Abstract- It is known that the Kalman filter estimation performance heavily depends on the statistical parameters to both the dynamic and observation models, especially the covariance matrix Q. This paper presents a fast genetic algorithm (GA) to adapt extended Kalman filter (EKF) for real time tracking. In the proposed method, the covariance matrix Q can be real-time adjusted by GA to meet some specifications. The simulation results demonstrate that the adaptive EKF is capable of tracking maneuvering targets and reducing the bias and variance of error tracking remarkably.

Keywords- Adaptive Extended Kalman Filter; Maneuver Target Tracking; Fast Genetic Algorithm.

I. INTRODUCTION

The Kalman filter [1] is a well known optimal estimation scheme, satisfying a mean square error performance. Kalman filter has been widely used for radar tracking and navigation systems. The Kalman filter estimation based on certain assumptions about the system's mathematic model. These assumptions include the input and noise statistics. One of the key problems associated with Kalman filters are the statistical properties to both the dynamic and observation models. Also, when the maneuver is presented, it is hard to be predicted. Then, assign a constant noise levels for Kalman filter is not realistic. For many applications, the model statistics noise level are given before the filtering process and will maintain unchanged during the whole recursive process. If such a priori information is inadequate to represent the real statistics noise level, Kalman estimation is not optimal and may cause to an unreliable results, sometimes even leads to filtering divergence [2].

For tracking problems, sudden acceleration or deceleration and sudden change of directions are difficult to predict. Therefore, it is difficult to design a system with constant noise variance that will satisfy all situations. One of the common problems with tracking using Kalman filter is the so called “overshooting” problem. That is effect that the dynamic model keeps position estimation along with previous trend while a vehicle actually turns to another direction [3]. However, in spite of the recent advances in sensor technology, there are no devices that can detect the maneuver of the tracked target in the surveillance and guidance systems. This sudden maneuver of a target implies to a tracking system that it is accelerating unexpectedly and that acceleration may be time-varying and following an unknown profile. Even a short-term acceleration can cause a bias in the measurement sequence and will result in divergence [4]. To overcome the problem of maneuver, several approaches have been proposed. Generally, most of them modify the filter parameters or by using different structures to predict the maneuvers. Among these algorithms the most familiar adaptive filters are introduced by Chan et al [5] and the other by Bar-Shalom and Birmiwal [6]. Chan et al utilized the generalized least square estimation approach to estimate the acceleration input and used the estimate to update the Kalman filter directly. This algorithm is called input estimation (IE). It can track a constant velocity target to convergence very well, but under a noisy environment the target maneuver, it cannot be estimated accurately enough. Bar-Shalom and Birmiwal does not estimate the maneuver value to compensate the filter, but introduces extra state components in the state model when a maneuver is detected and reverts back to the quiescent state when it disappears. This is called variable dimension filter (VDF). It employs a sliding window to find the fading memory average of innovation from the estimator based on the quiescent state to detect the maneuver. In recent years the adaptive processing using the intelligent techniques have been interesting for many researchers, such as combining tracking filters with neural networks, fuzzy logic, and hybrid neural-fuzzy algorithms. Titi and Kaplan [7] used artificial neural networks trained by back propagation algorithm. They used the measured position, speed and acceleration as data training sets for six types of aircrafts. Owen and Stubberud [8] presented a neural extended Kalman filter for tracking maneuver targets. Bahari et al [9] proposed target maneuver detection and tracking technique with the use of hybrid Kalman filter-fuzzy logic architecture. Lho and Painter [10] applied the fuzzy logic to adapt the Kalman filter gain. Duh and Lin [11] used neural-fuzzy network based on Kalman filter to detect maneuver tracking. Zhu et al [12] combined the unscented Kalman filter (UKF) with the neuro-fuzzy inference system (ANFIS) to adjust system noise covariance matrix in target tracking system.

In this paper, we present an adaptive EKF by using a fast genetic algorithm, which adjusts the Q-covariance so that the filter is capable to track the targets, especially in presence of maneuvers.

The paper is organized as following: section II describes the target model and statement of the problem. Section III describes the proposed fast genetic algorithm for real-time applications. Section IV describes the adaptive EKF based fast GA. Simulation and results given in section V. Conclusions are in section VI.
II. STATEMENT OF THE PROBLEM

Consider the motion of a target being tracked modeled by the equations:

\[ X(k+1) = F(k)X(k) + G(k)v(k), \]

\[ Z_{sd}(k+1) = h(X(k+1)) + v(k+1), \]

where \( X(k) = \begin{bmatrix} x(k) & \dot{x}(k) & y(k) & \dot{y}(k) & z(k) & \dot{z}(k) \end{bmatrix}^T \) is the state vector representing the relative range and velocity of the target in the Cartesian coordinates. \( Z_{sd}(k) \) is the radar measurement vector of the range, azimuth, angle of elevation, and the radial velocity. \( w(k) \) is the process noise consisting of the acceleration components, \( v(k) \) is the measurement noise; both sequences \( w(k) \) and \( v(k) \) are assumed to be uncorrelated white Gaussian noise sequence with zero means and the covariance matrices \( Q(k) \) and \( R(k) \), i.e. \( w(k) \sim \mathcal{N}(0,Q(k)), v(k) \sim \mathcal{N}(0,R(k)) \), respectively. \( F(k) \) is the model state transition matrix; \( G(k) \) is the coupling matrix for maneuver inputs, \( h(X(k)) \) is the nonlinear function-vector:

\[
F(k) = \begin{bmatrix} 1 & T(k) & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T(k) & 0 & 0 \\ 0 & 0 & 0 & 1 & T(k) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},
\]

\[
G(k) = \begin{bmatrix} T(k)^2 & 0 & 0 \\ 2 & T(k) & 0 \\ 0 & T(k)^2 & 0 \\ 0 & 2 & T(k) \\ 0 & 0 & T(k)^2 \end{bmatrix},
\]

\[
h(X(k)) = \begin{bmatrix} v(k)^2 + y(k)^2 + z(k)^2 \\ \sqrt{x(k)^2 + y(k)^2 + z(k)^2} \arctg \frac{x(k)}{y(k)} \\ \arctg \frac{z(k)}{x(k)} \\ z(k) - \dot{z}(k) \\ x(k) \cdot \dot{x}(k) + y(k) \cdot \dot{y}(k) + z(k) \cdot \dot{z}(k) \\ \sqrt{x(k)^2 + y(k)^2 + z(k)^2} \end{bmatrix},
\]

\[ T(k) \] is the sampling time interval (or radar update interval). \( k = 0,1,2,... \).

As it is known, tracking task with moving model (1) and (2) can be solved using EKF. EKF estimates the states at some time then obtains feedback in form of noisy measurements; so, estimation falls into two groups, time update equations (predictor) and measurement update equations (corrector) [13]. The filtering algorithm can be represented by the following sequence of calculations at each \( k+1 \) radar measuring step.

**Predicator equations:**

\[ P_{s}(k+1) = F(k)P(k)F^T(k) + G(k)Q(k)G^T(k) \]  

\[ \tilde{X}_p(k+1) = F(k)\tilde{X}_p(k) \]  

\[ \tilde{Z}(k+1) = h(\tilde{X}_p(k+1)) \]  

**Corrector equations:**

\[ P_{s}(k+1) = P_{p}(k+1)H^T(\tilde{X}_p(k+1)) \times \left\{ H(\tilde{X}_p(k+1))P_{p}(k+1)H^T(\tilde{X}_p(k+1)) + R(k) \right\}^{-1} \]

where \( H(\tilde{X}_p(k+1)) = \frac{\partial h(\tilde{X}_p(k+1))}{\partial \tilde{X}_p(k+1)} \) is the Jacobian of nonlinear function (3), defined in predicted point \( \tilde{X}_p(k+1) \).

\[ r(k+1) = Z_{sd}(k+1) - \tilde{Z}(k+1) \]

\[ \tilde{X}_d(k+1) = \tilde{X}_p(k+1) + K(k+1)r(k+1) \]

\[ P_{p}(k+1) = (I - K(k)H(\tilde{X}_p(k+1)))P_{p}(k+1) \]

where \( I \) is the identity matrix.

It is known [6, 17], that the EKF algorithm described above (equations (4)-(10)) will provide a stable suboptimal estimate of the state vector only when the characteristic noise condition \( w(k) \), i.e. variance \( Q(k) \), will correspond to the current value of the target acceleration. Otherwise, the estimate may be strongly biased or even divergent. It should be remembered that, we usually do not have any information about the intensity maneuvering targets, and hence on the current values of its acceleration. Therefore, it is necessary to adjust in real time the covariance \( Q(k) \), thus to obtain satisfactory evaluation target path when adequate computational cost [17].

III. FAST GENETIC ALGORITHM FOR REAL TIME APPLICATION

Genetic algorithms (GAs) are powerful and widely applicable stochastic search and optimization methods based on the concepts of natural selection and natural evaluation. GAs are applied to those problems which either cannot be formulated in exact and accurate mathematical forms and may contain noisy or irregular data or it takes so much time to solve or it is simply impossible to solve by the traditional computational methods. Genetic algorithms were first invented by John Holland in 1960s and were developed by Holland and his students and colleagues at the university of Michigan in the 1960s and the 1970s [14]. GA shows great promise in complex domains because it operates in an iterative improvement fashion. The search performed by it is probabilistically concentrated towards regions of the given data set that have been found to produce a good classification behavior. GAs work on a population of individuals represents candidate solutions to the optimization problem. These individuals consist of strings (called chromosomes) of genes. The genes are a practical allele (gene could be a bit, an integer number, a real value or an alphabet character etc depending on the nature of the
problem. GAs apply the principles of survival of the fitness, selection, reproduction, crossover (recombining), and mutation on these individuals to get, hopefully, a new butter individuals (new solutions).

GA has been shown to be an effective strategy in the off-line design of many fields. In this paper, GA has been used to provide an adaptive decision algorithm for determining the optimum parameter of process noise covariance $Q$ of the Kalman filter. We present some modifications on the conventional genetic algorithm, which made it applicable in the real time optimization. The basic principle of the fast genetic algorithm is by creating an array called elite population, this population contains the best fitted chromosomes from the previous optimizations cycles. The adaptive EKF target tracking starts tracking with its predefined $Q$ value. When the prediction error $r$ (the residual or innovation) is greater than an acceptable value (desired error), which indicates that the filter starts diverge from tracking the target, then the genetic algorithm begin to find the filter’s process noise covariance value ($Q$) (optimization procedure). The best values of $Q$ will be saved in the elite population. In the next measurements, when the prediction error $r$ greater than an acceptable value, we check the elite population array. If elite population array results produced an error greater than the desired error, we start the genetic algorithm optimization to find another filter's noise covariance ($Q$) value, which it is saved in the elite population. This procedure is repeated for every prediction error. Fig. 1 shows the proposed algorithm flowchart.

The following steps describe the proposed fast genetic algorithm [18]:

1. Defines and initialize the variables and parameters of the EKF and GA (the most important of them are):
   
   $NIND$ - number of individuals or chromosomes in the population; $MAXELIT$ - maximum number of individuals in the elite population; $MAXGEN$ - the maximum number of generation for each optimization cycle; $\varepsilon$ - desired error value.

2. First generation initialization:
   
   First generation is initialized by a random real numbers. Each real number corresponds to gene in the individual (or chromosome). Number of individuals is equal to $NIND$, while the number of genes equals to number of variables to be optimized. The genes are randomly generated from predefined limits as follows:
   
   $B_{kk}' \leq \theta_{kk} \leq B_{kk}''$,  
   $k = 1,2,....,M$,  
   $\theta$ is representing a gene; $M$ is the number of genes in the chromosome; $B_{kk}'$ and $B_{kk}''$ are lower and upper limits of the gens respectively.

   The real value encoding scheme saves memory and improves processing speed [15, 16].

3. Read the new measurements:
   
   The new measurements are made to clarify the current values of the objective function (fitness function) in real-time tasks.

4. Fitness evaluation of each individual in a generation:
   
   For each individual in the current generation, calculate the fitness function using a predefined formula or procedure.

5. Adding the best individuals in the elite population:
   
   Elite population, which has a maximum size equal to $MAXELIT$, during operation of the genetic algorithm, it is constantly formed from the individuals with the best fitness function. Further, if the elite population has $MAXELIT$ individuals and we get an individual from the current
population, which has a best fitness function value better than the current fitness function of one or more individuals from the elite population, then this individual replaces the individual having the worst fitness of the current function in the elite population.

6. Checking the termination criterion optimization of elite population:

Decision to complete the procedure of searching an acceptable solution at the current step of the algorithm and output measurement results shall be accepted if the elite population has at least one such individual has a fitness function better than the specified precision optimization error (ε). If such individuals are more than one, of course, select the individual which has the best fitness function. Otherwise, the decision will be on the implementation of the main stages of the classical GA.

7. Classical genetic algorithm (Fig. 1 inner loop):

The classical or conventional genetic algorithm has the following major components [15]: selection of the parents - the most appropriate individuals to participate in the creation of a new generation (recombination); crossover: genes are exchanged or combined during recombination; mutation: selects probabilistically one of the fittest individuals and changes a number of its characteristics in a random way. The formation of a new generation and evaluation for its constituent is species of the fitness function. Finally, for each new generation a verifiable criterion formed at the end of the optimization process. This criterion is met in two ways: if at least one of the individuals of the current generation of fitness function is better than the specified precision optimization (ε), i.e. convergence of GA population [16]; or turn out of maximum generation (MAGEN). With this approach, in principle, complete the loop of optimization procedure, and then quit the GA.

8. Choosing the best individuals from the current genetic population or the elite population, then output of the result:

In this step, the current chromosome is selected from either the classical genetic algorithm population or from the elite-population individuals, which has the best fitness function. This chromosome is taken as the result of solving the problem at the current step measuring real-time algorithm.

We see that the proposed fast genetic algorithm for adaptive EKF algorithm at each step of measurement is able to provide some acceptable results, which, most often, will satisfy a predetermined requirement for accuracy (ε), and only in relatively rare cases where the elite population, and in all population (MAGEN) generations of classical GA fail to find the right individual, this requirement will be violated. In addition, from the scheme of the algorithm (Fig. 1) shows that in such cases when it is possible to find an acceptable solution in the elite population, the procedure is not performed classical GA, and this leads to the release of a significant amount of time between consecutive measurements for other concurrent real-time task.

IV. ADAPTIVE EXTENDED KALMAN FILTER ALGORITHM BASED ON FAST GA

In [17] it is shown that the maneuver targets to substantially increase the innovation process (equation 8). There’s also a simple procedure is proposed to detect this event, based on the calculation of the normalized square signal updates:

\[ e(k+1) = r^T(k+1)S^{-1}(k+1)r(k+1) \]  \hspace{1cm} (12)

where \( S(k+1) \) is the normalized innovation squared;

\[ S(k+1) = H(\hat{X}_p(k+1))P_r(k+1)H^T(\hat{X}_p(k+1)) + R(k) \]  \hspace{1cm} (13)

From equations (8), (12) and (13), it follows that the parameter \( e(k+1) \) is always a scalar quantity. This parameter is used as an indicator for maneuver detection purpose. When the value of \( e(k+1) \) exceeds a predefined level of error, this indicates that the target start in maneuver movement, and the tracking system needs to update its statistic parameters. The most effective parameter updating and give a satisfactory tracking result is the process covariance matrix \( Q(k) \).

For each new measurement \( (k+1) \), the algorithm is predict the target position based on the EKF algorithm (equations (3)–(10)). After that checks the maneuver detection equation (12) and (13). If a maneuver presented, i.e. equation (12) exceeds the predefined limits, then the GA try to optimize the process covariance matrix values \( Q(k) \), and this values are saved in a special population called elite population. In the following measurements, and whenever a maneuver is presented, the covariance matrix values \( Q(k) \) is taken from the elite population without GA evaluation. If the elite population has no satisfactory covariance matrix values \( Q(k) \), the algorithm evaluates the GA to find new covariance matrix values \( Q(k) \).

V. SIMULATION RESULTS

In order to evaluate the performance of the adaptive algorithm for the EKF, consider the following target movement model:

\[ X(k+1) = F(k)X(k) + G(k)a(k), \]  \hspace{1cm} (14)

\[ Z_m(k+1) = h(x(k+1) + v(k+1)), \]  \hspace{1cm} (15)

where \( a(k) = 15 \text{ m/s}^2 \) is the target acceleration.

Total time tracking was 25 sec or 100 discrete steps measurements \( k \). Moreover, the measurements were made cyclically to simulate the mechanical rotation of the radar antenna: 7 measurements after 0.1 seconds with a pause of 1.2 seconds.

The target is started in the following initial values in spherical coordinates: range is 30000 m, azimuth is \( \pi/4 \text{ rad} \), angle of elevation is 0.2 rad, and radial speed is -550 m/sec. Thus, the target first moved to the radar station direction
until about 26000 m, and then began to move away from it, because the radial velocity (due to the constant acceleration) during tracking changed sign and at the end of tracking was approximately 400 m/sec. Azimuth during tracking period varied from 45° to 80°, and angle of elevation from 10° to 23°.

Initial conditions for the filter (3)-(10) in Cartesian coordinates and covariance matrix $P_0(0)$ are calculated on the basis of known formulas [17].

The spherical measurement noise; which is represented by the matrix $R(k)$ has the following noise values: range 20 m, azimuth and angle of elevation 0.028 rad, and radial velocity 4 m/sec.

The genetic algorithm parameters are: $NIND=20$; $MAXELIT=20$, $MAXGEN=100$, $\varepsilon = 5$. The fitness function for evaluation of the predicted range with the measured range is based on minimization of the normalized innovation squared (equation (12)).

The genes are randomly generated from the limit regions and for each element:

$$100 < Q_i < 5000.$$  \hfill (17)

For a comparison purpose we used the conventional Kalman filter described by equations (3)-(10) with the same initial values and covariance matrices. The process covariance matrix in conventional EKF accepted diagonal:

$$Q(k)=\begin{bmatrix}
\sigma^2_{aX} & 0 & 0 \\
0 & \sigma^2_{aY} & 0 \\
0 & 0 & \sigma^2_{aZ}
\end{bmatrix},$$

where $\sigma^2_{aX}=10$, $\sigma^2_{aY}=10$, and $\sigma^2_{aZ}=10$.

Tables 1, 2 and 3 summarizes a comparisons between conventional EKF and the adaptive EKF-GA in the mean (bias) and variance target position in the spherical coordinates, target position in the Cartesian coordinates, and the speed in the Cartesian coordinates respectively.

Fig. 2 shows the covariance matrix $Q(k)$ variations, Fig. 3 shows the target position estimation error in the spherical coordinates, Fig. 4 shows the target position estimation error in the Cartesian coordinates, while Fig. 5 shows the estimation error for the target speed in the Cartesian coordinates.

**TABLE 1** STATISTICAL PERFORMANCE OF THE TARGET POSITION ESTIMATION ERROR IN THE SPHERICAL COORDINATES

<table>
<thead>
<tr>
<th>Axis-parameter</th>
<th>EKF</th>
<th>Adaptive EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>range</td>
<td>$\overline{x}_{conv}$</td>
<td>$\overline{\sigma}_{conv}$</td>
</tr>
<tr>
<td>azimuth</td>
<td>0.032</td>
<td>0.044</td>
</tr>
<tr>
<td>angle of elevation</td>
<td>0.021</td>
<td>0.014</td>
</tr>
<tr>
<td>radial velocity</td>
<td>3.649</td>
<td>2.667</td>
</tr>
</tbody>
</table>

**TABLE 2** STATISTICAL PERFORMANCE OF THE TARGET POSITION ESTIMATION ERROR IN THE CARTESIAN COORDINATES

<table>
<thead>
<tr>
<th>Axis-parameter</th>
<th>EKF</th>
<th>Adaptive EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-axis</td>
<td>444.07</td>
<td>565.34</td>
</tr>
<tr>
<td>y-axis</td>
<td>223.7</td>
<td>336.76</td>
</tr>
<tr>
<td>z-axis</td>
<td>360.3</td>
<td>262.87</td>
</tr>
</tbody>
</table>

**TABLE 3** STATISTICAL PERFORMANCE OF THE TARGET SPEED ESTIMATION ERROR IN THE CARTESIAN COORDINATES

<table>
<thead>
<tr>
<th>Axis-parameter</th>
<th>EKF</th>
<th>Adaptive EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-axis</td>
<td>93.12</td>
<td>127.3</td>
</tr>
<tr>
<td>y-axis</td>
<td>-38.83</td>
<td>100.6</td>
</tr>
<tr>
<td>z-axis</td>
<td>81.33</td>
<td>57.04</td>
</tr>
</tbody>
</table>
Fig. 2 Comparison of the error estimation between the EKF and adaptive EKF in the spherical coordinates. a – range estimation error, b – azimuth estimation error, c – angle of elevation estimation error, d – radial velocity estimation error.

Fig. 3 Comparison of the position error estimation between the EKF and adaptive EKF in the Cartesian coordinates. a – in x-axis estimation error, b – in y-axis estimation error, c – in z-axis estimation error.
optimize the process covariance noise \( Q \) values is by using a fast genetic algorithm. A modification was made to the classical genetic algorithm by introducing an elite population, which makes the search for an acceptable solution for most steps of measurement takes a minimum time. As seen from the figures that the adapted filter try to correct the estimated range and velocity by changing the \( Q \) values. We see that it gives good results in reduction the maneuvering affects when presented in the target trajectory, for both the range and velocity in both spherical and Cartesian coordinates. Also, it reduced the bias divergence and variance noticeably.

**ACKNOWLEDGMENT**

The authors wish to thank ministry of higher education and scientific research – Iraq, Sumner University for support.

**REFERENCES**


