

# Composite Data Fusion Algorithm for Miniature Vehicles Building Navigation Base in Formation Flying

Runle Du, Jiaqi Liu, Zhifeng Li, Zhenhong Niu, Zhiye Jiang  
National Key Laboratory of Science and Technology on Test Physics and  
Numerical Mathematics, Beijing, China (100076)  
Email: Jenniferdu@126.com

Yadong Yang  
China Academy of Launch Vehicle  
Technology  
Beijing, China (100076)

**Abstract**— When multiple miniature vehicles with individual position and inter-vehicle distance measurement ability collaborate in a formation, navigation base can be established by data fusion in a decentralized and standalone scheme. A Composite Data Fusion (CDF) algorithm which combines Least Square Error and Kalman Filtering is proposed in this paper to build navigation base with optimized computing stress. In CDF, Enhanced LSE is incorporated as the preprocessing stage to build a coarse estimation and handle temporary or permanent group number failure. KF stage is then built to further alleviate noises in the pre-processed estimations. In CDF, the dynamic model can be much simpler than KF, so the computation load is reduced while the result still has the advantage of high precision. Simulation results show that, when the fault rate of measurement in each vehicle goes 5 thousandth, the result is still acceptable. The computation time of the proposed method is less than three percent of that of KF, while its precision is almost the same to that of KF.

**Keywords**— composite data fusion; navigation base; cooperative navigation; Kalman Filtering; formation flying

## I. INTRODUCTION

Among various research interests related to formation flying, building navigation base is of great importance. Navigation base is the effective fused value of navigation data in formation flying, from which the feedback control signals to maintain formation structure can get better precision than individual navigation data.

The virtue of building navigation base through data fusion is the independence to external assistant. In article [1], Rajnikant presented a detailed cooperative navigation methodology for miniature air vehicles with low-weight low cost Inertial Measurement Unit (IMU) such that they can be operated without relying on Global Positioning System. Similar navigation data processing problem can be found in robots [2], sensor networks [3], spacecrafts [4] and underwater vehicles [5].

There are two fundamental requirements for navigation data fusion algorithms, namely flexibility and efficiency. To secure the robustness of the entire formation, decentralized scheme is analyzed in [6]. Covariance intersection also has the ability to handle uncertainty in measure and communication [7], with some overhead of computation. In covariance intersection algorithm

proposed here, measurement acquired by the relative position sensor and the estimation of the state received from predecessor spacecraft is distinguished, and treated separately. This algorithm addressed the correlation between fellow members and thus eliminated the drift that occurs when a standard Kalman Filter is used to estimate state of the whole formation. Multi agent method [8] is used to control and coordinate multiple satellites in a formation, which is declared to be robust and flexible. In the autonomous state estimation method from paper [9] robustness with respect to increasing formation dimensions and poor initial guess accuracy is investigated. Optimal formation reconfiguration from [10] not only focus on fuel consumption optimization, but also pointed out the potential of formation flying to expand functionality, distribute risk and to reduce overall cost. Yet in these published works, the flexibility problem is either just mentioned without a solution or tackled with coarse method that causes huge overhead.

Navigation data fusion algorithms are expected to optimize the computation burden. In [2][5][11-14], KF and adaptive KF are used to establish navigation base. Optimization is studied only to meet specific configuration. In article [12], distributed Kalman Filter was specially designed to handle large scale formation. Paper [13] tackled data fusion problem when the communication is under constraints. For these classes of data fusion, heavy computation is required due to the intrinsic of Kalman Filtering algorithm, rendering them not suitable for onboard computers.

To improve the efficiency of KF, article [15] used overlapping block-diagonal state space model to simplify the implementation of distributed Kalman Filter. However, this approach is in the implementation level, thus its improvement is insignificant.

Besides KF, other approaches had also been investigated in the literature. Multi-agent approach was discussed in paper [3] and [16]. Article [17] presented the parallel estimation problem in terms of the full system dynamics and illustrated both time-varying and time-invariant solutions to the estimator design problem. Article [18] posed the fusion problem as a convex optimization problem and addressed it. Approximate dynamic programming was employed in [19] to implement cooperative navigation for formation that consists of heterogeneous autonomous vehicles. Those approaches may or may not run with high efficiency, but the precision is not comparable to KF.

In this paper, to meet the flexibility and efficiency need, a Composite Data Fusion (CDF) algorithm to build navigation base without external assistant is proposed. In our scheme, each member of formation is equipped with inter-vehicle distance measurement instruments. Navigation data and the distance between vehicles are shared by inter-vehicle communication system. In the CDF algorithm, LSE is used as the first stage to build a coarse estimation and Kalman Filtering is used to alleviate noises from preprocessed results. In the LSE stage, enhancement is made to handle temporary or permanent member failure. In the KF stage, CDF uses much simpler state vector than Kalman Filtering, so its computation burden is significantly less than that of Kalman Filtering. The running speed of CDF is much higher than that of Kalman Filtering, but the CDF achieves the same navigation precision like KF does.

Rest of this paper is organized as follows. Section 2 formulates data fusion problem and section 3 discussed generic solution based on Kalman Filtering. Section 4 introduces Composite Data Fusion algorithm in detail. Section 5 gives theoretical analysis on quantity of computation. Experimental setup and results in section 6 will demonstrate the flexibility and efficiency. Finally, section 7 concludes this paper.

## II. NAVIGATION BASE PROBLEM

Multiple flying vehicles with individual position and inter-vehicle distance measurement ability can collaborate in establishing navigating base. Vehicles share their navigation data and distances between each vehicles pair.

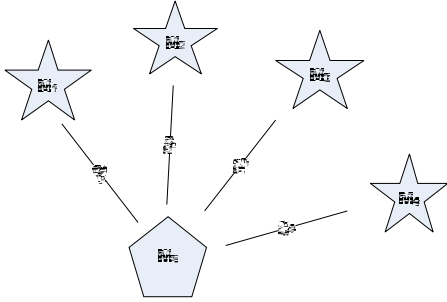


Fig. 1. Navigation base problem

As illustrated in Fig. 1, there are supposedly  $n+1$  miniature vehicles in the decentralized formation, namely  $M_0, M_1, \dots, M_n$ . Position is  $P(x, y, z)$ , and speed is  $V(V_x, V_y, V_z)$ , etc. When a certain vehicle, for example  $M_0$ , is under question, the distances to partner vehicles are  $d_1, d_2, \dots, d_n$ . Notably,  $M_0$  is marked with different icon in Fig. 1. Output of estimation position  $P(x,y,z)$  for vehicle  $M_0$  should subject to minimized estimation error.

## III. GENERIC SOLUTION

Kalman filtering is a signal processing algorithm to eliminate observation noise. In a decentralized scheme, when Kalman filter is used to estimate local state, the state vector must comprise the position and velocity of all vehicles [7, 8]. Direct observation data of distance to other vehicles and indirect observation data like position of peers need to be integrated into the state vector.

When referred in an inertial 3-D rectangular coordinates system, each of them can contribute 6 elements, position and speed, to the state vector of Kalman filter. So the state vector of Kalman filter is as follows, within which the subscript denotes these  $n+1$  vehicles.

$$X = \left\{ \left\{ x_0, y_0, z_0, V_{x0}, V_{y0}, V_{z0} \right\} \dots \left\{ x_n, y_n, z_n, V_{xn}, V_{yn}, V_{zn} \right\} \right\} \quad (3-1)$$

State equations are:

$$\dot{X} = \begin{cases} \dot{x}_0 = V_{x0}, \dot{y}_0 = V_{y0}, \dot{z}_0 = V_{z0} \\ \dot{V}_{x0} = a_{x0} + g_{x0} + \omega_{vxi} \\ \dot{V}_{y0} = a_{y0} + g_{y0} + \omega_{vyi} \\ \dot{V}_{z0} = a_{z0} + g_{z0} + \omega_{vzi} \\ \dots \\ \dot{x}_n = V_{xn}, \dot{y}_n = V_{yn}, \dot{z}_n = V_{zn} \\ \dot{V}_{xn} = a_{xn} + g_{xn} + \omega_{vxi} \\ \dot{V}_{yn} = a_{yn} + g_{yn} + \omega_{vyi} \\ \dot{V}_{zn} = a_{zn} + g_{zn} + \omega_{vzi} \end{cases} \quad (3-2)$$

In the equation above,  $a_{xi}, a_{yi}, a_{zi}$  are the measurements of onboard accelerometer.  $\omega_{vxi}, \omega_{vyi}, \omega_{vzi}$  are the dynamic noises of model.  $g_{xi}, g_{yi}, g_{zi}$  are the gravity vectors at current location.

Observation equation is built in geocentric inertial 3-D rectangular coordinates system. Observation of Kalman Filter consists of the positions of vehicles and distances between current vehicle and fellow vehicles.

$$Z = h(X) = \left\{ \begin{array}{l} \left\{ x_0 + v_{x0}, y_0 + v_{y0}, z_0 + v_{z0} \right\}, \\ \dots, \\ \left\{ x_n + v_{xn}, y_n + v_{yn}, z_n + v_{zn} \right\}, \\ \left\{ d_1 + v_{d1} \right\}, \\ \left\{ d_2 + v_{d2} \right\}, \\ \dots, \\ \left\{ d_n + v_{dn} \right\} \end{array} \right\} \quad (3-3)$$

where

$$d_i = \text{sqrt} \left( (x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 \right)$$

In the equation above,  $v_{xi}, v_{yi}, v_{zi}$  are the noises of individual navigation data.  $v_{di}$  is the error of distance measurement.

The nonlinear model can be derived as:

$$\begin{cases} \frac{dX(t)}{dt} = f(X(t), t) + W(t) \\ Z(t) = h(X(t), t) + V(t) \end{cases} \quad (3-4)$$

$$E[W_k W_j^T] = Q_k, E[V_k V_j^T] = R_k$$

Extended Kalman Filtering algorithm is as follows.

$$\begin{cases} \hat{X}_{k+1/k} = \hat{X}_{k/k} + f(\hat{X}_{k/k})\Delta t + \frac{1}{2} F f'(\hat{X}_{k/k})\Delta t^2 \\ \Phi_{k+1/k} = I + F\Delta t + \frac{1}{2} F^2 \Delta t^2 \\ P_{k+1/k} = \Phi_{k+1/k} P_{k/k} \Phi_{k+1/k}^T + Q_k \\ K_{k+1} = P_{k+1/k} H_{k+1}^T (H_{k+1} P_{k+1/k} H_{k+1}^T + R_{k+1})^{-1} \\ \hat{X}_{k+1/k+1} = \hat{X}_{k+1/k} + K_{k+1} [Z_{k+1} - h(\hat{X}_{k+1/k})] \\ P_{k+1/k+1} = (I - K_{k+1} H_{k+1}) P_{k+1/k} (I - K_{k+1} H_{k+1})^T + K_{k+1} R_{k+1} K_{k+1}^T \end{cases} \quad (3-5)$$

Within equation above,

$$F = \left. \frac{\partial f[X(t)]}{\partial X(t)} \right|_{X = \hat{X}_{k/k}}$$

$$H_{k+1} = \left. \frac{\partial h[X(t)]}{\partial X(t)} \right|_{X = \hat{X}_{k+1/k}}$$

#### IV. COMPOSITE DATA FUSION ALGORITHM

CDF is designed to have two stages. In the first stage LSE is used to build a coarse estimation and in the second stage Kalman Filtering is used to alleviate noises from preprocessed results. In the LSE stage, enhancement is made to handle temporary or permanent member failure. In the KF stage, CDF uses much simpler state vector than Kalman Filtering, so its computation burden is significantly less than that of Kalman Filtering. The running speed of CDF will be much higher than that of Kalman Filtering, but the CDF achieves the same navigation precision like KF does.

##### A. LSE stage

Distance from vehicle  $M_i$  to current vehicle  $M_0$  can be calculated by the following equation.

$$x^2 + y^2 + z^2 + x_i^2 + y_i^2 + z_i^2 - 2xx_i - 2yy_i - 2zz_i = d_i^2 \quad (4-1)$$

Specially, distance from  $M_0$  to itself is 0.

$$x^2 + y^2 + z^2 + x_0^2 + y_0^2 + z_0^2 - 2xx_0 - 2yy_0 - 2zz_0 = d_0^2 = 0 \quad (4-2)$$

Taking (4-2) away from (4-1), we can get a single linear equation about  $x$ ,  $y$  and  $z$ .

$$\begin{aligned} & x_i^2 + y_i^2 + z_i^2 - d_i^2 - (x_0^2 + y_0^2 + z_0^2) \\ & = 2x(x_i - x_0) + 2y(y_i - y_0) + 2z(z_i - z_0) \end{aligned} \quad (4-3)$$

Juxtapose all similar equations from all pairs relative to vehicle  $M_0$ , equation set about  $x$ ,  $y$  and  $z$  can be formed as:

$$\begin{bmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_i - x_0 & y_i - y_0 & z_i - z_0 \\ \dots & \dots & \dots \\ x_n - x_0 & y_n - y_0 & z_n - z_0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 - d_1^2 - (x_0^2 + y_0^2 + z_0^2) \\ x_i^2 + y_i^2 + z_i^2 - d_i^2 - (x_0^2 + y_0^2 + z_0^2) \\ \dots \\ x_n^2 + y_n^2 + z_n^2 - d_n^2 - (x_0^2 + y_0^2 + z_0^2) \end{bmatrix} \quad (4-4)$$

In (4-4), if number of cooperating vehicles,  $n > 3$ , the equation set is over-determined. Under this condition, LSE method can be applied to solve it.

$$\mathbf{x} = (\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{b}) \quad (4-5)$$

In (4-4), if  $n \leq 3$ , the equation is not over determined. This kind of degenerated equation need to be augmented before LSE can be applied. In (4-6), equation about vehicle  $M_0$  can be appended to build following equation set.

$$\begin{bmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ x_i - x_0 & y_i - y_0 & z_i - z_0 \\ \dots & \dots & \dots \\ x_n - x_0 & y_n - y_0 & z_n - z_0 \\ 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 - d_1^2 - (x_0^2 + y_0^2 + z_0^2) \\ x_i^2 + y_i^2 + z_i^2 - d_i^2 - (x_0^2 + y_0^2 + z_0^2) \\ \dots \\ x_n^2 + y_n^2 + z_n^2 - d_n^2 - (x_0^2 + y_0^2 + z_0^2) \\ 2x_0 \\ 2y_0 \\ 2z_0 \end{bmatrix} \quad (4-6)$$

After augmentation, the solvent is:

$$\mathbf{x} = ((\mathbf{A}^*)^T (\mathbf{A}^*))^{-1} ((\mathbf{A}^*)^T (\mathbf{b}^*)) \quad (4-7)$$

##### B. Enhanced LSE to handle vehicle failure and recover

LSE algorithm is stateless, which means it use only temporal observation data to calculate the estimation without caring about history observations. The change of input data dimension caused by member failure can be handled in LSE algorithm by modifying the equation set.

Vehicles can fail temporarily or permanently, either caused by communication or instrumentation error. Judgment on vehicle failure can be done by checking communication timeout or data validity verification. For instance, if data from  $M_i$  is detected to be invalid, the equation set of LSE need to be changed by setting corresponding value in matrix  $\mathbf{A}$  and vector  $\mathbf{b}$  to zero, as shown in (4-8).

$$\begin{bmatrix} x_1 - x_0 & y_1 - y_0 & z_1 - z_0 \\ 0 & 0 & 0 \\ \dots & \dots & \dots \\ x_n - x_0 & y_n - y_0 & z_n - z_0 \\ 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_1^2 + y_1^2 + z_1^2 - d_1^2 - (x_0^2 + y_0^2 + z_0^2) \\ 0 \\ \dots \\ x_n^2 + y_n^2 + z_n^2 - d_n^2 - (x_0^2 + y_0^2 + z_0^2) \\ 2x_0 \\ 2y_0 \\ 2z_0 \end{bmatrix} \quad (4-8)$$

Whenever the vehicle  $M_i$  recovered from its own problem or communication jam, its contribution can be turned on by simply insert the shared data into that slot.

From equation set 4-8, enhanced LSE is designed to be tolerable to member failure. When the total remaining valid vehicle is equal or bigger than 2, in equation set 4-8, the number of equations will be equal or bigger than 5, because local vehicles contributes 3 equations. Considering the number of unknown variables is 3, equation set 4-8 is still over-determined and thus guaranteed to be solvable.

Enhanced LSE will degenerate into local measurement only, if the number of remaining vehicle is 1. As shown in equation set 4-9, if all peer vehicles are down and the number of remaining vehicle is 1, it will degenerate into linear equation:

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2x_0 \\ 2y_0 \\ 2z_0 \end{bmatrix} \quad (4-9)$$

### C. Kalman Filtering Stage

Kalman Filtering stage is built on the output of LSE result. So, the model vector can be simplified as 6 elements: position and velocity of current vehicle, only.

$$X = \{x, y, z, V_x, V_y, V_z\}^T \quad (4-10)$$

State equation of Kalman Filtering stage is as follows.

$$\dot{X} = \begin{cases} \dot{x} = V_x \\ \dot{y} = V_y \\ \dot{z} = V_z \\ \dot{V}_x = a_x + g_x + \omega_{vx} \\ \dot{V}_y = a_y + g_y + \omega_{vy} \\ \dot{V}_z = a_z + g_z + \omega_{vz} \end{cases} \quad (4-11)$$

In the equation above,  $a_x, a_y, a_z$  are the measurements of onboard accelerometer.  $\omega_{vx}, \omega_{vy}, \omega_{vz}$  are the dynamic noises of model.  $g_x, g_y, g_z$  are the gravities at current location.

Observation equation is also built in geocentric inertial 3-D rectangular coordinates system. Observation of Kalman Filtering stage is only the position of vehicle under question.

$$Z = \begin{cases} x + v_x \\ y + v_y \\ z + v_z \end{cases} \quad (4-12)$$

In the equation above,  $v_x, v_y, v_z$  is the noise of LSE output. Kalman Filtering stage use nonlinear model as

$$\begin{cases} \frac{dX(t)}{dt} = f(X(t), t) + W(t) \\ Z(t) = H(X(t)) + V(t) \end{cases} \quad (4-13)$$

$$E[W_k W_j^T] = Q_k, E[V_k V_j^T] = R_k$$

## V. THEORETICAL ANALYSIS ON QUANTITY OF COMPUTATION

Computation in navigation base problem involved 5 types operations, namely matrix multiplication, matrix multiply vector, matrix addition or subtraction, vector addition or subtraction and matrix inversion. All major computations are list in table 1 for comparison.

In table 1, column 1 is the computation type. Following columns are organized into two groups, representing Composite Data Fusion and Kalman Filtering respectively. For each algorithm, scale and frequency of each type of computation inside one round of iteration are listed.

In each iteration, Composite Data Fusion and Kalman Filtering contain similar matrix calculations. But the matrix of Composite Data Fusion is much simpler than Kalman filtering, so the overall amount of computation is smaller. As can be seen in the table, the scale for Kalman Filtering is enlarged by a factor that is not less than  $n*n$ . The amount of computation of Kalman Filtering is significantly bigger than that from proposed CDF.

TABLE I COMPUTATION COMPARISON

Comp. Type	Composite Data Fusion				Kalman Filtering		
	L.S.E.		Kalman		Scale	Freq	
	Scale	Freq	Scale	Freq	Scale	Freq	
Matrix mul.	$3*n$	<b>1</b>	$6*6$	<b>4</b>	$6(n+1)*6(n+1)$	<b>5</b>	
	by $n*3$		by $6*6$		by $6(n+1)*6(n+1)$		
			$6*6$	<b>2</b>	$6(n+1)*6(n+1)$	<b>2</b>	
			by $6*3$		by $6(n+1)*(4n+3)$		
			$6*3$	<b>1</b>	$6(n+1)*(4n+3)$	<b>2</b>	
			by $3*3$		by $(4n+3)*(4n+3)$		
			$6*3$	<b>1</b>	$6(n+1)*(4n+3)$	<b>2</b>	
			by $3*6$		by $(4n+3)*6(n+1)$		
			$3*6$	<b>1</b>	$(4n+3)*6(n+1)$	<b>1</b>	
			by $6*6$		by $6(n+1)*6(n+1)$		
		$3*n$	<b>1</b>	$6*6$	<b>1</b>	$6(n+1)*6(n+1)$	<b>1</b>
		by $n$		by $6$		by $6(n+1)$	
Matrix add.sub.			$3*6$	<b>1</b>			
			by $6$				
			$6*3$	<b>1</b>	$6(n+1)*(4n+3)$	<b>1</b>	
			by $3$		by $(4n+3)$		
		$3*n$	<b>1</b>	$6*6$	<b>3</b>	$6(n+1)*6(n+1)$	<b>4</b>
				$3*3$	<b>1</b>	$(4n+3)*(4n+3)$	<b>1</b>
		$n$	<b>1</b>	$6$	<b>2</b>	$6(n+1)$	<b>2</b>
				$3$	<b>1</b>	$4n+3$	<b>1</b>
		$3*3$	<b>1</b>	$3*3$	<b>1</b>	$(4n+3)*(4n+3)$	<b>1</b>
Matrix inv.							

## VI. EXPERIMENT

Experimental formation flying is carried out on a simulation system implemented by C language under MS Windows.

### A. Experiment setting up

In the simulated scheme, 11 virtual exoatmospheric miniature vehicles are deployed in the formation. The individual positions are supposed to be provided by inertial measurement unit. Communication and distance measuring between vehicles are provided by radio frequency instruments. In a decentralized scheme, individual measurement is broadcasted to all fellow vehicles. Each vehicle using either Kalman Filter or CDF to estimate its local state, and uses shared navigation data to determine the relative position inside the formation.

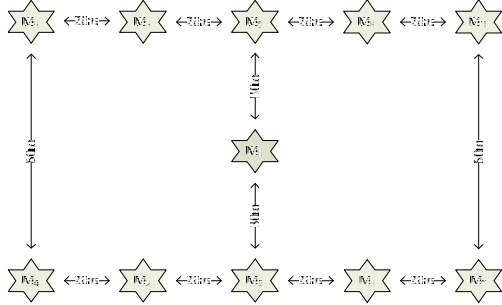


Fig. 2. Structure of simulated formation.

Formation structure and attitude are regulated by maintaining control algorithms. Fig. 2 is the projection of formation structure,

which is actually a 3-D structure. In Fig. 2,  $M_0$  is the investigated vehicle, whose navigation base will be build by Composite Data Fusion and Kalman Filtering algorithm.

Simulated flight lasted about 1400 seconds. Distances between vehicles are annotated in Fig. 2. Error of navigation data complies with Gauss distribution. Standard deviation of navigation data error is 1.0 meter. Noise of distance measurement also complies with Gauss distribution, with standard deviation as 0.001 meter.

To demonstrate the flexibility, a failure rate of measurement in each vehicle is set to be unrealistically high as 5 thousandth. Typical wireless communication in similar environment like WIFI, has error rate of only  $1E-8$  to  $1E-5$ . In the simulated environment, 5 out of thousand measurements or shared data package are invalid. Under this condition, CDF and Kalman Filter are used to estimate the state of current vehicle.

### B. Navigation base precision

Error of navigation base is the distance in 3-dimension space between estimated position and the real position that maintained by the simulation system. In fig. 3, black curve is the error of navigation data. Blue curve is the result from LSE. Red curve stands for Composite Data Fusion and green curve is from Kalman Filtering.

As can be seen from Figure 3, Kalman Filtering can effectively eliminate the observation noise and distance measurement error. LSE, due to its intrinsic deflection of amplifying input noise, the error level is the biggest. The proposed Composite Data Fusion algorithm takes the result of LSE as input. After filtering out the noise of LSE result, the error level matches Kalman Filtering.

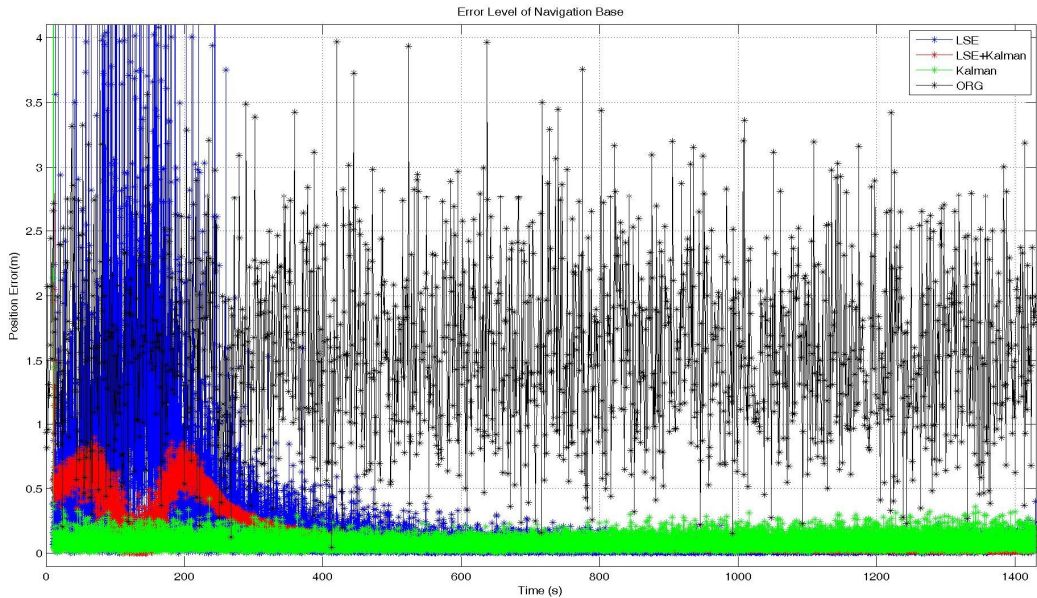


Fig. 3. Comparison of navigation base error from different algorithms

### C. Runtime comparison

To demonstrate the merit of shorter runtime of Composite Data Fusion, runtime for different algorithms is measured in the simulation system. Time measurement is implemented by the function of GetTickCount(). Since time measuring precision is limited to micro second, an indirect measuring method is applied in this paper. To get the runtime of an algorithm, two round of simulation is made. One is with target algorithm turned on, and the other with it turned off. The difference between these two times is the runtime of corresponding algorithm.

TABLE II  
RUNTIME COMPARISON BETWEEN KALMAN FILTERING  
AND COMPOSITE DATA FUSION

No.test	Kalman Filtering(ms)	Composite Data Fusion(ms)	Runtime Percentage%	Speedup Ratio
1	<b>2465689</b>	<b>2606</b>	<b>0.11</b>	<b>946.2</b>
2	<b>2450198</b>	<b>4555</b>	<b>0.19</b>	<b>537.9</b>
3	<b>2485798</b>	<b>5039</b>	<b>0.20</b>	<b>493.3</b>
4	<b>2458483</b>	<b>5693</b>	<b>0.23</b>	<b>431.8</b>
5	<b>2457405</b>	<b>7597</b>	<b>0.31</b>	<b>323.5</b>
6	<b>2455644</b>	<b>2449</b>	<b>0.10</b>	<b>1002.7</b>
7	<b>2466859</b>	<b>4788</b>	<b>0.19</b>	<b>515.2</b>
8	<b>2481898</b>	<b>3542</b>	<b>0.14</b>	<b>700.7</b>
9	<b>2473956</b>	<b>2669</b>	<b>0.11</b>	<b>926.9</b>
10	<b>2496592</b>	<b>3770</b>	<b>0.15</b>	<b>662.2</b>
Avr.	<b>2469252</b>	<b>4270</b>	<b>0.17</b>	<b>654.0</b>

In table 2, first column is number of tests. Column 2 and 3 show runtimes of Kalman filtering and Composite Data Fusion separately. Column 3 is the ratio of runtime from CDF to that from Kalman Filtering. Speed-up factor in column 4 is the result of runtime from Kalman Filtering divided by that from CDF. As illustrated in table 2, the runtime of Composite Data Fusion is significantly smaller than Kalman Filtering. On average only 0.17% of runtime is required by Composite Data Fusion, or acquire a speed up ratio of 654.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, a Composite Data Fusion algorithm is proposed to build navigation base. It employed LSE and Kalman Filtering as two stages. In each round of iteration, LSE is used as the preprocessing stage to build a coarse estimation and enhanced to handle unpredicted failure of measurement or communication. Kalman Filtering stage is then used to further eliminate noises from preprocessed estimations. By combination of these two algorithms to build CDF, member failure is handled and computation amount and precision requirement is mediated.

Simulation results confirmed that, proposed method uses only 0.17% of computation time of Kalman Filtering and achieved matching precision. When the failure rate of measurement in each

member is 5 thousandth, CDF still can yield satisfying estimation precision.

## REFERENCES

- [1] Rajnikant Sharma, Clark Taylor, Cooperative navigation of MAVs in GPS denied areas, in proceedings of IEEE international conference on multisensory fusion and integration for intelligent systems, Seoul, Korea, August 20-22, 2008.
- [2] Stergios I. Roumeliotis, George A. Bekey, Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots, in proceedings of the 2000 IEEE international conference on Robotics and automation, San Francisco, CA, April 2000.
- [3] Nicolas Singer, Jean-Marie Pecatte, Sylvie Trouilhet, the multi-agent cooperative navigation system Mawa: a model of dynamic knowledge specialization for a user-centric analyze of the Web, in proceedings of the 2005 international conference on computational intelligence for Modeling, Control and Automation, and international conference on Intelligent Agents, Web technologies and Internet commerce (CIMCA-IAWTIC'05).
- [4] Joseph Nicosia, Decentralized cooperative navigation for spacecraft.2006.
- [5] Darren K. Maczka, Aditya S. Gadre, Daniel J. Stilwell, Implementation of a cooperative navigation algorithm on a Platoon of autonomous underwater vehicles, 2007.
- [6] John L. Crassidis, Yang Cheng, Decentralized attitude estimation using a quaternion covariance intersection approach, AAS 08-284.
- [7] Pablo O. Arambel, Constantino Rago, Raman K. Mehra, Covariance intersection algorithm for distributed spacecraft state estimation, in proceedings of the American control conference, Arlington, VA June 25-27, 2001.
- [8] Hongwei Xia, Guangcheng Ma, Weinan Xie, Baomin Feng, Multiple satellite formation based on multi-agent, 2005.
- [9] Marco Sabatini, Fabrizio Reali, Giovanni B. Palmerini, Autonomous state estimation in formation flight, IEEE AC paper, 2006.
- [10] Jichao Tian, Naigang Cui, Rongjun Mu, Optimal formation reconfiguration using genetic algorithms, in proceedings of International conference on computer modeling and simulation, 2009.
- [11] Roy S. Smith, Fred Y. Hadaegh, Closed-loop dynamics of cooperative vehicle formations with parallel estimators and communication, IEEE transaction on automatic control, Vol. 52, No. 8, August 2007.
- [12] Usman A. Khan, Jose M. F. Moura, Distributing the Kalman filter for large scale systems, IEEE transactions on signal processing, Vol. 56, No. 10, October 2008.
- [13] Patricio Nebot, Enric Cervera, Cooperative navigation using the optical flow and time-to-contact techniques, in proceedings of 2008 10th International conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 17-20 December, 2008.
- [14] Pan Feng, Leader-Follower cooperative navigation with communication delays for multi AUVs, 2010.
- [15] S.M. Azizi, K. Khorasani, A sub-optimal distributed Kalman filter with fusion feedback for acyclic systems, in proceedings of Joint 48th IEEE conference on decision and control and 28th Chinese control conference, Shanghai, P.R. China, December 16-18, 2009.
- [16] Maria Carmela De Gennaro, Ali Jadbabaie, Formation control for a cooperative multi-agent system using decentralized navigation functions, in proceedings of the 2006 American control conference, Minneapolis, Minnesota, USA, June 14-16, 2006.
- [17] Roy S. Smith, Fred Y. Hadaegh, A distributed parallel estimation architecture for cooperative vehicle formation control, in proceedings of the 2006 American Control Conference, Minneapolis, Minnesota, USA, June 14-16, 2006.
- [18] Michael A. Enright, An analysis of advanced optimization algorithm for multiuser cooperative navigation, 2010.
- [19] Silvia Ferrari, Michael Anderson, Rafael Fierro, Wenjie Lu, Cooperative navigation for heterogeneous autonomous vehicles via approximate dynamic programming, in proceedings of 2011 5th IEEE conference on decision and control and European control conference (CDC-ECC), Orlando, FL, USA, December 12-15, 2011.