Adaptive Neuro-Fuzzy Extended Kalman Filtering for Robot Localization

Ramazan Havangi*, Mohammad Ali Nekoui and Mohammad Teshnehlab

* K.N. Toosi University of Technology, Tehran, Iran, e-mail: havangi@ee.kntu.ac.ir
K.N. Toosi University of Technology, Tehran, Iran, e-mail: manekoui@eetd.kntu.ac.ir
K.N. Toosi University of Technology, Tehran, Iran, e-mail: teshnehlab@eetd.kntu.ac.ir

Abstract—Extended Kalman Filter (EKF) has been a popular approach in localization of a mobile robot. However, the performance of the EKF and the quality of the estimation depends on the correct a priori knowledge of process and measurement noise covariance matrices \((Q_k\) and \(R_k\), respectively). Imprecise knowledge of these statistics can cause significant degradation in performance. In this paper, the Adaptive Neuro-Fuzzy Inference System (ANFIS) supervises the performance of the EKF with adjusting the matrix \(Q_k\) and \(R_k\). The ANFIS is trained using the steepest gradient descent (SD) to minimize the differences between the outputs of ANFIS and desired outputs. The simulation results show the effectiveness of the proposed algorithm.

Keywords—Kalman Filter, localization, fuzzy Inference System.

I. INTRODUCTION

Mobile robot localization is the problem of estimating a robot pose (position, orientation) relative to its environments. Two different kinds of localization exist: relative and absolute. Relative localization which is also known as dead-reckoning (DR) is realized through the measures provided by sensors measuring the dynamics of internal variables of the vehicle. Typical internal sensors are encoders which are fixed to the axis of the driving wheels. The basic drawback of this method is the error of robot’s position and orientation generally grows unbounded with time. Absolute localization is performed by processing the data provided by a proper set of sensors measuring some parameters of the environment in which the vehicle is operating. The methods to obtain absolute measurements can be divided into methods based on the use of landmarks and methods based on the use of maps. The main drawback of absolute measures is their dependence on the characteristics of the environment. Possible changes to environmental parameters may give rise to erroneous interpretation of the measures provided by the localization algorithm. In this paper, we integrate the advantages of "the relative localization" and "the absolute localization" and make them complementary, which will enable the mobile robot to localize itself more accurately. To this purpose, data provided from odometric, laser range finder and MAP are combined together through extended Kalman filter (EKF). The localization based on EKF has been proposed in the literatures in [1], [2], [3], [4], [5], [6], [7], [8] and [9] for the estimation of robot pose. However, a significant difficulty in designing an EKF can often be traced due to incomplete a priori knowledge of the process covariance matrix \(Q_k\) and measurement noise covariance matrix \(R_k\) [10], [11], [12], [13]. In most robot localization application these matrices are unknown. On the other hand, it is well known how poor estimates of noise statistics may seriously degrade the Kalman filter performance [12]. One of the efficient ways to overcome the above weakness is to use an adaptive algorithm. In this paper the EKF coupled with adaptive Neuro-Fuzzy Inference System to adjust the matrix \(Q_k\) and \(R_k\) is presented. Simulation validation shows the effectiveness of the proposed algorithm.

II. KINEMATICS MODELING ROBOT AND ITS ODOMETRY

The state of robot can be modeled as \((x, y, \theta)\) where \((x, y)\) are the Cartesian coordinates and \(\theta\) is the orientation to global environment respectively. The kinematics equations for the mobile robot are in the following form [1], [2], [4]:

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
(l' + v_r \cos(\phi + \gamma) + v_{\gamma}) \\
(l' + v_r \sin(\phi + \gamma) + v_{\gamma}) \\
B \gamma
\end{bmatrix}
\]

(1)

Where \(X = [x, y, \phi]^T\) is vector state, \(B\) is the base line of the vehicle, \(u = [v, \gamma]^T\) is the control input consisting of a velocity input \(V\) and a steer input \(\gamma\). The process noise \(v = [v_r, v_{\gamma}]\) is assumed to be applied to the control input, \(v_r\) to velocity input, and \(v_{\gamma}\) to the steer angle input. The vehicle is assumed to be equipped with a sensor (range-laser finder) that provides a measurement of range \(r\) and bearing \(\rho\) to an observed feature \(\rho\), relative to the vehicle as the following:

\[
\begin{bmatrix}
r \\
\rho
\end{bmatrix} = h(X) =
\begin{bmatrix}
\sqrt{(x-x_i)^2 + (y-y_i)^2 + \omega_r} \\
\tan^{-1} \frac{y-y_i}{x-x_i} - \phi + \omega_{\phi}
\end{bmatrix}
\]

(2)
Where \((s_i, y_i)\) is the position landmark in map and \(W = [\omega_1 \, \omega_2]^T\) is related to the observation noise.

III. EXTENDED KALMAN FILTER (EKF)
Kalman filter (KF) is widely used in studies of dynamic systems, analysis, estimation, prediction, processing and control. Kalman filter is an optimal solution for the discrete data linear filtering problem. KF is a set of mathematical equations which provide an efficient computational solution to sequential systems. The filter is very powerful in several aspects: It supports estimation of past, present, and future states (prediction), and it can do so even when the precise nature of the modeled system is unknown. The filter is derived by finding the estimator for a linear system, subject to additive white Gaussian noise. However, the real system is non-linear; Linearization using the approximation technique has been used to handle the non-linear system. This extension of the nonlinear system is called the Extended Kalman Filter (EKF). The general non-linear system and measurement form is as given by (3) and (4) as follows:

\[
x_{k+1} = f(x_k, u_k) + w_k \tag{3}
\]

\[
z_k = h(x_k) + v_k \tag{4}
\]

The system, measurement noises are assumed to be Gaussian with zero mean and are represented by their covariance matrices \(Q_k\) and \(R_k\):

\[
E[w_j w_j^T] = Q_k \quad k = j
\]

\[
E[w_j w_j^T] = 0 \quad k \neq j
\]

\[
E[v_j v_j^T] = R_k \quad k = j
\]

\[
E[v_j v_j^T] = 0 \quad k \neq j
\]

The Extended kalman filter algorithm has two groups of equations [14]:

1) The prediction equations:
   The extended Kalman filter predicts the future state of system \(\hat{x}_{k+1}\) based on the available system model \(f(.)\) and projects ahead the state error covariance matrix \(P_{k+1}^{-}\) using the time update equations:

\[
\begin{align*}
\hat{x}_{k+1}^- &= f(\hat{x}_k, u_k) \\
P_{k+1}^- &= \nabla f_k P_k \nabla f_k^T + G_u Q_u G_u^T
\end{align*}
\]

Where

\[
\begin{align*}
\nabla f_k &= \frac{\partial f}{\partial X} \\
G_u &= \frac{\partial f}{\partial u}
\end{align*}
\]

2) Measurement updates equations
   Once measurements \(z_k\) become available the Kalman gain matrix \(K_k\) is computed and used to incorporate the measurement into the state estimate \(\hat{x}_k\). The state error covariance for the updated state estimate \(P_k\) is also computed using the following measurement update equations:

\[
K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{11}
\]

\[
\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-)) \tag{12}
\]

\[
P_k = (I - K_k H_k) P_k^- \tag{13}
\]

Where \(I\) is an identity matrix and \(H_k\) is following:

\[
H_k = \frac{\partial h}{\partial X} \tag{14}
\]

In the above equations, \(\hat{x}_k\) is an estimation of the system state vector \(x_k\) and \(P_k\) is the covariance matrix corresponding to the state estimation error defined by

\[
P_k = E[(x_k - \hat{x}_k) (x_k - \hat{x}_k)^T] \tag{15}
\]

The difference between the prediction and observed measurements is called the measurement innovation, or residual, generally denoted as \(r_k\):

\[
r_k = z_k - h(\hat{x}_k^-) \tag{16}
\]

The innovation represents the additional information variable to the filter in consequence to the observation \(z_k\). For an optimal filter the innovation sequence is a sequence of independent Gaussian random variables.

IV. LOCALIZATION BASED ON EKF
We assume that robot knows map of environment. The EKF estimates a robot pose given a map of the environment and range-bearing of landmarks measurements. For this purpose, the data provided by odometric, map and laser range-finder are fused together by means of an EKF. The algorithm consists of such steps as position prediction, observation, measurement prediction, matching and estimation. We now proceed to discuss of these steps in detail.

Step1) Initialization
   Initialize the state vector \(\hat{x}_0\) and covariance matrix \(P_0\) of the mobile robot

Step 2) Robot Position Prediction: The robot position at time step \(k+1\) is predicted based on its old localization (at time step \(k\)) and its movement due to the control input \(u_k\):

\[
\hat{x}_{k+1} = f(\hat{x}_k, u_k) \tag{17}
\]

\[
P_{k+1}^- = \nabla f_k P_k \nabla f_k^T + G_u Q_u G_u^T \tag{18}
\]

Where

\[
\nabla f_k = \frac{\partial f}{\partial X} \tag{19}
\]

\[
G_u = \frac{\partial f}{\partial u} \tag{20}
\]

Step3) Measurement prediction
   Using the predicted robot position \(\hat{x}_{k+1}^-\) and the current map we can generate the predicted measurement \(\hat{z}_k\) according to (2). The error between the actual measurement \(z_k\) and the predicted measurement based on estimate of the state is:

\[
v_k = z_k - \hat{z}_k \tag{21}
\]
Where \( v_j \) is innovation sequence (or residual) and its covariance is:
\[
S_k = \nabla h_i P_k \nabla h_i^T + R_k
\]
(22)

Where
\[
\nabla h_i = \frac{\partial h_i}{\partial X}
\]
(23)

Step 4) Matching
The goal of the matching procedure is to produce an assignment from measurements to the landmarks (stored in the map). If the measurement result satisfies the following inequality, it is thought to be eligible, otherwise, it is not, and it will be abnegated.
\[
v_j S_j v_j^T \leq G
\]
(24)

Step 5) Estimation
\[
\hat{x}_i = \hat{x}_i + K_i (z_i - h(\hat{x}_i))
\]
(25)

Where \( K_k \) is the gain of the kalman gain:
\[
K_k = P_k \nabla h_i^T (\nabla h_i P_k \nabla h_i^T + R_k)^{-1}
\]
(26)

The new state covariance matrix is:
\[
P_k = (I - K_k \nabla h_i) P_k
\]
(27)

Step 6) Return to step 2

V. LOCALIZATION BASED ON ADAPTIVE NEURO-FUZZY EKF (ANFEKF)

As stated earlier, localization based on EKF assumes complete a priori knowledge of the process and measurement noise statistics; matrices \( Q_k \) and \( R_k \). However, in most applications these matrices are unknown. An incorrect a priori knowledge of \( Q_k \) and \( R_k \) may lead to performance degradation [12] and it can even lead to practical divergence [13]. One of the effective ways to overcome the above mentioned weakness is to use an adaptive algorithm. Two major approaches that have proposed for adaptive EKF are Multiple Model Adaptive Estimation (MMAE) and Innovation Adaptive Estimation (IAE) [12]. In this paper IAE adaptive scheme of the EKF coupled with ANFIS is used to adjust the filter from divergence.

A. Localization based on ANFEKF (\( Q_k \) is fixed)
The covariance matrix \( R_k \) represents the accuracy of measurement instrument. Assuming that the noise covariance \( Q_k \) is completely known, an algorithm to estimate the measurement noise covariance \( R_k \) can be derived. In this case, an innovation based adaptive estimation (IAE) algorithm to adapt the measurement noise covariance matrix \( R_k \) is derived. In particular, the technique known as covariance matching is used. The basic idea behind this technique is to make the actual value of the covariance of the residual to be consistent with its theoretical value [12]. The innovation sequence \( r_i \) has a theoretical covariance \( S_i \) where obtained from EKF algorithm. The actual residual covariance \( \hat{C}_k \) can be approximated by its sample covariance, through averaging inside a moving window of size N as following.
\[
C_i = \frac{1}{N} \sum_{n=-N}^{N} (\hat{r}_i \hat{r}_i^T)
\]
(28)

Where \( i_0 \) is first sample inside the estimation window. If the actual value of covariance \( C_k \) has discrepancies with its theoretical value, then the diagonal elements of \( R_k \) based on the size of this discrepancy can be adjusted. The objective of these adjustments is to correct this mismatch as far as possible. The size of the mentioned discrepancy is given by a variable called the degree of mismatch (\( DOM_k \)), defined as
\[
DOM_k = S_k - \hat{C}_k
\]
(29)

The basic idea used by an ANFIS, to adapt the matrix \( R_k \) is as follows: from (22) an increment in \( R_k \) will increase \( S_k \) and vice versa. Thus, \( R_k \) can be used to vary \( S_k \) in accordance with the value of \( DOM_k \) in order to reduce the discrepancies between \( S_k \) and \( \hat{C}_k \). The adaptation of the \((i,i)\) element of \( R_k \) is made in accordance with the \((i,i)\) element of \( DOM_k \). The general rules of adaptation are as following:

If \( DOM_k (i,i) \equiv 0 \) then maintain \( R_k \) unchanged.
If \( DOM_k (i,i) > 0 \) then decrease \( R_k \).
If \( DOM_k (i,i) < 0 \) then increase \( R_k \).

Fig.1 and Fig.2 presents membership functions for \( DOM_k (i,i) \) and \( R_k (i,i) \).

1) The ANFIS Architecture (\( Q_k \) is fixed)
The ANFIS model has been considered as a single-input-single-output system. This ANFIS is a tree layers network as shown in Fig.3. Let \( u_i^l \) and \( o_i^l \) denote the input to output from the \( i \) th node of the \( l \) th layer, respectively. To provide a clear understanding of an ANFIS, the function of layer 1 to layer 3 are defined as follows:

Layer 1: The node in this layer only transmits input values to the next layer directly, i.e.,
\[
o_i^1 = u_i^1
\]
(30)
Layer2: In this layer, each node only performs a membership function. Here, the input variable is fuzzified employing five membership functions (MFs). The output of the \(i^{th}\) MF is given as:

\[
o_i = \mu_i(u_i) = \mu_i(DOM_i(i,i))
\]  

(31)

Layer3: Output Layer
This layer performs defuzzification where the difuzzified output is calculated as a weighted average of all its input is calculated. Here the output from the solitary node in this layer can be calculated as

\[
AdjR = \sum_{i=1}^{n} \alpha_i^2 \mu_i^2
\]  

(32)

Fig.3. The ANFIS Structure.

Finally, adjustments of \(R_k\) is performed using the following relation

\[
R_i = R_k + \Delta R_k
\]  

(33)

Where \(\Delta R_k\) is ANFIS output and \(DOM_i\) is ANFIS input. Fig.4 shows the block diagram of Localization based on ANFEKF.

```
Robot
```

```
Odometry
```

```
Range Laser
```

```
Matching
```

```
MAP
```

```
EKF
```

```
\(S_k\)
```

```
\(C_k\)
```

```
\(\Delta R_k\)
```

```
\(DOM_i\)
```

```
\(\hat{C}_k\)
```

```
\(\hat{S}_k\)
```

Fig.4. Localization Based on ANFEKF (\(Q_k\) Fixed).

2) Learning Algorithm
The aim of the Learning algorithm is to adjust the network weights through the minimization of following cost function:

\[
E = \frac{1}{2} e^2
\]  

(34)

Where

\[
e = S_k - \hat{C}_k
\]  

(35)

By using the back propagation (BP) learning algorithm, the weighting vector of the ANFIS is adjusted such that the error defined in (35) is less of than a desired threshold value after a given number of training cycles. The well-known BP algorithm may be written as:

\[
y^{-1}(k + 1) = y^{-1}(k) + \eta(-\frac{\partial E(k)}{\partial y^{-1}(k)})
\]  

(36)

Where \(\eta\) and \(y^{-1}\) represent, respectively, the learning rate and tuning parameter of ANFIS. The gradient of \(E\) with respect to an arbitrary weighting vector \(y^{-1}\) as following

\[
\frac{\partial E(k)}{\partial y^{-1}(k)} = \varepsilon o^2
\]  

(37)

B. Localization based on ANFEKF (\(R_k\) Fixed)
Assuming that the noise covariance matrix \(R_k\) is completely known an algorithm to estimate matrix \(Q_k\) can be derived. The idea behind the process of adaptation of \(Q_k\) is as follow (18) can be rewritten as

\[
S_k = \nabla_h^T \nabla_f^T P_k \nabla_h^T + \nabla_h G_k Q_k \nabla_f^T \varepsilon h_k^T + R_k
\]  

(38)

Fig.5. Localization Based on ANFEKF (\(R_k\) Fixed).

Fig.6. The ANFIS Structure (\(R_k\) fixed).

It may be deduced from (38) that a variation in \(Q_k\) will affect the value of \(S_k\). If \(Q_k\) is increased, then \(S_k\) is increased, and vice versa. Thus if a mismatch between \(S_k\) and \(\hat{C}_k\) is observed then a correction can be made through augmenting or diminishing the value of \(Q_k\). The two general adaptation rules are defined as following

1. If \(DOM_i (1,1)\) is L and \(DOM_i (2,2)\) is L then \(AdjQ_k\) is H
2. If \( \text{DOM}_1 (1,1) \) is \( Z \) and \( \text{DOM}_2 (2,2) \) is \( Z \), then \( \text{AdjQ}_k \) is \( Z \). Thus \( \Delta Q_k \) is adapted in this way
\[
\Delta Q_k = Q_k \Delta Q_k
\] (39)
Where \( \Delta Q_k \) is the ANFIS output and \( \text{DOM}_1 (1,1) \) and \( \text{DOM}_2 (2,2) \) are ANFIS input.

1) The ANFIS Architecture (\( R \) is fixed)
The ANFIS model has been considered as a two-input-single-output system that shown in Fig.5. The Training algorithm is such as case previous.

VI. IMPLEMENTATION AND RESULTS
Experiments have been carried out to evaluate the performance of the proposed approach in comparison with classical method. Fig.7 shows the robot trajectory and landmark location.

![Fig.7. Trajectory Robot.](image)

The star points (*) depict location of the Landmarks that are known. The initial position of the robot is assumed to be \( x_0 = 0 \). The uncertainties in control inputs are specified as \( \sigma_\omega = 0.3 \text{ m/s} \) and \( \sigma_\theta = 3 \text{ deg} \). The uncertainties range-bearing sensor is specified as \( \sigma_r = 0.1 \text{ m/s} \) and \( \sigma_\theta = 1 \text{ deg} \). The performance of the proposed method is compared with localization based on EKF where matrices \( Q_k \) and \( R_k \) are kept static throughout the experiment. First we consider the situation where the sensor statistics are wrongly considered as: \( \sigma_r = 2.0 \) and \( \sigma_\theta = 0.5 \) and the noise covariance \( Q_k \) is completely known. To verify the consistency of both algorithms, average Normalized Estimation Error Squared (NEES) is used as a measure factor. For an available ground truth \( \hat{x}_k \) and an estimated mean and covariance \( \{ \hat{x}_k, \hat{P}_k \} \), we can use NEES to characterize the filter performance:
\[
\epsilon_k = (x_k - \hat{x}_k) \hat{P}_{k|k}^{-1} (x_k - \hat{x}_k)
\] (40)
Consistency is evaluated by performing multiple Monte Carlo runs and computing the average NEES. Given \( N \) runs, the average NEES is computed as
\[
\bar{\epsilon} = \frac{1}{N} \sum_{i=1}^{N} \epsilon_i
\] (41)
Given the hypothesis of a consistent linear-Gaussian filter, \( N \bar{\epsilon} \) has a \( \chi^2 \) density with \( N \text{ dim}(x_k) \) degrees of freedom \( \beta_{15} \). Thus, for the 3-dimensional vehicle pose with Twenty Monte Carlo simulations, the two sided 95% probability concentration region for \( \bar{\epsilon}_k \) is bounded by interval [2.02, 4.17].

![Fig.8. Localization Based on AFEKF (\( R \) fixed).](image)

Finally, the consistency of both localizations Based on AFEKF and EKF are compared. In this regard, we have considered the situation where the sensor statistics are set wrongly as: \( \sigma_r = 2.0 \) and \( \sigma_\theta = 0.5 \) and the noise covariance \( Q_k \) is completely known. To verify the consistency of both algorithms, average Normalized Estimation Error Squared (NEES) is used as a measure factor. For an available ground truth \( x_k \) and an estimated mean and covariance \( \{ \hat{x}_k, \hat{P}_k \} \), we can use NEES to characterize the filter performance:
\[
\epsilon_k = (x_k - \hat{x}_k) \hat{P}_{k|k}^{-1} (x_k - \hat{x}_k)
\] (40)
Consistency is evaluated by performing multiple Monte Carlo runs and computing the average NEES. Given \( N \) runs, the average NEES is computed as
\[
\bar{\epsilon} = \frac{1}{N} \sum_{i=1}^{N} \epsilon_i
\] (41)
Given the hypothesis of a consistent linear-Gaussian filter, \( N \bar{\epsilon} \) has a \( \chi^2 \) density with \( N \text{ dim}(x_k) \) degrees of freedom \( \beta_{15} \). Thus, for the 3-dimensional vehicle pose with Twenty Monte Carlo simulations, the two sided 95% probability concentration region for \( \bar{\epsilon}_k \) is bounded by interval [2.02, 4.17].

![Fig.9. Localization Based on AFEKF (\( Q \) is fixed).](image)

Now, we consider the situation where the uncertainties in control inputs are wrongly considered as: \( \sigma_\omega = 0.03 \text{ m/s} \) and \( \sigma_\theta = 0.5 \text{ deg} \). Fig.9 shows RMSE for this situation. Such as previous situation, it is observed that localization based on FAEKF is more accurate than localization based on EKF.
Fig. 10 and Fig. 11 show that the consistency of localization based on AFEKF is more than that of localization based on EKF.

VII. CONCLUSION

This paper proposed a new method for the accurate localization of a mobile robot. The approach is based on the use of an EKF with ANFIS for the adjustment of the process and measurement noise covariance matrices. The main advantage of the proposed method is that consistency of this approach is more than that of localization based on EKF. This is because that the theoretical value of the innovation sequence is matched with its actual value in the proposed method. The simulation results are shown that localization based on AFEKF is more accurate than localization based on EKF.

REFERENCES

[1] F. Kong, Y. Chen, J. Xie, and Gang, “Mobile robot localization based on extended Kalman filter,” in Proc. of the 6th World Congress on Intelligent Control and Automation, June 21-23, 2006.

[2] J. Kim, Y. Kim, and S. Kim, “An accurate localization for mobile robot using extended Kalman filter and sensor fusion,” in Proc. of the Int. Joint Conf. on Neural Networks, 2008.

[3] T. H. Cong, Y. J. Kim and M. Lim, “Hybrid Extended Kalman Filter-based Localization with a Highly Accurate Odometry Model of a Mobile Robot,” Int. Conf. on Control, Automation and Systems 2008.

[4] S. Kwon, K. W. Yang and S. Park, “An Effective Kalman filter Localization Method for mobile Robots,” in Proc. of the 2006 IEEE/RSJ, Int. Conf. on Intelligent Robots and Systems.

[5] G. Reina, A. Vargas, K. Nagatani, and K. Yoshida “Adaptive Kalman Filtering for GPS-based Mobile Robot Localization,” in Proc. of the IEEE, Int. Workshop on Safety, Security and Rescue Robotics, 2007.

[6] M. Betke and L. Gurvits, “Mobile robot localization using landmarks,” IEEE Trans on Robotics And Automation, Vol. 13, No. 2, APRIL 1997.

[7] I. Roumeliotis and A. Bekey, “An extended Kalman filter for frequent local and infrequent global sensor data fusion,” in SPIE Int. Symposium on Int. Systems and Advanced Manufacturing, 1997.

[8] W. Jin and X. Zhan, “A modified Kalman filtering via fuzzy logic system for ARVs Localization,” in Proc. of the 2007 IEEE, Int. Conf. on Mechatronics and Automation.

[9] L. Jetto, and S. Longhi, “Development and experimental validation of an adaptive extended kalman filter for the localization of mobile robots,” IEEE Trans on robotics and automation, vol. 15, No. 2, April 1999.

[10] E. E. El Madbouly, A. E. Abdalla, and Gh. M. El Banby, “Fuzzy adaptive Kalman filter for multi sensor system,” IEEE, 2009.

[11] W. Qin, W. Yuan, H. Chang, L. Xue, and G. Yuan, “Fuzzy Adaptive Extended Kalman Filter for miniature Attitude and Heading Reference System,” 4th IEEE Int. Conf. on Nano/Micro Engineered and Molecular Systems, 2009.

[12] R. K. Mehra, “On the identification of variances and adaptive Kalman filtering,” IEEE Trans. Autom. Control, Vol. AC-15, No. 2, pp. 175–184, Apr. 1970.

[13] R. J. Fitzgerald, “Divergence of the Kalman filter,” IEEE Trans. Autom. Control, Vol. AC-16, No. 6, pp. 736–747, Dec. 1971.

[14] M. S. Grewal and A. P. Andrews, “Kalman filtering: Theory and practice,” Prentice-HALL, 1993.

[15] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, “Estimation with Applications to Tracking and Navigation,” John Wiley and Sons, 2001.