ARTIFICIAL NEURAL NETWORK EMBEDDED KALMAN FILTER BEARING ONLY PASSIVE TARGET TRACKING

A. SURENDRA RAO Scientist – E
Naval Science and Technological Laboratory, Visakhapatnam - 530 027, India
Fax : 0091-0891-559464; E-mail : nstl.vizag@rmy.sprintrpg.ems.vsnl.net.in

ABSTRACT

Target tracking is an important issue in underwater surveillance systems. The tracking systems in Sea Warfare utilize Passive sonar to have bearing only information contaminated with noise, which is assumed here as additive zero mean Gaussian noise. In underwater warfare two dimensional target motion analysis is familiar.

The Kalman Filter (KF) is used to obtain the target parameters with the help of bearing data coming from sensor. The error in target parameters of velocity, range, heading and bearing are estimated. For some of the scenarios the errors are unacceptable to real time combat systems.

Hence alternative methods are surveyed and Artificial Neural Network (ANN) is coupled with Kalman filter to reduce the creeping errors in the solution in spite of Kalman adaptive filters exist. The network selected for this purpose is Backpropagation neural network. The network is pre-trained using different inputs to predict the said target parameters. The simulation results are presented and comparative studies are conducted. The ANN provides the adaptive capability the filter model needs.

1. INTRODUCTION

Passive target tracking will be defined as the ability to reconstruct, in some operational time frame, the location of an underwater target in both space and time using bearing history [2]. The knowledge of complete System State is a requirement in the implementation of optimum control strategy. Varieties of techniques to estimate the target state are Least Square Estimate, Maximum Likelihood Estimate, Maximum Apostoriori Probability and Kalman Filter derivatives. Filter is an estimate of signal (data). A process is called observable if from the measured data of the output, it is possible to determine the state of the system [1]. The Kalman filter has a recursive form suited to digital computer. Initially the co-variance matrix is singular (state unobservability) causing Eigen values of Fisher information matrix / covariance matrix tends to zero.

Many tracking systems employing Kalman filters were having trouble with longer bias errors or filter divergence. The basic requirement for the target motion analysis is the system observability i.e. the existence of unique tracking solution [3]. Even though system model is linear and the observer equation is non-linear as is well known, utilization of KF requires explicit mathematical models for both measurement process and system dynamics. In operation of the filter, current estimates of state and co-variance are of particular interest since they statistically characterize system state vector. However for the purpose of analysis, it is more convenient to work directly with initial estimates, which do not vary explicitly with time. In most of the practical tracking applications accurate initialization data are usually unavailable, thus an examination of filter under such circumstances is of interest. The demands of high precision tracker system for future strategic systems, require accurate target state.

Although KF yields satisfactory results, the erratic behavior of KF is seen in literature. Premature collapse of co-variance matrix has been observed even under favorable conditions [6]. This anomaly detrimental to the filter performance leads to filter divergence. The Artificial Neural Network has become handy in these situations whereby coupling 'ANN' model to KF, the errors in the state variables can be minimized in most of the scenarios. The network is pre-trained using different inputs i.e. Kalman gain, difference between predicted and estimated state vectors and predicted state vector. The errors between the simulated and estimated values are compared with and without the aid of neural network. The results are shown.

2. SYSTEM MODEL AND MEASUREMENT MODEL

2.1 System Model

$X(k/k) = A(k/k-1)*X(k/k-1) + B(k-1)U(k-1)$; A and B are constant matrices. –(1)

$X(k) = [\, r_X \;\; r_Y \;\; v_X \;\; v_Y \,]$ State Vector, A/ ϕ : Transition matrix. –(2)

r_x, r_y : Relative Range Components, v_x, v_Y : Relative Velocity Components

$$A(k/k-1) = \begin{vmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}, \quad B(k) = \begin{vmatrix} T/2 & 0 \\ 0 & T/2 \\ 1 & 0 \\ 0 & 1 \end{vmatrix} \qquad -(3)$$

T : sample interval,

U : Dynamic Vector,

2.2 Measurement Model

$Z(k) = H(k)*X(k/k)+n(k)$ -(4)

$H(k) = \{ \cos[b_m(k)], \; -\sin[b_m(k)], \; 0, \; 0 \}$ -(5)

Bearing $b(k) = \arctan\{r_x(k)/r_y(k)\} = b_m(k) + v(k)$ (noise) -(6)

b_m = Measured Bearing, k = Sample Number, n = Noise vector.

3. KALMAN ALGORITHM & LIMITATIONS

3.1 Kalman Algorithm

$$X(k+1/k) = \phi(k+1/k) * X - \begin{vmatrix} 0010 \\ 0001 \end{vmatrix}^{T} * \Delta V_0(k); \quad -(7)$$

ΔV_0 : Observer velocity change, P : Covariance matrix (Positive semi-definite)

$$P(k+1/k)=\phi(k+1/k)*P(k/k) * \phi^{T}(k+1/k); \quad -(8)$$

$$H(k+1) = [\ \cos\beta(k+1) \quad -\sin\beta(k+1) \quad 0 \quad 0\ ] \quad -(9)$$

$$K(k+1)=P(k+1)*H^{T}(k+1)*[H(k+1)*P(k+1)*H^{T}(k+1)+1]^{-1}; \quad -(10)$$

K : Kalman gain, Superscript 'T' refers to transpose ;

$$X(k+1/k+1)=X(k+1/k)-K(k+1)*H(k+1)*X(k+1/k) \quad -(11)$$

$$P(k+1/k+1)=[I - K(k+1)* H(k+1)] * P(k+1); \quad -(12)$$

3.2 Target Parameters v_{Tx}, v_{Ty} are target velocities, R_x, R_y are Ranges.

$$R(k+1/k+1)=[r_X^{2}(k+1/k+1)+r_Y^{2}(k+1/k+1)]^{1/2}; \quad \text{Range} \quad -(13)$$

$$C(k+1/k+1)=\tan^{-1}[v_{Tx}^{2}(k+1/k+1)/v_{Ty}^{2}(k+1/k+1)]; \ \text{Course} \quad -(14)$$

$$S(k+1/k+1)=[v_{Tx}^{2}(k+1/k+1)+v_{Ty}^{2}(k+1/k+1)]^{1/2}; \ \text{Speed} \quad -(15)$$

$$\beta(k+1/k+1)=\tan^{-1}[R_x(k+1/k+1)/R_y(k+1/k+1)]; \ \text{Bearing} \quad -(16)$$

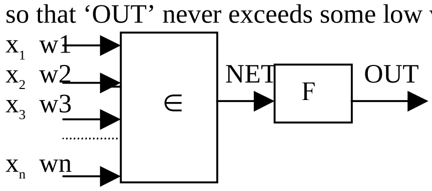
3.3 Limitations: Very succinctly, a Kalman Filter. KF is a recursive, linear, optimal, real-time sequential data processing algorithm used to estimate states of a dynamic system in a noisy environment. In many cases measurements occur only at discrete points in time so that for most practical applications the KF operates on a discrete set of observations to estimate state vector at the same time points. It is assumed that inputs, outputs and system dynamics are driven by additive Gaussian noise. Correct modeling of the system's random noise is crucial to the success of any KF design. If the model is linear then a co-variance analysis is all that is required in order to analyze filter performance. However, if the model is non-linear then an extended KF results, whose performance must be evaluated through a series of Monte-Carlo runs. The tuning of KF plays a critical role in the KF design. There exist KF derivatives to account for these limitations to some extent.

The performance criteria of interest are 1. Convergence speed 2. Hardware complexity 3. Quality of estimates. Convergence time of an algorithm, in the case of non-linear problem depends on true state vector, the modeling of noise present in signal and the initialization of state and covariance matrices. The recursive algorithms are faster than the batch / block algorithms apart from hardware complexity are less in recursive one. The two primary statistical measures employed for quality of estimates are estimator mean and the variance. These measures may lead to accurate / unbiased and precise estimates. An unbiased estimator is one whose expected value is identical to the parameter being estimated. Still there exists filter misbehavior for some scenarios.

4. AN OVER VIEW OF ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks are biologically inspired, i.e. they are composed of elements that perform in a manner that is analogous to the most elementary functions of the biological neuron. ANN exhibits a surprising number of the brain's characteristics. For example they learn from experience, generalize from previous examples to new ones and extract essential characteristics. ANNs are capable of abstracting the essence of a set of inputs. A network can be trained on a sequence of distorted versions of letter A. After adequate training, network will recognize a deformed letter [8]. In one sense, it has learnt to identify something that it has never seen before.

Neuron is the fundamental building block of the nervous system. A biological neuron can be modeled mathematically as an artificial neuron. Each neuron is given a set of inputs. The inputs are multiplied with corresponding weights and the sum of the products is output NET. The NET signal is processed by an activation function 'F' to produce neuron's output signal "OUT". The activation function is chosen to be sigmoid logistic function. This is an 'S' shaped function whose values range from 0 to 1. The function compresses the range of NET, so that 'OUT' never exceeds some low value.



$$NET=x_1w1+x_2w2+\text{---}+x_nwn \quad -(17)$$

$$OUT=F(NET)=1/[1+EXP(-NET)] \quad -(18)$$

Training is the ability of network to learn from experience. Training is accomplished by sequentially applying input vectors, while adjusting network weights according to a predetermined procedure. During the process, the weights gradually converge to values such that each input vector produces the desired output. Training algorithms are categorized as supervised and unsupervised. Supervised training requires the pairing of each input vector with a target vector representing the desired output where as in unsupervised training no target vector is required. Back Propagation (BP) is a supervised training algorithm. The activation function usually used for back propagation is sigmoidal (squashing) function. This network requires a function that is every where differentiable. It has the additional advantage of providing a form of automatic gain control. A multilayer network is suitable for BP training. Before starting the training process, all of the weights must be initialized to small random numbers [5].

5. IMPLEMENTATION OF ANN WITH KALMAN FILTER

To demonstrate the improvement in error reduction, a single / multi hidden layer(s) can be used. Here the BP is a multi-layered feed forward neural network. Assume different geometries of target initial position and generate training plans over a number of samples and store these values in a file. Randomly select some samples from each geometry and put them in another file, using which, the network is trained until the average error converges to less than 0.01.

$$\text{Error} = \in_s \in_k (X_k - OUT_k)^2, \ \in \text{ symbolizes summation} \qquad \text{-(19)}$$

X_k / OUT_k is desired /actual output for node k, s varies from 1 to number of samples.

The range, velocity and gain values are all normalized to zero to one in order not to overflow/underflow before feeding them to nodes. The outputs of the network are range and velocity along X and Y directions using which the percentage errors of target parameters are obtained after the de-normalization of the data. In this paper it is proposed to implement a Back Propagation neural network learning mechanism into the standard Kalman Filter. Through simulation, a comparison was made to show the performance difference of KF with and without ANN incorporation. The activate function F squashes NET to produce the OUT value for each neuron in that layer. BP trains hidden layer(s) by propagating the output error back through the network layer by layer adjusting weights at each layer. Each hidden layer calculates the weighted sum of the error derivatives to find its contribution to the output layer. The convergence criteria chosen, is mean square error. The momentum coefficient / factor has to fixed depending upon the quantity of error generated while training the network.

6. RESULTS AND OBSERVATIONS

The Monte Carlo simulations provide the statistical characterization of the filter, necessary to optimize design. For the Cartesian bearing only filters, it is necessary to tune state/covariance to achieve optimum results [7]. The complete statistical picture of errors from these Monte-Carlo trials is presented in the graphs. Some typical geometries are assumed and the corresponding results are shown. The results of simulation runs, on a number of scenarios have demonstrated good convergence and robustness to large errors in the initial state / covariance guess. Filter lock up and divergence have not been observed. This approach allows for easier manipulation of filter equation and ANN algorithms and provides insight into bearing-only tracking problem.

Number of iterations vs. training error shows the effect of training rate coefficient on training. The effect of number of nodes Vs training error in a single / multi hidden layer shows the impact of number of nodes in hidden layer. With complex structure of ANN and more number of samples, this may not be the case. The sample time is 10 sec. FIG 1.1 and FIG 1.2 show two chosen typical scenarios of observer and target. FIG 2.1 and FIG 2.2 give the percentage errors of range, bearing, heading and velocity of target using only KF and FIG 3.1 and FIG 3.2 show the same percentage errors in case of ANN coupled with KF correspondingly. The convergence time and quantity of error are less in the later case when compared with former case. This adaptive filter is stable and modest.

The network is trained to give the accurate output even if there is a slight change in the input data. Here the input mainly depends upon the dynamics of observer –target geometries. The network is pre-trained with observer maneuver and target straight-line motion. This adaptive filter produces output accurately even if there is a sudden target maneuver.

7. SUMMARY AND FUTURE SCOPE

One problem that has been addressed in this paper is that of convergence for both KF and Back Propagation. Practically speaking, however of the several hundred training sessions conducted for this research, only some of these cases are favorable to convergence. The BP network employs a trial and error method for training. It has to be trained for a longer time to converge which depends on training rate coefficient, number of hidden layers and number of nodes in hidden layers. In all the tests conducted, this adaptive algorithm behaved in a completely predicted manner. In many cases it is desirable to provide each neuron with a trainable bias. This offsets the origin of logistic function thereby permitting more rapid convergence of the training process. The drawback of training process of BP is time uncertainty causing non-optimal step size.

Most of the effort in any practical KF software design will be directed towards the implementation of the real-time version of the filter coupled with ANN. This adaptive filter designer has to take into account the usual real-time, on-line constraints involving computer precision / memory / speed together with data drop outs, failure of sub-systems and operator interaction with the filter. Many software designs do not progress beyond the off-line simulation stage due to significant effort required in successfully adapting a given design to the real-word environment [4]. Techniques for imbedding fault detection procedures and adaptive procedures for on-line 're-tuning' of the filter are essential for filter software. This study can be extended to multi-target and multi-sensor scenarios. This study is spin off for maneuvering target dynamics cases. The Hopfield network can also be wedded to the Kalman filter in order to improve the simulation results further.

8. REFERENCES

1. Lindgre, A.G. and K.F. Gong (July ‘78). ``Position and Velocity estimation via bearing observations,” IEEE Trans. AES – 14, pp. 564 – 577.
2. Nardon, C., A.G. Lindgren, and K.F. Gong (Sep ‘84). ``Fundamental properties and performance of conventional bearing only target motion analysis,” IEEE Trans. AES – 29, pp. 775 – 787.
3. Vincent J. Aidala (Jan ‘79). ``Kalman Filter Behavior in bearing only tracking applications,” IEEE Trans. AES – 15, pp. 29-39.
4. Patel. S.B. ``Real Time Passive Tracking for Tactical Applications,” J. N. S, Vol. – 8, No: 3. pp. 192 – 201.
5. Leonard CHIN (Jan ‘94). ``Application of Neural Networks in Target Tracking data fusion,” IEEE Trans. AES – 30, pp. 281 – 286.
6. Gholson. N and R.L. Moose (May ‘77). ``Maneuvering target tracking using adaptive state estimation,” IEEE Trans. AES – 13, pp. 310 – 317.
7. Walter Grossman (May – June 94). ``Bearing – only Tracking A hybrid Co-ordinate system approach,” JGCD, VOL: 17 No.3, pp. 451 – 458.
8. Surendra Rao. A., Babu Thomas, and U. K. Singh. ``Identification of hand written text using BP and ART networks,” CSI –95, Conference in India.

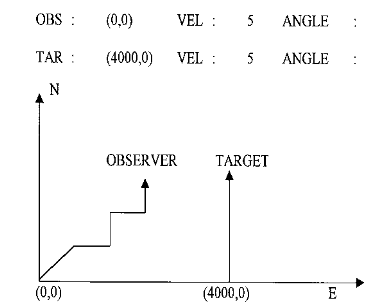


Fig 1.1

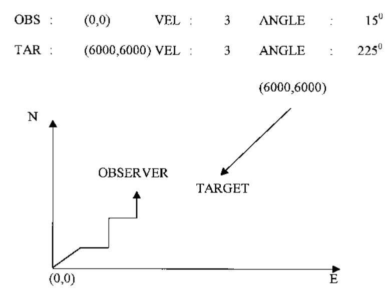
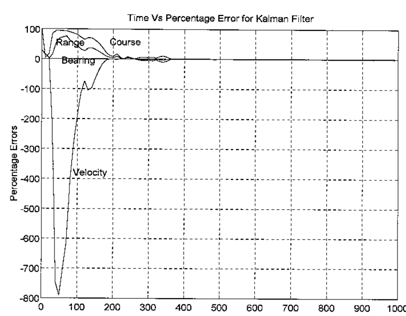
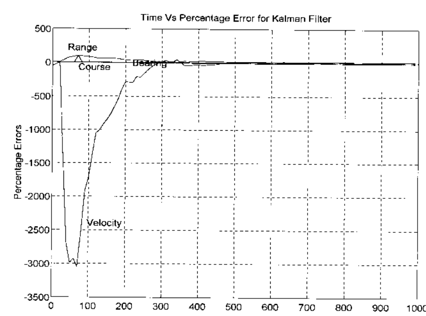


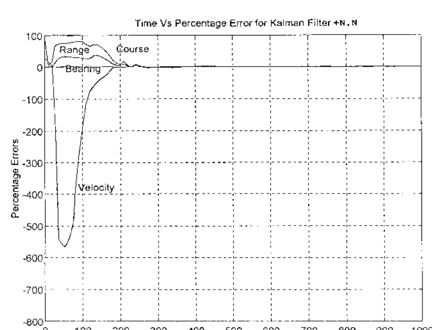
Fig 1.2



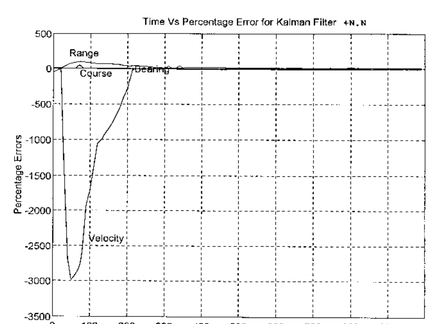
Time (Sec)
Fig 2.1



Time (Sec)
Fig 2.2



Time (Sec)
Fig 3.1



Time (Sec)
Fig 3.2