

Under Water Active Target Tracking Using Kalman Filter

D.V.A.N Ravi Kumar,
Asst. Professor,
E.C.E Department,
GVP College of Engineering
For Women, Vizag.

Dr.S.KoteswaraRao,
Principal, Senior professor,
E.C.E Department ,
vignan institute of engineering
for Women,Vizag.

Dr.K.Padma Raju
Principal, Professor,
E.C.E Department,
University College of Engg.
JNTUK, Kakinada.

Abstract- Tracking is the process of finding the future position of the target. If this is done with the help of the available range and bearing measurements (from a SONAR) then the tracking is called the active tracking. The problem with the technique is, the received measurements are contaminated by some noise. Based upon these measurements if we try to hit the target, definitely we are going to miss and since there will be no second chance in the war environment, it is essential to remove the noise present in the measurements. Here an attempt is made to solve the mentioned problem using kalman filter and the authors were successful. The implementation was done in MATLAB 7.8.0(R2009a) environment and simulation results are presented.

Keywords- Tracking, Range, Bearing, SONAR, Kalman Filter.

I. INTRODUCTION

In military and war field tracking enemy object is very important for blasting it. Tracking is done in 2 ways, they are active tracking and passive tracking. In active tracking we transmit a signal from our own ship, These signals hit the enemies object and some portion of signal is reflected back. We collect these echo signals to identify the location of enemy's target. In passive tracking without sending any signals, we collect some acoustic signals generated from enemy ship like mechanical vibrations or sound signals etc. From these signals we find the target location. But these signals are corrupted with noise. If we use directly these noisy data for tracking, we get misleading results and can't hit the enemy target at all. So before tracking the enemy target, we must minimize the noise to the maximum extent possible because we can't remove noise completely from the measurements.

For minimising noise we use different adaptive algorithms like LMS, WLMS and RLS. An **adaptive filter** is a filter that self-adjusts its transfer function according to an optimization algorithm driven by an error signal.

Least mean squares (LMS) algorithms are a class of adaptive filters used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal (difference between the desired and the actual signal). It is a stochastic gradient descent method in

that the filter is only adapted based on the error at the current time.

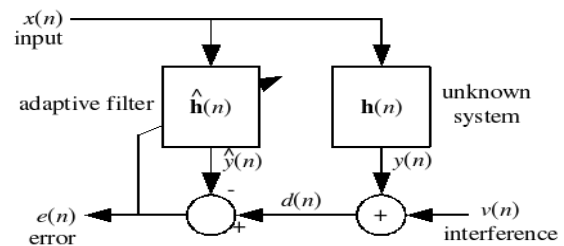


Fig 1 Block diagram of the least mean squares filter

In least squares (LS) estimation, the unknown values of the parameters, β_0, β_1, \dots in the regression function, $f(\vec{x}; \vec{\beta})$, are estimated by finding numerical values for the parameters that minimize the sum of the squared deviations between the observed responses and the functional portion of the model. Mathematically, the least (sum of) squares criterion that is minimized to obtain the parameter estimates is

$$Q = \sum_{i=1}^n [y_i - f(\vec{x}_i; \vec{\beta})]^2 \quad (1)$$

As previously noted, β_0, β_1, \dots are treated as the variables in the optimization and the predictor variable values, x_1, x_2, \dots are treated as coefficients. To emphasize the fact that the estimates of the parameter values are not the same as the true values of the parameters, the estimates are denoted by $\hat{\beta}_0, \hat{\beta}_1, \dots$. For linear models, the least squares minimization is usually done analytically using calculus. For nonlinear models, on the other hand, the minimization must almost always be done using iterative numerical algorithms.

Advantages

Linear least squares regression has earned its place as the primary tool for process modelling because of its effectiveness and completeness.

Disadvantages

The main disadvantages of linear least squares are limitations in the shapes that linear models can assume over long ranges, possibly poor extrapolation properties, and sensitivity to outliers.

In **weighted least squares** parameter estimation, as in regular least squares, the unknown values of the parameters, β_0, β_1, \dots , in the regression function are estimated by finding the numerical values for the parameter estimates that minimize the sum of the squared deviations between the observed responses and the functional portion of the model. Unlike least squares, however, each term in the weighted least squares criterion includes an additional weight, w_i , that determines how much each observation in the data set influences the final parameter estimates. The weighted least squares criterion that is minimized to obtain the parameter estimates is

$$Q = \sum_{i=1}^n w_i [y_i - f(\vec{x}_i; \vec{\beta})]^2 \quad (2)$$

Advantages

Like all of the least squares methods discussed so far, weighted least squares is an efficient method that makes good use of small data sets. It also shares the ability to provide different types of easily interpretable statistical intervals for estimation, prediction, calibration and optimization. In addition, as discussed above, the main advantage that weighted least squares enjoy over other methods is the ability to handle regression situations in which the data points are of varying quality.

Disadvantages

The biggest disadvantage of weighted least squares, which many people are not aware of, is probably the fact that the theory behind this method is based on the assumption that the weights are known exactly. This is almost never the case in real applications, of course, so estimated weights must be used instead. The effect of using estimated weights is difficult to assess, but experience indicates that small variations in the weights due to estimation do not often affect a regression analysis or its interpretation. However, when the weights are estimated from small numbers of replicated observations, the results of an analysis can be very badly and unpredictably affected. This is especially likely to be the case when the weights for extreme values of the predictor or explanatory variables are estimated using only a few observations.

The **Recursive least squares (RLS)** adaptive filter is an algorithm which recursively finds the filter coefficients that minimize a weighted linear

least squares cost function relating to the input signals. This is in contrast to other algorithms such as the least mean squares (LMS) that aim to reduce the mean square error. In the derivation of the RLS, the input signals are considered deterministic, while for the LMS and similar algorithm they are considered stochastic. Compared to most of its competitors, the RLS exhibits extremely fast convergence. However, this benefit comes at the cost of high computational complexity.

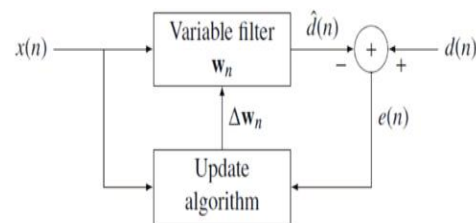


Fig 2 Block diagram of RLS

Advantages:

Unlike the LMS algorithm and its derivatives, the RLS algorithm directly considers the values of previous error estimations. RLS algorithms are known for excellent performance when working in time varying environments.

Disadvantages

These advantages come with the cost of an increased computational complexity and some stability problems.

The problems associated with the RLS can be solved with the advanced stochastic algorithm called the kalman filter. This algorithm is used in this paper to track the target when the sonar is operating in the active mode.

Section II reviews the kalman filter theory, section III deals with the mathematical modelling, section IV presents the simulation and results and finally the paper is concluded in the section v.

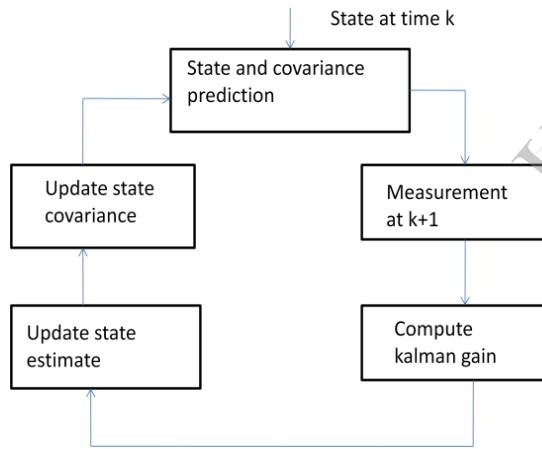
II. KALMAN FILTER

The **Kalman filter** is an algorithm which operates recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The Kalman filter is a very powerful tool when it comes to controlling noisy systems. The basic idea of a Kalman filter is: Noisy data in => hopefully less noisy data out.

PROCESS INVOLVED IN KALMAN FILTER

Sensors are used to make measurements of the system's state (position and velocity, in the case of a vehicle), but the measurements are always corrupted with some amount of error, including random noise. The Kalman Filter algorithm is an optimized method for determining the best estimation of the vehicle's state. The basic concept is similar to simple mathematical curve-fitting of data points using a least-squares approximation and enables predictions of the state into future time steps. The most basic concepts of the filter are related to interpolation and extrapolation. The algorithm works in a two-step process: in the prediction step, the Kalman filter produces estimates of the true unknown values, along with their uncertainties. Once the outcome of the next measurement is observed, these estimates are updated using a weighted average, with more weight being given to estimates with higher certainty. This method produces estimates that tend to be closer to the true unknown values than those that would be based on a single measurement alone or the model predictions alone.

BLOCK DIAGRAM OF KALMAN FILTER



Because the certainty of the measurements is often difficult to measure precisely, it is common to discuss the filter's behavior in terms of gain. The Kalman gain is a function of the relative certainty of the measurements and current state estimate, and can be "tuned" to achieve particular performance. With a high gain, the filter place more weight on the measurements, and thus follows them more closely. With a low gain, the filter follows the model predictions more closely, smoothing out noise but decreasing the responsiveness. At the extremes, a gain of one causes the filter to ignore the state estimate entirely, while a gain of zero causes the measurements to be ignored.

KALMAN FILTER EQUATIONS

state equation:

$$X(n) = \phi(n-1)X(n-1) + W(n)$$

observation equation:

$$Z(n) = H(n)X(n) + V(n)$$

Initialization:

$$\hat{X}(0|0) = E\{X(0)\}$$

$$P(0|0) = E\{x(0)x^T(0)\}$$

Computation: for $n=1,2,\dots$

$$\hat{X}(n|n-1) = \phi(n-1)\hat{X}(n-1|n-1)$$

$$P(n|n-1) = \phi(n-1)P(n-1|n-1)\phi^T(n-1) + Qw(n)$$

$$k(n) = P(n|n-1)H^T(n)[H(n)P(n|n-1)H^T(n) + Qv(n)]^{-1}$$

$$\hat{X}(n|n) = \hat{X}(n|n-1) + K(n)[Z(n) - H(n)\hat{X}(n|n-1)]$$

$$P(n|n) = [I - k(n)H(n)]P(n|n-1)$$

NOMENCLATURE

$\hat{X}(k|k-1)$: Estimate of State vector at time 'k' without considering the measurement at time 'k'.

$\hat{X}(k|k)$: Estimate of State vector at time 'k' considering the measurement at time 'k'.

ts : Time sample , $t(k+1) - t(k)$.

$\phi(k|k-1)$: Transition Matrix.

$P(k|k-1)$: Covariance matrix of $\hat{X}(k|k-1)$.

$P(k|k)$: Covariance matrix of $\hat{X}(k|k)$.

$H(k)$: Measurement matrix.

III. MATHEMATICAL MODELLING

In order to make the kalman filter remove the noise present in the measurements it is essential to represent the situation in the form of the two equations one of them being the State equation while the other is the Measurement Equation.

Before doing the job mentioned above the state vector should be chosen. For the present situation the state vector is considered as follows

$$X(k) = [\dot{x}_t(k) \quad \dot{y}_t(k) \quad R_x(k) \quad R_y(k)]^T \quad (3)$$

Where $\dot{x}_t(k)$, $\dot{y}_t(k)$ represent the velocity components of the target in the x and y directions respectively. $R_x(k)$, $R_y(k)$ Represent the range components in the x and y directions respectively.

The reason for choosing the above as the state vector is if the $\dot{x}_t(k)$ and $\dot{y}_t(k)$ are known the course and the velocity can be determined and the knowledge of $R_x(k)$ and $R_y(k)$ will help in determining the range and the bearing. once the range ,bearing ,course and the velocity are found the future position of the target can be determined and hence the tracking is said to be done successfully.

State Equation: It is the equation that gives the relationship between the present state and the next state. for the state vector chosen as in (3) the state transition matrix will be as follows

The transition matrix is given by

$$\phi(k+1|k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ t_s & 0 & 1 & 0 \\ 0 & t_s & 0 & 1 \end{bmatrix} \quad (4)$$

It is assumed that the target is moving at fixed course with constant velocity, and we can write that

$$\dot{x}_t(k+1) = \dot{x}_t(k)$$

$$\dot{y}_t(k+1) = \dot{y}_t(k)$$

$$R_x(k+1) = R_x(k) + t_s \dot{x}_t - [x_0(k+1) - x_0(k)]$$

$$R_y(k+1) = R_y(k) + t_s \dot{y}_t - [y_0(k+1) - y_0(k)] \quad (5)$$

Where $x_0(k)$ and $y_0(k)$ are position components of own ship. Arranging the above eqns. in matrix form

$$\begin{bmatrix} \dot{x}_t(k+1) \\ \dot{y}_t(k+1) \\ R_x(k+1) \\ R_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ t_s & 0 & 1 & 0 \\ 0 & t_s & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_t(k) \\ \dot{y}_t(k) \\ R_x(k) \\ R_y(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -[x_0(k+1) - x_0(k)] \\ -[y_0(k+1) - y_0(k)] \end{bmatrix} \quad (6)$$

The above relation can be written as

$$x(k+1|k) = \phi(k+1|k)x(k|k) + b(k+1) \quad (7)$$

Where $b(k+1)=[0 \ 0 \ -\{x_0(k+1) - x_0(k)\} \ -\{y_0(k+1) - y_0(k)\}]^T$

Equation (7) represents the state equation for our active tracking process.

Measurement Equation: It is the equation that gives the relation between the true state and the measured state.

Here the measurement equation is as follows

$$Z(k)=H(k)x(k)+\delta \quad (8)$$

$Z(k)$ is the measurement vector=

$$\begin{bmatrix} R_m(k) \sin B_m(k) \\ R_m(k) \cos B_m(k) \end{bmatrix} \quad (9)$$

The reason for choosing $z(k)$ as given in (9) is since we are performing the active tracking process the measurements available to us are the noisy range and the bearing.

It is assumed that the noise in the bearing measurement and the noise in the range measurement are uncorrelated. $\sigma_B^2(k+1)$ and $\sigma_R^2(k+1)$ are the variances of the noises at time $t(k+1)$ in the bearing and range measurements respectively.

For the $z(k)$ taken as above the $H(k)$ will be in the form of

$$H(k) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

δ in (8) is the measurement noise which is assumed to have a mean of zero and it is a function of $\eta(k)$, $\gamma(k)$ which represent the noise in the range and the bearing measurements as given in (11)

$$R_m(k) = R(k) + \eta(k)$$

$$B_m(k) = B(k) + \gamma(k) \quad (11)$$

IV. SIMULATION AND RESULTS

When the algorithm is working in the real environment the range and the bearing inputs are received from the sonar. but since we are testing the algorithm in the pc environment so it is essential to generate the noisy range and bearing measurements for a particular scenario, This is done with the help of a simulator.

ASSUMPTIONS

Following are the assumptions made in the simulator.

1. Initially the own ship is at origin thereafter it moves with a constant course and velocity.
2. Target is moving at constant velocity and course.

- All angles are considered with respect to Truth North.

OWN SHIP MOTION

The own ship motion is introduced as follows consider the figure shown below

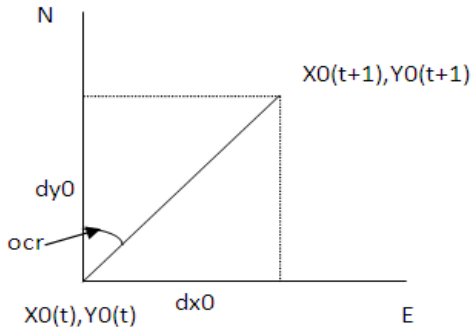


Figure 1 :Plot for computing own ship positions

Slant height in the figure is equal to v_0 .

$(X_0(t), Y_0(t))$ is the position of observer at time 't'.

$(X_0(t+1), Y_0(t+1))$ is the Position of observer at time 't+1'.

dx_0 is the component of the distance covered by the vehicle in 1 sec in East direction.

dy_0 is the component of the distance covered by the vehicle in 1 sec in North direction.

Ocr is the ownship course.

v_0 is the velocity of the ownship.

Then

$$X_0(t+1) = X_0(t) + dx_0$$

$$Y_0(t+1) = Y_0(t) + dy_0$$

Where

$$dx_0 = v_0 * \sin(ocr)$$

$$dy_0 = v_0 * \cos(ocr)$$

CALECULATION OF THE INITIAL POSITION OF THE TARGET

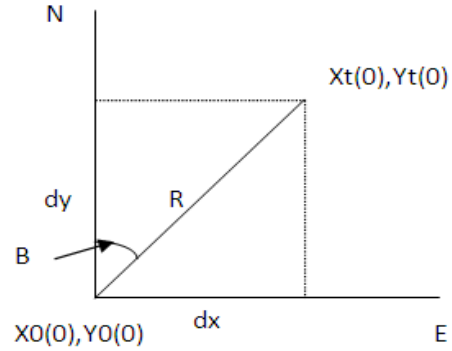


Figure 2: Plot for computing Initial position of the target.

$(X_0(0), Y_0(0))$ is the initial position of the ownship.

$(X_t(0), Y_t(0))$ is the initial position of the Target.

R is the Range.

B is the Bearing.

dx is the Range component in the East direction.

dy is the Range component in the North direction.

Then

$$X_t(0) = X_0(0) + dx$$

$$Y_t(0) = Y_0(0) + dy$$

Where

$$dx = R * \sin(B)$$

$$dy = R * \cos(B)$$

TARGET MOTION

The target motion is introduced as allows.

Consider the figure shown below.

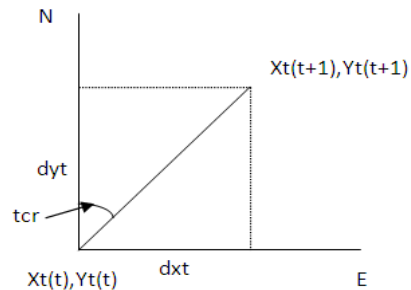


Figure 3: Plot for computing Target positions

Slant height in the figure is equal to v_t .

$(X_t(t), Y_t(t))$ is the position of the target at time 't'.

$(X_t(t+1), Y_t(t+1))$ is the position of the target at time 't+1'.

d_{xt} is the component of the distance covered by the target in 1 sec in the East direction.

d_{yt} is the component of the distance covered by the target in 1 sec in the North direction.

t_{cr} is the Target Course.

v_t is the velocity of the Target.

Then

$$X_t(t+1) = X_t(t) + d_{xt}$$

$$Y_t(t+1) = Y_t(t) + d_{yt}$$

Where

$$d_{xt} = v_t \cdot \sin(t_{cr})$$

$$d_{yt} = v_t \cdot \cos(t_{cr})$$

SCENARIO

The target and own ship are moving with constant velocity and course. We already know the initial positions of the target and ownship. Target and own ship motions parameters are given below

- Initial position of the ownship is origin.
- Initial range between own ship and Target = 10000 KM
- Initial bearing between own ship and Target = 0DEG
- Constant velocity of the own ship = 10 mps
- Constant course of the own ship = 45DEG
- Constant velocity of the target = 15 mps
- Constant course of the target = 100DEG

OUTPUT PLOTS

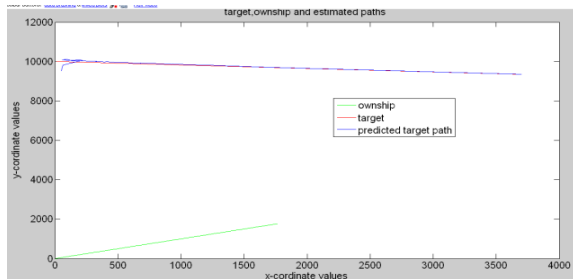


Figure 4: Target Estimated and True Paths

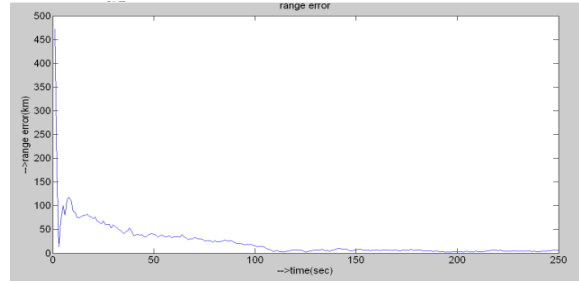


Figure 5: Range Error Vs Time

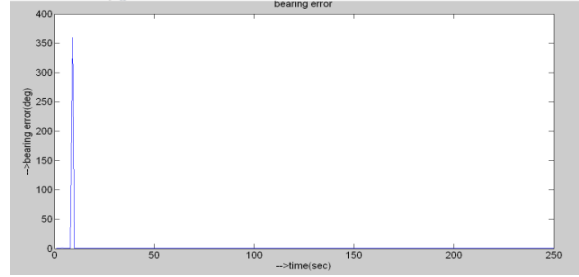


Figure 6: Bearing Error Vs Time

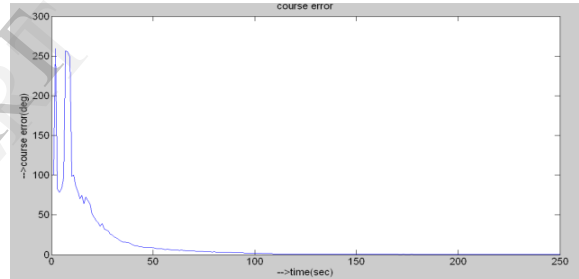


Figure 7: Course Error Vs Time

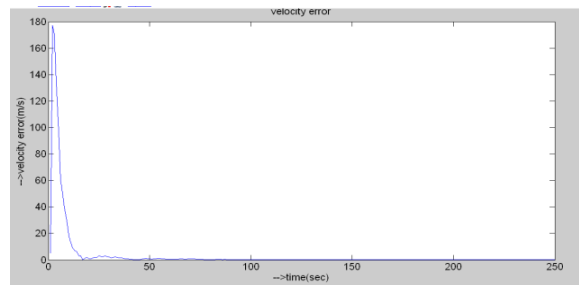


Figure 8: Velocity Error Vs Time

V. CONCLUSION

The results show that the error in the range, bearing, course and velocity has fallen to zero after a maximum duration of 150 seconds which suggests that the kalman filter has indeed cleaned the noise present in the range and the bearing measurements given by an active sonar. A better

kalman filter can be designed if mean of δ in equation (8) is not assumed to be zero.

REFERENCES

- [1] Ali H. Sayed, Adaptive Filters, Wiley, NJ, 2008, ISBN 978-0-470-25388-5.
- [2] Anderson, Brian D. O.; Moore, John B. (1979). *Optimal Filtering*. New York: Prentice Hall. pp. 129–133. ISBN 0-13-638122-7.
- [3] Matisko P. and V. Havlena (2012). Optimality tests and adaptive Kalman filter. Proceedings of IFAC System Identification Symposium, Belgium.
- [4] MATLAB Simulations for Radar Systems Design- Bassem R.Mahafza, At Z.Elsherbeni.
- [5] Simon, D. (2006). *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Wiley-Interscience.
- [6] Stratonovich, R.L. (1960). *Conditional Markov Processes*. Theory of Probability and its Applications, 5, pp. 156–178.

IJERT