ON THE IMPLEMENTATION OF A TRANSIENT MODEL FOR AN INTELLIGENT TARGET MOTION ANALYSIS SYSTEM

Lars Nolle

School of Computing and Informatics
Nottingham Trent University
Clifton Lane, Nottingham, NG11 8NS, UK
lars.nolle@ntu.ac.uk

KEYWORDS

Target Motion Analysis, Stochastic modelling, random number generation, probability density function.

ABSTRACT

This paper presents details of an implementation of a computer simulation for the Target Motion Analysis problem where the error of time-delay measurements derived from passively sensed transients is modelled using a Gaussian-mixture probability density function. In order to improve accuracy of the simulation results, a pseudo-random number generator based on shift-register operations was implemented and compared to the standard C++ random number generator. Von Neumann's accepting-rejection method was used to generate the required non-uniform distribution function. The model was validated and its performance evaluated.

INTRODUCTION

In the oceanic context, the aim of Target Motion Analysis (TMA), also known as Contact Localisation and Motion Analysis (CLMA), is to estimate the state, i.e. location, bearing and velocity, of a sound-emitting object (Hassab 1983). These estimates are based on a series of passive measures of both the angle and the distance between an observer (ownship) and the source of sound (target). It is assumed that both the ownship and the target are moving in horizontal plane. A typical TMA application is, for example, a submarine that cannot use active sonar to localise a target, because it would immediately give away its existence and position. Another application would be the monitoring of maritime creatures without interfering with their environment, for example whales or shrimps (Ferguson and Cleary 2001). If an active sonar system were be used, those creatures could be harmed or influenced in their behaviour.

TMA is usually carried out by measuring differences in arrival time of short-duration acoustic emissions of the target using hydrophones that are mounted at some distance onboard an observer platform, e.g. a ship or a submarine. These measurements are corrupted by noise and false readings caused by a small Signal-to-Noise Ratio (SRN) and inhomogeneous pathways. Usually, sequences of measurements are taken and statistical methods are applied to estimate the target's state. For example, for robust state estimation, the residual in range and bearing have to be minimised for a series of measurements. A standard approach would be to apply Least Square (LS) methods. But because of the noisy data and possible outliers in the readings from the hydrophones, LS methods cannot be used directly to minimise the residuals. Often so-called M-estimators are used here, which replace the original error function with a symmetric positive-definite objective function (Carevic 2003). Other methods used are Maximum Likelihood and Bayes estimators.

In order to be able to test Computational Intelligence methods, like Genetic Algorithms (Goldberg 1989) or Artificial Neural Networks (Picton 2000), to the TMA problem and to compare their effectiveness with traditional statistical methods, a simulation of the transient model for the time delays and their measurement errors was developed and implemented in an object oriented way in C++.

TRANSIENT MODEL

This section introduces the geometry used to determine the distance and the bearing of a contact in relation to an observer platform and describes the probability density function used to model the measurement errors for the time delays.

Geometrical Model

Figure 1 shows the geometrical model used for the localization of a sound-emitting target T (Hassab et. al. 1981). Three hydrophones are mounted onboard the observer platform, equally spaced with distance D. The direct paths of the transient signals from target T to the hydrophones are R_1, R, and R_2.

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

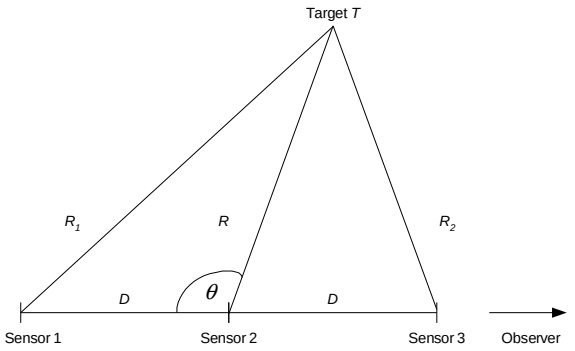

Figure 1 – Geometrical model of the TMA problem.

τ_1 is the difference in propagation time between path R_1 and R (Equation 1) and τ_2 represents the time delay between R and R_2 (Equation 2). The velocity of sound in water c is approximately 1500 m/s.

.

$$\tau_1 = \frac{1}{c}\left(R - \sqrt{R^2 + D^2 - 2RD\cos(\theta)}\right) \quad (1)$$

$$\tau_2 = \frac{1}{c}\left(\sqrt{R^2 + D^2 + 2RD\cos(\theta)} - R\right) \quad (2)$$

Using Equation 1 and 2, the distance R between the target T and the observer platform can be calculated as a function of the time delays, the distance between the hydrophones, and the velocity of sound in water (Equation 3):

$$R = \frac{2D^2 - c^2(\tau_1^2 + \tau_2^2)}{2c(\tau_1 + \tau_2)} \quad (3)$$

The bearing θ can be calculated as follows:

$$\theta = \cos^{-1}\left(\frac{c^2(\tau_2^2 - \tau_1^2) + 2cR(\tau_1 + \tau_2)}{4RD}\right) \quad (4)$$

Modelling of the Noise

In a real application, the time delay measurements are disturbed by noise, caused, for example, by the cross-correlation function used for finding a common signal in a pair of sensors or by the environment (Carevic 2003). Therefore, the measured time delay Δt_i is a linear combination of the real time delay τ_i and a measurement error e_i:

$$\Delta t_i = \tau_i + e_i \quad (5)$$

The time delay error distribution function is a non-Gaussian distribution with long wide tails. Carevic (2003) proposed a probability density function for modelling this error as independent random variables based on a two-component Gaussian-mixture model:

$$p(e_i) = (1-\varepsilon)\mathrm{N}(e_i, \mu_1, \sigma_1) + \varepsilon\mathrm{N}(e_i, \mu_2, \sigma_2) \quad (6)$$

Where

$$N(e, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(e-\mu)^2}{2\sigma^2}} \quad (7)$$

Figure 2 gives a graphical comparison of the standard Gaussian probability density function with $\mu = 0$ and $\sigma = 1$, and the Gaussian-mixture probability density function described above (Equation 6) with $\mu = 0$, $\sigma_1 = 1$, $\sigma_2 = 5$, and $\varepsilon = 0.2$.

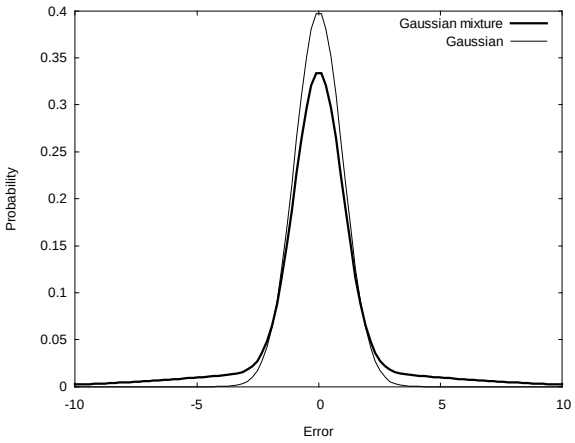

Figure 2 – Comparison between Gaussian and Gaussian mixture probability density function.

It can be seen that the tails are much wider for the Gaussian-mixture distribution, which implies that there is always a certain measurement error. This noise makes it difficult for traditional Least Squares Methods to solve the TMA problem.

In order to obtain realistic results from the simulations, σ_1 in the probability density function (Equation 6) is modelled as a function of the target range R (Carevic 2003). Figure 3 shows the dependency of σ_1 on target range R.

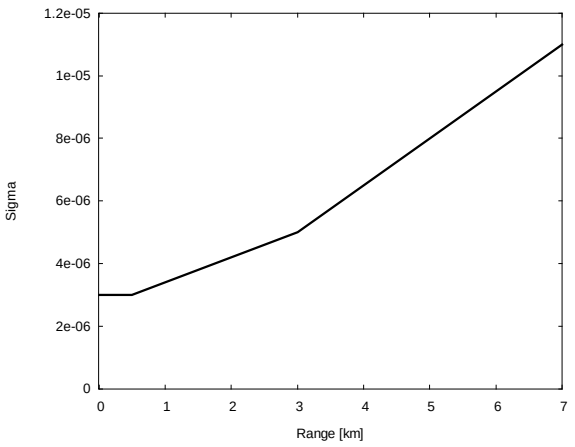

Figure 3 – Dependency of σ_1 on target range R.

Modelling of the Clutter

Another source of errors is false readings or clutter (Carevic 2003). This clutter is usually assumed to be uniformly distributed over an area *A* and to follow a Poisson probability density function with a known parameter λ. (Jauffret and Bar-Shalom 1990).

For the simulation the mean percentage of clutter can be selected in order to be able to reproduce the scenarios reported by (Carevc 2003). Clutter occurs randomly and the probability that it is miss-clssified as target is 30%. If it is labeled as target, it is added to the measurements.

IMPLEMENTATION

The model described in the previous section was implemented in C++. One requirement was not to use any non-standard libraries, as this would make it more difficult to port the model to other computer platforms.

Evaluation of the standard `rand()` function

Because the accuracy of the simulation depends crucially on the accurate modelling of the measurements errors, the standard C++ pseudo-random number generator was evaluated using a Chi-square test (Rubinstein 1981). The hypothesis *H0* was that the random numbers generated by the `rand()` function (Deitel and Deitel 2003) are uniformly distributed. The number of categories was chosen to be 101, which results in a degree of freedom of 100. 10,000,000 samples were generated and the Chi-square value was compared to the critical value for 5%, the conventional level of significance adopted by hypothesis testing.

The Chi-square value was 135.964, which is worse than 124.3, the critical value for 0.05 (Göhler 1996). In fact, it is even worse than the critical value for 0.01, which is 135.8. Therefore, the hypothesis *H0* was rejected. As a consequence, a more sophisticated pseudo-number generator had to be implemented, which passes the Chi-square test.

Uniform pseudo-random number generation

In order to solve the TMA problem using soft computing techniques, pseudo-random numbers with a very long period are needed. A shift-register based method was chosen, because these methods have a very long period (Knut 1997). Shift-register based methods generate 1 bit random numbers r_k according to equation 8. For random integer numbers, the bitwise addition can be carried out in parallel by replacing the bits in (8) with integers and the bitwise addition by the exclusive-or operator.

$$r_k = c_1 r_{k-1} + c_2 r_{k-2} + \ldots + c_p r_{k-p} \qquad (8)$$

The period of this method is 2^{p-1} and hence large values for p are required in order to obtain a long period. Kirkpatrick and Stoll (1981) proposed the R250 algorithm, where most of the constants in (8) are set to zero apart from two so that only two terms remain on the right hand side of the equation (Equation 9).

$$r_k = r_{k-q} + r_{k-p} \qquad (9)$$

The suggested values for p and q are p=250 and q=103. Hence, the R250 algorithm uses a history of 250 previous generated pseudo random numbers to generate a new random number r_k by adding number r_{k-250} bitwise to number r_{k-103}.

In order to avoid the movement of a large amount of data in an array or to compute indices to cells in an array, and hence to achieve a high processing speed, the history of random numbers is stored in a circular single-linked list (Figure 4), which contains 251 nodes to store the pseudo-random numbers. If a new number r_k is generated, only three pointer values have to be adjusted, for the pointers to the current node and the two pointers to the nodes containing the *k*-103 and *k*-250 random numbers (P_k, P_{k-103}, and P_{k-250} in Figure 4).

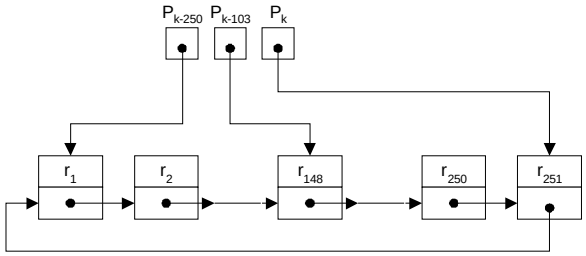

Figure 4 – Circular linked-list for fast pseudo random number generation.

One problem with the R250 algorithm is that it requires an initial set of 250 random numbers before the algorithm can be used. Obviously, these numbers cannot be generated using the R250 algorithm itself and hence it is necessary to initialise them using a second random number generator. But because the use of a poor random number generator can compromise the quality of the R250 algorithm, it was decided to use true-random numbers instead of a second pseudo-random number generator.

The initial 250 random numbers were generated using a hardware true-random number generator based on atmospheric noise (Foley 2001). This sequence of numbers is hard-wired into the code and hence the algorithm needs to be initialised using a seed number. This is achieved by discarding the first n random numbers, where n is the remainder for the system time in seconds divided by 10,000.

Evaluation of the R250 implementation

The R250 implementation was evaluated using the same Chi square test as for the standard `rand()` function. The hypothesis *H0* was that the random numbers generated by the R250 algorithm are uniformly distributed. The number of categories was also chosen to be 101, which results in a degree of freedom of 100. 10,000,000 samples were generated and the Chi-square value was compared to the critical value for 5%. The value for the R250 implementation was 95.8075, well below the critical value of 124.3 and hence Hypothesis *H0* was accepted and therefore the algorithm passed the Chi-square test.

Generating Gaussian-mixture distributions

As stated above, the error probability distribution for the transients is modelled using a Gaussian-mixture function. The pseudo-random numbers generated by the R250 algorithm are uniformly distributed and hence need to be mapped into the Gaussian-mixture distribution. The analytical approach of finding the inverted cumulative distribution function (Devroye 1996) cannot be applied, because the Gaussian function is not explicitly invertible. Therefore, only a numerical approach was used. In this work, von Neuman's acceptance and rejection method (Rubinstein 1981) was applied to generate random numbers from the Gaussian-mixture probability distribution. This method allows the generation of any known random number distribution from a finite interval [min, max]. For every trial, two random numbers are generated, the first one drawn from the range [min, max] determining the non-uniform random number, and the second one represents its probability, which must be below the required probability in order for the non-uniform random number to be accepted. If the probability is greater than the required one, the trial is rejected. As a consequence, the method is expensive in terms of resources, because many trials are rejected for small probabilities and hence the algorithm is less efficient.

One problem associated with acceptance and rejection method is that the interval [min, max] must be defined in a way that all the expected non-uniform random numbers are drawn from this interval. In a normal distribution, this range would be approximately $[-3\sigma, +3\sigma]$, because 99.7% of the values would fall into this interval. For the Gaussian-mixture density distribution used in this work, the cumulative distribution function cannot be found analytically. Therefore, the required range was estimated by numerically integrating the distribution function and by determining when this function converges towards 0.5.

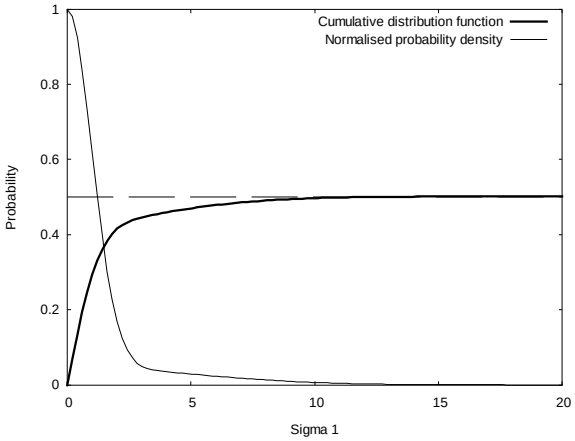

Figure 5 –Estimating the input range for the acceptance and rejection method.

As it can be seen from Figure 5, the cumulative distribution function approaches 0.5 after approximately $10\sigma_1$. Therefore, the range was chosen to be $[-10\sigma_1, +10\sigma_1]$.

MODEL EVALUATION AND VALIDATION

In order to evaluate the efficiency of von Neumann's acceptance and rejection method, 10,000,000 pseudo-random numbers have been generated from the Gaussian-mixture distribution and the average number of rejections was measured. On average, for each valid pseudo-random number that was generated, an average of 9.09 trails was rejected, representing 18.18 pseudo-random numbers that were wasted. This has not only consequences on the efficiency of the simulation, but also on its quality, because it might exhaust the period of the pseudo-random number generator.

To verify the correctness of the probability density distribution generated by the simulation, another Chi-square test was carried out. The *H0* hypothesis was that the pseudo-random numbers generated by the model are from the Gaussian-mixture probability density distribution described (Equation 6). The number of categories was chosen to be 101, resulting in a degree of freedom of 100. For the test, 10,000,000 pseudo-random numbers were generated of which 500,000 were valid ones (see above). The Chi-square value was calculated to be 123.347. This is below the critical value of 124.3 for a confidence level of 5%. Hence the hypothesis *H0* was accepted and the method passed the Chi-square test. Figure 6 shows a graphical representation of the observed and the expected probabilities for the Gaussian-mixture distribution with $\mu = 0$, $\sigma_1 = 1$, $\sigma_2 = 5$, and $\varepsilon = 0.2$.

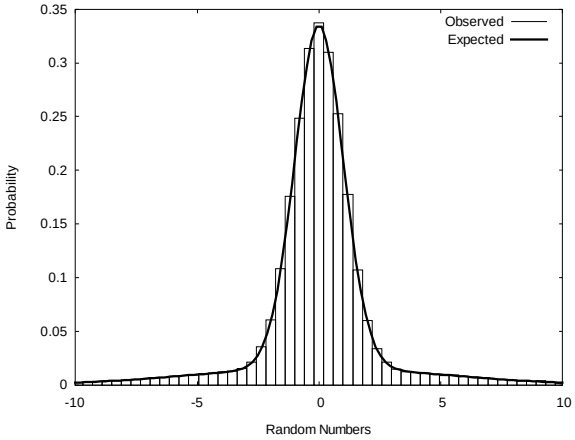

Figure 6 – Observed and expected probabilities for the Gaussian mixture probability distribution.

It can be seen that the observed probability density distribution agrees with the expected Gaussian-mixture distribution and hence the simulation is effective but might lack efficiency.

CONCLUSION AND FURTHER WORK

As it was shown above, a transient model for the Target Motion Analysis problem was successfully implemented in a C++ simulation program. This model incorporates a non-Gaussian model of the error probability density distribution for the time-delays and a model of false readings. The standard pseudo-random number generator could not be used for generating uniformly distributed random numbers due to poor performance. A shift-register based generator was developed instead, which is initialised using a hard-coded sequence of true-random numbers and a seed value, which depends on the system clock. The quality of the random numbers produced by the new generator is superior to the ones generated by the standard generator. The mapping from the uniform distribution to the Gaussian-mixture distribution was achieved using a numerical method that was proven to be effective for the problem at hand.

Figure 7 shows a world-centric representation (Cunningham and Thomas 2005) of the simulation output for a typical test scenario. The white diamonds represent the true positions of the target over time and the black diamonds represent the true observer positions. Figure 8 provides the noisy measurements of the transients for the same scenario.

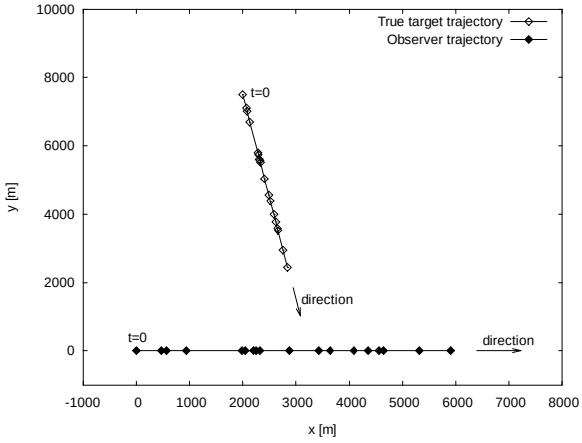

Figure 7 – Typical test scenario.

As it can be seen, the noisy measurements, which are all time stamped, could differ significantly from the real target positions.

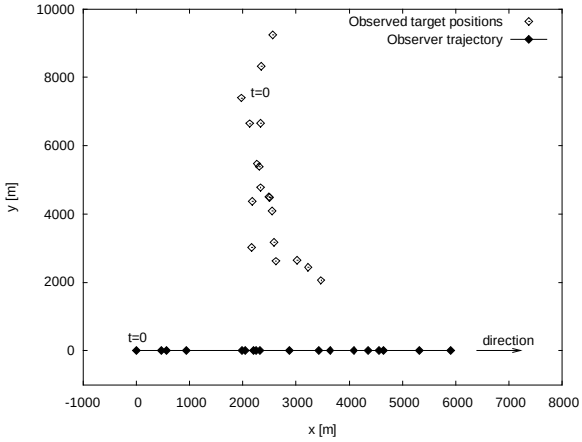

Figure 8 – Simulated transient measures.

In Figure 9 the crosses represent false readings, i.e. clutter. As it can be seen from the figure, the problem of estimating the state of the target based on the noisy measurements of the transient signals is very complex.

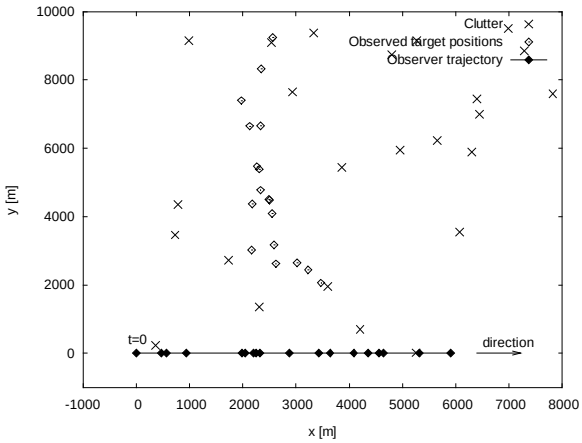

Figure 9 – Noisy measurements and clutter.

The next step in this research will be to use it to generate the scenarios described by Carevic (2003). This data will then be used to test and to compare computational optimisation methods for the TMA problem. The first method that will be tested on the original error function as state estimator directly will be a Genetic Algorithm (Goldberg 1989). A second approach will be to use Artificial Neural Networks (Picton 2000) to predict the target state based on previously trained examples. Both approaches will be evaluated, based on experimental results, and compared with standard TMA methods.

However, if the amount of random numbers necessary for these soft computing techniques exceeds the period of the random number generator, it might be necessary to revisit the mapping from a uniform distribution to the Gaussian-mixture distribution. One approach to reducing the number of wasted random numbers is to use an Artificial Neural Network to learn the inverted cumulated probability function. This should be possible, since Hornik et.al. (1989) have proven that standard multilayer feedforward networks with one hidden layer using arbitrary squashing functions are capable of approximating any measurable function from one finite dimensional space to another to any desired degree of accuracy and even to approximate their derivatives (Hornik 1991).

AUTHOR BIOGRAPHY

Lars Nolle graduated from the University of Applied Science and Arts in Hanover in 1995 with a degree in Computer Science and Electronics. After receiving his PhD in Applied Computational Intelligence from The Open University, he worked as a System Engineer for EDS. He returned to The Open University as a Research Fellow in 2000. He joined the Nottingham Trent University as a Senior Lecturer in Computing in February 2002. His research interests include: applied computational intelligence, distributed systems, expert systems, optimisation and control of technical processes.

REFERENCES

Carevic, D. (2003): Robust Estimation Techniques for Target Motion Analysis Using Passively Sensed Transient Signals, IEEE Journal of Oceanic Engineering, Vol. 28, No. 2, April, pp 262-270

Cunningham, A., Thomas, B. (2005): Target Motion Analysis Visualisation, Proceedings of the Asia-Pacific Symposium on Informatin Visualisation, Sydney, Australia, 27-29 January, pp 81-90

Deitel, H. M., Deitel, P. J. (2003): C++ How to program, 4th Edition, Prentice Hall

Devroye, L. (1996): Random Variate Generation in One Line of Code, Proceedings of the 28th Winter Simulation Conference, Coronado, USA, 8-11 December, pp 256-272

Ferguson, B. G., Cleary, J. L. (2001): In situ source level and source position estimates of biological transient signals produced by snapping shrimp in an underwater environment, Journal of the Acoustical Society of America, Vol. 106, No. 6, June, pp 3031-3037

Foley, L. (2001): Analysis of an On-line Random Number Generator, Project Report, Trinity College Dublin, Ireland

Göhler, W. (1996): Formelsammlung Höhere Mathematik, Verlag Harry Deutsch, Frankfurt am Main

Goldberg, D. E. (1989): Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley

Hassab, J. C., Guimond, B.W., Nardone, S. C. (1981): Estimation of location and motion parameters of moving source observed from a linear array, Journal of the Acoustical Society of America, Vol. 70, No. 4, October, pp 1054-1061

Hassab, J. C. (1983): Contact Localization and Motion Analysis in the Ocean Environment: A Perspective, IEEE Journal of Oceanic Engineering, Vol. 8, No. 3, July, pp 136-147

Hornik, K., Stinchcombe, M., White, H. (1989): Multilayer Feedforward Networks are Universal Approximators, Neural Networks, Vol. 2, pp 359-366

Hornik, K. (1991): Approximation Capabilities of Multilayer Feedforward Networks", Neural Networks, Vol. 4, pp 251-257

Jauffret, C., Bar-Shalom, Y. (1990): Track Formation With Bearing and Frequency Measurements in Clutter, IEEE Transactions on Aerospace and Electronic Systems, Vol. 26, No. 6, November, pp 999-1010

Kirkpatrick, S., Stoll, E. (1981): A Very Fast Shift-Register Sequence Random Number Generator, Journal of Computational Physics, Vol. 40, pp 517-526

Knuth, D. E. (1997): The art of computer programming, volume 2 (3rd ed.): seminumerical algorithms, Addison-Wesley Longman Publishing Co., Inc., Boston, MA

Picton, P. D. (2000): Neural Networks, 2nd Edition, Palgrave, London

Rubinstein R. Y. (1981): Simulation and the Monte Carlo Method. John Wiley & Sons, New York