euronoise

A track-before-detect algorithm for active sonar based on a hidden Markov model

Nigel Parsons

Thales Underwater Systems Ltd., Dolphin House, Ashurst Drive, Bird Hall Lane, Cheadle Heath, SK3 0XB Stockport, UK
nigel.parsons@uk.thalesgroup.com

Abstract – An active sonar track-before-detect algorithm is described. It is based on a hidden Markov model which uses a Viterbi algorithm to estimate the log-likelihood ratio of the presence or absence of a target in tracks within a state space representing a set of ranges, bearings, range rates and bearing rates, assuming a set of transition probabilities of changes in range rate and bearing rate. A detection is declared if the log-likelihood ratio exceeds a certain threshold and subsequently an HMM tracker, operating on a much smaller state space, is then employed. The performance of this algorithm on simulated data is evaluated. It is shown that, for moving and manoeuvring targets, the detection performance is significantly better than that of a conventional algorithm.

1 Introduction

In conventional automatic detection and tracking (ADT), data $y(t,\omega)$, depending on time, t, and observational variables, ω, such as range and bearing, is used to decide whether a target signal is present or not at any time t and any location ω. This process is known as detection or track initiation. After a target has been detected at time t_0, the ADT has to track the target by estimating the location, $\omega(t)$, of the signal at later times $t > t_0$.

Track initiation relies on thresholding and integration. Conventionally, the integration is either performed by evaluating the mean power over some fixed number, N, of consecutive time intervals, before thresholding, or by applying an M out of N criterion after thresholding (or by a mixture of the two). In either case, the false alarm probability depends on the threshold, and on the extent, $\delta\omega$, of the test window in the data domain. Increasing $\delta\omega$ increases tolerance to drift of the signal (or target manoeuvres) with time, but at the expense of increasing the false alarm rate. In practice, $\delta\omega$ is usually given by the sensor resolution, and N, the integration time is determined by how long the target is likely to stay within one resolution cell (or sometimes by the duration of the shortest signal to be detected). Typically, N is less than 10.

In track-before-detect (TBD) processing a model of the dynamics of probable targets is used in order to integrate signal contributions over tentative tracks linking different resolution cells, (i.e. the position of the test window, $\delta\omega$, is allowed to evolve with time). Potentially, this allows the integration time, N, to be increased, thus allowing lower SNR targets to be detected. In most TBD algorithms processing is done on data blocks of fixed duration (i.e. N fixed) [1, 2]. But, in this case we use a new track initiation scheme, the sequential Markov detector (SMD), combining a hidden Markov model (HMM) and sequential detection. At each time step a HMM calculates transition probabilities between different 'hidden' states in a space of states. The HMM allows for the testing of any path $(x(t))_{t_0\le t\le t_0+\Delta t}$ in the state space from an exact expression of the joint likelihood ratio of this path and the data sequence $(y(t,\omega_x(t)))_{t_0\le t\le t_0+\Delta t}$ along the corresponding path $(\omega(t))_{t_0\le t\le t_0+\Delta t}$ in the data domain. In the SMD, the interval Δt is not fixed.

2 Theory

2.1 HMM Detection

The time behaviour of the target is assumed to be a Markov process taking its values from a finite state space $\{x_1 \ldots x_N\}$. The *a priori* probability of any path $X(t_0,\Delta t) = (x(t))_{t_0\le t\le t_0+\Delta t}$ being the path of the target is given by the product of the initial state probability and the transition probabilities between successive states on the path, $x(t\text{-}1)$ and $x(t)$ for $t_0+1\le t\le t_0+\Delta t$. This can be evaluated recursively as

$$P[X(t_0,\Delta t)] = P[x(t_0+\Delta t)\,|\,x(t_0+\Delta t-1)]P[X(t_0,\Delta t-1)] \quad (1)$$

with $P[X(t_0,0)] = P[x(t_0)]$, where the N initial state probabilities $P_n = P[x(t_0) = x_n]$ and N^2 transition probabilities $P_{n,m} = P[x(t) = x_n \,|\, x(t-1) = x_m]$ are assumed parameters of our Markov model.

Assuming that the data series $Y_X(t_0,\Delta t) = (y(t,\omega_x(t)))_{t_0\le t\le t_0+\Delta t}$ can be modelled as an independent random process with probability densities p_0 for background noise and p_1 for a mix of noise and signal, we can represent the probability density of the data at a point on the path under the hypotheses of noise only and noise plus target signal as

$$\begin{aligned} p_0(y(t,\omega_x(t))) &\equiv P[y(t,\omega_x(t))\,|\,H_0], \\ p_1(y(t,\omega_x(t))) &\equiv P[y(t,\omega_x(t))\,|\,x(t)], \end{aligned} \quad (2)$$

respectively. Now the likelihood of the data series $Y_X(t_0,\Delta t)$, contingent upon the path $X(t_0,\Delta t)$ is

$$\begin{aligned} P[Y_X(t_0,\Delta t)\,|\,X(t_0,\Delta t)] = {} & P[y(t_0+\Delta t,\omega_x(t_0+\Delta t))\,|\,X(t_0,\Delta t)] \\ & \times P[Y_X(t_0,\Delta t-1)\,|\,X(t_0,\Delta t)]. \end{aligned} \quad (3)$$

In an HMM the information from the state process about the data at a particular time comes from the state at that time. Therefore, Eq.(3) can be expressed recursively as

$$\begin{aligned} P[X(t_0,\Delta t), Y_X(t_0,\Delta t)] = {} & p_1(y(t_0+\Delta t,\omega_x(t_0+\Delta t))) \\ & \times P[x(t_0+\Delta t)\,|\,x(t_0+\Delta t-1)] \\ & \times P[X(t_0,\Delta t-1), Y_X(t_0,\Delta t-1)]. \end{aligned} \quad (4)$$

Now since $P[Y_X(t_0,\Delta t)\,|\,H_0] = \prod_{t=t_0}^{t=t_0+\Delta t} p_0(y(t,\omega_x(t)))$, and the likelihood ratio, $\Lambda_{X,Y}(t_0,\Delta t)$ of $(X(t_0,\Delta t),Y_X(t_0,\Delta t))$ is obtained by dividing by $p_0(y(t,\omega_x(t)))$ for each point on the path, we have

$$\Lambda_{X,Y}(t_0,0) = \frac{p_1(y(t_0,\omega_x(t_0)))}{p_0(y(t_0,\omega_x(t_0)))} P[x(t_0)],$$
$$\Lambda_{X,Y}(t_0,\Delta t) = \frac{p_1(y(t_0+\Delta t,\omega_x(t_0+\Delta t)))}{p_0(y(t_0+\Delta t,\omega_x(t_0+\Delta t)))} \times P[x(t_0+\Delta t)\,|\,x(t_0+\Delta t-1)]\Lambda_{X,Y}(t_0,\Delta t-1) \quad (5)$$

For each state x_n, we consider the maximum value of $\Lambda_{X,Y}(t_0,\Delta t)$ for all paths $X(t_0,\Delta t)$ ending at x_n:

$$\Lambda(x_n,t_0,\Delta t) = \max\{\Lambda_{X,Y}(t_0,\Delta t)\,|\,x(t_0+\Delta t)=x_n\} \quad (6)$$

which can be computed recursively by means of the Viterbi algorithm:

$$\Lambda(x_n,t_0,0) = \frac{p_1(y(t_0,\omega_n))}{p_0(y(t_0,\omega_n))} P_n,$$
$$\Lambda(x_n,t_0,\Delta t) = \frac{p_1(y(t_0+\Delta t,\omega_n))}{p_0(y(t_0+\Delta t,\omega_n))} \max_{1\le m\le N}\{P_{n,m}.\Lambda(x_m,t_0,\Delta t-1)\}, \quad (7)$$

ω_n being the location in the data domain corresponding to the state x_n.

By comparing $\Lambda(x_n,t_0,\Delta t)$, for $1 \le n \le N$, to a threshold, we perform a detection test which maximises the detection probability for a fixed false alarm rate given by the detection threshold value. This is the case for any signal stating at or before t_0 and ending at or before $t_0 + \Delta t$, but for signals starting and/or ending within this interval, better detection performance would be achieved with a shorter processing time window, i.e. one which is matched to the signal duration. In principle, the best detection performance would be obtained by performing the processing over all possible vales of t_0 and Δt, but in practice a compromise must be found between detection performance and computation cost by choosing $(t_0,\Delta t)$ from a reduced subset of the possible values. A scheme for doing just that is the sequential Markov detector described in the next section.

2.2 Sequential Markov Detector

The number of paths increases exponentially with Δt. This leads to a false alarm rate which varies with Δt (depending on the balance between this exponential increase and the damping effect of the assumed transition probabilities) if $\Lambda(x_n,t_0,\Delta t)$ is compared to a fixed threshold. Thus in order to keep the false alarm rate constant we need to either vary the threshold by a fixed factor for each unit increase in Δt, or vary $\Lambda(x_n,t_0,\Delta t)$ at each time step by a constant factor K.

If the target signal is present only for part of the time interval of the data the likelihood $\Lambda(x_n,t_0,\Delta t)$ should be reset, ideally immediately before the target signal begins, in order to prevent the signal detection from being jeopardised by the noise data before the starting time of the signal. If the reset does not coincide with the start of the signal then at least it should be shortly before to minimise the effect of the noise data. Thus a logical reset process would be to reset $\Lambda(x_n,t_0,\Delta t)$ (i.e. disregard the past data) if, by doing so, the current test value (and consequently future ones because of the recursive computation) are increased.

These principles are applied in the SMD [3]. Its test value $\Lambda_n(t)$ at each time t and each state n in the HMM state space is calculated recursively as

$$\Lambda_n(0) = \frac{p_1(y(0,\omega_n))}{p_0(y(0,\omega_n))} P_n,$$
$$\Lambda_n(t) = \frac{p_1(y(t,\omega_n))}{p_0(y(t,\omega_n))} \max\{K \max_{1\le m\le N}\{P_{n,m}.\Lambda_m(t-1)\}, P_n\}, \quad (8)$$

where K is a constant, chosen so as to make the reset probability approximately 50%.

2.3 Application to Active Sonar Data

We make the simplest choice for $p_0(y)$ and $p_1(y)$, in which y is chosen as the unintegrated energy in a range-bearing cell[1], and

$$p_0(y) = \exp(-y),$$
$$p_1(y) = \frac{1}{1+S_0}\exp\left(-\frac{y}{1+S_0}\right), \quad (9)$$

where it has been assumed that the noise has been normalised to a mean value of 1, and the signal to noise ratio is S_0.

Instead of dealing with the likelihood ratio, it is more convenient to use the log-likelihood ratio. Thus we take the natural logarithm of both sides of Eq.(8), and substitute the forms for $p_0(y)$ and $p_1(y)$ given by Eq.(9). After some algebra it can be shown that Eq.(8) is equivalent to

$$\ln\Lambda_n(0) = y(0,\omega_n) + K;$$
$$\ln\Lambda_n(t) = y(t,\omega_n) + K + \max\left\{\max_{1\le m\le N}\left[\ln P_{n,m} + \ln\Lambda_m(t-1)\right], 0\right\} \quad (10)$$

where K has been redefined. K is regulated to maintain the reset probability close to 0.5.

Unfortunately it is not sufficient just to continue to let the log-likelihood evolve, unchecked, after the threshold has been exceeded. The log-likelihood peak does not remain localised, it spreads out in all directions in the state space. This would lead to additional false detections and prevent the detection of real targets. Upon detection of a contact (real or false) it is necessary to inhibit further detection. Thus, after clustering of closely spaced threshold crossings, the log-likelihood ratio is reset to a background level (e.g. zero). Each cluster is used to initialise a new track in a dedicated HMM tracker operating in a much smaller state space. The data into the SMD is inhibited around the track positions. Fig. 1 gives a schematic overview of the processing chain.

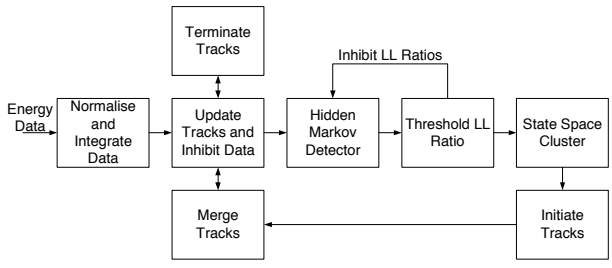

Fig. 1 Track-Before-Detect Processing Chain

[1] The corresponding results for normalised, integrated data can be derived. For rectangular integration by M, the pdf generalises to a chi-squared distribution of 2*M* degrees of freedom The set of recursive equations turns out to be identical to those of the unintegrated case.

3 Simulation

In order to compare the performance of the SMD with a conventional detector, normalised sonar data was simulated. For each ping, independent samples of exponentially distributed background noise of unit mean (and standard deviation) were generated in a 2-D grid of 2500 range cells by 20 bearing cells. This would correspond reasonably well, for example, to an active sonar operating with a range resolution of 10m, dead range 3km, maximum range 28km, and 20 beams. A reasonable choice for the state space is a 4-D space representing range, bearing, range rate and bearing rate. A SMD was initialised with the following available states:

Number or range cells	2500
Number or bearing cells	20
Number of range rates	41
Number of bearing rates	3
Number of hidden Markov states	2500×20×41×3 = 6150000

Table 1 States chosen for SMD processing simulated data

The bearing cells, numbered 1 to 20, are arranged to wrap round cyclically so that the next beam after beam 20 is beam 1. Assuming a PRI of 40sec, the natural range rate resolution for the set of states would be 10m/40s = 0.25m/s, and the natural bearing rate resolution would be 360°/(20beams×40s) = 0.45°/s. Thus this set of states corresponds to a max/min range rate = ±5m/s, max/min bearing rate = ±0.45deg/s.

Next it is necessary to choose a transition matrix for the probabilities of different transitions in range rate and bearing rate. In these experiments we test two different transition matrices. The first is appropriate to a non-manoeuvring target. It is a 1 × 1 matrix equal to 1. This implies that the probability of the target range rate or bearing rate changing by more than ±0.125m/s or ±0.225°/s is zero; i.e. there is a probability of 1 that the target has the same range rate and bearing rate, to within the accuracy of a resolution cell, at the next ping. This is, of course, an extreme choice. The second choice we test is a manoeuvring target model with a uniform probability of 1/9 for the range rate to change by up to ±4 range rate units per PRI, corresponding to a maximum change in range rate of ±1m/s from one ping to the next. We also impose a probability of ¼ each for the bearing rate to change by ±1 bearing rate units per PRI and ½ for no change in bearing rate. Thus 9 × 3 = 27 different transitions in the range rate/bearing rate combination are possible. Range and bearing transitions are also constrained by simple kinematics.

The simulated target was generated as a point target, occupying one range/bearing cell and of constant strength (amplitude) from ping to ping and added to the noise background in that cell. The strength of the target is the actual SNR, as the mean background noise power is 1. The occupied range/bearing cell changes from ping to ping, as appropriate to the target manoeuvre which we are simulating.

4 Results

4.1 Detection Threshold

In both the manoeuvring target model and the non-manoeuvring target model we found that a value of $K_0 = 0$, gave a reset probability sufficiently close to 50% for simulated noise data. In order to estimate the false alarm probability versus the threshold 200 simulated pings of exponentially distributed data were processed by the SMD in order to allow the log-likelihood statistics of the recursive estimate to settle down to a steady state. In the case of the non-manoeuvring target model a further 400 pings were then processed. A number of different thresholds were applied and, for each of them, the average number of threshold crossing per ping was measured. This was measured on every 20th ping, in order to obtain sufficiently decorrelated samples. Each ping has 50,000 range/bearing cells. Therefore the false alarm probability per cell is the mean number of false alarms per ping divided by 50,000. The results are illustrated in Fig. 2.

In the case of the manoeuvring target model a further 80 pings were processed and the number of threshold crossings was measured on every 4th ping, in order to obtain sufficiently decorrelated samples. The results are illustrated in Fig. 3.

The decorrelation interval between measurements can be estimated from the relaxation time of the recursive filter represented by Eq.(10). In the case of the non-manoeuvring target model the transition probability matrix $P_{n,m}$ is equal to zero unless the states n and m have the same range rate and bearing rate, in which case it is equal to one. The kinematics thus constrain the possible transitions to a one-to-one mapping between the old states m and the new states n. We can disregard the occurrence of a reset for a large value of $\ln\Lambda$, and in such a situation Eq.(10) implies a change in $\ln\Lambda$ between two consecutive connected states of $\delta\ln\Lambda(t) = y_n(t) + \ln P_{m,n} + K$. In running the algorithm we find, for the non-manoeuvring model, that K, settles down to a value of about –1.3. The input data, $y_n(t)$, has been generated with a mean value of 1. Thus, on average, $\delta\ln\Lambda(t)$ will be approximately –0.3. Over 20 pings, a high value of $\ln\Lambda(t)$ will, therefore, reduce by about 6. Looking at Fig. 2, we see that this will lead to about 10% of false alarms being redetections of previous higher values, 20 pings earlier. This amount of correlation will slightly reduce the effective number of independent samples, but should not affect the results significantly.

For the manoeuvring target model, the relaxation time is more difficult to estimate. However, the kinematics allow the target to reach 5 different range/bearing cells with equal probability. Thus, assuming a current state which is a local maximum, the biggest cell in the neighbourhood at the next time step will be the one out of the 5 which has the largest value of $y_n(t)$. The mean value of this will be given by

$$\bar{y} = \int_0^\infty \left\{ 5ye^{-y} \left[\int_0^y e^{-x} dx \right]^4 \right\} dy = 2.2833 \qquad (13)$$

using the standard peak picking statistics.

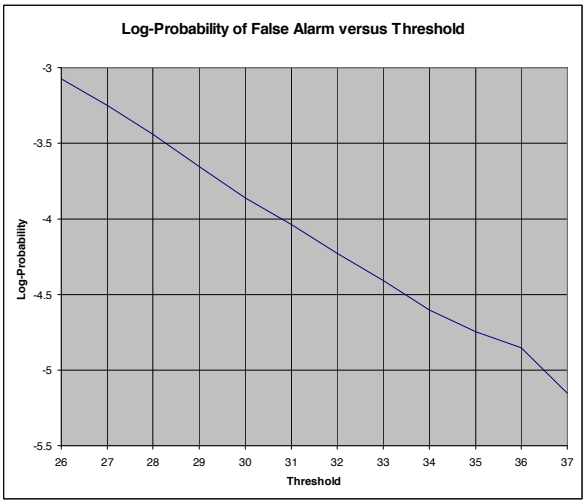

Fig. 2 Log-Probability of False Alarm versus Threshold for Non-Manoeuvring Target Model

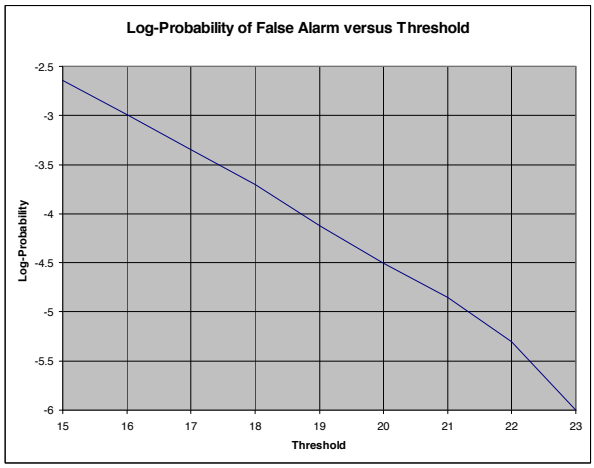

Fig. 3 Log-Probability of False Alarm versus Threshold for Manoeuvring Target Model

The mean and maximum values of $\ln P_{n,m}$ are given by –3.35 and –2.89, respectively and the value of K is found to be around –1.45, on average. Thus, on average, $\delta\ln\Lambda(t)$ will be approximately –2.52. (This allows us to predict that for a target of strength y_s the growth in $\ln\Lambda(t)$ per ping will be approximately y_s – 2.52 and hence a target of SNR less than $10\log_{10}2.52 = 4$ dB will not integrate in this model.) Even with the maximum value of $\ln P_{n,m}$, $\delta\ln\Lambda(t)$ will be –2.06. This higher damping factor in the manoeuvring target model leads to lower threshold values for a particular false alarm rate than in the non-manoeuvring target model. Of course these damping factors do not imply that the mean value of $\ln\Lambda(t)$ continues to fall globally with t, as we have neglected the reset to zero which occurs for about 50% of states. Unfortunately, a high value of $\ln\Lambda(t)$ will produce a large effect on the P_{FA} of the lower threshold values as it decays in subsequent time steps. The peak will decay to 27 lower peaks at the next time step (the transition probability allows transitions to 27 different range/bearing rate states) and each of these will then decay to a further 27 peaks at the next time step, etc., thus producing a large cascade or avalanche effect on the number of false alarms at lower threshold values. Our measurements over 20 pings produce a maximum $\ln\Lambda(t)$ of 23.1, and the average peak value in a ping is 20.5. The lowest threshold value we are testing is 15, therefore, if we measure the crossings every 4th ping, the expected decay of a previous peak is at least $4\times2.06 = 8.24$, which should take the value below 15. This should provide sufficiently decorrelated samples.

Fig. 2 & Fig. 3 indicate that threshold values of 30.7 and 18.7 will achieve a false alarm probability of about 10^{-4} in the non-manoeuvring and manoeuvring models, respectively. For conventional processing, the threshold which must be applied to the raw data to achieve a P_{FA} of 10^{-4} is $-\ln P_{FA} = 9.2$.

4.2 Probability of Detection

Next we introduced a target moving at constant range and bearing rates of 3 range cells and 1 bearing cell per PRI. SNRs of 5, 7 and 9dB (non-fluctuating) were used. In Fig. 4 and Fig. 5 we plot the P_D versus number of pings for the non-manoeuvring and manoeuvring models, respectively. The P_D was estimated using 28 independent runs with the target starting from different ranges. The estimate was made by counting the fraction of these 28 which had exceeded the threshold after a particular number of pings. Thus the measurement of P_D is rather coarsely quantised in steps of about 3.5% in these results.

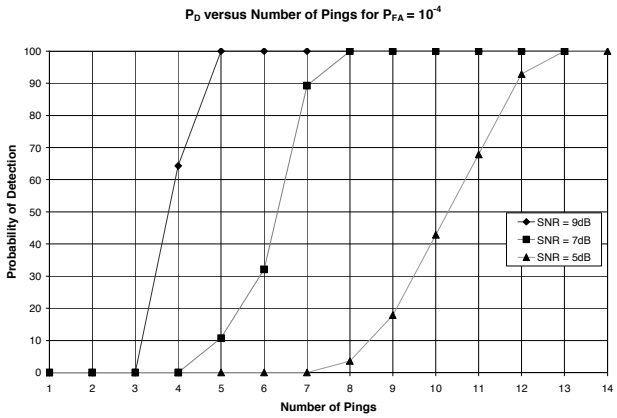

Fig. 4 P_D versus Number of Pings for Non-Manoeuvring Model: SNR = 9, 7 & 5 dB.

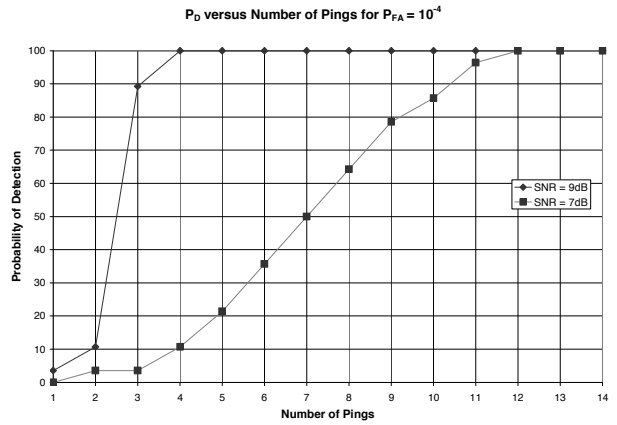

Fig. 5 P_D versus Number of Pings for Manoeuvring Model: SNR = 9 & 7 dB.

In Fig. 5 we have omitted the curve for SNR = 5dB because we found that, over 25 pings, though there was a slight general upward trend in the log-likelihood, the noise

fluctuations were high and the P_D was small and unmeasureable with the 28 samples we were using. Consequently the few observed threshold crossing were not sustained.

The P_D of the conventional detector acting on the raw (normalised) data, on the other hand was calculated, assuming the threshold to be 9.2. For example, given a target of 7dB, the target strength is $10^{7/10} \approx 5$. To obtain a level above 9.2 we need an additional noise contribution of at least 4.2. The probability of this occurring is $\int_{4.2}^{\infty} e^{-x} dx = e^{-4.2} \approx 1.5\%$. This will be the P_D on this and every subsequent ping. The P_D for conventional processing are listed in Table 2.

SNR	5dB	7dB	9dB
P_D	0.24%	1.5%	30.1%

Table 2 P_D for conventional processing

For a 9dB target the P_D is barely sufficient for subsequent tracking and 5 and 7 dB targets would be, to all intents and purposes, undetectable by conventional processing.

It is possible to argue that for N pings (with N>1) the total P_D will be larger than this because there are N chances, so that $P_D(N) = 1 - (1-P_D)^N$. However, this is over optimistic as, for example, if the processing only detects a target every 5 or 6 pings, it is unlikely that the tracker will succeed in tracking it. With the SMD, on the other hand, once a detection has been made, it is usually sustained on subsequent pings, either by the detector itself, or in the implementation illustrated in Fig. 1, by a hidden Markov tracker, 'seeded' by the current log-likelihood ratios above threshold, and continuing to update using unthresholded data and, therefore, continuing to benefit from the ping-to-ping integration inherent in this approach. Fig. 4 shows that, with the non-manoeuvring model, even a 5dB target would be detectable by the SMD, albeit with a latency of 11-12 pings. For the manoeuvring model, Fig. 5 shows that a 7dB target is detectable with a latency between 7 and 11 pings.

5 Conclusions

We have described the implementation of a track-before-detect algorithm and its application to active sonar data. Using simulated data, plots of P_D versus number of pings, for a fixed threshold, demonstrate a much better detection performance for weak targets than can be obtained using conventional processing. For strong targets there is no difference in performance, however, for targets intermediate in strength there are indications that there may be more latency in the track-before-detect algorithm (e.g. a conventional algorithm may detect the target at the first ping, whereas the TBD algorithm may require more pings). This may not be a problem in practice as real targets do not suddenly come into existence from nothing in active sonar. In any case it is always possible to run a conventional algorithm in parallel with a TBD algorithm.

It has been pointed out that, for a fair comparison, the false alarm rate after contact following in the conventional system, should be used for setting the threshold in the case of the SMD, as no additional false alarm reductions are achieved up to this stage in the SMD. This would imply a false alarm probability of about 10^{-7} (i.e. 3 orders of magnitude lower). Assuming that the relationship between $\log P_{FA}$ and threshold remains approximately linear, one can infer from Fig. 2 that this implies an increase in the threshold to about 43 instead of 30.7. On the other hand, one could argue that 2 orders of magnitude of this decrease in P_{FA} are due to single ping shape recognition in the conventional sonar system. This shape recognition could be used with the SMD processing. So perhaps the appropriate P_{FA} for fair comparison should be 10^{-5}. However, a fundamental property of the SMD is that, due to integration, increasing the threshold does not affect the P_D, it just affects the latency of the detections. Bearing all this in mind, it is apparent that it is very difficult to compare the performance of conventional and SMD processing.

As expected, the detection probability was adversely affected by allowing target manoeuvres in the HMM (c.f. Fig. 4 & Fig. 5). This suggests that best overall performance may require a compromise in the manoeuvres which the model can accommodate or better still the use of multiple target manoeuvre models in parallel.

The processing chain illustrated in Fig. 1 has been implemented and tested on real sonar data, with promising results. Future work will concentrate on reducing the P_{FA} by refinements in this processing chain.

References

[1] S. M. Tonissen, R. J. Evans, "Performance of Dynamic Programming Techniques for track-before detect", *IEEE Trans. AES,* 32, 1440-1451, (1996).

[2] R. F. Barrett, D. A. Holdsworth, "Frequency Tracking Using Hidden Markov Models with Amplitude and Phase Information", *IEEE Trans. AES,* 41, 2965-2976, (1993).

[3] D. Billon, "A new detection scheme: the sequential Markov detector", Proceedings of the XIth International Symposium on Applied Stochastic Models and Data Analysis, May 2005, Brest, France.