TRELLIS EXTRACTED SYNCHRONISATION TECHNIQUES APPLICABLE TO AUTOMATIC LINK ESTABLISHMENT

Bernard Hunt
Bahram Honary
Communications Research Centre
University of Lancaster, UK.

Mel Maundrell
Paul Arthur
Defence Research Agency
Malvern, UK.

ABSTRACT

With the ever increasing interest in Automatic Link Establishment (ALE) and adaptive radio systems, there is much research being carried out to try and improve the performance of such systems. As well as looking at completely new systems, there is also interest in improving the performance of systems complying to the US MIL-STD 188-141A [1], as this is the currently established standard in this area.

It has previously been shown how a new trellis structure for the (24,12) Golay code can be used to obtain soft decision maximum likelihood of this code [2], which allows an improvement of a MIL-STD compliant system, without affecting it's compliance.

This paper will address several techniques for making use of this trellis structure to allow maximum likelihood, or near maximum likelihood, block synchronisation within a MIL-STD compliant system.

INTRODUCTION

Using a Generalised Array Code (GAC) format of the (24,12) Golay Code, we are able to generate a trellis for decoding this code [2]. This trellis has a structure of eight parallel sub-trellises shown in Figure 1, with a sub-trellis structure shown in Figure 2.

In addition to performing full soft maximum likelihood decoding of the Golay code using this trellis, we have also shown that it is possible to perform reduced complexity, near maximum likelihood decoding [3]. By performing a simple manipulation on the received data, we are able to grade the sub-trellises of the structure into the order in which they are most likely to contain the correct decoded word. By only testing the best five sub-trellises, we are able to reduce the complexity of the decoder by about 40%, whilst retaining over 90% performance in conditions of 0dB Eb/No (AWGN).

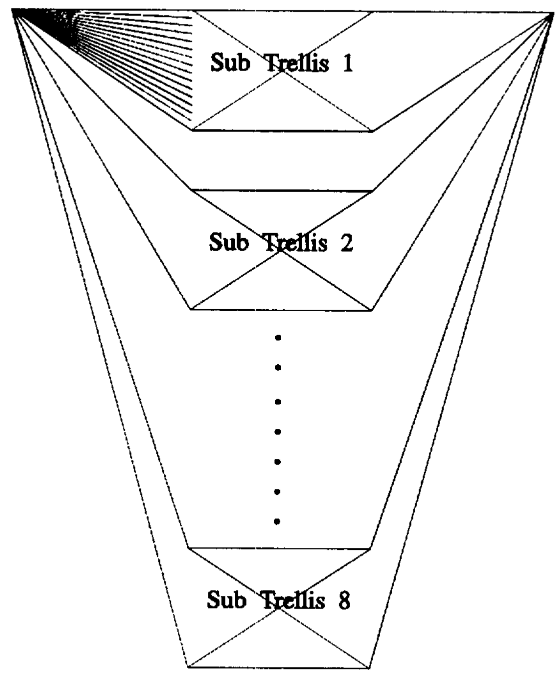

Figure 1. Overall trellis structure for decoding Golay code.

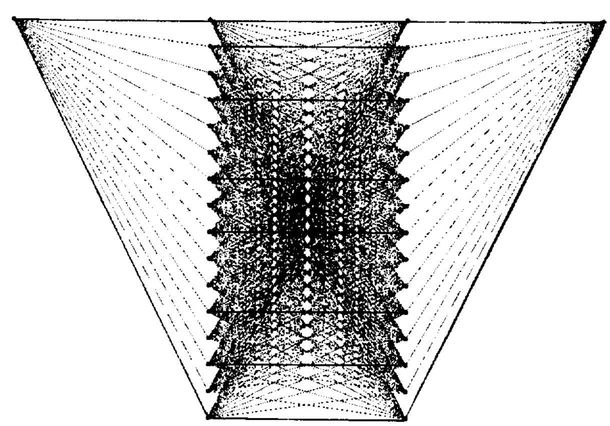

Figure 2. Structure of sub-trellis for decoding Golay code.

This trellis structure ensures that the system always returns to a zero state at codeword boundaries - which fact allows us to carry out maximum likelihood block synchronisation, at the same time as performing soft decision decoding.

TRELLIS EXTRACTED SYNCHRONISATION TECHNIQUES

We have identified four different trellis extracted synchronisation techniques (TEST) [4,5] to achieve block synchronisation of the Golay code. All of these techniques work on the principle of shifting a window across the incoming data stream, and using the resulting trellis metrics to decide on the synchronisation point. Figure 3 shows the operation of such a window over one frame period of the data stream. A 24 bit window is used, as this is the length of the Golay codeword.

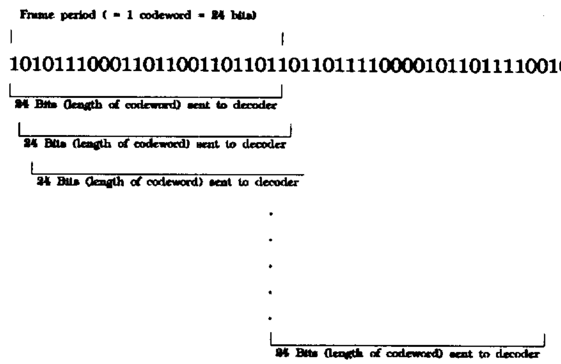

Figure 3. Windowing technique for TEST.

In the first synchronisation technique, the contents of the window are passed through the decoding trellis at every shift position. The results of the decode, and its path metric are recorded in a buffer. After 23 shifts, or 24 tests, we know that we must have been correctly synchronised in one of these positions. By looking for the decoded result with the lowest path metric we can identify the correct synchronisation point, and also the correct decoded data. Figure 4 shows how the value of the path metric varies against shift position under noise free conditions. Figure 5 shows the same technique, with Gaussian noise added to give an Eb/No of 5 db. It can be seen that there are clear synchronisation points generated by the technique, and although these are less obvious to the eye when noise is added, they can still be tracked by the algorithm.

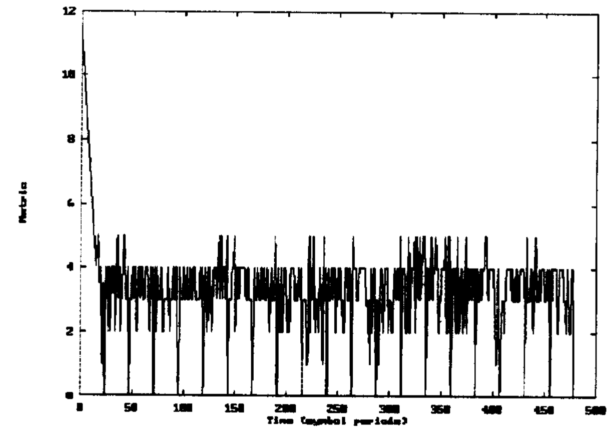

Figure 4. Variation of Path Metric v Time Shift. No noise.

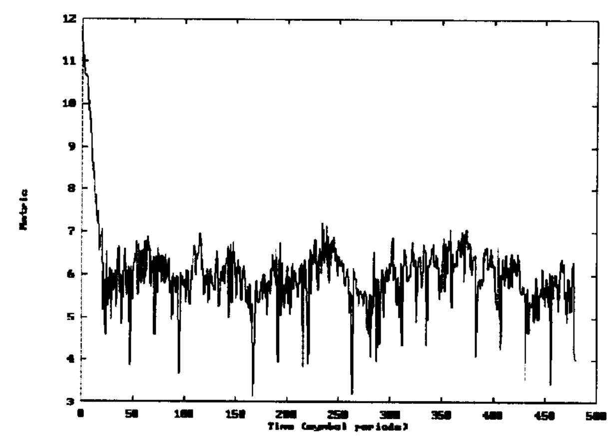

Figure 5. Variation of Path Metric v Time Shift. Eb/No = 5 db.

In order to improve the performance of the system under noise we can add a memory buffer. If we only look for the best synchronisation point within one symbol period, it is possible that noise will corrupt this point, and we will try to synchronise to the wrong point. Instead we add a buffer into which we place the values of the derived synchronisation point. Unless there are severe Doppler effects, we expect the synchronisation point to be the same for every symbol within a continuous transmission - or changing slowly with mild Doppler, or crystal drift etc. When a new synchronisation point is calculated, we compare it to the previous values, and if it is further on in the symbol than the previous point, we increment the value of the previous synchronisation point, and use that. If the newly calculated point is less than previously calculated, we decrement the old value and use that. In this way the system gradually shifts the position of the synchronisation point to follow slippage, without allowing noise to significantly corrupt it. The cost of this technique is a longer period to acquire initial synchronisation at the start of a transmission, although it may be possible to make the tracking speed adaptive to improve this response.

The second variation of TEST investigated again uses a windowing technique on the incoming data stream, but this time a different metric is required from the decoder. Rather than looking at the metric of the best path, we look at the difference between the metrics of the two best paths. At the best synchronisation point, the gap between these will be at its greatest, so can be used to obtain synchronisation. Figure 6 shows the variation in this metric under noise free conditions, and Figure 7 shows the effect of 5 dB noise.

Again, the addition of a memory buffer for the synchronisation points will improve the performance of the system under noise.

This technique is also expected to yield valuable information regarding the noise conditions on the channel, as the presence of noise will cause the gap between the two best paths to be reduced at the synchronisation point.

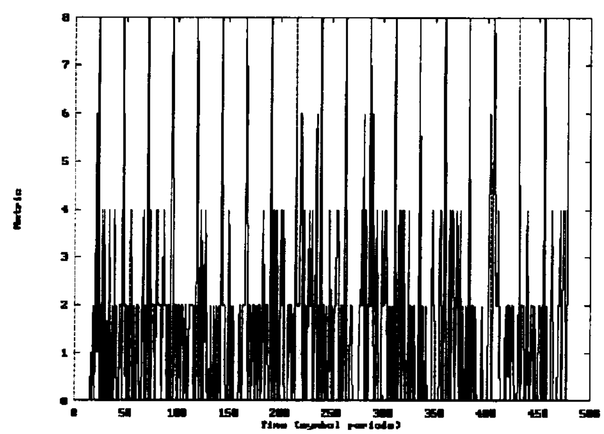

Figure 6. Variation of Best 2 path metric v time shift. No noise.

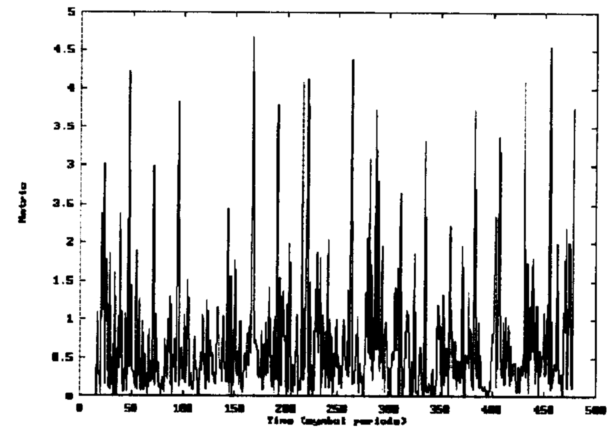

Figure 7. Variation of Best 2 path metric v time shift. Eb/No = 5 dB.

In the third variation of TEST, we again use data related to two paths through the trellis to try and obtain our synchronisation point. In this case, we use the difference between the best path through the trellis, and the worst path. The position of the least likely codeword in Euclidean space is such that it should have virtually no relationship to the correct codeword, and hence its path through the trellis should be much more indicative of the noise present on the channel. Again this should allow us to obtain useful channel condition data as well as synchronisation. Figures 8 and 9 show the variations of this metric under noise free, and Eb/No = 5 dB conditions.

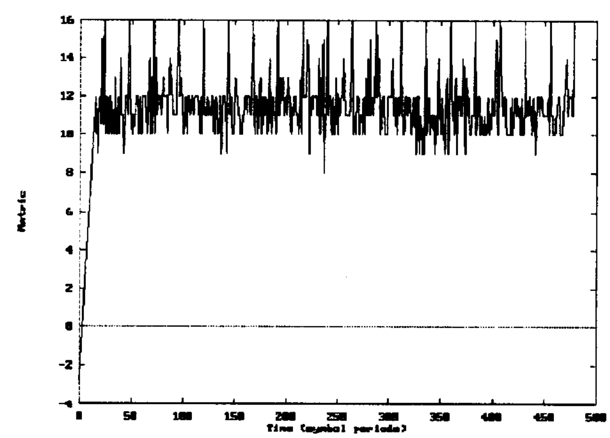

Figure 8. Variation of Best/Worst path metric v time shift. No noise.

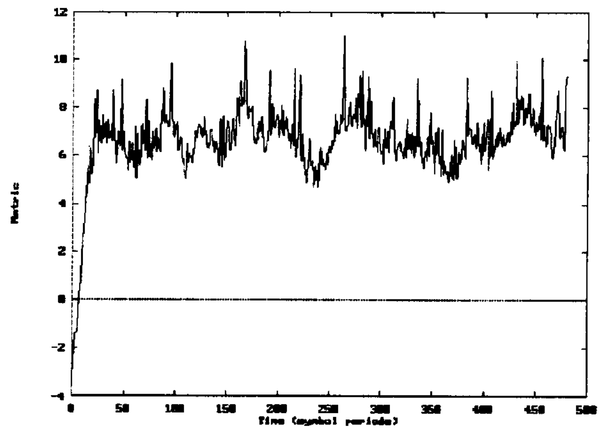

Figure 9. Variation of Best/Worst path metric v time shift. Eb/No = 5 dB.

The final variation of TEST under investigation again uses the difference between the best and worst paths through the decoder, but this time we use the low complexity decoder which only searches through 5 of the 8 possible sub-trellises of the code. Figures 10 and 11 show the performance of this technique. Visual inspection of these results show that they are very similar to those obtained using a full trellis decode. Further investigations by simulation have shown that there are negligible differences between the two techniques over a wide variety of noise levels, so we can effectively discount the use of a full search if we wish to reduce complexity, without affecting the performance of the synchroniser.

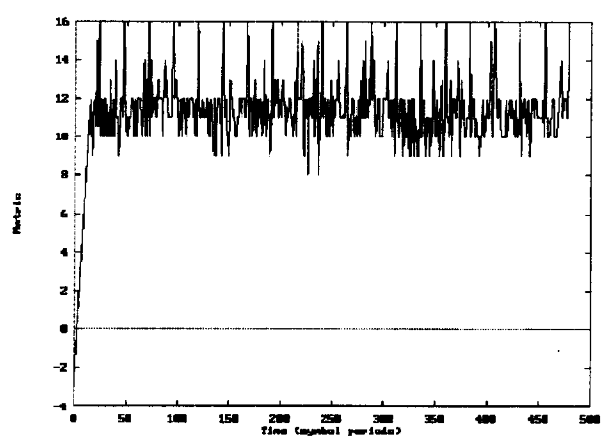

Figure 10. Variation of Best/Worst of 5 path metric v time shift. No noise.

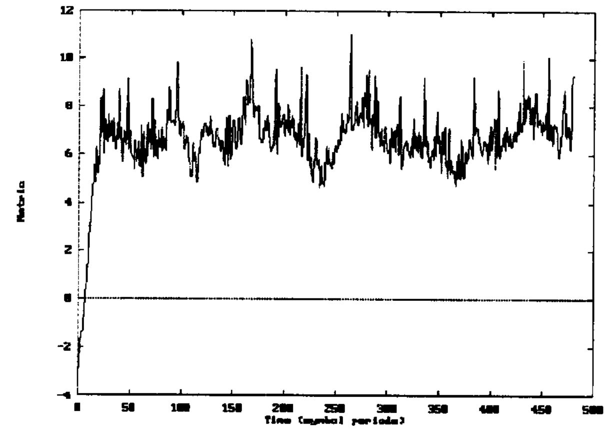

Figure 11. Variation of Best/Worst of 5 path metric v time shift. Eb/No = 5 dB.

To allow a comparison of these synchronisation techniques to be made, a system was modelled with random data source being encoded and passed through a bipodal channel, where Gaussian noise was added. At the receiver side, the synchroniser was used to calculate the position of the start of each codeword within the data window. The value of this position was compared with the true offset, and the error rate of this point was calculated. No memory buffer was used for these tests. Figure 12 shows the results of these simulations, with the curves for all four techniques superimposed.

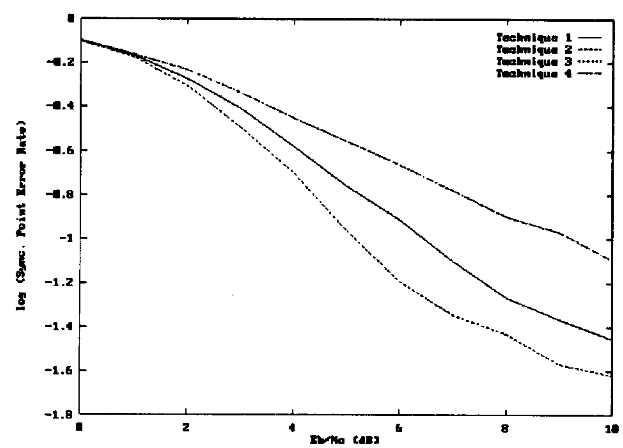

Figure 12. Performance Comparison of Synchronisation Techniques.

CONCLUSIONS

In this paper we have presented several techniques to allow us to obtain maximum, or near maximum, likelihood synchronisation in a system using the (24,12) Golay code. By simulation we have shown that we can obtain the best synchronisation by looking at the difference between the two best path weights through the decoding trellis for this code. It is also hypothesized that this technique will also allow information regarding the channel conditions to be obtained from the received data.

REFERENCES

1. US Military Standard 188-141A "Interoperability and performance standards for medium and high frequency radio equipment".

2. Honary B, Hunt B, Maundrell M, "Improving Automatic Link Establishment through a new soft decision trellis decoder for the (24,12) Golay code", Proc IEE 6th International Conference on HF Radio Systems and Techniques, York, UK. (pp182-185).

3. Street M, Darnell M, Hunt B, Honary B, Arthur P, Maundrell M, "Design Considerations and Performance of an Adaptive HF Systems Architecture", Accepted for IEE 7th International Conference on HF Radio Systems and Techniques, Nottingham, UK.

4. Coulton P, Hannaford C, Honary B, "Coding for Both Protection and Synchronisation", IEE Proc. Comm., Vol 142, No 6, (pp 352-356).

5. Honary B, Darnell M, Markarian G, "Trellis Extracted Synchronisation Technique", UK provisional Patent No 9414275-9.