2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

# A New Adaptive Kalman Filter by Combining Evolutionary Algorithm and Fuzzy Inference System

Yudan Huo, Zhihua Cai, Wenyin Gong and Qin Liu

*Abstract*— **The performance of the Kalman filter (KF), which is recognized as an outstanding tool for dynamic system state estimation, heavily depends on its parameter $R$, called the measurement noise covariance matrix. However, it's difficult to get the exact value of $R$ before the filter starts, and the value of $R$ is likely to change with the measurement environment when the filter is working. To solve this problem, a new parameter adaptive Kalman filter is proposed in this paper. In this new Kalman filter, the initial value of $R$ is offline decided by Evolutionary Algorithm (EA), and the value of $R$ decided by EA is online updated by Fuzzy Inference System (FIS). A simulation experiment based on target tracking is carried out, and the results demonstrate that the new adaptive Kalman filter proposed in this paper (HydGeFuzKF) has a stronger adaptability to time-varying measurement noises than regular Kalman filter (RegularKF), Sage-Husa adaptive Kalman filter (SageHusaKF), the adaptive Kalman filter only based on genetic algorithm (GeneticKF) and the adaptive Kalman filter only based on fuzzy inference system (FuzzyKF).**

## I. Introduction

KALMAN FILTER (KF) is one of the most popular methods for estimating the states of dynamic system from an incomplete and noisy measurement. As a recursion algorithm, Kalman filter has a small requirement on calculation and memory space, which makes it more favorable in the real-time system application. Since it was proposed in 1960s, Kalman filter has been widely applied in many fields, such as navigation, signal processing, control system and information fusion. It also has many improved variants such as Extended Kalman Filter (EKF) [1] and Unscented Kalman Filter (UKF) [2].

The Kalman filter works well in the condition that the $a$ $priori$ statistics of the stochastic errors in both dynamic process and measurement models are assumed to be available, which is very difficult in practical applications, especially the measurement noise covariance $R$. First, it is not easy to get accurate noise statistics data before the filter starts to work. And second, the noise statistics may change with time when the filter is working. To solve this problem, many adaptive mechanisms are used into Kalman filter, which is called Adaptive Kalman Filter (AKF). According to the filter results, adaptive Kalman filter can optimize or estimate its noise statistics parameters adaptively to adjust to the change of process and measurement noise. The traditional adaptive Kalman filter algorithms include Sage-Husa Kalman filter [3] based on maximum $a$ $posteriori$ probability estimation, Bayes adaptive filter [4] based on Bayesian estimation, Robust Kalman filter [5] and so on. Mehra [6] classified the different methods of adaptive filter into four categories: Bayesian, maximum likelihood, correlation and covariance matching.

In recent years, Evolutionary Algorithms (EA) and Fuzzy Inference System (FIS) have been successfully used in adaptive Kalman filter. In literature, Szabat [7], Salvatore [8], Jatoth [9], and Mosavi [10] published some of the earlier work on using evolutionary algorithms to optimize the initial values of the parameters of Kalman filter, but they did not consider that the initial values they obtained may change when filter is working. Ali [11], Jwo [12], Yadaiah [13], Shi [14] and Talel [15] used fuzzy inference system to adjust the parameters of Kalman filter in real time to meet the change of process and measurement noise, but they did not pay any attention on how to decide the initial values of these parameters.

However, the initial values of the parameters and their online adjustment are both very important to the Kalman filter, especially the measurement noise covariance matrix $R$. So in this paper, we proposed a new adaptive Kalman filter by combining evolutionary algorithm and fuzzy inference system. In this new adaptive Kalman filter, we utilized the evolutionary algorithm to determine the initial value of parameter $R$. Furthermore, the fuzzy inference system is used to adjust the value of $R$ with time based on the filter performance.

The rest of this paper is organized as follows: Section II introduces the Kalman filter algorithm and its parameter adaptability problem. Section III introduces the new adaptive Kalman filter we proposed (HydGeFuzKF) by combining EA and FIS. Some simulation results are presented in Section IV to show its performance. Finally, Section V concludes the paper.

## II. Kalman Filter

### A. Kalman Filter Algorithm

Kalman filter is one of the most popular algorithms in the control area. It is always been used to estimate the state of a dynamic system. The system model and measurement model for a simple linear discrete-time Kalman filter are represented as:

$$x_k = \Phi x_{k-1} + \omega_k \tag{1}$$

Yudan Huo, Zhihua Cai, Wenyin Gong and Qin Liu are with the Department of Computer Science, China University of Geosciences, Wu Han, 430074, PR China (email: yudanhuo@163.com, zhcai@cug.edu.cn).

This work is supported in part by the National Natural Science Foundation of China under Grant No.61203307, the Natural Science Foundation of Hubei Province under Grant No. 2013CFA004, and the Fundamental Research Funds for the Central Universities at China University of Geosciences (Wuhan) under Grant Nos. CUG130413 and CUG090109.

978-1-4799-1488-3/14/$31.00 ©2014 IEEE

$$z_k = Hx_k + \nu_k \tag{2}$$

where $x_k \in R^n$ is the system state vector, $\omega_k \in R^n$ is the system noise vector, $z_k \in R^m$ is the measurement vector to system state, and $\nu_k \in R^m$ is the measurement noise vector. $\Phi$ is the state transition matrix, which reflects the mathematical or physical relationship between system state $x_k$ and $x_{k-1}$. $H$ is the measurement matrix, which represents the relationship between the measurement $z_k$ and system state $x_k$. The vector $\omega_k$ and $\nu_k$ are both white noise sequences with zero means and mutually independent:

$$E[\omega_k \omega_i^T] = \begin{cases} Q, & i = k \\ 0, & i \neq k \end{cases}; \tag{3}$$

$$E[\nu_k \nu_i^T] = \begin{cases} R, & i = k \\ 0, & i \neq k \end{cases}; \tag{4}$$

$$E[\omega_k \nu_i^T] = 0, \text{for all } i \text{ and } k, \tag{5}$$

where non-negative definite matrix $Q$ is the system noise covariance matrix, positive definite matrix $R$ is the measurement noise covariance matrix, $E[\bullet]$ represents expectation, and superscript "T" denotes matrix transpose.

The purpose of Kalman filter is to estimate the actual value of $x_k$ in equation (1). Based on the model equations (1)-(5), the key five equations of discrete-time Kalman filter is summarized as follows:

$$\hat{x}_k^- = \Phi \hat{x}_{k-1}; \tag{6}$$

$$P_k^- = \Phi P_{k-1} \Phi^T + Q; \tag{7}$$

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}; \tag{8}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-); \tag{9}$$

$$P_k = (I - K_k H) P_k^-. \tag{10}$$

In the above equations, $\hat{x}_k$ is the estimation value of the system state $x_k$, $P_k$ is the error covariance matrix defined by $E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T]$, and weighting matrix $K_k$ is the Kalman gain matrix. The Kalman filter algorithm starts with an initial condition value $\hat{x}_0$ and $P_0$. Equations (6)-(7) are the time update equations of Kalman filter from step $k$-1 to $k$. These equations generate $a\ priori$ estimation of system state at step $k$. Equations (8)-(10) are the measurement update equations of the algorithm. They incorporate the measurement value $z_k$ into $a\ priori$ estimation to obtain an improved $a\ posteriori$ estimation, which is the output of Kalman filter at step $k$.

The procedure of Kalman filter algorithm is showed by Algorithm 1.

**Algorithm 1** Kalman Filter

Set the parameters $\Phi$, $H$, $Q$ and $R$;
Initialize the $\hat{x}_0$, $P_0$, $k$ = 1;
**while** (need to estimate the system state) **do**
  Time Update:
    $\hat{x}_k^- = \Phi \hat{x}_{k-1}$;
    $P_k^- = \Phi P_{k-1} \Phi^T + Q$;
  Get the measurement $z_k$;
  Measurement Update:
    $K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$;
    $\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-)$;
    $P_k = (I - K_k H) P_k^-$;
  $k = k + 1$;
**end while**

### B. Parameter Estimation Problem of Kalman Filter

Kalman filter is a very powerful method to estimate the system state. But it only works well in the condition that the parameters $\Phi$, $H$, $Q$ and $R$ in the equations (6)-(10) are precisely known. Inaccurate values of these parameters will reduce the filtering accuracy, increase the filtering error, and even cause filter divergence.

Generally, we can get the $\Phi$ and $H$ by building accurate system and measurement models, and the value of $Q$ is stable in a given system in most cases. The most difficult, and also important is to obtain the value of $R$, because of its variability. First, there is no direct method to estimate its value. Second, its value will change with time. For example in the navigation system, one very important application of Kalman filter, once the external environment of the target which is being navigated has changed, the value of $R$ will change immediately. So in the HydGeFuzKF we proposed, evolutionary algorithm and fuzzy inference system is used to estimate the value of $R$, both before the filter starts to work and when it is working.

## III. The New Adaptive Kalman Filter Algorithm: HydGeFuzKF

In the HydGeFuzKF proposed in this paper, the initial value of the measurement noise covariance matrix $R$ is optimized offline by genetic algorithm, and the optimal value obtained by genetic algorithm is adjusted online by fuzzy inference system.

### A. Offline Optimization of $R$ by Genetic Algorithm

Genetic algorithm (GA), a very popular branch of evolutionary algorithm, was first developed by Holland in 1970s. Modeled on the natural biological evolution process, GA is a random search procedure to find the global optimal solution for some optimization problems. It works well, especially in dealing with complicated nonlinear problems. In the HydGeFuzKF, we utilize real-coded GA to decide the initial value of parameter $R$. The procedure is outlined as follows:

Step 1. Encoding. The measurement noise $\nu_k$ is a $m$-dimension vector and its covariance matrix $R$ is a positive definite matrix. So the $m$ diagonal elements of the parameter $R$ are coded into a whole chromosome, which is a long real-valued string and is given as follows:

$$chromosome = [r_1, r_2, \ldots, r_i, r_{i+1}, \ldots, r_{m-1}, r_m] \quad (11)$$

where $r_i$ is the $i$-th diagonal element of matrix $R$.

Step 2. Population Initialization. Generate an initial population of the chromosomes by randomly selecting values between the upper bound and lower bound of the element of matrix $R$.

Step 3. Fitness Evaluation. In the current generation, each of the chromosomes are decoded back to the corresponding values of $R$. Then, these values are separately used to different Kalman filters to yield the fitness function.

The fitness function is designed by the mean of the residual vectors $\varepsilon_k$ of Kalman filter. At filter step $k$, the residual $\varepsilon_k$ is defined as:

$$\varepsilon_k = \tilde{z}_k = z_k - \hat{z}_k^- = z_k - H\hat{x}_k^- \quad (12)$$

where $z_k$ is the measurement at step $k$, $H$ is the measurement matrix, $\hat{x}_k^-$ is the predictive value of state vector $x_k$, and $\hat{z}_k^-$ defined by $H\hat{x}_k^-$ is the estimation of measurement $z_k$. From the definition we can see that the $\varepsilon_k$ includes the new information from measurement $z_k$. From equation (9), if $\varepsilon_k = 0$, then the predictive value of state vector $x_k$ is the estimation value. That means the estimation at step $k$-1 is very accurate. In certain extend, the residual $\varepsilon_k$ symbolizes the filtering error. So the fitness function is designed as follows:

$$fitness = \frac{1}{N}\sum_{k=1}^{N} \varepsilon_k^T \varepsilon_k \quad (13)$$

where $N$ is the total step number of filtering. A smaller value of the fitness function indicates a more outstanding chromosome.

Step 4. Selection. According to the value of fitness function, rank the chromosomes by ascending order. Then copy the $P_s \times S$ chromosomes which are in the front of the population to replace the $P_s \times S$ ones which are in the back of the population, where $P_s$ is the probability of selection between 0 and 0.5, and $S$ is the size of population.

Step 5. Crossover. In this step, part-discrete crossover method is used to exchange the information between two chromosomes.

Step 6. Mutation. Non-uniform mutation method [16] is utilized in this step to increase the diversity of the population.

Step 7. Iteration. The real-coded GA repeats the Step 3 - 6 until the maximum number of iterations is reached.

The flow chart of using genetic algorithm to decide the initial value of parameter $R$ is illustrated in Fig. 1.

The parameters of the real-coded GA used in the simulation experiments in Section IV are set as follows:

(1) size of the population: 20
(2) maximum number of iterations: 300
(3) probability of selection: 0.2
(4) probability of crossover: 0.3
(5) probability of mutation: 0.2
(6) the range of the elements of matrix $R$: [0.04,400]

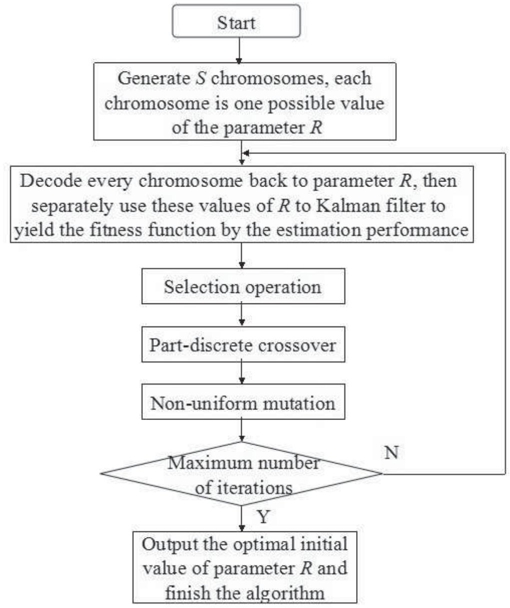


Fig. 1. The flow chart of using GA to decide the initial value of $R$

*B. Online Adjustment of $R$ by Fuzzy Inference System*

Fuzzy logic was first developed in the 1960s for representing uncertain and imprecise knowledge. It is an approximate but very effective method to describe the state of some systems which are too complex or not easily to be analyzed mathematically. In order to improve the estimation performance of the filter, we utilize a fuzzy inference system to carry out the online adjustment of the parameter $R$ in the HydGeFuzKF.

The fuzzy inference system used for the adjustment of $R$ is based on a adaptive approach called covariance-matching techniques [6]. Its basic idea is to make the actual value of the covariance of the residual $\varepsilon_k$ consistent with its theoretical value. From Kalman filter equations, the theoretical covariance of residual $\varepsilon_k$ is

$$C_t = HP_k^- H^T + R. \quad (14)$$

The actual covariance of residual $\varepsilon_k$ is approximated by its sample covariance

$$C_a = \frac{1}{M}\sum_{i=1}^{M} \varepsilon_i \varepsilon_i^T \quad (15)$$

where $M$ is the window size which is chosen empirically to give some statistical smoothing. When filtering, if the value of $R$ is accurate, $C_t$ and $C_a$ are basically the same. In this paper, we defined a variable called Degree of Matching ($DoM$) to indicate the degree of matching between $C_t$ and $C_a$. The variable $DoM$ is defined as:

$$DoM = \frac{trace(C_a)}{trace(C_t)} \quad (16)$$

where $trace(\bullet)$ represents calculating the trace of the matrix in the bracket.

In the HydGeFuzKF, assume matrix $R$ is a time-varying parameter and its value at step $k$ is obtained by

$$R_k = \alpha_k R_{k-1} \tag{17}$$

where $\alpha_k$ is the adjustment factor at step $k$ and $R_1$ is the initial value of measurement noise covariance matrix which is obtained by genetic algorithm.

Select $DoM$ as the input of the fuzzy inference system and $\alpha_k$ the output. According to the covariance-matching techniques, the actual value of the covariance of the residual $\varepsilon_k$ should be made to consistent with its theoretical value, viz., the value of $DoM$ should be basically equal to 1. Based on equation (14)-(17), we can meet the requirement of covariance-matching techniques by adjusting the value of $\alpha_k$. When $DoM$ is larger than 1, we need a bigger $R$ to increase the value of $C_t$, so the fuzzy inference system will output a $\alpha_k$ which is larger than 1. When $DoM$ is smaller than 1, we need a smaller $R$ to decrease the value of $C_t$, so the fuzzy inference system will output a $\alpha_k$ which is smaller than 1. When $DoM$ is equal to 1, the value of $R$ has no need to change, so $\alpha_k$ will be equal to 1.

The fuzzy set of the input variable $DoM$ is described by three linguistic variables, namely {Large1 (larger than 1), Equal1 (equal to 1) and Small1 (smaller than 1)}, so is the fuzzy set of the output variable $\alpha_k$. The rule base of the fuzzy inference system is described as follows:

1. IF ($DoM$ is Large1) THEN ($\alpha_k$ is Large1)
2. IF ($DoM$ is Equal1) THEN ($\alpha_k$ is Equal1)
3. IF ($DoM$ is Small1) THEN ($\alpha_k$ is Small1)

The membership functions for $DoM$ and $\alpha_k$ are shown in Fig. 2. Centroid method is used in the process of defuzzification.

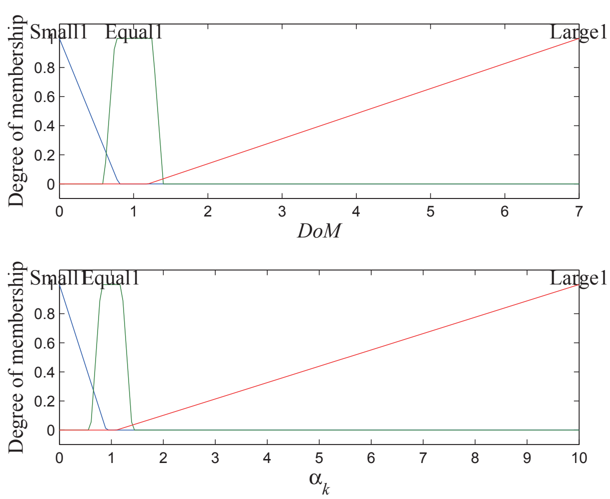


Fig. 2. Membership functions for $DoM$ and $\alpha_k$

The flow chart of using fuzzy inference system to adjust the value of $R$ online is illustrated in Fig. 3.

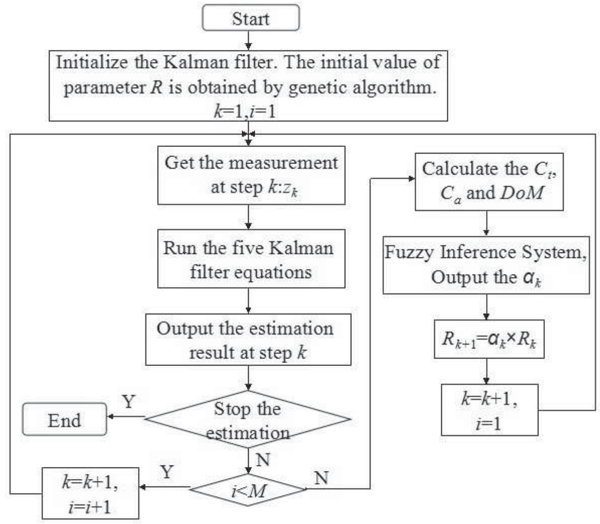


Fig. 3. The flow chart of using FIS to adjust the value of $R$ online

## IV. Simulation Experiment

In this section, simulation experiment has been carried out to compare the performances of the following five Kalman filter algorithms:

HydGeFuzKF: The new adaptive Kalman filter algorithm we proposed in this paper.

GeneticKF: In this algorithm, the initial value of $R$ is decided by genetic algorithm offline, but there is no any online adjustment.

FuzzyKF: In this algorithm, the value of $R$ is adjusted by fuzzy inference system online, but the initial value of $R$ is selected randomly.

RegularKF: The regular Kalman filter algorithm, in which the initial value of $R$ is selected randomly and there is no any online adjustment.

SageHusaKF: The Sage-Husa adaptive Kalman filter algorithm [3] based on maximum *a posteriori* probability estimation.

### *A. Simulation Model Based on Target Tracking*

The experiment is based on a simulation of rocket target tracking. Assume a rocket is doing uniformly accelerated motion escaping from the Earth. Its acceleration is $20m/s^2$. The fluctuations of engine thrust always cause some fluctuations of the acceleration. A radar on the ground is tracking the rocket and it gives the observation of the distance of the rocket from the ground every second. The observations are noisy. Now we need to estimate the displacement, velocity and acceleration of the rocket every second using Kalman filter.

So the system state at $k$-th second is $x_k = [d_k, v_k, a_k]^T$, where $d_k$ indicates the displacement of the rocket at $k$-th second, $v_k$ indicates the velocity and $a_k$ indicates the acceleration. According to the physical model of uniformly accelerated motion, the system model is

$$x_k = \begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \times x_{k-1} + w_{k-1} \tag{18}$$

where $w_{k-1}$ is zero-mean Gaussian white noise with covariance matrix $Q = diag([0, 0, 0.1])$ caused by engine thrust fluctuations. The measurement model is

$$z_k = [1 \quad 0 \quad 0] \times x_k + \nu_k \tag{19}$$

where $z_k$ is the observation of the rocket displacement at $k$-th second and $\nu_k$ is the measurement noise with covariance matrix $R$. The initial conditions in the simulation are assumed to be $x_0 = [1000, 50, 20]^T$, $\hat{x}_0 = [990, 0, 0]^T$ and the initial error covariance matrix is given by $P_0 = diag([30, 20, 10])$.

We need generate two sets of data by simulation for the experiment. The first set of data is used for genetic algorithm to decide the initial value of $R$. This set of data is simulated from 0 to 200s with the measurement noise covariance being four. The second set of data is used for estimating the state of the rocket. This set of data is simulated from 0 to 300s, and from $0 \sim 100$s the measurement noise covariance $R = 4$, from $100 \sim 150$s the covariance is $5R$, from $150 \sim 200$s the covariance is $10R$, from $200 \sim 250$s the covariance is $15R$ and from $250 \sim 300$s the covariance changes back to $R$.

### B. Experiment Results and Analysis

The comparison of GeneticKF and HydGeFuzKF is shown in Fig. 4, Fig. 5 and Fig. 6. In Fig. 4 the y-coordinate is the displacement estimation error defined by $\hat{d_k} - d_k$, in Fig. 5 the y-coordinate is the velocity estimation error defined by $\hat{v_k} - v_k$ and in Fig. 6 the y-coordinate is the acceleration estimation error defined by $\hat{a_k} - a_k$, where $\hat{d_k}$, $\hat{v_k}$ and $\hat{a_k}$ are the estimates of filter, $d_k$, $v_k$ and $a_k$ are the real state value of the rocket. In all the three figures, the x-coordinate represents the filter steps from $0 \sim 300$s. HydGeFuzKF and GeneticKF share a same optimized initial value of $R$, so the performances of the two filters at the beginning are almost the same. But HydGeFuzKF has much smaller error than GeneticKF when the value of $R$ changes significantly from $100 \sim 300$s, because of its online adjustment mechanism based on fuzzy inference system.

The comparison of FuzzyKF and HydGeFuzKF is shown in Fig. 7, Fig. 8 and Fig. 9. Both the two filters carry out online adjustment to the parameter $R$, so basically they have the same ability to meet the change of the value of $R$. However, HydGeFuzKF converges faster than FuzzyKF at the beginning of filter, which can be seen from the ellipse parts in the figures, because the initial value of $R$ of HydGeFuzKF is optimized by genetic algorithm.

The comparison of RegularKF and HydGeFuzKF is shown in Fig. 10, Fig. 11 and Fig. 12. RegularKF has neither the offline optimization of the initial value of $R$ nor the online adjustment to it, so HydGeFuzKF has not only faster convergence speed of filtering, but also greater capability to deal with the variability problem of the value of parameter $R$ than RegularKF.

The comparison results shown in Fig. 13, Fig. 14 and Fig. 15 have proven that HydGeFuzKF has better performance than SageHusaKF too.

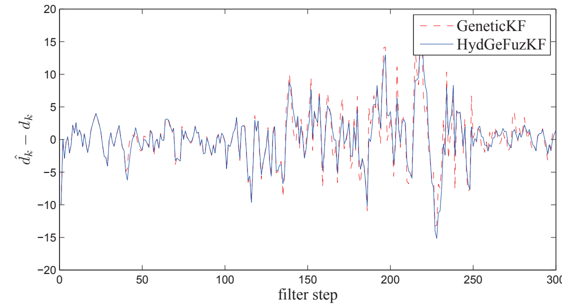


Fig. 4. Comparison of GeneticKF and HydGeFuzKF on the estimation of rocket displacement

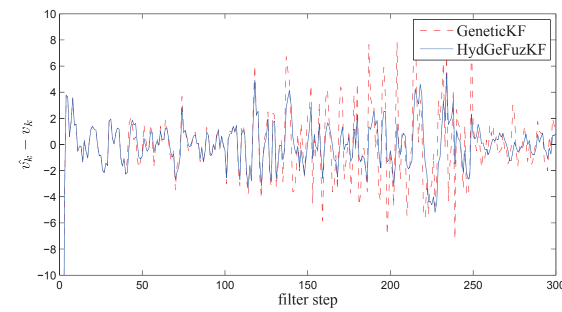


Fig. 5. Comparison of GeneticKF and HydGeFuzKF on the estimation of rocket velocity

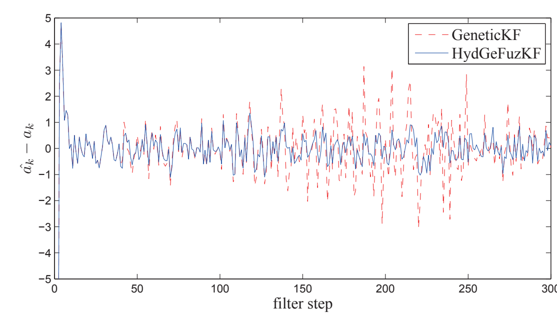


Fig. 6. Comparison of GeneticKF and HydGeFuzKF on the estimation of rocket acceleration

Each filter algorithm runs 20 times and calculate their root mean square error (RMSE) of filtering. The results are shown in Table. I respectively, where the smallest error each time is marked in bold. From Table. I we can see, the HydGeFuzKF we proposed got the smallest error 18 times. It has more capabilities to reduce the filtering error than RegularKF, SageHusaKF, FuzzyKF and GeneticKF.

TABLE I

COMPARISON OF FIVE KALMAN FILTER ALGORITHMS ON RMSE

| Run time | RegularKF | SageHusaKF | FuzzyKF | GeneticKF | HydGeFuzKF |
|---|---|---|---|---|---|
| 1 | 6.1956 | 6.4922 | **5.8032** | 6.3100 | 5.8066 |
| 2 | 6.5292 | 6.3861 | 6.5429 | 6.3109 | **5.8103** |
| 3 | 6.4670 | 6.3882 | 6.4553 | 6.3103 | **5.8065** |
| 4 | 6.5925 | 6.3847 | 6.6103 | 6.3104 | **5.8075** |
| 5 | 6.2136 | 6.4102 | 6.1444 | 6.3105 | **5.8080** |
| 6 | 6.3117 | 6.5036 | 5.8097 | 6.3103 | **5.8065** |
| 7 | 6.1677 | 6.4189 | 6.0800 | 6.3102 | **5.8065** |
| 8 | 6.9137 | 6.3855 | 6.9392 | 6.3103 | **5.8065** |
| 9 | 6.7531 | 6.3838 | 6.8055 | 6.3103 | **5.8065** |
| 10 | 6.2093 | 6.4109 | 6.1378 | 6.3106 | **5.8091** |
| 11 | 6.6767 | 6.3838 | 6.7035 | 6.3107 | **5.8093** |
| 12 | 6.4712 | 6.3880 | 6.4545 | 6.3092 | **5.8149** |
| 13 | 6.6304 | 6.5159 | **5.7624** | 6.3101 | 5.8065 |
| 14 | 6.3078 | 6.3985 | 6.2621 | 6.3106 | **5.8090** |
| 15 | 6.3916 | 6.5083 | 5.8565 | 6.3102 | **5.8065** |
| 16 | 6.7339 | 6.3837 | 6.7714 | 6.3103 | **5.8065** |
| 17 | 6.9741 | 6.3866 | 6.9997 | 6.3102 | **5.8065** |
| 18 | 6.6188 | 6.3843 | 6.6893 | 6.3099 | **5.8064** |
| 19 | 6.7284 | 6.3837 | 6.7642 | 6.3097 | **5.8064** |
| 20 | 6.5076 | 6.3867 | 6.5228 | 6.3102 | **5.8065** |
| mean | 6.5197 | 6.4142 | 6.4057 | 6.3102 | **5.8076** |

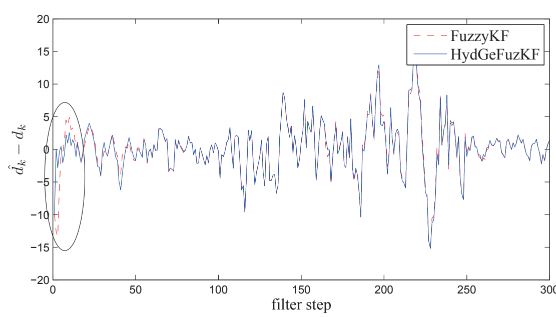


Fig. 7. Comparison of FuzzyKF and HydGeFuzKF on the estimation of rocket displacement

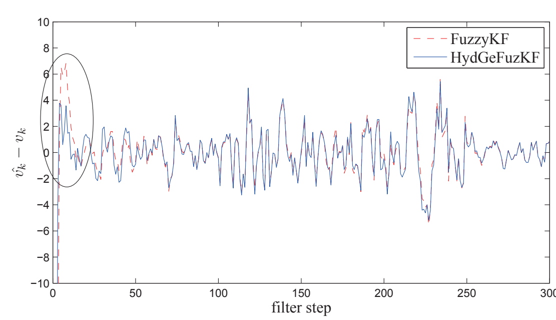


Fig. 8. Comparison of FuzzyKF and HydGeFuzKF on the estimation of rocket velocity

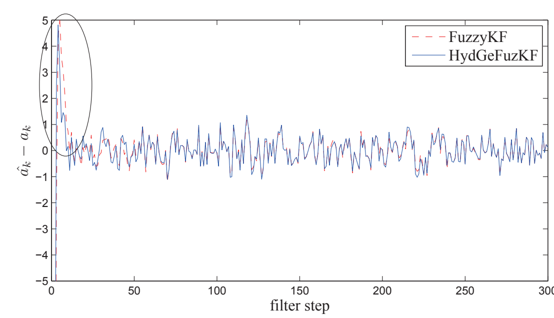


Fig. 9. Comparison of FuzzyKF and HydGeFuzKF on the estimation of rocket acceleration

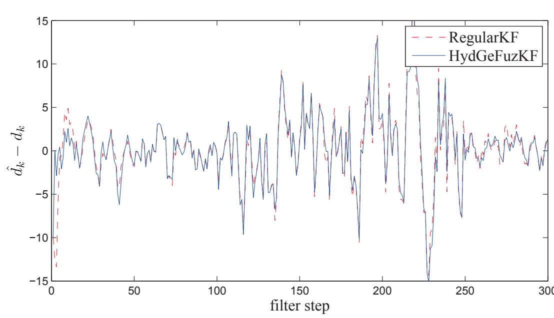


Fig. 10. Comparison of RegularKF and HydGeFuzKF on the estimation of rocket displacement

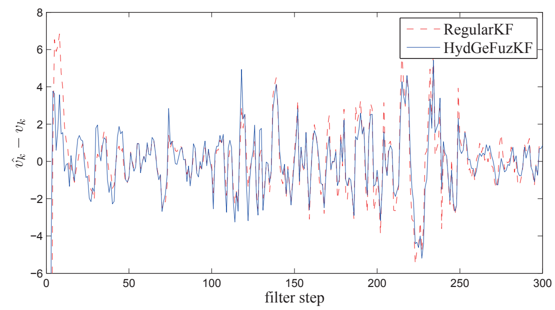


Fig. 11. Comparison of RegularKF and HydGeFuzKF on the estimation of rocket velocity

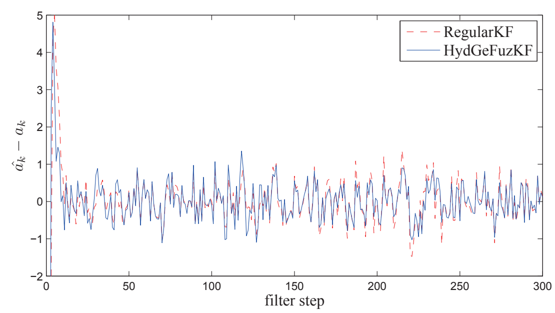


Fig. 12. Comparison of RegularKF and HydGeFuzKF on the estimation of rocket acceleration

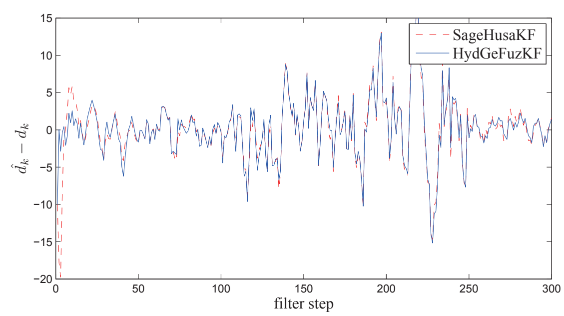


Fig. 13. Comparison of SageHusaKF and HydGeFuzKF on the estimation of rocket displacement

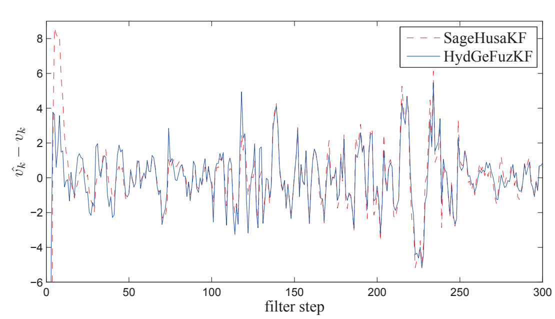


Fig. 14. Comparison of SageHusaKF and HydGeFuzKF on the estimation of rocket velocity

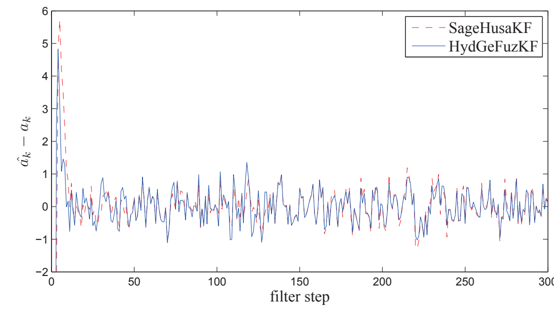


Fig. 15. Comparison of SageHusaKF and HydGeFuzKF on the estimation of rocket acceleration

In the simulation based on target tracking, the initial value of parameter $R$ was unknown, and its value changed four times in the tracking process. To solve this problem, HydGeFuzKF got the initial optimal value by genetic algorithm at first, and then kept adjusting this optimal value in real-time using fuzzy inference system. The combination of GA and FIS is the reason why HydGeFuzKF got the better results than another four Kalman filter algorithms.

Without a doubt, it will spend more time to run HydGeFuzKF than RegularKF. However, using offline optimization mode, it only takes a little time before the filter works to run the genetic algorithm, ensuring the real-time performance of Kalman filter when it is tracking the target. Fuzzy inference system is a very popular technique used in real-time systems to adjust the system parameters, because of its low requirement on calculation. Besides, the times that FIS is used can be changed by adjusting the parameter $M$ in equation (15). So generally, the HydGeFuzKF algorithm can satisfy the requirement of Kalman filter on real-time capability in practical applications.

## V. Conclusions

In this paper, a new adaptive Kalman filter by combining evolutionary algorithm and fuzzy inference system, namely HydGeFuzKF is presented. In this new algorithm, the initial value of the measurement noise covariance matrix $R$ is decided offline by evolutionary algorithm, and the optimal initial value of $R$ is adjusted online by fuzzy inference system to meet the changeable measurement noise. Simulation results indicate that the new filter algorithm we designed has more capabilities to reduce the filtering error. It has many potentials in practical applications.

## References

[1] R. Dhaouadi, N. Mohan, and L. Norum, "Design and implementation of an extended kalman filter for the state estimation of a permanent magnet synchronous motor," *IEEE Transactions on Power Electronics*, vol. 6, no. 3, pp. 491–497, 1991.

[2] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 477–482, 2000.

[3] M. Narasimhappa, P. Rangababu, S. L. Sabat, and J. Nayak, "A modified sage-husa adaptive kalman filter for denoising fiber optic gyroscope signal," in *2012 Annual IEEE India Conference (INDICON 2012)*, pp. 1266–1271, 2012.

[4] S. Sarkka and J. Hartikainen, "Non-linear noise adaptive kalman filtering via variational bayes," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2013)*, pp. 1–6, 2013.

[5] X. M. Qu, "Robust kalman track fusion in target tracking with uncertainties," in *International Conference on Computational Problem-Solving (ICCP 2011)*, pp. 1–4, 2011.

[6] R. Mehra, "Approaches to adaptive filtering," *IEEE Transactions on Automatic Control*, vol. 17, no. 5, pp. 693–698, 1972.

[7] K. Szabat and T. Orlowska-Kowalska, "Performance improvement of industrial drives with mechanical elasticity using nonlinear adaptive kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 3, pp. 1075–1084, 2008.

[8] N. Salvatore, A. Caponio, F. Neri, S. Stasi, and G. L. Cascella, "Optimization of delayed-state kalman-filter-based algorithm via differential evolution for sensorless control of induction motors," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 1, pp. 385–394, 2010.

[9] R. K. Jatoth, D. N. Rao, and K. S. Kumar, "Particle swarm optimization aided unscented kalman filter for ballistic target tracking," in *IEEE International Conference on Communication Control and Computing Technologies (ICCCCT 2010)*, pp. 455–460, 2010.

[10] M. Mosavi, M. Sadeghian, and S. Saeidi, "Increasing dgps navigation accuracy using kalman filter tuned by genetic algorithm," *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 246–252, 2011.

[11] J. Ali, "Strapdown inertial navigation system/astronavigation system data synthesis using innovation-based fuzzy adaptive kalman filtering," *IET Science, Measurement & Technology*, vol. 4, no. 5, pp. 246–255, 2010.

[12] D. J. Jwo and F. C. Chung, "Fuzzy adaptive unscented kalman filter for ultra-tight gps/ins integration," in *International Symposium on Computational Intelligence and Design (ISCID 2010)*, vol. 2, pp. 229–235, 2010.

[13] N. Yadaiah, T. Srikanth, and V. S. Rao, "Fuzzy kalman filter based trajectory estmation," in *11th International Conference on Hybrid Intelligent Systems (ICHIS 2011)*, pp. 566–571, 2011.

[14] E. Shi, "An improved real-time adaptive kalman filter for low-cost integrated gps/ins navigation," in *International Conference on Measurement, Information and Control (ICMIC 2012)*, vol. 2, pp. 1093–1098, 2012.

[15] B. Talel and B. H. Faycal, "Fuzzy kalman filter for non linear stochastic systems," in *10th International Multi-Conference on Systems, Signals & Devices (SSD 2013)*, pp. 1–7, 2013.

[16] Z. Michalewicz, *Genetic algorithms+ data structures= evolution programs*. springer, 1996.