, M. F.; Machado, R. Z. F.; and Silva, J. R. M., "HYDRO-ACOUSTIC COMMUNICATION SYSTEM BASED ON A NEURAL NETWORK ERROR CORRECTOR" (1992). *ECE Technical Reports*. Paper 283.

<http://docs.lib.purdue.edu/ecetr/283>

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.



Hydro-Acoustic Communication System Based on a Neural Network Error Corrector

M. F. Tenorio
R. Z. Machado F.
J. R. M. Silva

TR-EE 92-7
February 1992

**HYDRO-ACOUSTIC COMMUNICATION SYSTEM BASED ON A NEURAL
NETWORK ERROR CORRECTOR**

M. F. Tenorio (Member)**

Parallel Distributed Structures Laboratory

School of Electrical Engineering

Purdue University

W. Lafayette, In. 47907

R. Z. Machado F., (Member)*

Subsea Engineering Group

Petrobrás R&D Center

Rio de Janeiro, R. J.

Brazil

J. R. M. Silva***

Brazilian Naval Research Center

Ilha do Governador, R.J.

Brazil

ABSTRACT

This study describes an application of Neural Networks for transmission errors identification and correction in binary messages. The network is used as a classifier of detected hydro-acoustic signals into one of a possible alphabet of symbols. The algorithm used is a Hamming-type Neural Networks classifier associated with the transmission of a Hamming code. This system can detect and correct all transmission errors if the number of errors is less than or equal to half the Hamming distance between transmitted symbols minus one. Symbols to be transmitted are chosen and associated to messages, assuring that bit to bit non similarities result on the prescribed Hamming distance. The auto-associative error correcting scheme can be used to generate a teaching signal to a supervised learning equalizer tracking the channel non-stationary characteristics. The proposed system is intended for in hydro-acoustic communication applications, and it is currently undergoing sea tests.

I. INTRODUCTION

Exploration in deep seas of mineral and energy resources, as well as associated ambient monitoring and subsea engineering activities, requires data transmission between subsea stations and the surface. Due to very strong absorption by the Ocean of the electromagnetic spectra in the radio band, cables have been the natural choice for transmission of signals in water. Cables have been used by the industry for several decades, but in several deep sea operations, the drag imposed on the umbilical cables by the Ocean currents, cable laying and connection problems at the Ocean floor have led the

investigation into the use of sound carriers and acoustic telemetry [HAA70], [BAG81], [ZIE86], [SCO87], [CAT90b].

Ocean sound propagation have been studied in relation to sonar technology. Its properties and uses are well known, although severe constraints in their use are imposed by several varying, and sometimes unknown ambient conditions [LAG71], [URICK], [GOT83]. The main characteristics of the Ocean's acoustic channel are: low and varying propagation speed; increasing absorption of the higher frequencies; strong ambient noise; and presence of reverberation due to multi-path propagation. Range and carrier frequency selection is a design trade-off. Ranges of a few kilometers are possible with the use of carriers around and below 10 kHz. Higher frequency carriers will decrease the range. Using frequency carrier between 10 and 100 kHz, limited by range and absorption considerations, restricts the channel bandwidth and consequently its data transmission capacity.

The slow propagation speed (1500 m/s) imposes a limitation on the time required to data message travel to the receiver and back to the sender. To further complicate the design process, propagation speed variations, due to variations of temperature, salinity and pressure with depth along the channel, combined with surface and volume reverberations, introduce several anomalous propagation conditions, such as: ray bending, shadow zones, and multiple receptions of the same signal (fading). These properties of the acoustic fading channel limits maximum communication ranges, imposing operation with low SNR at longer ranges, and allowing conditions for occurrence of errors in the transmitted messages.

One possible application of acoustic telemetry is related with deep waters subsea oil field exploration, such as the giants Albacora and **Marlim** fields discovered at the Brazilian Continental Slope (depths from 400 to 1200 m), where development plans consider a large

number of subsea wells controlled and monitored by a central, semi-submersible production platform. Parameters to be monitored are valve position, temperature, and pressure, and valve actuation must be commanded. Conventional design employs umbilical cables between surface and subsea units with hydraulic supply and control lines and with electric supply and communication lines. In deep water operations, such umbilical cables can be very expensive and impose several additional restraints to the installation, handling, and operation of the production system. Another approach employs untethered subsea well concept, where the subsea well communicates with the floating platform through acoustic telemetry and generates electric energy locally (Figure 1). Acoustic telemetry has been already employed in subsea well control [FRA88].

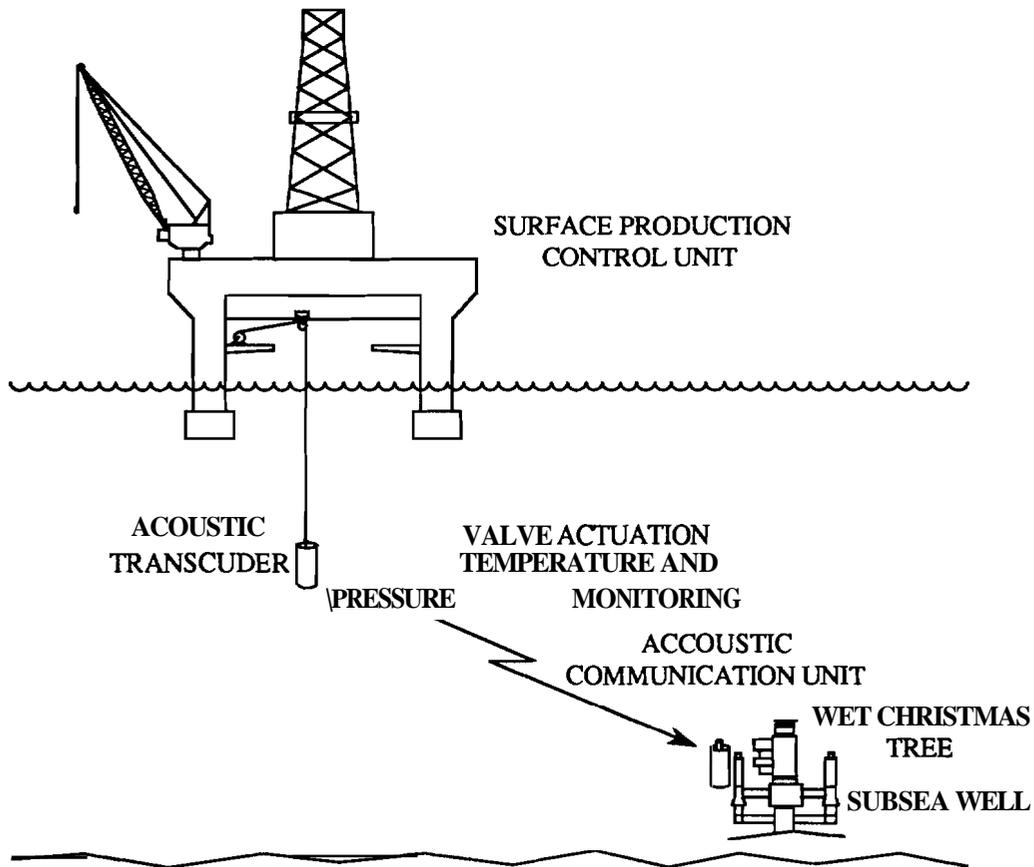


Figure 1. Acoustic Communication in Subsea Oil Production

Data transmissions associated to subsea oil production control applications requires high error immunity. If acoustic communication is considered, there will be restrictions to the use of data exchange protocols which requires message re-transmissions after error detection. The restrictions are due to the time required for message exchange between surface and subsea station, and to power availability at the subsea stations. Message encoding and decoding techniques allowing error correction to decrease the probability of error without re-transmission, for systems operating in the time-varying hydro-acoustic channel, was the main motivation for this study.

II. THE HYDROACOUSTIC COMMUNICATION SYSTEM

The Hydro-acoustic Communication System (HCS) being considered for the implementation of the proposed coding and decoding method uses MFSK modulation, mapping each bit position and value into a carrier frequency. MFSK modulation has been shown to be an effective technique for the underwater environment [WAX81], [GAR81].

Since the channel is noisy, fade-prone and reverberant, messages with transmission errors are likely to occur. The usual method to increase immunity to transmission errors is to introduce some sort of redundancy that can be implemented even at the modulation level [PRO91]. In our case, the final HCS design will probably have time or frequency diversity combined with an efficient modulation scheme [PIE78].

There are two preferred techniques used to avoid transmission errors in digital messages between two systems: automatic re-transmission, when an error is detected (Automatic Repeat Request - ARQ), and coding, that allows error correction (Forward Error Correction - FEC) [COU87]. To further increase immunity to errors, communication protocols normally include: parity bits of information in the message to detect transmission

errors, or signalling bits for message re-transmission (ARQ). These form of immunity leads, in the end, to higher energy consumption at the subsea units, which are powered by limited local power cells. For deep sea environment, such as in our application, this high power consumption is undesirable. An alternative is coding which usually requires that each bit of information or bit group be associated to some redundancy (parity bits, for example), through a compact coding scheme. The messages, \mathbf{M} , are translated into a pre-computed alphabet code, \mathbf{K} , that compactly encodes the message and associated redundancy into the alphabet symbols. So the encoding process is:

$$\mathcal{E}(\mu_j) = \kappa_j \quad \kappa_j \in \mathbf{K} \text{ and } \mu_j \in \mathbf{M}$$

and the associated transmission process, $\mathcal{T}(\cdot)$, which induces noise in the symbols, κ_j , forms words, ω_j , that are corrupted versions of the transmitted symbols κ_j

$$\mathcal{T}(\kappa_j) = \omega_j \quad \langle \omega_j \rangle = \kappa_j$$

The symbols, after decoding and classification in the receiving end, $\mathcal{D}(\cdot)$ and $\mathcal{C}(\cdot)$, are applied to a look-up table to reconstruct the original message.

$$\mathcal{D}(\omega_j) = \hat{\epsilon}_j \text{ and } \mathcal{C}(\hat{\epsilon}_j) = \hat{\kappa}_j,$$

where $\hat{\epsilon}_j$ is an estimate of the digital form of the original message (may be a non valid symbol in \mathbf{K}), and $\hat{\kappa}_j$ is an estimate of κ_j . The decoding and classification phases incorporate the reliability introduced by the code selected, thus increasing robustness, whereas the processing phase after the reconstruction of the message, $\mathcal{P}(\hat{\epsilon}_j)$, can, if desired, contain the actions for special handshake protocols.

Since there is a requirement to avoid automatic re-transmission of messages, we will focus on the coding technique. The idea behind coding is to associate the message to symbols with enough redundancy or orthogonality as to: reinforce message particularities,

and allow error identification and correction. Symbols derived from the message data are transmitted, κ_i , instead of the original message, μ_i . A block diagram of the HCS is presented in figure 2.

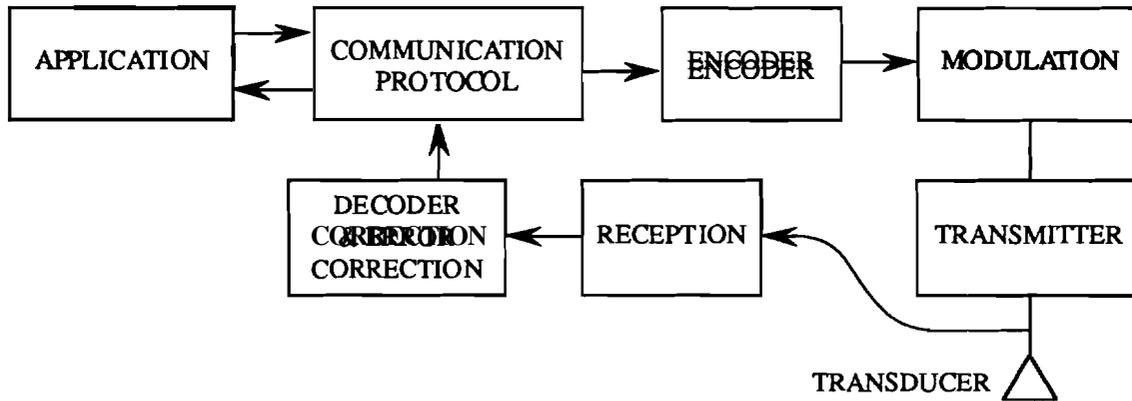


Figure 2. Block diagram of the transmission and reception process for the HCS module.

A communication protocol handles the data exchange between the application and the HCS. Before transmitting a message, the whole message or its characters are mapped to a symbol at the encoder. Each symbol is mapped to a modulation scheme to be transmitted. At the reception, the modulation frequency patterns is detected and converted to a symbol, that can be corrupted by errors. The decoder performs the correction and delivers the message to the protocol handler.

In the proposed HCS, the reception will include N channel parallel receivers able to detect and decide on the existence of each of the N message bits. The larger the number of receivers, the larger K is, and the more orthogonal the code can be made. With a larger alphabet, more messages can be sent per instance of time, by allowing message sequences to be assigned to symbols though savvy message-to-symbol encoding schemes. But with the increase in N, practical design and detection considerations, as well as limited bandwidth for the desired range, become the limiting factor, imposing restrictions on the

size of the alphabet with the desired orthogonality properties. After preprocessing and detection, the detected word will be decoded by the Neural Net into an estimated symbol.

III. CODING AND ERROR CORRECTION

Messages can be transmitted associated to block (memory-less) or to convolutional (with memory) codes. Properties and advantages of convolutional codes for hydro-acoustic transmission have been described [CAT90a]. In this study, we have concentrated on block codes.

Block codes are a mapping of binary sequence \underline{u} of length L into binary sequences \underline{x} of length N , $N > L$. These codes are referenced as codes (N,L) . Code rate is defined as $R = \frac{N}{L}$.

Since extensive use the concept of Hamming distance is used, it is now restated. *Hamming distance* between two binary code symbols is the number of "1" after XORing the symbol words, in other words, the number of differing positions. *Hamming weight* is the number of bits "1" in a code. Being s the number of errors that can be detected, and t the number of errors that can be corrected, the possibility of error detection and correction is defined by the relation (the distance between two code words) [COU87]:

$$d \geq t + s + 1 \quad s \geq t$$

Simultaneous detection and correction of the same number of errors imposes that the distance between the code words or symbols be:

$$d \geq 2t + 1$$

Systematic parity check codes are defined [GAL68] as any binary block code with length N where the set of message symbols is the set of 2^L binary sequences length L ($L < N$) and for each message \underline{u} (u_1, \dots, u_N), the associate code symbol \underline{x} (x_1, \dots, x_N) is given by:

$$x_n = u_n \quad 1 \leq n \leq L$$

$$x_n = \sum_{\kappa}^2 u_{\kappa} g_{\kappa,n} \quad L+1 \leq n \leq N \quad 1 \leq \kappa \leq L$$

The summation indicated above is the modulo 2. The first L digits of a message \underline{x} are **information** digits, and the last $N-L$ digits are check digits.

The code generation can be done algebraically using a code generator matrix \mathcal{G} :

$$\mathcal{G} = \begin{matrix} 1 & 0 & 0 & 0 & 0 & \dots & 0 & g_{1,L+1} & \dots & g_{1,N} \\ 0 & 1 & 0 & 0 & 0 & \dots & 0 & g_{2,L+1} & \dots & g_{2,N} \\ & & & & & & & \dots & & \\ 0 & 0 & 0 & 0 & 0 & \dots & 1 & g_{L,L+1} & \dots & g_{L,N} \end{matrix}$$

The code symbol is created by the multiplication (the associated summation is modulo 2) of the message sequence with the matrix \mathcal{G} :

$$\underline{x} = \underline{u} \cdot \mathcal{G}$$

Decoding and **error correction** can be also done algebraically with a parity check matrix \mathcal{H}

$$\mathcal{H} = \begin{matrix} g_{1,L+1} & \cdots & g_{1,N} \\ g_{2,L+1} & \cdots & g_{2,N} \\ \dots & & \\ g_{L,L+1} & \cdots & g_{L,N} \\ 1 & \cdots & 0 \\ 0 & 1 \cdots & 0 \\ \vdots & & \\ 0 & \cdots & 1 \end{matrix}$$

$$x_n = \sum_{\kappa} 2^{\kappa} x_{\kappa} g_{\kappa,n} \quad L < n \leq N \quad 1 \leq \kappa \leq L$$

For each code word \underline{x} , $\underline{x} \cdot \mathcal{H} = \mathbf{0}$ iff $\underline{x} = \kappa_j \in \mathbf{K}$, is expected. Occurring an error, the received sequence y will be associated to syndrome $S = y \cdot \mathcal{H}$, $S \neq \mathbf{0}$ iff $y \neq \kappa_j \in \mathbf{K}$. When the i th column of \mathcal{H} is equal to S that indicates an error in the i th message bit.

Sphere packed codes are those where any received sequence or is a valid code either is at a maximum distance d from only one valid code. Perfect codes are sphere packed codes where all errors combinations can be corrected. Hamming codes [HAM50] are a class of perfect codes. In the Hamming code all \mathcal{H} matrix lines are different and include all $N-L$ non zero sequences. For these characteristics, Hamming codes have been selected for this implementation.

It has been shown [GAL68] that in a symmetric binary channel, the maximum likelihood (ML) decoding is equivalent to choosing the code symbol k_m with the smaller Hamming distance from the received sequence \underline{y} . Hamming codes have 2^L different code symbols, with a minimum distance $d = 3$, allowing the correction of one and only one error. Unfortunately, Hamming codes are not available in all combinations of N and L . It can be verified that these codes only exists for a N and L relationship of $N = 2^{(N-L)} - 1$, or:

N	L
3	1
7	4
15	11
31	26
...	...

Substitution of each 0 or 1 code bit by different v length sequences, respectively, the code distances are increased to vd , allowing correction of more errors, but increasing the code rate.

Each L length code can be considered a vector as coordinates in a L dimensional space. These vectors are mapped to a N dimensional space ($L < N$), with the property that they are a set of d-spaced vectors in the N space. A vector corresponding to a message with **errors** will be always at distance less than d from a valid vector.

After the careful choice of the coding scheme to be used, $\mathcal{E}(\cdot)$, the next step in the design is the classifier, $C(\cdot)$, such that the original message symbols can be estimated and corrected. This is done on the HCS through a neural network classifier.

IV. NEURAL CLASSIFIER DESIGN

Neural networks are computing paradigm that loosely mimics the design of animal nervous systems. These networks are formed by grouping in a coupled system, a large number of stereotypical neurons, called neurodes, where: each link between neurodes has an associated weight; each **neurode** performs a simple quasi-linear integration of its inputs, and the behavior of the system can be easily modified by altering the associated weights through learning laws. This types of networks can be further subdivided in auto-associative networks (usually with recursive connections), which associate its inputs with a new copy of the same input, not corrupted by noise; and hetero-associative networks (usually of a feed-forward nature), which associates its inputs

with a related output that can be quite different in size and form of the pattern. For a more detailed explanation of the various types, the readers should refer to the excellent overview in [SIM88].

The objective of our investigation was to determine which network would be more suited for an underwater communication application, with the constraints as described before. Different learning algorithms are available to train networks: pre-wired, unsupervised, supervised, and penalty-reward. The first one, the weights are obtained by either design, or by off-line application of one of the other methods. The second case, the weights converge to a set of statistics of the input, and the last two cases require a feedback from a teaching signal. For underwater applications, it is difficult to generate a teaching signal remotely, which poses difficulties in the utilization of the last two paradigms. The unsupervised learning could work well, but due to constantly changing conditions in the water, tracking of a new set of parameters with non-stationary statistics may not be sufficiently fast or trivial. That leaves the off-line design of the classifier to be the best choice. When the set of possible patterns in a population is not known *a priori*, the best design is accomplished by creating the widest possible attractors per pattern. This is attained by the use of possibly orthogonal patterns, such as the ones in the coding scheme described before, and a nearest-neighbor classifier design, which in this case is the Hamming network [LIPP87]. The Hamming network is an auto-associative, recursive network that can be used on an off-line application for which the input pattern will converge to the nearest-neighbor pattern, pre-encoded on the network. It is very similar to the **Hopfield** network [SIMP89], except that the distances between patterns are measured in the Hamming sense. As pointed out by [LIPP87], the choice of the Hamming network over the **Hopfield** network was made due to the former's basic advantages over the latter. The **Hopfield** network is often tested on corrupted versions of its memorized binary patterns by flipping their bits randomly and independently given a certain probability. This is exactly the communication problem that is presented here: a binary fixed-length signal traveling over a memoryless binary symmetric channel. The optimum solution to this problem comes from calculating the Hamming distance of the exemplar to the prototypes of the

classes, and **choosing** the class with minimum Hamming distance. Since the code selected for the prototype patterns (valid symbols) guarantees a minimum Hamming separation between the classes to be selected by design, the use of such a scheme would work very well. The Hamming net calculates the Hamming distances between the exemplars and the store **memory** points, in our case, the prototype symbols for each class. Further, as a pure auto-associative device, the Hamming net has a number of advantages over the **Hopfield** net. As a classifier, the Hamming net implements the optimum **minimum** error classifier when bit errors are random and independent. It also requires fewer connections than the **Hopfield** net. For a 100 bit symbol with 10 possible symbols, the difference between the number of connections is almost an order of magnitude ($1,1 \times 10^3$ versus 10^4).

The Hamming network can perform nearest-neighbor classification of symbols using the Hamming metric, with the maximum distance property [KOH84]. The restriction that the patterns of the symbols would have maximum Hamming distance among them (maximum dissimilarity) is equivalent to the choice of orthogonal vectors, and it is a necessary restriction in creating a coding scheme between the messages and the symbols, $\mathcal{E}(\cdot)$. In designing this code, first the messages need to be identified, or their maximum number estimated, in order to determine the size of the symbols in bits to accomplish the coding task. Each symbol, κ_i , is to have the property of $\max(d_{\mathcal{H}am}(\kappa_i, \kappa_j))$ with any other symbol, κ_j , which would correspond, or be greater than, the number of messages to be transmitted. This would permit the design of attractors as large as $\max(d_{\mathcal{H}am}(\kappa_i, \kappa_j))/2$, to be greater or equal to the distance between a valid symbol and a received symbol, $d_{\mathcal{H}am}(\kappa_i, \hat{e}_i)$.

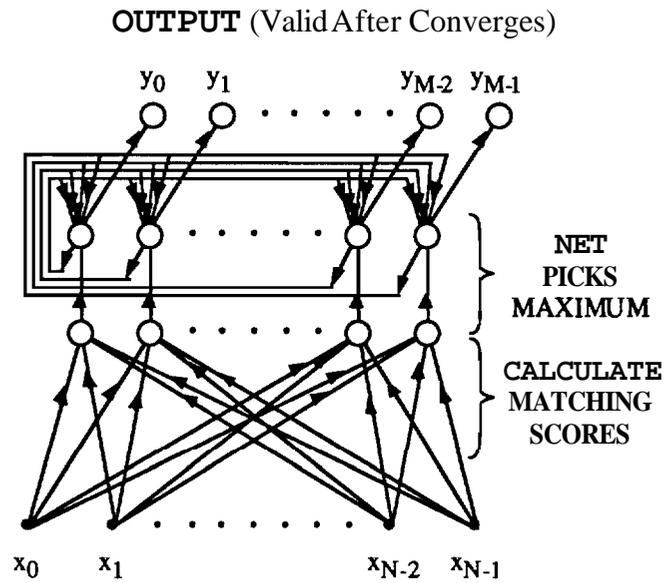


Figure 3. The feed-forward Hamming net performs a maximum likelihood classification for binary inputs **corrupted** by noise.

A Hamming network with N inputs and M symbols has an expected number of connections equal to $C_{Ham} = M \cdot (M + N)$. The network would have N input neurodes, followed by an intermediary layer and an output layer of M neurodes each. For simplicity, to maintain a small number of sub-indexes, the symbol \mathbf{k}_i can be refereed as:

$$\mathbf{k}_i = \underline{\mathbf{x}}^i = \underline{\mathbf{x}} = (x_1, x_2, x_3, \dots, x_N)$$

Let \mathbf{K} be the M prototype vectors, to be modified from a binary encoding $(0,1)$ to a bipolar encoding $(-1,1)$. The off-line weight design can be done by adjusting the input weights via:

$$w_{j,i} = \frac{x_i^j}{2} \quad i = 1..N-1, \quad j = 1..M-1 \quad \text{and}$$

$$w_{j,0} = \frac{N}{2} \quad j = 1..M-1 \quad (\text{bias term})$$

and the output weights via:

$$t_{kl} = \begin{cases} 1, & k=l \\ -\epsilon, & k \neq l, \quad \epsilon < \frac{1}{M}, \quad 0 \leq k, l \leq M-1 \end{cases}$$

where w_{ij} is the connection weight from input i to node j in the input layer, and t_{kl} is the connection from node k to node l in the output layer. During recall, the input vector places at the input of the j th element:

$$O_j(o) = f_t \left(\sum_{i=0}^{N-1} w_{ij} x_i - w_o \right) \quad 0 \leq j \leq M - 1$$

where $O_j(t)$ is the output of the node j in the output unit at time t , x_i is the i th element of the input, and f_t is the threshold logic nonlinearity (a linear output between 0 and u , and constant outside the linear range). The computing process is repeat until convergence, after which the output of only one onde remains positive:

$$O_j(t+1) = f_t \left(O_j(t) - \epsilon \sum_{k \neq j} O_k(t) \right) \quad 0 \leq j, k \leq M - 1$$

The convergence is obtained in only a few iterations, and the process of receiving a new symbol is restarted.

V. IMPLEMENTING THE NETWORK

The input layer will need as many neurodes as the number of bits of the code vectors. The neurodes in that layer will have a Threshold Linear (or sigmum) transfer function (a linearized version of the ubiquitous sigmoid function). As in figure 3, all output neurodes receive connections from the input layer. There are as many nodes in the output layer as the number of desired prototypes, or in our case, valid symbols, or $2L$. In the output layer, each neurode is connected to its neighboring counterpart. The output layer can be also substituted during implementation for a winner-takes-all function.

The concept was tested considering an alphabet of 16 different messages, which corresponds to the different valve commands, therefore $L = 4$ bits. For a minimum Hamming distance of 3 (and consequently correction of one error per message), the code to be defined was the (7,4). Tables 1 and 2 present the (7,4) code and the Hamming distance between the 16 components. notice that some components have distance higher than 3, but never lower. In this case, the net will have 7 neurodes at the input layer and 16 neurodes at the output layer. Extensions to a larger alphabet or larger minimum distance are immediate. To prepare experimental data, files of the (7,4) code has been prepared, for all 16 combinations of 4 bits, as well as Hamming distances higher than N for corrections of 2 and 3 errors per pattern.

Table 1: Hamming Codes (7,4)

0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	1	0	0	1	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	1	0	0
1	0	1	0	1	0	1
1	0	1	1	0	1	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	1	0	0	0	0
1	1	1	1	1	1	1

It is well known that the probability of error per bit in a message is a function of the **signal-to-noise ratio**. To complete our design procedure, we need to establish the probability of multiple errors in a message at a real off-shore environment, where other aspects like reverberation will be present, and some of the assumption about the noise model do not hold. A field experiment is

being performed at the Campos Basin, using acoustic data transmission between an oil production platform and a subsea unit. Data consists of exemplar files with codes for correction of 1, 2 and 3 errors; and it is being transmitted in different geometries (varying depths from 20 to 400 meters and horizontally, in ranges from 0 to 2000 meters).

Table 2: Hamming Distance Matrix for Code (7,4), Between all Symbols

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	4	3	3	3	3	4	4	3	3	4	4	4	4	3	7
2	4	0	3	3	3	3	4	4	3	3	4	4	4	4	7	3
3	3	3	0	4	4	4	3	3	4	4	3	3	3	7	4	4
4	3	3	4	0	4	4	3	3	4	4	3	3	7	3	4	4
5	3	3	4	4	0	4	3	3	4	4	3	7	3	3	4	4
6	3	3	4	4	4	0	3	3	4	4	7	3	3	3	4	4
7	4	4	3	3	3	3	0	4	3	7	4	4	4	4	3	3
8	4	4	3	3	3	3	4	0	7	3	4	4	4	4	3	3
9	3	3	4	4	4	4	3	7	0	4	3	3	3	3	4	4
10	3	3	4	4	4	4	7	3	4	0	3	3	3	3	4	4
11	4	4	3	3	3	7	4	4	3	3	0	4	4	4	3	3
12	4	4	3	3	7	3	4	4	3	3	4	0	4	4	3	3
13	4	4	3	7	3	3	4	4	3	3	4	4	0	4	3	3
14	4	4	7	3	3	3	4	4	3	3	4	4	4	0	3	3
15	3	7	4	4	4	4	3	3	4	4	3	3	3	3	0	4
16	7	3	4	4	4	4	3	3	4	4	3	3	3	3	4	0

VI. ENHANCEMENTS WITH SUPERVISED LEARNING NETWORK

As discussed before, supervised learning networks are difficult to apply in a remote communication design due to the difficulty of generating on-line training signals. In our case, the Hamming net may be able to provide such a signal. Ocean channel characteristics tend to be very susceptible to variation in temperature and other environment conditions. This makes the channel characteristics non-stationary. The performance of the HCS system can be much enhanced with the use of a tracking equalizer that would correct the received signal for such variation.

Initially, the protocol would send known messages, κ_{i-n} to κ_i , that would **try** to excite the different frequencies of the channel. Since the message is known, it can be used as a training signal to a supervised learning network, such as a multi-layer perceptron. To account for the larger linear component of this equalization procedure, linear connections directly between the input and output are added to the classical design. The back-propagation algorithm is used to adjust the weights in order to minimize the error between the messages received and the internal copy, $\mathcal{B}(\omega_i, \omega_{i-n}, \hat{e}_i, \hat{e}_{i-n}, \kappa_i, \kappa_{i-n}, \mathcal{D}^i) = \mathcal{D}^{i+1}$. Notice that more than one message in time is needed to create a delay line input with parallel input nodes for each bit at each time instance. After convergence of the network, the channel is equalized for the first command message, assuming that the channel varies slower than the time to compute the equalizer.

During normal operations, since the first few messages are equalized, it is reasonable to assume that less than the maximum tolerable number of errors will occur. At each message reception within this prescribed bound, the HCS system is still capable of producing the correct output code, $\hat{\alpha}_i$. This is then used by the supervised learning algorithm as a teaching signal, literally tracking the changing conditions of the environment on-line. The equalizer has two objectives: first, to keep \hat{e}_i exactly like α_i in the presence of error; second, to track the channel by equalizing ω_i as to make it easy to detect \hat{e}_i . This synergy of the associative design with a supervised design will give the system robustness to a variety of Ocean conditions. The **supervised** result will be tested after the channel data acquisition with the Hamming code has taken place.

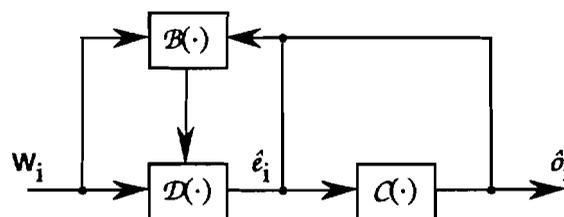


Figure 4. The automatic equalizer tracks the channel on-line using the output of the Hamming net

V. CONCLUSIONS

Hamming networks are one solution for the transmission error correction problem in acoustic telemetry, provided that a code with attractors with the required Hamming distance is chosen, to correct a determined maximum number of errors in a message. Software implementation of the network for subsea oil well control, where the messages are few and short and transmission speed is slow, allows almost real-time operation. For higher transmission speeds and larger messages, a hardware implementation will be necessary. A network implemented in hardware will probably be sufficiently fast to be considered for data transmission applications other than acoustic.

The acoustic transmission error in Ocean media is related to time variations of the signal-to-noise ratio and or to reverberation ratio, due to man-made noise (platform industrial activity or ship traffic). A third possibility is ambient parameters variation such as wind, rain, variation in salinity and temperature (altering sound speed profiles and allowing reflection, refractions, multiple propagations, and signal fading). The combination of associative and supervised networks will be investigated as a powerful paradigm to counter-act the changes in the environment, making the system robust to more errors than specified in the code design.

Another interesting area of future investigation is related to the selection of subset of attractors with a specified Hamming distance and smaller length. This problem arises when in attending a design specification of a certain number of errors to be corrected, and the number of different messages, the existing block codes can present larger sets that necessary. This length can be so large as to render the raw transmission of the message with redundancy far more efficient. Neural networks can be a tool for the optimized search of these simplified attractors.

This design experiment has shown that Neural Networks can be a powerful tool for application in hydro-acoustic transmission and telemetry. The combination of various paradigms, perhaps not obvious from the theoretical literature, can be of great value to the system designer, in search of more robust and economical solutions.

REFERENCES

- [BAG81] Baggeroer, A.B., "A Survey of Acoustic Telemetry," **Oceans'81 Proc.**, pp. 48-54.
- [CAT90b] Catipovic, J.A., "Performance Limitations in Underwater Acoustic Telemetry," **IEEE Journal of Oceanic Engineering**, vol. 19, no. 03, July 1990, pp. 205-216.
- [COU87] Couch, L.W., "Digital and Analog Communication System," Macmillan, 1987.
- [FRA88] Franceschini, G., **Chimisso, C., Dal Lago, C.**, "Subsea Wells Acoustic Control System," **ROV/UTC Conf.**, Ap., 1988.
- [GAL68] Gallager, R.G., "Information Theory and Reliable Communication," John Wiley & Sons, 1968.
- [GAR81] **Garrod, D.J.**, "Application of the MFSK Acoustical Communication System," **Oceans'81**, 67-71.
- [GOT83] Gotherstrom, L., "Underwater Telemetry and the Acoustical Transmission Channel," **Swed. Trade Fair & Intern. Offshore & Marine Technology**, 1983.
- [HAA70] Haase, W.H., "Acoustic Valve Operating System (AVOS)," **Offshore Technology Conference Proc.**, 1970, pp. 269-278.
- [HAM50] Hamming, R.W., "Error Detecting and Error Correcting Codes," **Bell Syst. Tech. J.**, vol. 29, April 1950, pp. 147-160.
- [LAG71] **Lagoe, J.A.**, "The Subsea Acoustic Communication Environment," **Proceedings, 7th Annual Conf. Marine Techn. Soc.**, 1971, pp. 259-264.
- [LIPP87] Lippman, R.P., "An Introduction to Computing with Neural Nets," **IEEE ASSP Magazine**, vol. 4, n. 2, April 1987.
-

- [PIE78] Pieper, J.F., Proakis, J.G., Reed, R.R., Wolf, J.K., "Design of Efficient Coding and Modulation for a Rayleigh Fading Channel," IEEE Trans. Inf. Theory, vol. IT-24, no. 4, July 1978, pp. 457-468.
- [PRO91] Proakis, J.G., "Coded Modulation for Digital Communication over Rayleigh Fading Channels," IEEE Journal of Ocean. **Engin.**, vol. 16, no. 1, Jan. 1991, pp. 66-73.
- [SIM89] Simpson, P.K., Artificial Neural Systems, Pergamon Press, 1989.
- [SCO87] Scofield, J.R., Baxter, C.F.G., "Recent Innovations Improve Offshore Operations: Bottomhole Pressure Monitor Aids Development Planning," Ocean Industry, Nov. 1987, pp. 26-27.
- [URI82] Urick, R.J., "Sound Propagation in the Sea," Peninsula Publishing, 1982.
- [WAX81] Wax, D.W., "MFSK - The Basis for Acoustical Communication System," Oceans '81 Proc., pp. 61-66.
- [ZIE86] Zielinski, A., "Acoustic Telemetry and Communication," 2nd. Mar. Techn. **Soc.** et al. Mar. Techn. Pacific Congr. **Proc.**, 1986, pp. 31-35.
-