DISTRIBUTED ORBIT DETERMINATION VIA ESTIMATION CONSENSUS

Ran Dai, Unsik Lee, and Mehran Mesbahi

This paper proposes an optimal algorithm for distributed orbit determination by using discrete observations provided from multiple tracking stations in a connected network. The objective is to increase the precision of estimation by communication and cooperation between tracking stations in the network. We focus on the dynamical approach considering a simplified Low Earth Orbital Satellite model with random perturbations introduced in the observation data which is provided as ranges between tracking stations and satellite. The dual decomposition theory associated with the subgradient method is applied here to decompose the estimation task into a series of suboptimal problems and then solve them individually at each tracking station to achieve the global optimality.

INTRODUCTION

With the development of Global Positioning System (GPS), the power of this system brings immense benefit and convenience to today’s navigation world. However, without accurate information of the GPS satellite location, the users, who determine their position according to the relation to the satellite, will be totally lost. Generally, there are two classified approaches to determine the satellite orbit, the geometric approach and the dynamical approach. The geometric approach requires instruments with high accuracy to collect data in a high sampling rate and obtain the determined orbit purely relied on the collected data without considering the satellite motion. While the dynamical approach considers the two-body motion model assuming forces acting on the satellite. The well-known batch and sequential estimation algorithms are applied in the dynamical approach to estimate states or unknown parameters in the nonlinear system equations of the orbit determination problems. In this paper, we consider multiple tracking stations in a connected network to monitor the satellite locations in a cooperative way to improve the precision of orbit determination without centralized data collecting and processing.

The distributed estimation technology can be commonly seen in process control, signal processing and information systems. A subset of these efforts have generally been focused on the integration of measurements from all the sensors into a common estimate without centralized processor. For example, the well known information filter algorithm approaches the local estimate asynchronously and assimilate the innovation covariance in the inverse covariance form to simplify

---

1Research Associate, Department of Aeronautics and Astronautics, University of Washington, Box 352400, Seattle, WA, 98195-2400. Email: dairan@u.washington.edu
2Research Assistant, Department of Aeronautics and Astronautics, University of Washington Seattle, Box 352400, WA, 98195-2400. Email: unsik@u.washington.edu
3Professor, Department of Aeronautics and Astronautics, University of Washington, Seattle, Box 352400, WA, 98195-2400. Email: mesbahi@u.washington.edu
the resulting algebraic expressions. Nies et al. applied the data fusion strategy in distributed orbit determination where the individual sensor in the fully connected network will process its estimates before communicate to the other nodes. The summary of the centralized, decentralized and hierarchic approaches for states estimation of formation flying spacecrafts can be referred to Reference.

However, the consensus term between the neighbors of the sensor in the network has not been well examined in the information filter. Moreover, there is no guarantee that the common estimate is the optimal solution to minimize the overall mean squared estimation error under consensus constraints. Insight of consensus based distributed estimation can be found in the work of Reference and 11. By introducing the consensus condition as additional constraints in the objective function, the optimization problem is transformed to the problem of searching the minimal estimation error while satisfying the consensus constraints. applied the consensus strategy in overlapping estimator and designed a control law for vehicle formations based on the decentralized consensus estimation.

In order so solve the optimization problem with consensus constraints, the dual decomposition methodology and the associated subgradient method are introduced. Dual decomposition has been applied in many optimization problems where each component in the dual function is a strict convex/concave problem and can be solved independently. The subgradient method, initiated by Shor in 1970s, can then drive the dual function to a convergent solution by updating the dual variables which is related to the affine of the network in each iteration. has applied the dual decomposition in the distributed estimation where parts of the observation data are tracked by multiple sensors. Each suboptimal problem in their formulation is a concave log-likelihood function with consensus constraints for the multiple tracked data. In this paper, we implement the dual decomposition in the distributed orbit determination problem in the network focusing on the global optimality properties of the distributed estimation.

In the following, we will first introduce the formulation of distributed orbit determination problem in Section II and the conventional local estimation solution followed by the notation in Section III. Then we proceed to present the dual decomposition and subgradient method and its application in distributed orbit determination problems in Section IV. Simulation results will be provided in the final version.

PROBLEM FORMULATION

The problem in the paper is a two-dimensional, two-body problem under the assumption that there are no perturbations and the earth is non-rotating. The dynamical equations of the satellite in earth orbit are expressed as

\[
\dot{X} = F(X) = \begin{bmatrix}
\dot{x} = u \\
\dot{y} = v \\
\dot{u} = -\mu \frac{x}{(x^2 + y^2)^{3/2}} \\
\dot{v} = -\mu \frac{y}{(x^2 + y^2)^{3/2}}
\end{bmatrix}
\]  

(1)

where \((x, y)\) are the satellite position in the fixed Earth coordinate, \(u\) and \(v\) are the radial and transverse velocity with respect to the center of the Earth and \(\mu\) is the gravitational parameter. A set of \(m\) discrete observations are provided in the form of range measurement from \(n\) different tracking stations and labeled as \(R_{i,k}\), where \(i\) is the sequence of the observation points and \(k\) is the tracking station serial number. When the initial condition of the state variables in Eq. (1) is defined, we can
proceed to calculate the states at desired observation points by numerical integration. After that, the observational range between the satellite and the tracking stations \( k \) at observation points \( i \) are expressed as

\[ Y_{i,k} = G(X) = \sqrt{(x_i - x_{track k})^2 + (y_i - y_{track k})^2} + \epsilon_i, \quad k = 1, \ldots, n, \quad i = 1, \ldots, m \] (2)

where \( \sqrt{(x_i - x_{track k})^2 + (y_i - y_{track k})^2} \) corresponds to the actual range \( R_{i,k} \), \( (x_{track k}, y_{track k}) \) are the tracking station position and \( \epsilon_i \) is the observation error at point \( i \). Due to the view constraints, each tracking station can only obtain the ranges in sequence when the satellite stays in their observable area as illustrated in Figure 1. When the satellite leaves the first tracking station view sight, no range data will be provided from the first station and the next tracking site will have range data as long as the satellite enters its view scope. We denote the set of observable points for tracking station \( k \) as \( m_k \) and \( \sum_{k=1}^{n} m_k = m \). Our objective is to estimate the initial position \( (x_0, y_0) \) and velocity \( (u_0, v_0) \) of the satellite at \( t_0 \) to minimize the sum of the squares of the observation errors, such that the states of the satellite at any points thereafter can be propagated from the initial points by the dynamical equations. Mathematically, the objective function is written as

\[ J = \sum_{k=1}^{n} \sum_{i=1}^{m_k} (R_{i,k} - Y_{i,k})^2 \] (3)

This problem is analogous to a scenario that a group of people are watching one movie and each people can only record one section of the movie. After the movie is finished, people communicate with each other randomly and try to figure out the complete story of this whole movie. We assume the communication between the tracking stations is constructed by the topology of the graph \( G = (V, E) \) where stations are represented by the node set \( V \) with cardinality \( n \). If there is a communication channel between node \( v_k \) and \( v_h \), \( (k \neq h) \), then we assume the connection between \( v_k \) and \( v_h \) belong to the edge set \( E \) and denote it as \( \{v_k, v_h\} \in E \). At the same time, we define the \( n \times n \) adjacency matrix \( A \) with entries \( [A(G)]_{k,h} = 1 \) indicating the existence of communication
between node \( v_k \) and \( v_h \) and \([A(G)]_{k,h} = 0\) otherwise. Since the communication is assumed to be bidirectional, \( A \) is a symmetric matrix with \([A(G)]_{k,h} = [A(G)]_{h,k}\).

**CENTRALIZED ESTIMATION**

In the problem formulation, the measurement data collected is in time sequence and the estimate is the initial state before the starting of the observation. Therefore, batch filter is applied here to solve this type of historical observation and estimation problem. In batch Filter, nonlinear system equations are linearized at the reference points \( X^* \) by first-order Taylor expansion such that the corresponding nonlinear system can be approximated by

\[
\begin{align*}
\dot{X} &= \dot{X}^* + \left[ \frac{\partial F}{\partial X} \right]^* [X - X^*] + h.o.t \\
Y &= G(X^*) + \left[ \frac{\partial G}{\partial X} \right]^* [X - X^*] + h.o.t + \epsilon
\end{align*}
\]

(4)

where the partial differential matrices are defined as

\( A(t) = [\partial F / \partial X]^*, \quad \tilde{H}(t) = [\partial G / \partial X]^* \)

(5)

The deviation between the unknown states \( X \) and nominal states \( X^* \), named as \( x \), is in the linear expression format

\[
\dot{x} = A(t)x
\]

(6)

as well as the observation deviation \( y \)

\[
y = \tilde{H}(t)x + \epsilon
\]

(7)

By defining the state transition matrix \( \Phi(t_i, t_0) \), the state at a single epoch \( t_0 \) can be mapped to another epoch \( t_i \) using

\[
x_i = \Phi(t_i, t_0)x_0
\]

(8)

and the state transition matrix can be derived by integrating the following differential equation

\[
\dot{\Phi}(t, t_0) = A(t)\Phi(t, t_0), \quad \Phi(t_0, t_0) = I
\]

(9)

Substituting \( x \) in Eq. (7) by Eq. (8) can simplify Eq. (7) as

\[
y = Hx_0 + \epsilon
\]

(10)

Figure 2 demonstrates the data collection and calculation framework of centralized estimator. After collecting the ranges \( R_{k,i} \) from all tracking stations, the centralized processor obtained \( \tilde{H}_i \) for all observation points \( t_i \) and assemble them in one measurement sensitivity matrix \( H \) as

\[
H = \begin{bmatrix}
\tilde{H}_1\Phi(t_1, t_0) \\
\tilde{H}_2\Phi(t_2, t_0) \\
\vdots \\
\tilde{H}_i\Phi(t_i, t_0)
\end{bmatrix}
\]

(11)
The objective now is to find the best estimate of the state correction, which will minimize the sum of the squares of the observation errors $J(\hat{x}(t_0)) = \epsilon^T\epsilon$. By solving the least square problem, the best linear unbiased estimate is expressed as

$$\hat{x}(t_0) = (H^T H)^{-1} H^T y$$

(12)

The optimal state solution $\hat{X}(t_0)$ is then obtained by

$$\hat{X}(t_0) = X^*(t_0) + \hat{x}(t_0)$$

(13)

Figure 2. Data Collection and Framework of Centralized Estimator

**CONSENSUS BASED DISTRIBUTED ESTIMATOR**

In this section, we will proceed with the distributed estimation with consensus. We will firstly look into the framework of the distributed estimator and then introduce the dual decomposition and subgradient method which is applicable in solving this type of decomposable concave subproblems with constraints. The detailed algorithm is summarized in the following subsection.

**Framework**

In a connected tracking station network systems as shown in Figure 3, we assume each station can communicate information with its neighbor stations. There is no central processor and the network is not fully connected. However, as long as there is connection which is defined by the entries of the adjacent matrix $A(G)$ between any two nodes in the network, the two connected stations can communicate information with each other. Furthermore, they are able to spread the information among the connected network finally. In such system, the information filter or some other data fusion algorithm that require fully connected network cannot be applied in such partially connected system. In addition, the fully connected network requires large data communication and storage which may bring difficulty for implementation with the increase of data numbers. Furthermore, without consensus constraints, there is no limitation to converge the final result to the average consensus.

For a network system described in Figure 3, the index of neighbors for station $k$ is determined by the entries $A(G)_{k,h}, \ h = 1, \ldots, n, \ h \neq k$. If $A_{k,h} = 1$, then station $k$ can communicate
with station \(h\) and we will expect the estimates obtained from sensor \(k\), \(\hat{x}_k(t_0)\), is identical to \(\hat{x}_h(t_0)\) as well as the other estimates from the connected neighbors. By passing the identity request from one node to the other in the network, we can transfer the consensus request in the system. With the neighborhood consensus constraints on the estimates, we have the following relationships:

\[
a_{k,h} \hat{x}_k(t_0) - a_{k,h} \hat{x}_h(t_0) = 0, \quad k = 1, \ldots, n, \quad k > h
\]  

(14)

where \(a_{k,h}\) denotes the element \(A_{k,h}\) in matrix \(A\). If there is connection between station \(k\) and \(h\), the consensus condition expressed in Eq. (14) will have \(\hat{x}_k(t_0) = \hat{x}_h(t_0)\), otherwise such consensus constraint between node \(k\) and \(h\) does not exist. Since matrix \(A\) is symmetric, we will have \(a_{k,h} = a_{h,k}\). Therefore, \(\hat{x}_k(t_0) = \hat{x}_h(t_0)\) is equivalent to \(\hat{x}_h(t_0) = \hat{x}_k(t_0)\). In order to avoid repeating of consensus constraint, only upper triangular elements of \(A\) is included in Eq. (14). That is how the \(k > h\) term comes from.

As no central processor is provided in the distributed framework, each station will propagate the nominal states from \(t_0\) to their corresponding observation point \(t_i\) and calculate \(H_{k,i}\) independently. The objective function in Eq. (3) is rewritten as

\[
J = \sum_{k=1}^{n} \sum_{i=1}^{m_k} (R_{i,k} - Y_{i,k})^2
= \sum_{k=1}^{n} \sum_{i=1}^{m_k} (y_{i,k} - H_{i,k} \hat{x}_k(t_0))^T (y_{i,k} - H_{i,k} \hat{x}_k(t_0))
\]  

(15)

We assign \(y_k = [y_{1,k}, \ldots, y_{m_k,k}]^T\) and \(H_k = [H_{1,k}, \ldots, H_{m_k,k}]^T\), for all the \(y_{i,k}\) and \(H_{i,k}\), \(i = 1, \ldots, m_k\) obtained at station \(k\), respectively. The consensus relationships specified in Eq. (14) are additional constraints when solving the Batch filtering problem. By introducing the Lagrangian multipliers \(\lambda_{k,h}\) for the first order equality constraints in the objective function, the Lagrangian function of Eq. (15) is expressed as

\[
L = \sum_{k=1}^{n} (y_k - H_k \hat{x}_k(t_0))^T (y_k - H_k \hat{x}_k(t_0)) + \sum_{k=1}^{n} \sum_{h=k+1}^{n} \lambda_{k,h} [a_{k,h} \hat{x}_k(t_0) - a_{k,h} \hat{x}_h(t_0)].
\]  

(16)

Before we proceed to solve this typical problem, we will review the contents of dual decomposition and subgradient method first.
Dual Decomposition and Subgradient Method

For a nonlinear optimization problem of minimizing the objective function $f(x)$ under equality constraints $h_i(x) = 0 (i=1, \ldots, p)$ and inequality constraints $g_i(x) \leq 0 (i=1, \ldots, m)$, its Lagrangian has the form:

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{p} \mu_i h_i(x) + \sum_{i=1}^{m} \lambda_i g_i(x). \quad (17)$$

If the objective and constraint functions can be expressed in the summation form as

$$f(x) = \sum_{k=1}^{K} f_k(x^k), \quad g_i(x) = \sum_{k=1}^{K} g_i^k(x^k), \quad h_i(x) = \sum_{k=1}^{K} h_i^k(x^k), \quad \quad (18)$$

by partition the state vector $x$ into subvectors $x = (x^1, \ldots, x^K)$, then the Lagrangian is reformulated as

$$L(x, \lambda, \mu) = \sum_{k=1}^{K} (f_k(x^k) + \sum_{i=1}^{p} \mu_i h_i^k(x^k) + \sum_{i=1}^{m} \lambda_i g_i^k(x^k)). \quad (19)$$

The above function can be decomposed into $K$ subproblems according to the subvector $x^k$, where $k=1, \ldots, K$. For each subproblem, it can be solved by minimizing the dual function

$$L^D_k(\lambda, \mu) = \min_{x^k} (f_k(x^k) + \sum_{i=1}^{p} \mu_i h_i^k(x^k) + \sum_{i=1}^{m} \lambda_i g_i^k(x^k)) \quad (20)$$

for a given pair of multipliers $(\lambda^k, \mu^k)$. The dual function defined as

$$L^D_k(\lambda, \mu) = \inf_x L(x, \lambda, \mu) \quad (21)$$

is always concave, thus the dual problem of the summation of the subproblems

$$\max_{(\lambda, \mu)} L^D_K(\lambda, \mu) \quad (22)$$

is a convex optimization problem and can be solved by the subgradient method.

The subgradient method is an iterative procedure to gradually process the optimization solution by finding the ascent direction for the dual problem. At each sequence $j$, assuming the multipliers $(\lambda_j, \mu_j)$ are given, the subgradient at this point is expressed as

$$d_j = \begin{pmatrix} g(x_j) \\ h(x_j) \end{pmatrix}. \quad (23)$$

Then the multipliers are updated as follows:

$$\lambda_{j+1} = \max(0, \lambda_j + \alpha_j g(x_j))$$
$$\mu_{j+1} = \mu_j + \alpha_j h(x_j)) \quad (24)$$

where $\alpha_j$ is the step size that will control the convergent speed of the subgradient method. The maximum number of sequence $j$ is generally defined to satisfy the stopping criteria of iteration.

The new algorithm proposed in this paper is expected to build a framework to search for a more precise optimal solution by communication between tracking stations without central processor. In the following, we will explain how to solve distributed orbit determination under estimate consensus using dual decomposition method.
Application in Distributed Orbit Determination

As we introduced above, at epoch \( i \), the Lagrangian function for the distributed batch filtering problem under consensus is represented by Eq. (16). By introducing the linear model in Batch filtering algorithm and sorting \( x_k(t_0) \) in Eq. (16), the Lagrangian function can be rewritten as

\[
L = \sum_{k=1}^{n} \left[ (y_k - H_k \hat{x}_k(t_0))^T (y_k - H_k \hat{x}_k(t_0)) + \sum_{h=k+1}^{n} \lambda_{k,h} a_{k,h} \hat{x}_k(t_0) \right] - \sum_{k=1}^{n} \sum_{h=k+1}^{n} \lambda_{k,h} a_{k,h} \hat{x}_k(t_0) \tag{25}
\]

From the above reformatted expression, it is easy to tell that Eq. (25) is composed of \( n \) subproblems which can be summarized as

\[
L_k = \min_{\hat{x}_k(t_0)} \left[ (y_k - H_k \hat{x}_k(t_0))^T (y_k - H_k \hat{x}_k(t_0)) + \sum_{h=k+1}^{n} \lambda_{k,h} a_{k,h} \hat{x}_k(t_0) \right] k = 1, \ldots, n. \tag{26}
\]

For a given Lagrangian multiplier vector \( \lambda_{k,h} \), the above subproblems are independent of each other with its own estimates and observations, therefore they can be solved individually. After a careful comparison of these subproblems with the local Batch filtering formulation in Eq. (3), we found that they are very similar to each other except the additional term \((\sum_{h=k+1}^{n} \lambda_{k,h} a_{k,h} - \sum_{h=1}^{k-1} \lambda_{h,k} a_{h,k}) \hat{x}_k(t_0)\). However, the estimates need to be recalculated to obtain the new optimal solution for each subproblem with the additional term in the original objective function. In order to make \( L_k \) maximum, the estimates \( \hat{x}_k(t_0) \) are updated as

\[
\hat{x}_k(t_0) = (H_k^T H_k)^{-1} (H_k^T y_k - \sum_{h=k+1}^{n} \lambda_{k,h} a_{k,h} + \sum_{h=1}^{k-1} \lambda_{h,k} a_{h,k}), \quad k = 1, \ldots, n \tag{27}
\]

The above estimation can be performed by each station providing the lagrangian multiplier vector and its local observations. Comparing the local estimates of distributed Batch filtering using dual decomposition in Eq. (27) with the estimates of local Batch filtering in Eq. (12), the additional term in Eq. (27) comes from the lagrangian multipliers. This extra term works as a correction to the local batch filtering estimates to adjust them to a uniform solution. Using the subgradient method, the lagrangian multipliers are updated by

\[
\lambda_{k,h}^{j+1} = \lambda_{k,h}^j + \alpha_j a_{k,h} (\hat{x}_k(t_0) - \hat{x}_h(t_0)), \quad k = 1, \ldots, n, \quad h = k + 1, \ldots, n \tag{28}
\]

The estimates and lagrangian multiplier vectors for the neighborhood sensors are coupled with each other. The iterations are repeated procedure of adjustment and comparison. The adjustment will add compensation to estimates under average and at the same time counteract the extra part of estimates above average. The adjusted estimates from neighborhood sensors are compared in the subgradient method to find the new adjustment in the next loop. Ideally, the difference between the estimates are getting smaller with the increasing number of iteration. When the estimates are convergent to each other in the network, we can expect \( \lambda^{j+1} = \lambda^j \). The stopping criteria for the convergent solution can use a fixed maximum iteration number \( j_{\text{max}} \) and make \( j \leq j_{\text{max}} \) or set the sum of least square difference \( \sqrt{\sum \left( \hat{x}_k(t_0) - \hat{x}_h(t_0) \right)^2} \) less than a threshold, \( \epsilon \). In each iteration, the local batch filter will calculate it’s own estimates based on the new optimal formulation. Then they exchange this information with its neighbor sensors to obtain the updated multipliers in the correction term. Different from the information filter that requires the sensors to exchange large
information, for example, information matrix, observations, estimates, et al., as the public variables, the new algorithm only requires information of estimates and lagrangian multipliers. Furthermore, the local estimation can be carried out in parallel calculation by individual processor, which will save the overall calculation time. Finally, we summarized the algorithm in the following protocol chart:

```plaintext
for iter ← 1 to iter_max do
    Initialization: X = X*, λ = 0;
    for k ← 1 to number of all tracking stations do
        Calculate y_k, H_k at each tracking station k with current X;
    end
    for j ← 1 to j_max do
        for k ← 1 to number of all tracking stations do
            \[ \hat{x}_k(t_0) = (H_k^T H_k)^{-1}(H_k^T y_k - \sum_{h=k+1}^n \lambda_{k,h} a_{k,h} + \sum_{h=1}^{k-1} \lambda_{h,k} a_{h,k}); \]
        end
        for k ← 1 to number of all tracking stations do
            for h ← k + 1 to number of all tracking stations do
                \[ \lambda_{j+1}^{k,h} = \lambda_{j}^{k,h} + \alpha_j a_{k,h}(\hat{x}_k(t_0) - \hat{x}_h(t_0)); \]
            end
        end
        Update: \[ \hat{x}(t_0) = \frac{1}{n} \sum_{k=1}^n \hat{x}_k(t_0), \; X^* = X + \hat{x}(t_0); \]
    end
end
```

Algorithm 1: Decentralized Batch Filtering Algorithm with Estimates Consensus using Dual Decomposition and Subgradient Method

SIMULATION EXAMPLE

In this section, we will apply the distributed estimation framework in a real scenario orbital determination problem where observation of ranges are provided by two tracking stations over time period \([0, 3105]\) seconds. Due to the view constraints, the first tracking station observe ranges from 0 to 1320 seconds by every 15 seconds. After that, the satellite go to a area with no observation data. From 1680 to 3105 seconds, the satellite enters the view zone of the second tracking station which will provide range data also by every 15 seconds. The observed range data is demonstrated in Figure 4 with respect to time. The objective is to estimate the complete trajectory for the satellite during the whole time period \([0, 3105]\) seconds.

In order to estimate the complete trajectory during the specified period, we select \(t_0 = 1510\) which is the middle time point between the ending of observation from station 1 and starting of observation from station 2 and then try to solve the consensus based estimates at \(t_0\) first. States at any other time points can be propagated forward or backward from \(t_0\) by the dynamical integration based on the achieved estimation vector. Since states at \(t_0\) is after the observation points of station 1, we integrate states and corresponding elements in transition matrix backward from \(t_0\) to the observation points. While for station 2, we integrate the same set of variables forward to the observation points of station 2. We follow the algorithm proposed in Protocol 1 and get \(X(t_0)\) from centralized estimator, single tracking station estimator and distributed estimator with consensus listed in Table 1.
Figure 4. Time History of Observed Ranges by Tracking Station 1 and 2

Table 1. State Estimates at $t_0$ using Different Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>$x(t_0)(m)$</th>
<th>$y(t_0)(m)$</th>
<th>$v_x(t_0)(m/s)$</th>
<th>$v_y(t_0)(m/s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized Estimator</td>
<td>-2897.69411735</td>
<td>7182144.01656091</td>
<td>-7449.82637504</td>
<td>369.48640491</td>
</tr>
<tr>
<td>Local Estimator of Station 1</td>
<td>-2907.16521028</td>
<td>7182133.59162168</td>
<td>-7449.83580531</td>
<td>369.46778527</td>
</tr>
<tr>
<td>Local Estimator of Station 2</td>
<td>-2894.92052376</td>
<td>7182140.07737831</td>
<td>-7449.82938718</td>
<td>369.49360532</td>
</tr>
<tr>
<td>Distributed Estimator with Consensus</td>
<td>-2897.78458401</td>
<td>7182142.81635334</td>
<td>-7449.82675938</td>
<td>369.48644157</td>
</tr>
</tbody>
</table>

After comparison of the results calculated by different methods in Table 1, it is obvious to see that the estimates from distributed estimator with consensus constraints are more close to the estimates from the centralized estimator. For example, the position deviation between the centralized estimates and the distributed estimates is controlled under 2 meters, while the deviation between the centralized estimates and single estimator may reach 10 meters. In the next step, we propagate the four groups of states from $t_0$ to all observation points during $[0, 3105]$ and generate the time history of observation error respectively as illustrated in Figure 5. Results propagated from individual estimates will overlap the observation error processed by centralized estimator only within their observable time zone. It will diverge when the propagation is beyond their specific time zone. However, the results propagated by the distributed estimates with consensus coincide with the centralized one over the whole time period.

CONCLUSION

In this paper a novel way to view distributed batch filtering algorithm using dual decomposition and subgradient method has been proposed. The distributed batch filtering was first formulated as optimization problem with constraints. The distributed dual-decomposition algorithm then approaches the optimal solution with consensus in an iterative manner. The new algorithm was applied in the distributed orbit determination problem, which yields result close to the centralized estimates and has better performance than estimates from single tracking station. In the future, more tracking stations in a complex network will be tested for the distributed orbit determination problem.
Furthermore, application of the proposed algorithm in other areas will be considered.

**APPENDIX: PARTIAL DIFFERENTIAL AND TRANSITION MATRICES**

The partial differential matrix for dynamics Eq. (1) is:

$$
A = \left[ \frac{\partial F}{\partial X} \right]^* = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
-\mu(y^2-2x^2)/(x^2+y^2)^{7/2} & -3\mu xy/(x^2+y^2)^{7/2} & 0 & 0 \\
-3\mu xy/(x^2+y^2)^{7/2} & -3\mu(x^2-2y^2)/(x^2+y^2)^{7/2} & 0 & 0
\end{bmatrix}.
$$

The time derivative of the transition matrix is given as

$$
\dot{\Phi} = A\Phi
$$

where $A_{ij}$ denotes the $i,j$ component of the matrix $A$.

**REFERENCES**


