The Effect of Forward Error Correction on the Performance of Underwater Telemetry Systems

Douglas R. Sweet and Jimmy Wang

DSTO, Adelaide, Australia

ABSTRACT

A low probability of intercept (LPI) underwater acoustic telemetry system (UATS) is proposed to transmit sonar products from seabed arrays to submarines and surface relay buoys. A UATS simulator has been developed to implement all the steps in the processing chain. Underwater LPI systems have low transmitter power and hence low data rates. Techniques have been developed in DSTO to *compress* sonar products (2D images) to enable transmission at lower data rates without significant degradation. This paper studies bit error rates (BERs) and achievable bit rates for transmitting random messages and compressed sonar products (CSPs) from the standard UATS, with no channel coding, through Gaussian and fading channels. It also assesses the quality of the received and uncompressed CSP images. The effects of applying two types of forward error correction (FEC) - convolutional and *turbo* coding - in Gaussian and fading channels are then evaluated. A significant improvement in performance for both random messages and CSPs and hence data rates arises from using convolutional coding in Gaussian channels, but this improvement is lost in some fading channels. A greater improvement arises from using turbo coding in Gaussian channels. For fading channels, this improvement is less but still significant.

INTRODUCTION

An underwater acoustic telemetry system (UATS) is proposed to transmit sonar products from seabed arrays to submarines and surface relay buoys. A requirement of the system is to have a low probability of intercept (LPI). Underwater LPI systems have low transmitter power and hence low data rates, typically 100 bits/s (bps). Techniques have been developed in DSTO (Wang and Cox 2003) to *compress* sonar products - LOFARgrams and bearing-time records (BTRs) - to enable transmission at lower data rates without significant degradation. This paper studies bit error rates (BERs) and achievable bit rates for transmitting random messages and compressed sonar products (CSPs) from the standard UATS through Gaussian and fading channels. BERs are measured using a UATS simulator and compared with theoretical ones. The quality of the received and uncompressed CSPs (2D images) is also assessed.

The effect of applying two forward error correction (FEC) techniques - convolutional and *turbo* coding - in Gaussian and fading channels is then evaluated by simulation. Improvements (or otherwise) in BERs and available data rates are estimated. The quality of the received and uncompressed CSP images is also assessed. Some techniques are suggested which may mitigate the effect of fading channels.

The structure of the paper is as follows. In the next section, the UATS simulator used for estimating BERs is described. In the following section, the DSTO compression technique is introduced. Theoretical results are then presented for BERs and data rates in Gaussian and fading channels. In the section after, BER estimates and available data rates from UATS simulations are presented for uncoded transmission, convolutional coding and turbo coding, for Gaussian and fading channels. Finally, some conclusions are drawn and suggestions for future work presented.

UATS & SIMULATOR STRUCTURE

We have written a "virtual" underwater communications simulator which implements the UATS processing chain from the input message through channel encoding, modulation and transmission, channel simulation, reception and demodulation, and channel decoding. Details of the main simulator components are as follows.

Channel encoding

The simulator allows three options for FEC or channel coding: (1) no coding (the *standard* UATS), (2) convolutional coding and (3) turbo coding.

Convolutional coding.

In any FEC method, redundant *parity* or *error-correction bits* are added to the data stream to produce codewords. When the codewords are transmitted through the channel, some bits may be corrupted by noise. However, the decoder corrects these errors by finding the nearest valid codeword sequence according to some "distance" criterion. In a simple *convolutional* encoder, the input bit stream is passed through a *shift register*. The number of stages in the shift register is called the *constraint length.* The shift register has two or more sets of *taps* to combine the data passing through the shift register by modulo-2 addition, and the output of the adders is combined into codewords. Having more than one set of taps provides the codeword redundancy.

Turbo coding.

A comprehensive description of turbo coding is given by Barbulescu (Barbulescu 2004). The performance of turbo-coded systems can be significantly better than that of conventionally-coded systems, and can allow data rates close to the Shannon limit. In a typical turbo-coded system, the input data stream is sent to two parallel branches. In the first branch, the data stream is encoded by a conventional code such as a block code or a convolutional code. In the second branch, the data are first *interleaved* in a buffer, and then

encoded by another conventional code (which can be the same as in the first branch) – the codes in each branch are called *constituent* codes. The data streams are multiplexed and sent through the channel. In the turbo code used by the UATS simulator, an identical convolutional code is used in each branch.

UATS modulator and transmitter

The data in the system is encoded using differential phase-shift keying (DPSK) modulation. DPSK imparts tolerance to Doppler-induced phase drifts. The signal is then multiplied by the pseudonoise spreading code, which is a pseudorandom sequence of *chips* (-1s and 1s), with a chip rate much higher than the data rate. The modulated, spectrally-spread signal is then transmitted through the channel.

UATS simulator channel model

The current UATS simulator implements two simple channel models – direct path with Gaussian noise, and Rician fading with Gaussian noise. Future versions of the simulator will implement more sophisticated channel models, including multipath, variable sound speed profiles, reverberation and temporal channel variability.

Direct path with Gaussian noise

In this model, the received signal is simulated by two components – the transmitted signal with power scaled down by the transmission loss from the transmitter to the receiver, and ocean acoustic noise. Thus the received signal is

$$r(t) = \sqrt{l_T^{-1}(r)}\, s(t) + n(t) \qquad (1)$$

where $s(t)$ is the transmitted signal, $l_T(r)$ is the transmission loss at range r, and $n(t)$ is the ocean acoustic noise. Straight-line transmission loss has two components – spherical spreading and absorption. Let $TL(r)$ = 10 $\log_{10} l_T(r)$ be the transmission loss expressed in decibels. Then

$$TL(r) = 20 \log_{10} r + \alpha r \qquad (2)$$

where α is the attenuation coefficient. The UATS simulator implements the Thorp (Thorp 1965) absorption formula. The noise at each time sample t is generated by

$$n(t) = \sigma^2 \,\text{randn}(t)$$

where randn(t) is a normal random number generator with unity variance; and σ^2 is the noise power, given by $\sigma^2 = l_N B$ where l_N is the Cato (Cato 2000) noise power spectral level and B is the system bandwidth.

Rician fading with Gaussian noise

In a typical underwater acoustic channel, the received signal can be considered to be made up of a direct-path (nonfading) component plus multiple rays reflected from the bottom and/or the surface. Let p_M be the power of the multipath components relative to the power of the direct-path component. Then Rician fading can be simulated by multiplying the amplitude of the direct path signal by a factor f_R , the amplitude of the sum of two phasors, one of unity amplitude and fixed phase (representing the direct-path component), and one random amplitude and phase with average power $p_M / 2$ and random phase. The probability density function (PDF) of f_R can be derived from the Rician PDF (Greenwood & Hanzo 1994) and is given by

Error! Objects cannot be created from editing field codes. (3)

To simulate the fade period typical of the underwater channel, a low-pass filter is applied to the sequence $f_R(t)$ with cutoff frequency the inverse of the fade period, yielding a fading envelope $\hat{f}_R(t)$. Hence from (1), the received signal is simulated by

$$r(t) = \hat{f}_R(t)\sqrt{l_T^{-1}(r)}\, s(t) + n(t) \quad . \qquad (4)$$

UATS receiver and demodulator

The receiver maintains synchronization with the incoming data frames, and also tracks the Doppler arising from platform motions. Various algorithms are available for these operations. The carrier is then removed by demodulation, then the pseudonoise spreading is removed (*despread*) by a correlation process, leaving a phase sequence which is the transmitted sequence corrupted by noise. The phase difference between adjacent data frames is passed to the channel decoder.

Channel decoding

The UATS simulator implements decoding corresponding to the encoding scheme – no coding, convolutional coding or turbo coding.

No coding.

In this case, the phase difference between the adjacent frames is simply used to make a decision on the symbol. For example, for differential binary phase shift keying (DBPSK), the bit is set to 1 (0) if the phase difference is less than (greater than or equal to) $\pi/2$.

Convolutional decoding.

Most convolutional decoders use the Viterbi algorithm (Viterbi 1971) to find the nearest valid codeword sequence to that received using the maximum likelihood criterion. There are two variants of the Viterbi decoder – hard decoding, where received bits are quantized to two levels (0 or 1) before decoding, and soft decoding, where received bits are not quantized, or quantized to more than two levels between 0 and 1 before decoding. The latter decoder has a better performance than the former. The UATS simulator can use either hard or soft Viterbi decoding.

Turbo decoding.

At the receiver, the data streams are demultiplexed, and decoding is carried out by an iterative soft decoding technique. Details of the decoding algorithm are given in (Berrou et el 1993).

COMPRESSION FOR SONAR PRODUCTS

Typical sonar products which are transmitted from seabed arrays include LOFARgrams and bearing-time records (BTRs). A LOFARgram is a 2D display of power spectral density as a function of time, with frequency varying along the x-axis and time along the y-axis. A BTR is a time-history of integrated power versus bearing in a selected frequency band. It is a 2D display of power in the selected band, with bearing varying along the x-axis, and time varying along the y-axis. Target detections from LOFARgrams and BTRs are passed to trackers and classifiers to identify and track targets of interest.

These 2D displays contain a large amount of information – for example, a LOFARgram of 1 hour's duration, with an integration time of 61s and 600 frequency bins for each horizontal line of the display, typically requires about 73 KB. This cannot be transmitted in real time at typical LPI data transmission rates (~100 bps, see next section). Compression techniques have been developed in DSTO (Wang and Cox, 2003), which greatly reduce the data requirements of the images (a typical reduction factor is 50) without noticeably degrading the quality of the images. The user can select a "quality" factor (QF), which tunes the tradeoff between image quality and required data transmission rate. A low QF has a low data rate requirement, but the image may be noticeably degraded. A high QF has a higher data rate requirement, but it is still much lower than that required for raw, uncompressed data, and there is no noticeable difference between the reconstituted, decompressed image and the original image. Use of these compression techniques is expected to allow real-time transmission of several types of sonar products.

THEORETICAL LPI BER AND DATA RATES

We first give an analysis of the performance of the standard UATS in a Gaussian channel and for the system with convolutional and turbo coding. We then extend that to the case of a Rician fading channel with Gaussian noise.

Gaussian Channel – no FEC

The following analysis is a review of that given in (Sweet, 2005), with more detail in the development here. For the standard UATS operating in a Gaussian channel, the following well-known formulae relate bit error probability P_b (also termed the bit error rate, BER), to bit energy to noise spectral density ratio E_b/N_o for k bits per symbol DPSK :

$$P_b \cong F_k(E_b/N_0) \quad \text{, with} \tag{5}$$

$$F_1(x) \equiv \frac{e^{-x}}{2}, \text{and } F_k(x) \equiv \frac{1}{k}\text{erfc}[\sqrt{kx}\sin(\pi/2^{k+1/2})], \quad k \geq 2 .$$

For example, k=1 for binary phase shift keying (BPSK). Note that eq.(5) is exact for BPSK. In a spread-spectrum pseudonoise system, the ratio E_s/N_0, where $E_s = kE_b$ is the *symbol* energy, can be shown to be

$$E_s / N_0 = \rho_{in} L = \rho_{in} f_{ch} / R_s = \rho_{in} f_{ch} k / R_b \tag{6}$$

where ρ_{in} is the input signal-to-noise ratio (SNR) at the receiver (i.e. the ratio of maximum signal to noise power spectral density – PSD); L is the length of the pseudonoise sequence, f_{ch} is the pseudonoise chip rate, R_s is the symbol rate and R_b is the bit rate. Using (6) in (5), we obtain

$$P_b = F_k(10^{SNR_{in}/10} f_{ch} / R_b) \tag{7}$$

where $SNR_{in} \equiv 10 \log_{10}\rho_{in}$ is the input SNR in decibels. The dashed curves of Figure 1 plot P_b against bit rate in a Gaussian channel for various values of SNR_{in} .

CRITICAL AND MAXIMUM RANGES

In a LPI UATS, the transmitter power is limited by the requirement to keep SNR_{in} at the minimum presumed intercept range, at or below some intercept detection threshold, typically 0 dB. The range at which the input SNR equals this level is defined as the *critical range* r_{crit} . At ranges $r > r_{crit}$, SNR_{in} is less than 0 dB, and is given by $SNR_{in} = TL(r_{crit}) - TL(r)$, where $TL(r)$ is the transmission loss at range r. For a Gaussian channel, this is given by eq.(2). We define the *maximum range* r_{max} as the range at which the BER reaches some maximum acceptable level $P_{b,max}$, typically 10^4.

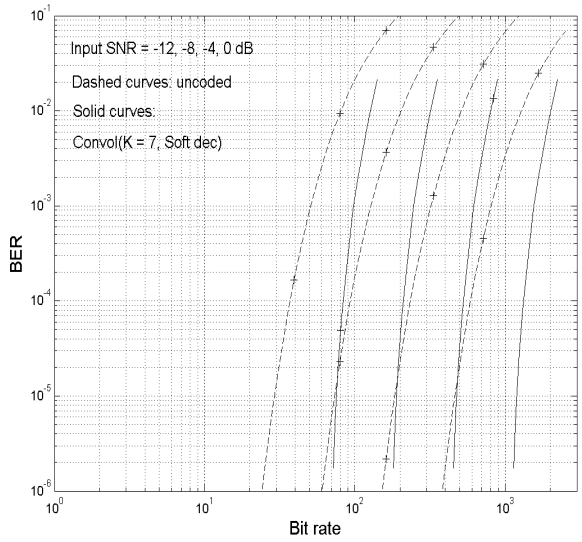

Figure 1. Bit error rate versus bit rate for uncoded and convolutionally coded UATS for various values of input SNR

Estimation of achievable bit rate at maximum range

If r_{crit} , r_{max} and $P_{b,max}$ are given, we can compute the achievable bit rate at r_{max} as follows:

1. Find $TL(r_{crit})$ and $TL(r_{max})$ from eq.(2)
2. Compute input SNR at maximum range by $SNR_{in} = TL(r_{crit}) - TL(r_{max})$
3. Solve (7) for R_b

As an example, for r_{crit} = 20 km and r_{max} = 27 km, we find from steps 1 and 2 of the above procedure that SNR_{in} = -7.5dB at r_{max}. If the maximum acceptable BER, $P_{b,max}$ is 10^{-4} , then from the dashed P_b versus R_b curves of figure 1, we find that the achievable bit rate is 100 bps.

Gaussian Channel with FEC

This analysis is a review of that given in (Sweet, 2005). The effect of FEC is to reduce the required ratio E_b/N_0 necessary to produce a given BER, and this reduction factor g_c is called the *coding gain*. It can be shown that for a coded system, eq. (7) can be replaced with

$$P_b = F_k(10^{SNR_{in}/10} g_c f_{ch} / R_b) \ . \tag{8}$$

Convolutional coding

The value of g_c depends on the modulation, the code and E_b/N_0 – for a convolutional code with constraint length 7, g_c ranges from 1 to 4.5 dB. Note that these estimates have been derived from published curves using *coherent* PSK and are only approximate. The solid curves of figure 1 plot P_b against bit rate for convolutional coding in a Gaussian channel for various values of SNR_{in} using these estimates of g_c .

The achievable bit rate at the maximum range can be calculated as above, with eq.(8) replacing eq.(7). For example, for r_{crit} = 20 km and r_{max} = 27 km, we find from steps 1 and 2 above that SNR_{in} = -7.5dB at r_{max} as above. If the maximum acceptable BER, $P_{b,max}$ is 10^{-4} , then from the solid P_b versus R_b curves of figure 1, we find that the estimated bit rate is about 200 bps, compared to 100 bps for the uncoded system.

Turbo coding

In a turbo coded system, coding gains can be considerably greater than for conventionally coded systems. For a typical turbo code with convolutional codes in each branch of the encoder and an interleaver buffer size of 512, the coding gain g_c varies from 2.5 dB to 6.7 dB. Note that these estimates have been derived from published curves using *coherent* PSK and are only approximate. The solid curves of figure 2 plot P_b against bit rate for turbo coding in a Gaussian channel for various values of SNR_{in} using these estimates of g_c . For r_{crit} = 20 km and r_{max} = 27 km, SNR_{in} = -7.5dB at r_{max} , as above. If the maximum acceptable BER, $P_{b,max}$ is 10^{-4} , then from the solid P_b versus R_b curve in figure 2 we see that the bit rate is predicted to be about 350 bps.

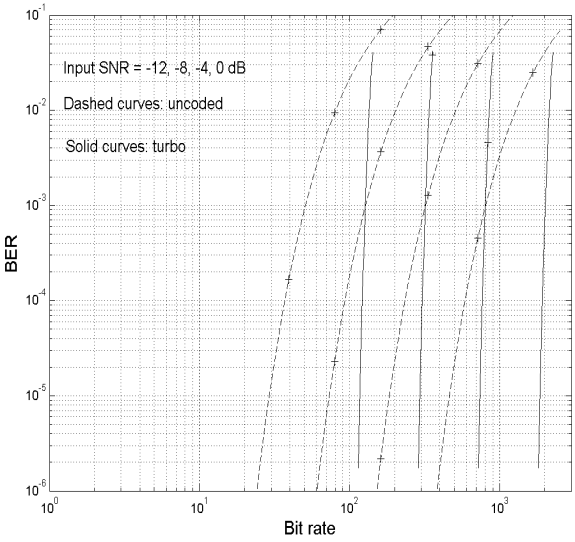

Figure 2. Bit error rate vs bit rate for uncoded and turbo coded UATS for various values of input SNR

Fading channel with Gaussian noise – no FEC

Consider the BER of the standard UATS at any fixed time in a channel with Rician fading. The actual value of E_b/N_o, say E_b'/N_o is related to the direct-path value of E_b/N_o by

$$E_b'/N_0 = f_R^2 E_b/N_0$$

where f_R is the fading factor as defined above, and the BER is given by

$$P_B(f_R) = F_k(f_R^2 E_b/N_0) \qquad (9)$$

where the function F_k is as defined in (5). The ensemble average of $P_B(f_R)$ over the distribution of f_R in the Rician distribution is

$$\tilde{P}_B = \int_0^\infty P_B(f_R)\,p(f_R)\,df_R \qquad (10)$$

with $p(f_R)$ as in (3). Putting (9) and (3) into (10) yields, for DBPSK

$$\tilde{P}_B = \frac{1}{2p_M}\int_0^\infty f_R \exp\left[-\frac{(f_R^2+1)}{2p_M} - \frac{f_R^2 E_b}{N_0}\right] I_0\left(\frac{f_R}{p_M}\right) df_R \qquad (11)$$

For a long message sequence or for multiple shorter messages, eq.(11) gives an average BER for a Rician fading channel. The solid curve of figure 3 gives the BER curve for p_M = 2, corresponding to a ratio σ/A of 1, where σ is the square root of the multipath power and A is the direct path amplitude. The dashed curve of Figure 3 gives the BER curve for σ/A = ½ . For comparison, the dash-dotted curve gives the BER curve for the non-fading channel. Note that for low E_b/N_0 , fading and non-fading BERs are similar. However, as E_b/N_0 increases, the gap between fading and non-fading BERs increases – at higher E_b/N_0 , there is a "error floor" effect in that BER is dominated by the deeper fading periods.

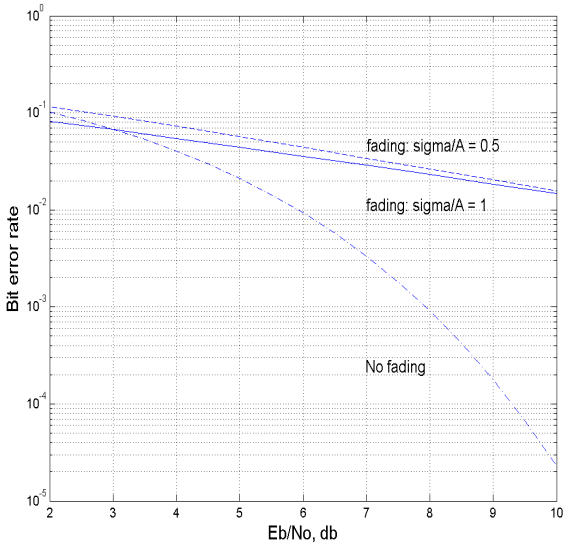

Figure 3. Theoretical BER curves for standard UATS: non-fading and Rician fading channels

Fading channel with Gaussian noise, FEC

There are no general theoretical results for the performance of the UATS with FEC in fading channels. Theoretical results are available for some cases, various bounds on BER are presented in the literature, and simulation results are presented for particular modulation and coding schemes. Our results on the performance of the UATS with convolutional and turbo coding are presented in the simulations below.

SIMULATION RESULTS

We first present BER estimates for random signals and CSPs transmitted through Gaussian and fading channels for the standard UATS (no coding), then repeat the simulations for convolutional and turbo coding. We have found that the CSP BER curves are very similar for LOFARgrams and BTRs for a range of QFs, except that there is a small scatter of about 0.5 dB at low BERs for coded transmissions in fading channels. This scatter can be attributed to the burstiness of errors from the decoders in a fading channel. For clarity, therefore, we present BER curves for a representative CSP – a LOFARgram with low QF. We also present representative displays of the transmitted images, and discuss their quality.

Standard UATS – no coding

The lower dashed curve of figure 4 shows the theoretical BER for the standard UATS. The simulated BER curve for *random* messages matches the theoretical curve, and is not presented here. Hence, the above analysis for available LPI bit rates applies, and the system can support a bit rate of 100 bps at BER 10^{-4}.

The lower dash-dotted curve shows the corresponding BER curve for the representative CSP, and it follows that for random messages closely. Hence the available data rates are the same as for random messages.

The solid curve shows the simulated BER for random messages sent through a Rician fading channel with a multipath power twice the direct path power. At higher values of E_b/N_0 , performance in a fading channel is much worse than that in a Gaussian channel. The corresponding theoretical curve is the "sigma/A = 1" curve of figure 3. The simulated curve is a little above this curve, and is closer to the "sigma/A = 0.5" curve of figure 3. This could be explained by the fact that the UATS simulator actually applies a low-pass filter to the Rician fading envelope to simulate the specified fade period, hence the fading variation from the multipath is reduced.

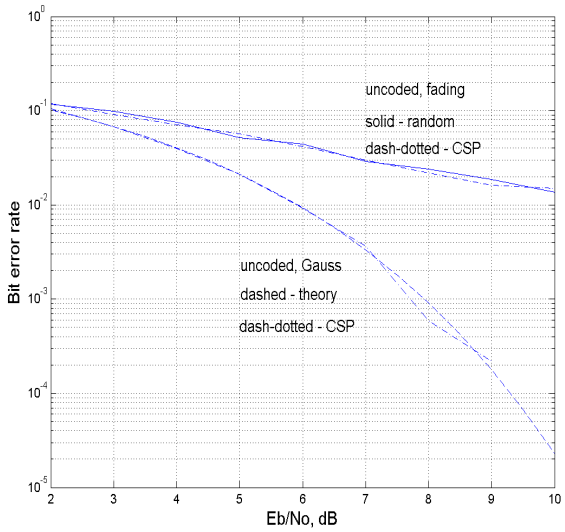

Figure 4. Simulated BERs for Gaussian (lower curves) and fading (upper curves) channel, no coding

The upper dash-dotted curve shows the corresponding BER curve for the representative CSP, and it follows that for random messages closely.

Figure 5 gives a typical low QF LOFARgram display for error-free transmission. Figure 6 displays the same data transmitted through the UATS at a bit error rate of 10^{-4}. There is some "noise" in the upper third of the picture; however, the major features are still clearly visible. This data was also transmitted at BER 10^{-2}, however, the received data could not be displayed. We assume that this was caused by corruption of vital header data at the higher BER. We therefore conclude that in the current UATS implementation, data rates available for BER = 10^{-4} (viz. 100 bps), but not those for BER = 10^{-2}, can be achieved for transmitting CSPs in an uncoded system in a Gaussian channel. We observe, however, that in a fading channel, required values of E_b/N_0 are much greater for a given BER (see figure 4), hence available data rates are much lower.

UATS with convolutional coding (soft decoding)

The middle dashed curve of figure 7 shows the theoretical BER for the standard UATS. The simulated BER curve for convolutional coding of random messages in a Gaussian channel is given by the lower solid curve. From this curve, the *coding gain* (reduction in required E_b/N_0) for BER = 10^{-4} is 2.2 dB. From eq.(8), this will allow an increase of $10^{0.22} = 1.7$ in data rate to 170 bps, a little less than that estimated in the section on theoretical BER and bit rate. However, the latter estimates are only approximate, and convolutional coding with soft decoding still allows a significant improvement in data rate. The coding gain increases as E_b/N_0 increases.

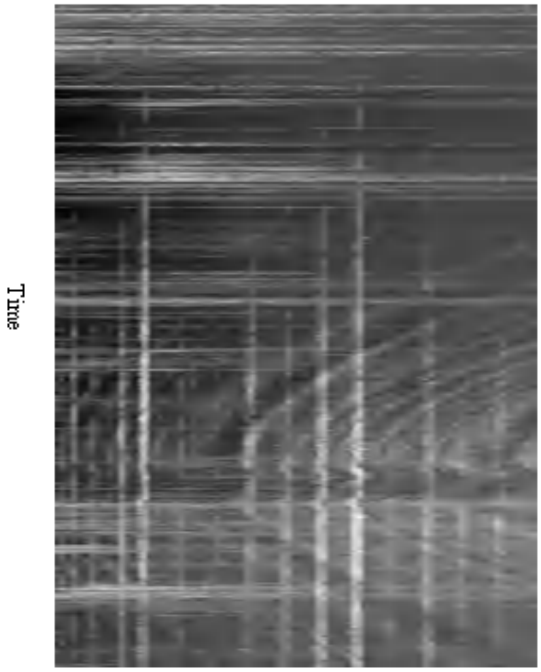

Figure 5. Error-free transmission of CSP (low QF LOFARgram)

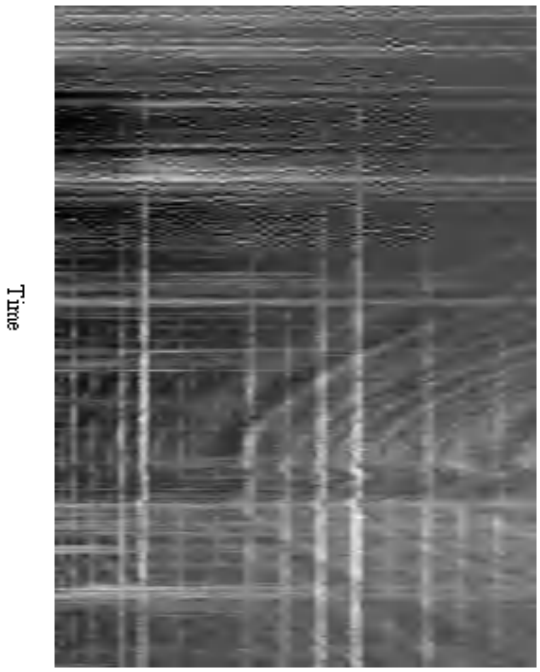

Figure 6. Uncoded transmission of CSP, standard UATS, Gaussian channel, BER = 10^{-4}

The simulated BERs for CSP transmission are very similar to those for random messages, so a similar improvement in data rate can be expected.

The upper curves show the simulated BER curves for random messages and CSPs in a fading channel. Clearly, fading greatly degrades the performance of convolutional coding. This could be mitigated by interleaving and decision-feedback equalization (DFE).

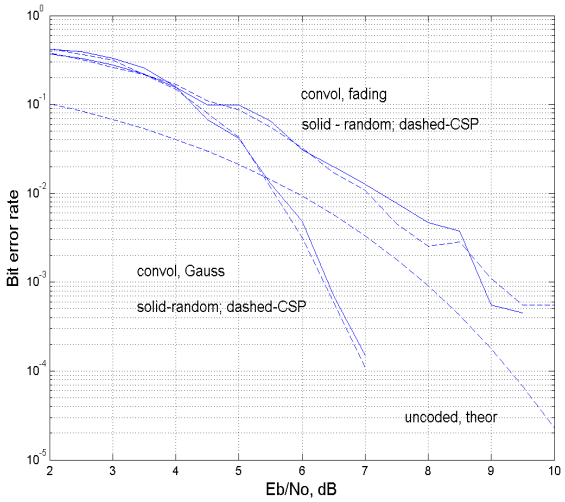

Figure 7. Simulated BERs for Gaussian (lower curves) and fading (upper curves) channel, convolutional coding

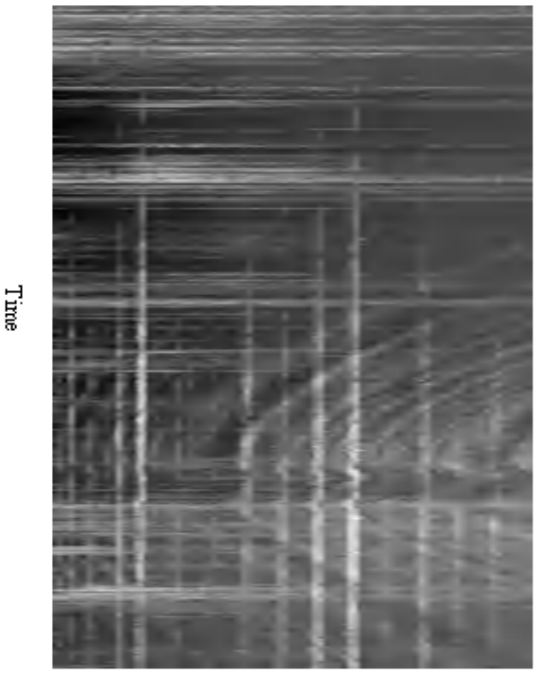

Figure 8. Convolutionally-coded transmission of CSP, standard UATS, Gaussian channel, BER = 10^{-4}

Figure 8 displays low QF LOFARgram data transmitted through the UATS with convolutional coding at BER 10^{-4}. This display is indistinguishable from that of figure 5 (error-free transmission), except for some very minor detail near the top of the image. However for BER 10^{-2}, the received data could not be displayed, as was the case for uncoded transmission – as above we attribute this problem to the corruption of vital header data at the higher BER. We therefore conclude that in the current UATS implementation, data rates available for BER = 10^{-4} (viz. 170 bps), but not those for BER = 10^{-2}, can be achieved for transmitting CSPs in a convolutionally-coded system in a Gaussian channel. We note, however, that transmission of data using convolutional coding in fading channels breaks down at all but the highest values of E_b/N_0, probably due to burst errors that were not handled well by the decoder. Techniques such as interleaving and DFE are required to overcome this problem.

UATS with turbo coding

The upper dashed curve (for $E_b/N_0 > 6$ dB) of figure 9 shows the theoretical BER for the standard UATS. The simulated BER curve for turbo coding of random messages in a Gaussian channel is given in the lower solid curve. From this curve, the coding gain for BER = 10^{-4} is 4 dB. From eq.(8), this will allow an increase of $10^{0.4}$ = 2.5 in data rate to 250 bps, a little less than that estimated in the section on theoretical BER and bit rate. However, the latter estimates are only approximate, and turbo coding still allows a significant improvement in data rate, and is better than that available with conventional (convolutional) FEC. Figure 9 shows that the improvement factor increases as E_b/N_0 increases, to 5 dB (a factor of 3.2) for a BER of 10^{-5}.

The simulated BERs for CSP transmission are very similar to those for random messages, so a similar improvement in data rate can be expected.

The middle curves (for $E_b/N_0 > 5.7$ dB) show the simulated BER curves for random messages and CSPs in a fading channel. The fading degrades the performance of turbo coding, but not by a large factor – typically 1 dB. This could also be mitigated if necessary by interleaving and decision-feedback equalization (DFE).

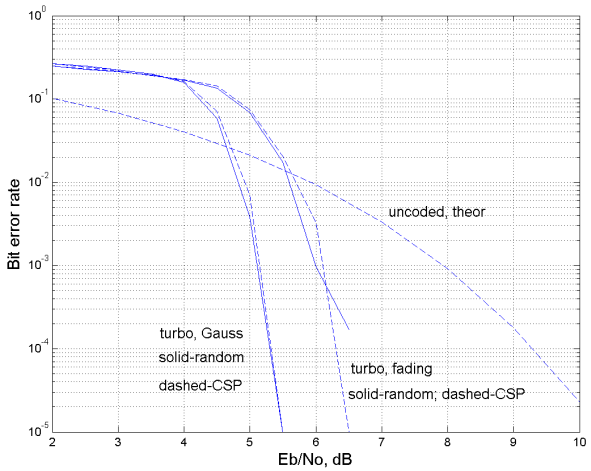

Figure 9. Simulated BERs for Gaussian (lower curves) and fading (upper curves) channel, turbo coding

Figure 10 displays low QF LOFARgram data transmitted through the UATS at a bit error rate of 10^{-4}. (Recall that figure 5 shows the corresponding display for error-free transmission). There is some "noise" in the lower half of the picture; however, the major features are still clearly visible. However, as above for BER 10^{-2}, the received data could not be displayed, and we attribute this problem to the corruption of vital header data at the higher BER. We therefore conclude that in the current UATS implementation, data rates available for BER = 10^{-4} (viz. 250 bps), but not those for BER = 10^{-2}, can be achieved for transmitting CSPs in a turbo-coded system in a Gaussian channel. In the fading channel examined here, coding gains and hence data rates are somewhat less than for a Gaussian channel.

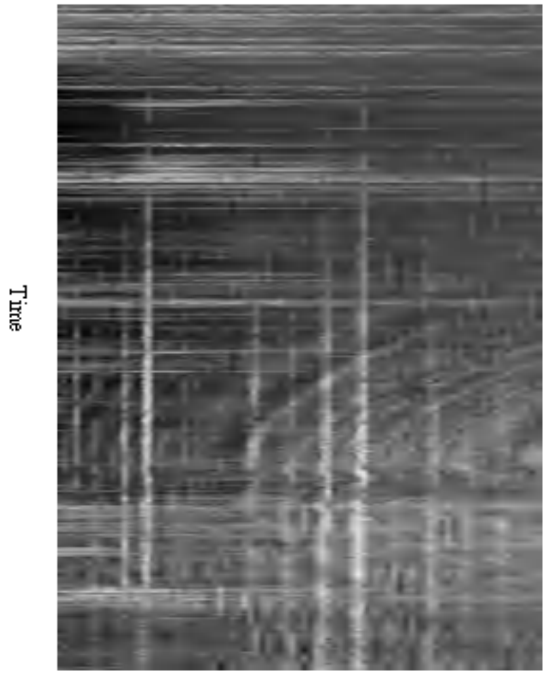

Figure 10. Turbo-coded transmission of CSP, standard UATS, Gaussian channel, BER = 10^{-4}

CONCLUSIONS AND FUTURE WORK

A simulator has been built to study the performance of a UATS in transmitting random messages and CSPs through Gaussian and fading channels. Three channel coding options are available – no coding, convolutional coding (hard and soft decoding) and turbo coding. A theoretical study of the performance of the UATS in a Gaussian channel for the three channel coding options has been reviewed, and extended to the consideration of Rician fading channels. Simulations show that for any coding and channel option, UATS performance in transmitting CSPs is close to that for transmitting random messages. Significant improvements are available in using convolutional coding, and further improvements arise from the use of turbo coding, though they are somewhat less than those estimated from published curves using coherent PSK modulation. Fading channels greatly degrade the performance of the uncoded system and the convolutional system, but the degradation is less for a turbo system. When the transmission of CSP data by any means results in a BER of 10^{-4} or less, satisfactory images are obtained. The quality of images decreases as BER increases beyond 10^{-4}, and for BER 10^{-2} and above, images cannot be transmitted because vital header data is generally corrupted.

It is intended to try interleaving and DFE to improve the robustness of the UATS to fading channels. Other turbo coding configurations, including increasing the interleaver size and using different constituent codes in the branches of the encoder, will be tried to improve its performance. Coherent PSK will be added to the UATS, as this is known to give very good performance with turbo coding. In this case, carrier synchronization will be required. It is also intended to implement stronger coding schemes for the header information of compressed images, so that higher error rates could be tolerated in the main compressed data information, allowing higher data rates.

REFERENCES

Barbulescu, S. A. (2004), *What a wonderful Turbo world*, E-book published on www.people.myoffice.net.au/ ~abarbulescu

Berrou, C., Glavieux, A. and Thitimajshima, P. (1993), "Near Shannon limit error-correcting codes and decoding: turbo codes", *IEEE Proc. Int.Conf. On Communications,* Geneva, Switzerland (ICC '93), pp. 1064-1070

Cato, D. (1995) "Ambient sea noise prediction curves – Australian waters (1995)".

Greenwood, D. and Hanso L. (1994), "Characterization of mobile radio channels", in *Mobile Radio Communications,* ed. R.Steele, Pentech Press, London.

Sweet, D. R. (2005), "The use of turbo coding in underwater telemetry systems", *Sea Technology,* Vol.45, No.5, pp.43-45.

Thorp, W. H. (1965), "Deep-ocean sound attenuation in the sub and low kilocycle per second region", *J. Acoust. Soc. Am.,* vol. 38, p. 648.

Viterbi, A. J. (1971), "Convolutional codes and their performance in communication systems", *IEEE Trans. Comm. Technol.*, vol. COM 19, no.5, pp. 751-772.

Wang, J. and Cox, I. (2003), "Sonar data communications thorough image compression and reconstruction: a potential solution to the data communications problem for network enabled undersea warfare", DSTO Technical Report DSTO-TR-1525.