Improved Divided Difference Filter based on Newton-Raphson Method for Target Tracking

Yong Shi  
Electronic and Information 
Engineering Department 
Xi’an Jiaotong University 
Xi’an, Shaan xi Province, China. 
wind-shi@163.com 

Chongzhao Han  
Electronic and Information 
Engineering Department 
Xi’an Jiaotong University 
Xi’an, Shaan xi Province, China. 
czhan@mail.xjtu.edu.cn 

Yongqi Liang  
Electronic and Information 
Engineering Department 
Xi’an Jiaotong University 
Xi’an, Shaan xi Province, China. 
yqi.liang@gmail.com 

Abstract - In this paper, improved divided difference filter, which will be called IDDF for brevity, is proposed for target tracking with nonlinear observation models. The new algorithm is derived from the Newton-Raphson method (or Newton’s method) to approximate maximum a posterior (MAP) estimation. We demonstrate the direct and intuitive relationship between the iterated extended Kalman filter and Newton-Raphson method and can extend the divided difference filter so that iteration is possible. Simulation results show that the proposed filter provides better performance in tracking accuracy when compared to standard DDF, iterated extended Kalman filter (IEKF) and extended Kalman filter (EKF) in presence of severe nonlinearity.

Keywords: Tracking, Nonlinear state estimation, divided difference filter, Newton-Raphson method

1 Introduction

In radar tracking applications, target dynamics are usually modeled in Cartesian coordinates, while the measurements are directly available in the original sensor coordinates. Hence, tracking in Cartesian coordinates using sensor coordinates measurements is actually a nonlinear state estimation problem. Traditionally, target tracking problems are solved using linearized tracking filters, mainly extended Kalman filters (EKF) [1, 2], which is based upon the principles of linearizing the nonlinear models via Taylor series expansions. Due to the assumptions of local linearization and the computation of the Jacobian matrix of the state vector, EKF may provide poor performance or diverges [3, 4].

In recent years, a large number of suboptimal approaches based on linearization techniques have been developed. Julier and Uhlmann have introduced a filter so called Unscented Kalman Filter (UKF) in which an approximation of the pdf’s representing state estimates by a set of the deterministically chosen weighed points are utilized [5, 6]. The divided difference filter (DDF), or the central difference filter (Nørgaard, et al., 2000), which uses divided difference approximations of derivatives based on Stirling's interpolation formula [7, 8], has attracted great attention. Derivatives based on Stirling’s interpolation formula rather than the derivative-based Taylor series approximation, results in a similar mean, but a different covariance from the EKF and using techniques based on similar principles to those of the unscented Kalman filter (UKF). DDF demonstrates relatively consistent stability, despite its similarities to the UKF. It can be shown [8, 9] that the second order DDF (DD2 filter) has the same a priori state as the UKF but a better covariance estimate. We discuss mainly about the DD2 filter in this paper.

However, in the application of target tracking, because of the high nonlinearity, the standard DDF also shows its weakness in robustness, convergence speed and tracking accuracy. To meet the needs of high filtering accuracy precision in target tracking, some improvement measures should be taken into account. In this paper, we proposed an improved DDF based on Newton-Raphson method to address these problems.

We formulate the maximum a posterior (MAP) estimation approach to nonlinear update. We demonstrate the direct and intuitive relationship between the iterated extended Kalman filter and Newton’s method so that we can apply the iteration method to divided difference filter under general Kalman filter framework. Simulation results show that the IDDF based on the Newton-Raphson iteration method performs better than standard DDF, IEKF and EKF.

This paper is organized as follows. The generic nonlinear filtering problem is described in Section 2. Section 3 introduces the divided difference filter. Then, we propose the iterated divided difference filter in Section 4. The results and analysis for the simulation are presented in Section 5. Finally, conclusions are provided in Section 6.

2 General nonlinear filtering problem

In the non-linear case, the general filtering problem in the state space is given by

$$x_k = f(x_{k-1}, v_{k-1})$$

(1)
where, $x_k$ and $y_k$ denote the state variable and observations, $f$ and $h$ are known nonlinear functions, $v_k$ and $w_k$ are independently distributed (i.i.d.) system noise and observation noise sequences respectively, and $v_k \sim \mathcal{N}(\bar{v}_k, Q), w_k \sim \mathcal{N}(\bar{w}_k, R)$.

The objective is to estimate unknown state $x_k$, based on a sequence of observations $Y_k$, where $Y_k$ is a set of received observations from time 0 to $k$.

3 The divided difference filter

3.1 Stirling’s interpolation formula

Consider a nonlinear function $y = f(x) \in \mathbb{R}^m$ with a random variable $x \in \mathbb{R}^n$ with mean $\bar{x}$ and covariance $P_{xx}$. The approximation utilizes the vector form of Stirling’s interpolation formula [8, 10]

$$y = f(\bar{x} + \Delta x) \approx f(\bar{x}) + \tilde{D}_{\Delta x} f + \frac{1}{2!} \tilde{D}_{\Delta x}^2 f$$  \hspace{1cm} (3)

The first and second divided difference operators can be written as

$$\tilde{D}_{\Delta x} f = \frac{1}{\lambda} \left( \sum_{p=1}^{n} \Delta x_p \mu_p \delta_p \right) f(\bar{x})$$  \hspace{1cm} (4)

The first and second divided difference operators can be written as

$$\tilde{D}_{\Delta x}^2 f = \frac{1}{\lambda^2} \left( \sum_{p=1}^{n} \sum_{q=p}^{n} \Delta x_p \Delta x_q (\mu_p \delta_q) (\mu_q \delta_p) \right) f(\bar{x})$$  \hspace{1cm} (5)

where $\lambda$ is an interval of length, $\lambda = \sqrt{3}$ is usually set for Gaussian distribution, and $\delta_p$ and $\mu_p$ denote the partial difference operator and the partial average operator respectively.

$$\delta_p f(\bar{x}) = f(\bar{x} + \frac{\lambda}{2} e_p) - f(\bar{x} - \frac{\lambda}{2} e_p)$$  \hspace{1cm} (6)

$$\mu_p f(\bar{x}) = \frac{1}{2} \left\{ f\left(\bar{x} + \frac{\lambda}{2} e_p\right) - f\left(\bar{x} - \frac{\lambda}{2} e_p\right) \right\}$$  \hspace{1cm} (7)

where, $e_p$ is the $p$th unit operator.

3.2 Divided difference filter

The second-order divided difference filter (DDF) is obtained by using the calculation of the mean and covariance in second-order polynomial approximation. Conceptually, the filter is much like the extended Kalman filter, while the implementation of this filter is simple because it is derivative-free and Jacobians of state and measurement equations are not needed. The idea presented in Schei [7] that the update of the Cholesky factors of the covariance matrices directly is also used in DDF.

First, the four square Cholesky factorizations are introduced.

$$Q = \bar{S}_y \bar{S}_y^T, R = \bar{S}_w \bar{S}_w^T,$$

$$\bar{P} = \bar{S}_x \bar{S}_x^T, \hat{P} = \hat{S}_x \hat{S}_x^T.$$  \hspace{1cm} (8)

The factorization of the noise covariance matrices $Q$ and $R$ can usually be made in advance. $\bar{S}_y$ and $\hat{S}_x$ are updated during application of the filter.

Four matrices containing divided differences are defined by

$$S^{(1)}_{xy}(k-1) = \{(f_i(\hat{x}_{k-1} + \hat{\lambda} \hat{s}_{x,j}, \bar{v}_{k-1})$$

$$- f_i(\hat{x}_{k-1} - \hat{\lambda} \hat{s}_{x,j}, \bar{v}_{k-1})) / 2 \hat{\lambda}\}$$  \hspace{1cm} (9)

$$S^{(1)}_{yx}(k-1) = \{(f_i(\hat{x}_{k-1}, \bar{v}_{k-1} + \lambda \hat{s}_{v,j})$$

$$- f_i(\hat{x}_{k-1}, \bar{v}_{k-1} - \lambda \hat{s}_{v,j})) / 2 \lambda\}$$  \hspace{1cm} (10)

$$S^{(2)}_{xx}(k-1) = \left\{ \frac{(\hat{\lambda}^2 - 1)^{1/2}}{2 \hat{\lambda}^2} \left( f_i(\hat{x}_{k-1} + \hat{\lambda} \hat{s}_{x,j}, \bar{v}_{k}) - 2 f_i(\hat{x}_{k-1}, \bar{v}_{k}) \right) \right\}$$  \hspace{1cm} (11)

$$S^{(2)}_{vy}(k-1) = \left\{ \frac{(\hat{\lambda}^2 - 1)^{1/2}}{2 \lambda^2} \left( f_i(\hat{x}_{k-1}, \bar{v}_{k} + \hat{\lambda} \hat{s}_{v,j}) + f_i(\hat{x}_{k-1}, \bar{v}_{k} - \hat{\lambda} \hat{s}_{v,j}) \right) \right\}$$  \hspace{1cm} (12)

where, $\hat{s}_{x,j}$ denotes the $j$th column of $\hat{S}_x$, and similarly for the other factors.

The DDF state predication is
\[ \bar{x}_k = \frac{\lambda^2 - n_x - n_v}{\lambda^2} f(\hat{x}_{k-1}, \bar{v}_{k-1}) \]

\[ + \frac{1}{2\lambda^2} \sum_{p=1}^{n_v} \left[ f(\hat{x}_{k-1} + \lambda \hat{s}_{x,p}, \bar{v}_{k-1}) + f(\hat{x}_{k-1} - \lambda \hat{s}_{x,p}, \bar{v}_{k-1}) \right] \]

\[ + \frac{1}{2\lambda^2} \sum_{p=1}^{n_v} \left[ f(\hat{x}_{k-1}, \bar{v}_{k-1} + \lambda \hat{s}_{v,p}) + f(\hat{x}_{k-1}, \bar{v}_{k-1} - \lambda \hat{s}_{v,p}) \right] \]

where, \( n_x \) and \( n_v \) denotes the dimension of the system state and system noise vector respectively.

The Cholesky factor of the predicted covariance is obtained by the House-holder transformation of the compound matrix [8, 11, 12]:

\[ \bar{S}_s(k) = \left[ \begin{array}{c} S_s^{(1)}(k-1) \\ S_s^{(2)}(k-1) \end{array} \right] \]

The state prediction covariance:

\[ \bar{P}(k) = \bar{S}_s(k) \bar{S}_s(k)^T \]

Update:

The four additional matrices containing divided differences are defined

\[ S_{s,1}^{(1)}(k) = \left\{ (h(\bar{x}_k, \lambda \bar{s}_{x,i}, \bar{w}_k) - h(\bar{x}_k - \lambda \bar{s}_{x,i}, \bar{w}_k)) / 2\lambda \right\} \]

\[ S_{s,1}^{(2)}(k) = \left\{ (\lambda^2 - 1)^{1/2} / 2\lambda^2 \right\} \left\{ (h(\bar{x}_k + \lambda \bar{s}_{x,i}, \bar{w}_k) + h(\bar{x}_k - \lambda \bar{s}_{x,i}, \bar{w}_k)) \right\} \]

\[ S_{s,1}^{(3)}(k) = \left\{ (\lambda^2 - 1)^{1/2} / 2\lambda^2 \right\} \left\{ (h(\bar{x}_k, \bar{w}_k - \lambda \bar{s}_{w,j})) - 2h(\bar{x}_k, \bar{w}_k) \right\} \]

The Cholesky factor of the state prediction covariance is calculated

\[ \bar{S}_s(k) = \left[ \begin{array}{c} S_s^{(1)}(k) \\ S_s^{(2)}(k) \end{array} \right] \]

The predicted observation

\[ \bar{y}_k = \frac{\lambda^2 - n_x - n_y}{\lambda^2} h(\bar{x}_k, \bar{w}_k) \]

\[ + \frac{1}{2\lambda^2} \sum_{p=1}^{n_v} h(\bar{x}_k + \lambda \bar{s}_{x,p}, \bar{w}_k) + h(\bar{x}_k - \lambda \bar{s}_{x,p}, \bar{w}_k) \]

\[ + \frac{1}{2\lambda^2} \sum_{p=1}^{n_v} h(\bar{x}_k, \bar{w}_k + \lambda \bar{s}_{w,p}) + h(\bar{x}_k, \bar{w}_k - \lambda \bar{s}_{w,p}) \]

4 Improved divided difference filter

4.1 Iterated extended Kalman filter

Before coming to the proposed algorithm, we give a review of the iterated extended Kalman filter [1, 13].

From the Bayesian perspective we have the conditional probability density function (PDF) of given \( Y_k \), provided that all the random variables are Gaussian

\[ p(x_k | Y_{1:k}) = p(x_k | Y_{1:k-1}, y_k) \]

\[ = \frac{1}{c} p(y_k | x_k, Y_{1:k}) p(x_k | Y_{1:k-1}) \]

\[ = \frac{1}{c} N(y_k ; h(x_k), R) N(x_k ; \hat{x}_{k|k-1}, \bar{P}_{k|k-1}^{-1}) \]

where \( c \) is normalization constant. An estimation based on MAP [1] would be equivalent to the minimization problem of some objective function. Then, maximizing this function is equivalent to minimizing the following cost function

\[ f(x) = \frac{1}{2} \left[ (y_k - h(x_{k-1}))^T \bar{R}^{-1} (y_k - h(x_{k-1})) \right] \]

\[ + \frac{1}{2} (x_k - \hat{x}_{k|k-1})^T \bar{P}^{-1} (x_k - \hat{x}_{k|k-1}) \]

so that we can use Newton’s method [14] for the cost functions to the minimum.

We expand the function above in Taylor series to a second order around the \( i \)-th estimate \( x_i \)

\[ f(x_i) \equiv f(x_i) + f'(x_i)(x - x_i)^T \]

\[ + \frac{1}{2} f''(x_i)(x - x_i)^T \]

According to the function, we take the derivatives

\[ f'(x_i) = \bar{P}^{-1}(x_i - \hat{x}_{k|k-1}) - \bar{H}^T \bar{R}^{-1} (y_k - h(x_i)) \]

\[ f''(x_i) = \bar{P}^{-1} + \bar{H}^T \bar{R}^{-1} \bar{H} \]

\[ f'(x_i) + f''(x_i)(x_i - x_i) = 0 \]
The Newton iteration for this problem is
\[ x_{j+1} = x_j - (f''(x_j))^{-1} f'(x_j) \]  
(33)
Substituting Eqs.(28) and (29) into Eq.(31) yields
\[ x_{j+1} = x_j - \left( P_{i-1}^{-1} + H^T R^{-1} H \right)^{-1} \left[ P_{i-1}^{-1} (x_j - \hat{x}_{i-1}-) - H^T R^{-1} \left( y_k - h(x_j) \right) \right] \]  
(34)
where, \( H = h_j (x_j) \) is the Jacobian of \( h(x) \) at \( x_j \).

Starting the iteration for \( j = 0 \) with \( x_0^0 = \hat{x}_{i-1} \) causes the second term in Eq. (34) to be zero and yields after the first iteration, that is, the same as the first order standard EKF.

It is worth noting that
\[ P_{i}^j = (P_{i-1}^{-1} + H^T R^{-1} H)^{-1} \]  
(35)
\[ K^j = P_{i}^j H R^{-1} \]  
(36)
According to the derivations mentioned above, the initialized nonlinear filter equation became.
\[ x_{j+1} = x_j - P_{i}^j P_{i-1}^{-1} (x_j - \hat{x}_{i-1}-) + K^j \left( y_k - h(x_j) \right) \]  
(37)

### 4.2 Improved Divided difference filter

All that discussions above reveals the substantial and intuitive relationship between the iterated nonlinear Kalman filter and Newton’s method, which may easily be overlooked. EKF is an instance of Newton’s iteration method. \( P_{i}^j \) and \( K^j \) can be regarded as the covariance and the Kalman gain associated with \( x_j^j \). Hence, Newton’s iteration method can be extended to the divided difference filter.

Enlightened by the IEKF, a natural idea is that improved performance may be expected if the Newton’s iteration method is implemented in DDF. \( P_{i}^j \) and \( K^j \) can be calculated by using the calculation of the mean and covariance in second-order polynomial approximation.

To make the improved divided difference filter perform as well as possible, some special steps should be taken. Then the improved divided difference filter will be developed to address the problem, using a different strategy. In practice, the number of the iteration steps \( d \) is usually set to be 3-5, which is typically sufficient for convergence. The proposed algorithm can be described as follows.

1. For each instant \( k (k \geq 1) \), evaluate the state predication and corresponding covariance matrix by Eqs. (7)-(15)
2. Update
   \[ l = 1, 2, \ldots d \]
   Initialization for iteration step
   \[ \hat{x}_k^l = \hat{x}_{i-1}- P_{i}^j \hat{x}_{i-1} \]
   (38)
   \[ S_{i+1}^{(l)}(k) = \left\{ (h_j(\hat{x}_{i-1}) + \lambda \hat{s}_{i-1}, \bar{w}_k \right) - h_j(\hat{x}_{i-1} - \lambda \bar{s}_{i-1}, \bar{w}_k) / 2\lambda \} \]  
(39)
\[ S_{i+1}^{(l)}(k) = \{(h_j(\hat{x}_{i-1}), \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k \}
- h_j(\hat{x}_{i-1}, \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k)/2\lambda \} \]  
(40)
\[ S_{i+1}^{(l)}(k) = \{(h_j(\hat{x}_{i-1}), \bar{w}_k + \bar{s}_{i-1}, \bar{w}_k \}
+ h_j(\hat{x}_{i-1}, \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k)/2\lambda \} \]  
(41)
\[ S_{i+1}^{(l)}(k) = \{(h_j(\hat{x}_{i-1}, \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k \}
+ h_j(\hat{x}_{i-1}, \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k)/2\lambda \} \]  
(42)
\[ S_{i+1}^{(l)}(k) = \{(h_j(\hat{x}_{i-1}), \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k \}
+ h_j(\hat{x}_{i-1}, \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k)/2\lambda \} \]  
(43)
The predicted observation
\[ y_{i-1}^l = \bar{y}_{i-1} + \frac{1}{2\lambda^2} \left( \sum_{p=1}^{n_w} h(\hat{x}_{i-1}, \bar{s}_{i-1}, \bar{w}_k + \lambda \bar{s}_{i-1}, \bar{w}_k) + h(\hat{x}_{i-1}, \bar{s}_{i-1}, \bar{w}_k) \right) \]  
(44)
where, \( n_w \) denotes the dimension of the measurement noise vector.
\[ P_{i}^{(l)} = S_{i+1}^{(l)}(k) S_{i+1}^{(l)}(k)^T \]  
(45)
Kalman gain
\[ K_j = P_{i}^{(l)} \left[ S_{i+1}^{(l)}(k) S_{i+1}^{(l)}(k)^T \right]^{-1} \]  
(46)
\[ \hat{x}_k = \hat{x}_{i-1}- K_j S_{i+1}^{(l)}(k) \]  
(47)
\[ P_{i}^{(l)} = \hat{x}_k S_{i+1}^{(l)}(k) \]  
(48)
A posterior update of the state estimate vector
\[ \hat{x}_k = \hat{x}_{i-1}^l - P_{i}^{(l)} P_{i-1}^{-1} (\hat{x}_{i-1}^l - \hat{x}_{i-1}-) + K_j \left( y_k - h(\hat{x}_{i-1}^l) \right) \]  
(49)
3. Outcome
\[ \hat{x}_k = \hat{x}_d \]  
(50)

### 5 Numerical simulations

#### 5.1 Simulation scenarios

Consider the scenario of a tracking problem as follows. For simplicity, assume the radar is located at the origin.

The system dynamics at discrete time can be described as
\[ x_k = f(x_{k-1}) + G v_{k-1} \]  
(51)
\( v_{k-1} \) is a zero-mean Gaussian noise vector with covariance \( Q = diag \left[ 0.1^2, 0.1^2 \right] \)

2071
The initial condition of the target, with state 
\[ \mathbf{x} = [x \ y \ \dot{x} \ \dot{y}]^T \] is, with position and velocity units m and m/s, respectively
\[ \mathbf{x}_0 = [50000 \ -120 \ 50000 \ 0]^T \]

The intervals between the samples are \( T = 2 \) s. An aircraft target has a nearly CV motion form (50000m, 50000m) with an initial velocity of 120m/s for 125s before executing a 1°/s coordinated turn for 90s. Then it flies south for another 125s, followed by a 3°/s turn for 30s. After the turn, it continues to fly at constant velocity. Measurements in a polar format taken by radar at discrete time include range and bearing, given by
\[
\mathbf{z}_k = \begin{bmatrix} r_k \\ \phi_k \end{bmatrix} = h(\mathbf{x}_k) + \mathbf{w}_k
\]
\[
= \begin{bmatrix} \sqrt{x_k^2 + y_k^2} \\ \tan^{-1}\left(\frac{y_k}{x_k}\right) \end{bmatrix} + \begin{bmatrix} v_{rk} \\ v_{\phi k} \end{bmatrix}
\]
\[(52)\]

where \( \mathbf{w}_k \) is an additive zero-mean Gaussian noise vector with variance \( \mathbf{R}_k = \text{diag}\left[\sigma_r^2 \ \sigma_\phi^2\right] \), \( \sigma_r = 20m \) and \( \sigma_\phi = 1° \) are standard deviations for range and bearing, respectively.

The following initialization of the state estimates and corresponding covariance matrix is used
\[ \hat{\mathbf{x}}_0 = [50010 \ -118 \ 50010 \ 2]^T \]
\[ \hat{\mathbf{P}}_0 = \text{diag}\left[100 \ 5 \ 100 \ 5\right] \]

### 5.2 Simulation results and analysis

The nonlinear filters, including EKF, IEKF, DDF and the proposed IDDF are applied to the scenario. The iteration number in this paper is set to be \( d = 5 \). Fifty Monte Carlo simulations are performed and all the filters use the same trajectories. For performance comparison, the root mean square error (RMSE) is utilized to evaluate the performances of the filters:
\[
\text{RMSE} = \left(\frac{1}{N} \sum_{i=0}^{N} |\hat{x}_k^i - x_k^i|^2\right)^{1/2}
\]
\[(53)\]

For reference, the true track of the target is shown in Fig.1. Figs.2-5 compare the RMSE position and velocity for different filters across 450 time steps.
Using the Newton-Raphson iteration method, the updated measurement is amended so that the linearization error is reduced via the estimation of the iteration operations to some extent. Therefore, the filtering performance can be improved with a second-order convergence rate.

The average computational times on dual 2.2GHz Intel processors for a tracking periods of 450 time steps are 1.8473s, 2.19014s, 2.1304s, and 3.1857s for EKF, IEKF, DDF and IDDF algorithms respectively. IDDF can reduce the bias and the estimation error greatly by adding only a few simple iterative operations, some increase in computational burden and complexity is considered acceptable.

Overall, according to the contents mentioned above, it can be concluded that the performance of the IDDF is superior to the other filters, especially in the noisier and more nonlinear situations.

6 Conclusions

In this paper, the IDDF has been developed to improve the tracking performance. The proposed algorithm uses Newton-Raphson method to approximate maximum a posterior (MAP) estimation so that the estimation error of prior DDF can be reduced. Applications of this new method to nonlinear target tracking are investigated with simulation examples. Simulation results demonstrate that the proposed iterated divided difference filter provides much better performance in convergence speed, tracking accuracy compared with DDF, IEKF and EKF.

Acknowledgement

This work is supported by the National Natural Science Foundation of China, under Grand No. 60602026 and 60574033. The authors thank the anonymous reviewers for their comments and recommendations.

References


