Asynchronous Track-to-Track Fusion by Direct Estimation of Time of Sample in Sensor Networks

Hadi Talebi and A. M. Afshin Hemmatyar

***Abstract*— Asynchronous data fusion is inevitable in track-to-track fusion for tracking high-speed targets. For low-speed targets, e.g., the movement of clouds, synchronization is insignificant and, depending on the application, may be disregarded. Real-time asynchronous fusion is a demanding task in sensor networks when the sensors are not synchronous in sampling-rate or in sampling-phase. In the method proposed in this paper, an estimator in the fusion center estimates the actual time of the sample with respect to the time-reference of the fusion center upon receiving the data from a sensor. Then, the computer of the fusion center uses predictions to transfer all the received data to the data corresponding to the start of the next fusion period. This process synchronizes the data, which is necessary for real-time uncorrelated track-to-track fusion. Finally, the pseudo-synchronized data of all the sensors are fused with an element-wise linear minimum variance unbiased estimator algorithm before the start of the next fusion period. Simulation and comparison with some benchmark algorithms are demonstrated to verify the effectiveness of the proposed algorithm.**

***Index Terms*— Asynchronous data fusion, sensor networks, real-time track-to-track fusion, high speed tracking, Kalman filter, linear minimum variance unbiased estimator.**

I. Introduction

SYNERGETIC use of different sensors in a sensor network helps to improve the quality of sensor-based tasks such as object tracking. Data fusion is a standard tool for reducing tracking errors in sensor networks. It also increases tolerance to missing data. If there are any corrupted data in a period - due to channel noise for example - it can be regarded as missing data, and it will be put aside from the fusion process in that period.

In general, only simultaneous sampled data can straightforwardly be fused in dynamic systems. If the samples are non-simultaneous the cost of fusion, regardless of the fusion algorithm, will be increased. In other words if the corresponding samples of the sensors are synchronous, they can be fused, otherwise because of the common process noise the samples are correlated, and they cannot be simply fused. Synchronization becomes more significant at higher speeds of the target. For instance, the maximum displacement of a target with a speed of 100 km/h is 2.78 m in 0.1 s, i.e. 0.1 s difference between the times of the samples results in 2.78 m difference between the samples of the two sensors, even with zero measurement noise. If the speed increases to 1000 km/h, the displacement is 27.78 m in 0.1 s.

To be more precise, one important problem in asynchronous track-to-track fusion is the dependency between the tracks - or state estimates - of different sensors in the fusion center. There are two reasons for dependency between state estimates of two sensors [1]–[3]. The first reason is data redundancy, which also known as common prior information. This happens when a state estimate from one sensor reaches the fusion center more than once, and contributes in a single fusion as separate state estimates. This may happen in data-centric [4], [5] sensor networks. The second reason is common process noise. In a simple case when the target is stationary and sensors sample the target position, and local estimators send the state estimates to the fusion center, there is no correlation between the state estimates in synchronous or asynchronous data fusion. If the target moves, as a dynamic phenomenon, the acceleration appears as the common process noise in the models. If sensors sample synchronously and send the state estimates to the fusion center after each sampling, the related state estimates in the fusion center are uncorrelated [3], [6]. If the samples are asynchronous and there is no synchronization between tracks, the related state estimates in the fusion center are correlated. This correlation is the result of the asynchronous sampling. In other words it represents the variations between samples related to different times. These differences are also known as the correlation between tracks and in some works are modeled as the bias of measurement in asynchronous fusion [7], [8]. Canceling biases, canceling correlation related to common process noise, and synchronizing the data are different ideas for zeroing or decreasing correlation between tracks for an independent-track fusion. Finally, in a distributed network the synchronized tracks can be considered independent.

Another problem in asynchronous track-to-track fusion is real-time tracking. In a sensor network, usually there is a fusion center in which the data of different sensors are fused. Even under synchronous sampling the result of the fusion is ready after receiving the last sample and some attention is required to be paid not to jeopardize the real-time performance of the tracking.

A general review of some asynchronous data fusion algorithms for tracking is available in [9].

Manuscript received July 8, 2013; revised August 15, 2013; accepted September 2, 2013. Date of publication September 11, 2013; date of current version November 1, 2013. The associate editor coordinating the review of this paper and approving it for publication was Dr. Amitava Chatterjee.

The authors are with the Department of Computer Engineering, Sharif University of Technology, Tehran 1458889694, Iran (e-mail: htalebi@ce.sharif.edu; hemmatyar@sharif.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSEN.2013.2281394

Many researches conducted on asynchronous tracking, especially those which use "multi-sensor" word, implicitly assume a common clock. These works are about a collection of sensors or a network of sensors which are very close to each other and to the fusion center, e.g. sensors in an airplane or an industrial plant. A common clock for pseudo-synchronization is used in these works [10]–[16].

Some researchers assume a constant or known time delay between each sensor and the fusion center [17], [18], thus pseudo-synchronization is an easy task for them.

Some researchers have solved the problem by estimating time delays or some time delay dependent parameters. In [7], an interesting solution is proposed for asynchronous fusion. The desired state variation is modeled as the sensor bias in this work. The state variation depends on the time difference between the time of sampling in a local sensor and the state arrival time in the fusion center, with respect to the time axis of the fusion center. Finally, the bias compensation is done after the bias estimation. In other words, the state estimates are made uncorrelated before fusion. In this work the bias is modeled by a white stationary Gaussian random process; however, a random process which depends on the time delay is generally neither Gaussian nor stationary, some other limitations of this work are discussed in [8]. Because of these limitations the solution of [7] is proper for tracking slow and nearly constant-speed targets, its simulation was done for a nearly constant-speed of 20 m/s (72 km/h) in the x and y directions. In [8] the authors have extended the work of [7] to reduce its limitations, they also supposed a stationary Gaussian model for bias estimation and its simulation was done for a nearly constant-speed of 20 m/s.

There are two general approaches for asynchronous track-to-track fusion that consider the state estimates correlation problem [1]; in the first approach the state estimates are made uncorrelated before fusion, while in the second approach the cross covariance between the state estimates is estimated and used in the fusion algorithm. Estimating the cross covariance between the tracks is a complex task with relatively high estimation error; moreover, it generally alters from sample to sample and often the result of the fusion is less informative [1]. If, somehow, the cross covariance between the state estimates are known, Bar-Shalom/Campo formulas represent the best fusion rule [1].

In order to solve the above mentioned problems in sensor networks, two methods are proposed in this paper. They are asynchronous track-to-track fusion methods that separate the task of making tracks synchronous, or uncorrelated, and the task of fusing. Moreover, in order to eliminate the data redundancy it is assumed that the sensor network is address-centric [4], [5]. Because of the nature of the track-to-track fusion the local filtering (local tracking) is needed before central fusion (central tracking). The type of the local filter does not affect the main contribution of the paper. Given the Gaussian noises, a Kalman filter is a relatively simple linear filter and is used at the sensor level tracking for both of the methods.

To the best of the authors' knowledge the first proposed method is novel; it is based on direct estimation of the actual time of sample. The main idea of this method is as follows: each sensor uses its own Kalman filter (K.F.) to estimate the tracking data locally and then transmits the estimated data to the fusion center. At the fusion center upon reception of new data the actual time of sample is directly estimated with respect to the time-reference of the fusion center. The fusion information from the previous fusion time together with the new sensor data are needed to estimate the actual time of sample. Using the estimated time of sample the received data is predicted for the start of the next fusion period. By doing so, all received data will be pseudo-synchronized for the start of the next fusion period. The independency of various sensors' state estimates is affected by the correctness of the time-estimation algorithm; the better the time-estimation, the more independent the pseudo-synchronized sensor data will be. The pseudo-synchronized sensor data can be fused immediately by a proper algorithm such as element-wise Linear Minimum Variance Unbiased Estimator (LMVUE).

The second proposed method is based on the use of a synchronization algorithm to achieve a common clock in the network with a maximum error of 1 ms. This idea is introduced as the representative of the methods which assume a common clock in asynchronous sensor networks and then they use prediction for pseudo-synchronization. This idea will be used for the evaluation in the simulation section. In this idea, tools such as Global Positioning System (GPS) receivers, the proper geostationary satellite receivers, a network synchronization protocol, or any other method that can help to achieve a common clock in the network can be used. Using such tools as GPS receivers in some sensor networks such as radar networks is straightforward, while for some other sensor networks the energy and dimension constraints should be taken into consideration. This method is a practical way for asynchronous track-to-track fusion based on a common clock in a distributed network. The maximum error of 1 ms is selected mainly because of the practical limitations and in order to mimic the error in the well-known clock synchronizing algorithms [19]. Moreover the maximum displacement of a target with a speed of 330 m/s is 33 cm in 1 ms. This method is based on the following idea: each node in the network uses a synchronization tool to achieve a common clock. Moreover, each sensor knows the start time of each fusion period in the fusion center. Each sensor uses its own Kalman filter to estimate the tracking data locally and then predicts it for the start of the next fusion period. The predicted data for the next fusion time is pseudo-synchronized with the data of the other sensors and is transmitted to the fusion center. In the fusion center the received data of the related sensors are fused by a proper algorithm such as element-wise LMVUE estimator.

These two methods make the state estimates independent before fusion. Besides basic differences, the methods differ in synchronization as well: the first method uses the centralized synchronization, while the second method uses the distributed (local) synchronization. The second method can synchronize the state estimates centrally if it uses the time-stamping technique, i.e. attaching the time of the sample to the data, in any sensor, before transmitting it to the center of fusion. However, whenever possible the distributed synchronization is a better choice, because it helps to balance the process load on the nodes. Moreover, in multi-target tracking it is an advantage for data association tasks.

The rest of the paper is organized as follows: the problem is formulated in section II. A novel method for direct estimation of the time of sample is proposed in section III. A method of pseudo-synchronization based on a common clock in the network is described in section IV. The results of computer simulation for the methods of sections III and IV, an inherently synchronous network, and an asynchronous network without any synchronization are demonstrated in section V. Finally, the conclusion of the paper is given in section VI.

II. Problem Formulation

Every sensor has its own sampling rate and its measurement is:

$$\boldsymbol{y}_i(k+1) = \boldsymbol{H}_i\boldsymbol{X}(k+1) + \boldsymbol{v}_i(k+1). \tag{1}$$

Where i is the sensor number, $\boldsymbol{y}$ is the 6×1 measurement vector in three-dimensional space:

$$\boldsymbol{y} = (x, \dot{x}, y, \dot{y}, z, \dot{z})^t. \tag{2}$$

The vector $\boldsymbol{y}$ has two elements of position and speed for each dimension. Theoretically $\boldsymbol{y}$ can have a single element of position for each dimension. Position is not observable through speed thus that single element cannot be the speed for target tracking. Measuring only position, or both position and speed depends on the sensor's ability. The state vector $\boldsymbol{X}$ is 9×1 in three-dimensional space:

$$\boldsymbol{X} = (x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z})^t. \tag{3}$$

The vector $\boldsymbol{X}$ has three elements of position, speed, and acceleration for each dimension. $\boldsymbol{H}$ is the measurement matrix and $\boldsymbol{v}$ is a white Gaussian measurement noise with zero mean and covariance matrix $\boldsymbol{R}$.

The corresponding discrete dynamic model is:

$$\boldsymbol{X}(k+1) = \boldsymbol{F}\boldsymbol{X}(k) + \boldsymbol{G}\boldsymbol{w}(k+1). \tag{4}$$

$\boldsymbol{F}$ is the state transition matrix, $\boldsymbol{G}$ is the noise input vector, and $\boldsymbol{w}$ is a white Gaussian process noise with zero mean and covariance matrix $\boldsymbol{Q}$. Whiteness is not necessary. In target tracking $\boldsymbol{w}$ is the target's acceleration vector, which is 3×1 in three-dimensional space:

$$\boldsymbol{w} = (w_x, w_y, w_z)^t. \tag{5}$$

As shown in Fig. 1, at sensor level tracking each sensor uses its own regular Kalman estimator to reduce the error or variance of the states. At the track-to-track level in the fusion center an element-wise LMVUE is used.

To use element-wise LMVUE, state estimates should be independent. This independency is achievable by synchronous or pseudo-synchronized state estimates which will be discussed later. Element-wise LMVUE is a static estimator and assuming Gaussian noises it is equivalent to an element-wise Maximum Likelihood (ML) estimator or an element-wise static Kalman estimator. Another name for this estimator is element-wise Optimum Weighted Averaging. This estimator needs to know the variance of each random variable contributing to the fusion process. These random variables are the outputs of a Kalman estimator; thus, the necessary main diagonal elements of the error covariance matrix, P generated by K.F., besides the state variables are sent to the fusion center. If N sensors are contributing to track a target and p_{11j} is the first element of the main diagonal of the covariance matrix for sensor j, then the fusion of x-position, element x, using element-wise LMVUE will be:

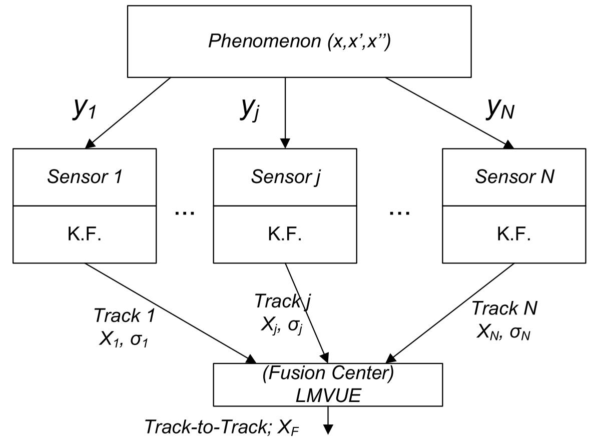

Fig. 1. The base of a sensor network for track-to-track fusion.

$$x_F = \sum_{j=1}^{N} \alpha_j x_j. \tag{6}$$

Where coefficients α_j are:

$$\alpha_j = \frac{\frac{1}{p_{11j}}}{\frac{1}{p_{111}} + \frac{1}{p_{112}} + \cdots + \frac{1}{p_{11N}}}\ ; \ j = 1, 2, \ldots, N. \tag{7}$$

The fusion variance is:

$$\frac{1}{\sigma_F^2} = \frac{1}{p_{111}} + \frac{1}{p_{112}} + \cdots + \frac{1}{p_{11N}}. \tag{8}$$

To derive equations (7) and (8), suppose all state estimates noises are zero mean; the condition for equation (6) to be unbiased would be:

$$\sum_{j=1}^{N} \alpha_j = 1. \tag{9}$$

Calculating the variance of both sides of equation (6); given the state estimates are uncorrelated, results:

$$\sigma_F^2 = \sum_{j=1}^{N} \alpha_j^2 p_{11j}. \tag{10}$$

Minimizing σ_F^2 in equation (10), given equation (9), results equations (7) and (8). If the state estimates cross covariance departs from zero, then equation (6) given equations (7) and (8) is element-wise convex estimation and is suboptimal with respect to the element-wise LMVUE. One advantage of using the LMVUE estimator is its ability for sequential

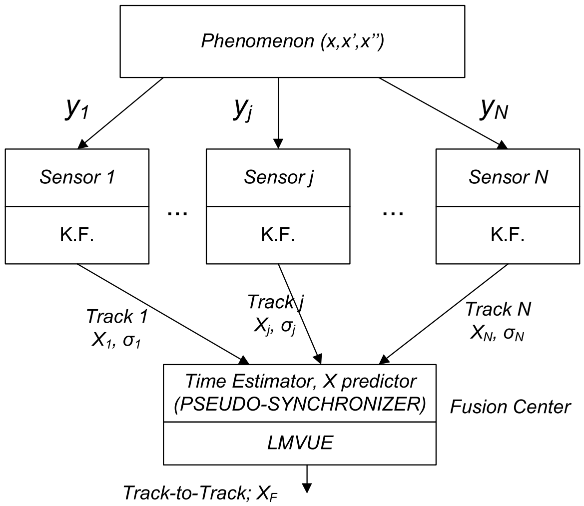

Fig. 2. A sensor network with pseudo-synchronization based on the estimation of time of sampling.

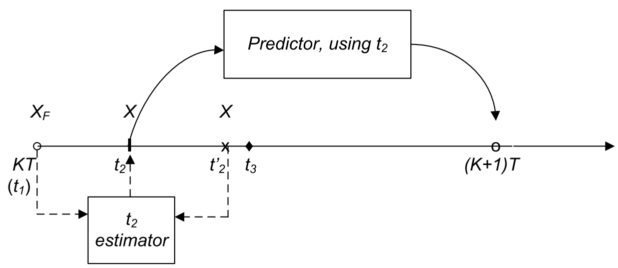

Fig. 3. Fusion center time axis, time estimator, and state predictor.

fusion. In other words, fusing data 1 and data 2 then fusing the result with data 3 is equal to fusing data 1, data 2, and data 3 at the same time. It helps to balance the process time over fusion period, which is necessary for real-time tracking and prevents processor overload at the time of receiving the last data.

III. Direct Estimation of Time of Sample

In the fusion center upon arrival of data from a sensor, the actual time of sample with respect to the time axis of the fusion center is estimated. The estimated time is used to predict the received data for the start of the next fusion period. This process will be done for all the related incoming data. These data are pseudo-synchronized for the start of the next fusion period, and can be fused in time. The fusion law is element-wise LMVUE. Fig. 2 shows the related block diagram.

The idea of estimating the time of sample is based on the kinematic principle of a moving object. Suppose the processing period of the fusion center is T. In Fig. 3, kT and $(k+1)T$ are two successive fusion times. A sample from a sensor arrives at time t_2', but the central tracking algorithm needs to know the actual time of sampling, t_2. Given deterministic data, t_2 can be simply calculated using the kinematic equation (11). In the case of stochastic data the task is not so straightforward and it involves stochastic estimation.

In a specific direction such as x equation (11) shows the kinematic equation for finding the actual time of the sample:

$$\Delta t = t_2 - t_1 = \frac{\Delta v_x}{a_x} = \frac{v_{x2} - v_{x1}}{a_x}; \quad a_x \neq 0. \tag{11}$$

Here t_2 is unknown and should be estimated; t_1 is kT and is known, the speed v_{x2} is extracted from the received stochastic data X; the speed v_{x1} is the result of the speed fusion from all the related sensors at time t_1 and it is stochastic. The denominator is the average acceleration from t_1 to t_2, i.e.

$$a_x = \frac{a_{x1} + a_{x2}}{2}. \tag{12}$$

Similar to speed, the acceleration a_{x2} is extracted from the received stochastic data X; the acceleration a_{x1} is the result of the acceleration fusion from all the related sensors at time t_1 and it is also stochastic. The stochastic variables assigned to t_2 are estimated by a local Kalman filter before transmission and those assigned to t_1 undergo one more estimation step in the fusion process.

Given all the related noises are Gaussian the numerator and the denominator of equation (11) are also Gaussian, but their ratio is not Gaussian. Moreover, it does not have a well defined random distribution [20]–[22]. However, knowing the exact form of the distribution is not necessary. According to many criteria, such as ML and Kalman, the best point-estimation for a random variable, given some continuous or discrete distributions, is its mean. For example according to the ML criterion for some distributions such as Gaussian, Exponential, and Poisson, for a random variable z, the sufficient statistic for the mean is [23]:

$$S = \sum_{j=1}^{n} z_j, \tag{13}$$

and the best estimation for the mean of the random variable is:

$$m_z = \frac{\sum_{j=1}^{n} z_j}{n}. \tag{14}$$

The process is real-time and the instantaneous estimation of Δt is useful, i.e. in the fusion center Δt is estimated based on the only sample available. In equation (14) when the mean is unknown and there is only one sample available, $n = 1$, the best estimate for the mean is the sample, $m_z = z_1$, and vice versa; if the mean is known and the random variable should be estimated the best estimate for it is the mean, $z_1 = m_z$. Generalizing this conclusion to Δt results that: the best instantaneous point-estimation for Δt is the mean of the ratio in equation (11). Although the distribution of the ratio in equation (11) is unknown and the ML estimation for some specific distributions results $z_1 = m_z$, for other distributions it is approximately true; for example, it can be easily shown that for Rayleigh distribution the sufficient statistic is:

$$S = \sum_{j=1}^{n} z_j^2, \tag{15}$$

and the best estimation for the mean of the random variable is:

$$m_z^2 = \frac{\pi}{4} \frac{\sum_{j=1}^{n} z_j^2}{n}, \tag{16}$$

with only one sample available, $n = 1$; $m_z = 0.9z_1$, $z_1 = 1.1m_z$. Therefore, with some heuristic, using equation (11) we adopt:

$$\Delta t = E(\frac{\Delta v_x}{a_x}). \tag{17}$$

To complete the instantaneous estimation of Δt the estimation of $E(\frac{\Delta v_x}{a_x})$ is needed. Thus instead of the distribution only its mean should be estimated. The best estimation for the mean of a ratio is the ratio of the means, i.e. the ratio of the mean of the numerator to the mean of the denominator excluding zero values from the denominator [20]–[22]. This can be verified by using Taylor series expansion. Rewriting equation (11) as (18), by replacing $\Delta v_x = v$ and $a_x = a$,

$$\Delta t = f(v, a) = \frac{v}{a}. \tag{18}$$

Using Taylor series expansion for equation (18) about $\frac{\mu_v}{\mu_a}$, Δt can be approximated by (19)

$$\Delta t \cong \frac{\mu_v}{\mu_a} + (\frac{\partial f}{\partial v})_{\mu_v,\mu_a}(v - \mu_v) + (\frac{\partial f}{\partial a})_{\mu_v,\mu_a}(a - \mu_a); \quad \mu_a \neq 0. \tag{19}$$

Taking expectation results approximation (20)

$$\Delta t \cong E(\frac{\Delta v_x}{a_x}) \cong \frac{\mu_v}{\mu_a}; \ \ \mu_a \neq 0. \tag{20}$$

Kalman and LMVUE estimators are unbiased; therefore,

$$\mu_v = v_{x2} - v_{x1}; \ \ \mu_a = a_x = \frac{a_{x1} + a_{x2}}{2}. \tag{21}$$

When a_x approaches zero the estimation law must be changed to:

$$if(|a_x| < \varepsilon) \rightarrow \Delta t = \frac{x_2 - x_1}{v_x}; \ \ v_x = \frac{v_{x1} + v_{x2}}{2}. \tag{22}$$

This implies a constant speed motion. In practice, selecting $\epsilon = 0.2\, m/s^2$ gives acceptable results. Similarly, if the average speed approaches zero the estimation reduces to $\Delta t = 0$, i.e. no motion.

Using the estimated actual time of the sample the state estimate X is predicted for time $(k+1)T$. This process is done for data of each sensor and the result is the pseudo-synchronized state estimates of all the sensors. The pseudo-synchronized data can be considered independent and can therefor be fused without need to consider the cross covariance between tracks. Using element-wise LMVUE the fusion process can be done sequentially upon receiving the new data, at a time like t_3 in Fig. 3. The time interval $t_3 - t_2'$ is the required time to complete the two tasks of Fig. 2. It also prevents processor overload at time $(k+1)T$.

Using the prediction for time $(k+1)T$ and fusing the data before this time can be used to compensate for some limitations such as electromechanical lags in the overall system, e.g. the movement of a tracking antenna. This is an advantage for real-time tracking. Moreover, the proposed algorithm is not a high time-consuming task which can be considered another advantage for this method, being an essential requirement for real-time tracking.

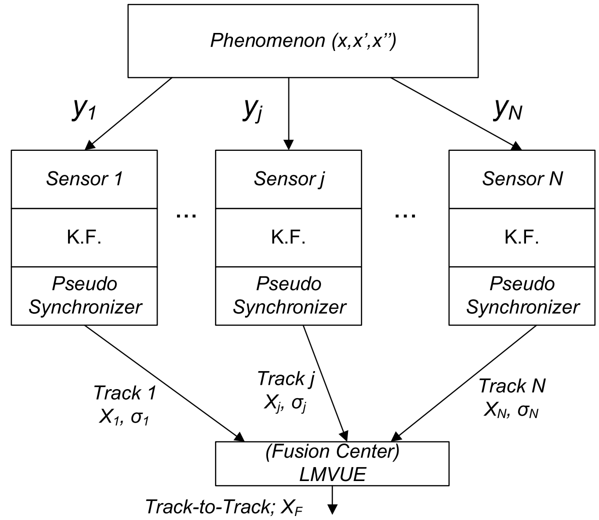

Fig. 4. A sensor network with pseudo-synchronization based on a common clock in sensors.

There is no restriction on the periods of the sensors T_j and the fusion period T, neither in duration nor in phase. A useful and feasible condition that limits the prediction error is selecting T equal to the greatest T_j. As a natural consequence of this method, given no missing data, the time of sample estimation-rate for each sensor in the fusion center is equal to the sampling-rate of that sensor.

IV. Pseudo-Synchronization Based on a Common Clock

A method based on synchronized clocks in all the sensors with at most one millisecond difference with respect to the fusion center clock is described in this section. Technically speaking, it is achievable by means of satellite positioning systems such as GPS receivers or a geostationary satellite system such as Geostationary Environment Operational Satellite receivers [19]. It is also possible by some network protocols such as NTP [19]. Many sensor networks do not meet energy and space constraints and using satellite receivers with sensors is practical; meanwhile, these constraints should be considered significant in some other networks. The proposed technique in this section is feasible in a lot of sensor networks. Moreover, it is a benchmark asynchronous track-to-track data fusion method, Fig. 4.

Here, each sensor knows the exact fusion times of the fusion center, e.g., 0, 1, 2, etc. seconds. The input data of each sensor passes through a Kalman filter. The filtered data are locally predicted for the next fusion time, i.e. $(k+1)T$, then the predicted data are transmitted to the fusion center. In the fusion center the related state estimates are pseudo-synchronized, and can be fused with the LMVUE estimator. The processor does not need to wait until time $(k+1)T$ to do the fusion and the fusion process can be done sequentially upon receiving a new state estimate.

In Fig. 5, X_j is the state of the target in sensor j corresponding to sample at time t; then predicted for time $(k+1)T$ which is shown by X_{jp} and transmitted to the fusion center. The desired state is received in the fusion center at time t'.

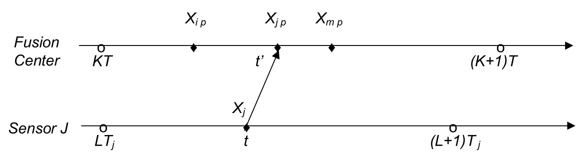

Fig. 5. The time of sample in sensor j, and the sample receiving time in the fusion center.

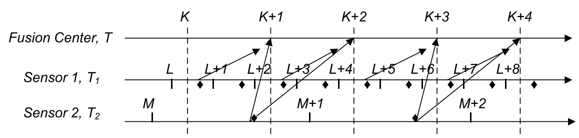

Fig. 6. Each sensor sends a sample for a target at every fusion period, the sensor periods relation is: $T_1 < T < T_2$.

The other state variables on the time axis of the fusion center are related to the other sensors.

Here also there is no restriction on the periods of the sensors T_j and the fusion period T, neither in duration nor in phase. Regardless of its period, each sensor sends one sample taken of a target at every fusion period. If a sensor period is greater than T that sensor repeats prediction for more than one fusion time. Fig. 6 shows various periods relations.

For correct operation, each sensor should know the worst case time-delay in the network to send its data at the right time. The worst case estimation of the network time-delay is approximately found by the fusion center and sent to all sensors.

V. Computer Simulation

This section is dedicated to computer simulation to validate the effectiveness of the novel method of section III, and compares it with three other methods of track-to-track data fusion. One simulation is about an ideal intrinsically synchronous network. Another simulation is about the method of section IV, i.e. pseudo-synchronization based on a common clock. The third simulation is about asynchronous track-to-track fusion without any type of pseudo-synchronization. The method of section IV, although expensive, is feasible in many sensor networks; moreover, it is a benchmark asynchronous track-to-track fusion method that separates the task of making tracks uncorrelated, or synchronous, and the task of fusing.

Comparison with algorithms which need some suppositions about the cross covariance model between tracks are not straightforward and can lead to an erroneous result. Moreover for the asynchronous tracks the cross covariance model usually changes from sample to sample. This is an important reason that most of the methods which use a model for cross covariance between tracks, simulate the slow and nearly constant-speed targets.

For all methods the fusion law is element-wise LMVUE. Three sensors and a fusion center are supposed to track a common target. Without losing generality the tracking is done in the x direction. The instantaneous speed of the target is assumed to be high and there is no restriction such as uniformity on the speed. The process noise or input noise is assumed to be white Gaussian with zero mean and standard deviation of 1.5 m/s^2. This amount is near the maximum

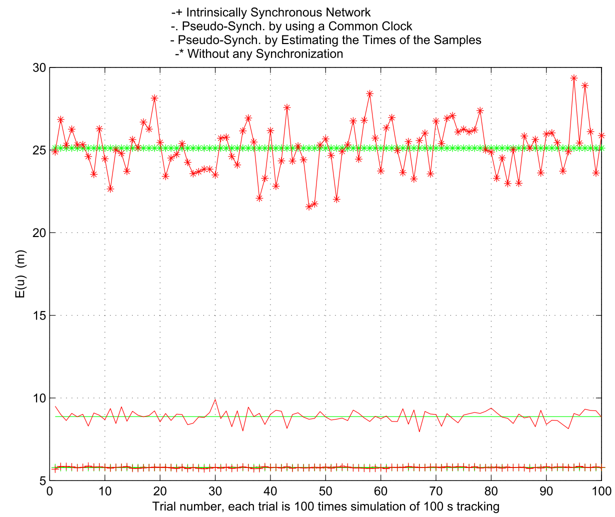

Fig. 7. Mean of tracking errors, $E(u)$; for all the four simulations.

acceleration of lots of aircrafts under control and it is usually selected for maneuvering targets [24], also it can be examined by any passenger in an airplane, by measuring the take off acceleration; the typical take off speed for an airplane is 270 km/h (75 m/s). The selected fusion period is $T = 1$ s. The measurement noises for position and speed are Gaussian with zero mean for all three sensors. The standard deviations are 20 m and 11 m/s for Sensor1, 15 m and 9 m/s for Sensor2, and 25 m and 15 m/s for Sensor3. These values are selected based on the guidelines of some earlier studies [7], [24], [25]. In real applications the process and measurement noises are estimated and updated continuously by using statistical methods with a period much larger than the sampling period of the sensor. The simulations have been examined with parameters considerably larger than the aforementioned values and there is a little variation in the results. The correlation between position-noise and speed-noise is zero. The initial values are $x = 7000$ m, $v_x = 330$ m/s, $a_x = 0$ m/s^2. The simulation time for each tracking is $100T$ or 100 s. The state transition model, F, is based on the second-order position linearization, i.e. for one dimension is:

$$F = \begin{pmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \tag{23}$$

The criterion for evaluation of the results is u and defined by equation (24),

$$u = \frac{\sqrt{\sum_{i=1}^{n} (x_i - x_{Fi})^2}}{n}. \tag{24}$$

In equation (24), x_i is the real data, x_{Fi} is the result of tracking, n is 100, and i shows the discrete time of simulation. The mean of u or $E(u)$, and the standard deviation of u are calculated for each trial of the 100 Monte Carlo tracking. This trial is repeated 100 times. There are two figures for all simulations. Fig. 7 demonstrates $E(u)$ and the mean of $E(u)$ versus the trial number. Fig. 8 shows the standard deviation of u and its mean versus the trial number. All the four methods are convergent and the differences are in the tracking errors.

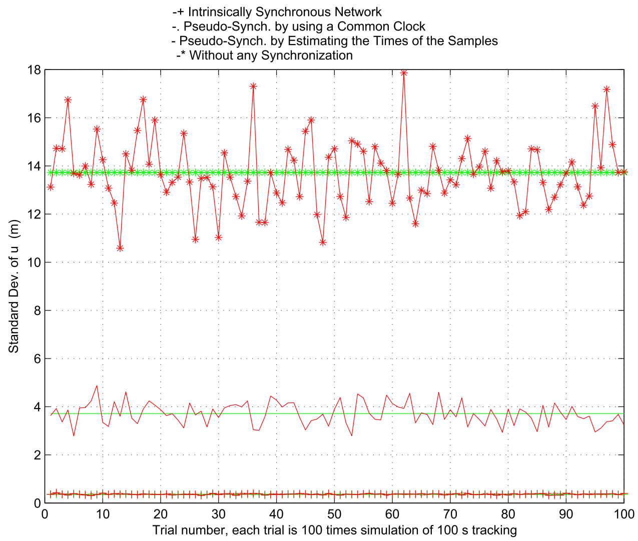

Fig. 8. Standard Deviation of u for all the four simulations.

A. The Ideal Intrinsically Synchronous Network

Here the periods of all sensors are equal to T and they are synchronized with the fusion center. The samples are taken at the same times and reach the fusion center with zero or exactly known time-delays. With respect to synchronicity the tracking result cannot outperform the result of this simulation. Thus this is the comparison basis. The results are shown in Fig. 7 and Fig. 8.

B. Pseudo-Synchronization Based on a Common Clock

Without loss of generality the sampling periods of the sensors 1, 2, 3 are chosen to be $0.7T$, $0.85T$, and T respectively; T is the fusion period. The sample times of the sensors 1, 2, 3 are put at $0.25T$, $0.5T$, $0.75T$ respectively, with respect to the start of the sensor period. The maximum error between the clock of each sensor and the fusion center clock is 1 ms. The results are shown in Fig. 7 and Fig. 8.

C. Data Fusion Without Synchronization

Here the sampling period of the sensors are set equal to the fusion period T. There is no synchronization between the sensors and the fusion center. The data of each sensor is received randomly at a uniformly distributed time during the period of the fusion center. This is the worst case scenario for the fusion, as the tracks are highly correlated. The results are shown in Fig. 7 and Fig. 8.

D. Pseudo-Synchronization of Data With Direct Estimation of Time of Sample

The theory behind this method, as explained in section III, does not impose any restriction on the sampling period of the sensors. The fusion period of the fusion center must be selected by the sensor network designer. There are some constraints to select this period, e.g. it should be less than the dominant time-constant of the phenomenon for a real-time tracking. On the other hand, in a well designed system the period of sampling for each sensor should also be less than the dominant time-constant of the phenomenon. The selection of sampling periods of the sensors is usually not in the control of the track-to-track fusion designer therefore a useful idea is selecting the fusion period equal to the greatest period among the periods of the sensors. The data of each sensor is received randomly at a uniformly distributed time during the period of the fusion center. The results are shown in Fig. 7 and Fig. 8.

E. Comparison

Referring to Fig. 7 and Fig. 8, the mean and variance of the tracking error which is defined by equation (16) for the reference network (an ideal intrinsically synchronous network) are nearly equal to those of an asynchronous network which uses the method of section IV (pseudo-synchronization based on a common clock), these two tracking methods are indistinguishable in the figures. The means of $E(u)$ for these two tracking methods are 5.79 m and 5.81 m, respectively. On the other hand, the tracking error of an asynchronous network without any kind of synchronization is much larger than the tracking error of the reference network, the means of $E(u)$ for these two tracking methods are 25.11 m and 5.79 m, respectively. Finally, the tracking error of an asynchronous network which uses the direct estimation of the time of sample for pseudo-synchronization is slightly more than the tracking error of the reference network, the means of $E(u)$ for these two tracking methods are 8.87 m and 5.79 m, respectively. In reality, contrary to the white noise, the adjacent samples of the input noise are not independent and there will be less error for the proposed algorithm.

To complete this comparison, it is useful to compare the two main methods in distributed networks. The novel method (“direct estimation of time of sample”) does not need high-cost common-clock facilities; this method is simple, and therefore imposes less load on the central processor, while not needing costly resources such as large memories. The second method (“pseudo-synchronization based on a common clock”) is more expensive and more complex, but it has a slightly better performance. The two methods can be used in tracking dynamic targets in distributed sensor networks.

VI. Conclusion

This paper proposes a method for asynchronous track-to-track fusion based on the direct estimation of the time of sample in a centralized asynchronous sensor network. Comparing this method with three other tracking methods, simulations show acceptable performance besides simplicity and low cost of algorithm implementation.

References

[1] M. Reinhardt, B. Noack, and U. D. Hanebeck, “On optimal distributed Kalman filtering in non-ideal situations,” in *Proc. 15th Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 1020–1027.

[2] M. Uney, D. E. Clark, and S. J. Julier, “Information measures in distributed multitarget tracking,” in *Proc. 14th Int. Conf. Inf. Fusion*, Chicago, IL, USA, Jul. 2011, pp. 1–8.

[3] C.-Y. Chong and S. Mori, “Convex combination and covariance intersection algorithms in distributed fusion,” in *Proc. Conf. Fusion*, Montreal, QC, Canada, Aug. 2001, pp. 1–8.

[4] S. K. Singh, M. P. Singh, and D. K. Singh, "Routing protocols in wireless sensor networks—A survey," *Int. J. Comput. Sci. Eng. Survey*, vol. 1, no. 2, pp. 63–83, Nov. 2010.

[5] X. Li, K. Lu, N. Santoro, I. Simplot-Ryl, and I. Stojmenovic, "Alternative data gathering schemes for wireless sensor networks," in *Proc. Int. Conf. ROGICS*, Mahdia, Tunisia, 2008, pp. 577–586.

[6] M. E. Liggins II, I. Kadar, and V. Vannicola, "Distributed fusion architectures and algorithms for target tracking," *Proc. IEEE*, vol. 85, no. 1, pp. 95–107, Jan. 1997.

[7] X. Lin, Y. Bar-Shalom, and T. Kirubarajan, "Multisensor-multitarget bias estimation for general asynchronous sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 899–921, Jul. 2005.

[8] A. Rafati, B. Moshiri, and J. Rezaei, "A new algorithm for general asynchronous sensor bias estimation in multisensor-multitarget systems," in *Proc. IEEE 10th Int. Conf. Inf. Fusion*, Jul. 2007, pp. 1–8.

[9] R. C. Luo, C.-C. Yih, and K. L. Su, "Multisensor fusion and integration: Approaches, applications, and future research directions," *IEEE Sensor J.*, vol. 2, no. 2, pp. 107–119, Apr. 2002.

[10] H. Xi-Feng and W. Qin-Zhang, "An algorithm of weighted covariance for centralized asynchronous fusion based on Kalman," *Amer. J. Eng. Technol. Res.*, vol. 11, no. 9, 2011.

[11] C.-L. Wen, Q.-B. Ge, and X.-L. Feng, "Optimal recursive fusion estimator for asynchronous system," in *Proc. 7th Asian Control Conf.*, Hong Kong, Aug. 2009, pp. 148–153.

[12] Q.-B. Ge and C.-L. Wen, "Optimal multiesensor asynchronous fusion based on state transform," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit.*, Beijing, China, Nov. 2007, pp. 360–365.

[13] H. Chen and Y. Bar-Shalom, "Track fusion with legacy track sources," in *Proc. 9th Int. Conf. Inf. Fusion*, Florence, Italy, Jul. 2006, pp. 1–8.

[14] G. Hui-Dong, Z. Xin-Hua, X. Lin-Zhou, S. Yuan, X. Ce, and T. Shao-Bo, "Asynchronous multisensor data fusion based on minimum trace of error covariance," in *Proc. IEEE 9th Int. Conf. Inf. Fusion*, Jul. 2006, pp. 1–5.

[15] A. T. Alouani and T. R. Rice, "On optimal asynchronous track fusion," in *Proc. 1st IEEE Austral. Symp. Data Fusion*, Nov. 1996, pp. 147–152.

[16] A. T. Alouani and T. R. Rice, "On asynchronous data fusion," in *Proc. IEEE 26th Southeastern Symp. Syst. Theory*, Mar. 1994, pp. 143–146.

[17] Z. Fengchun, "Asynchronous predict data fusion for sensor networks," in *Proc. Int. Conf. Intell. Comput. Technol. Autom.*, Changsha, China, May 2010, pp. 821–824.

[18] X. Tian and Y. Bar-Shalom, "On algorithms for asynchronous track-to-track fusion," in *Proc. Int. Conf. Inf. Fusion*, Edinburgh, U.K., Jul. 2010, pp. 1–8.

[19] A. S. Tanenbaum and M. Van Steen, *Distributed Systems, Principles and Paradigms*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2007.

[20] H. Morisugi, J. Romero, and T. Moriguchi, "Confidence interval for the ratio of two normal variables (an application to value of time)," *Interdisciplinary Inf. Sci.*, vol. 15, no. 1, pp. 37–43, 2009.

[21] G. Marsaglia, "Ratios of normal variables," *J. Stat. Softw.*, vol. 16, no. 4, pp. 1–10, May 2006.

[22] A. Stuart and K. Ord, *Kendall's Advanced Theory of Statistics*, vol. 1, 6th ed. London, U.K.: Arnold, 1998.

[23] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill, 2002.

[24] B. Moshiri and F. Besharati, "Maneuvering target tracking," *WSEAS Trans. Circuit Syst.*, vol. 3, no. 1, pp. 176–182, Jan. 2004.

[25] L.-W. Fong, "Multi-sensor track-to-track fusion via linear minimum variance sense estimators," *Asian J. Control*, vol. 10, no. 3, pp. 277–290, May 2008.

Hadi Talebi received the B.Sc. degree in electrical engineering from the Sharif University of Technology (SUT), Tehran, Iran, in 1989, and the M.Sc. degree in electrical engineering from Tehran University, Tehran, in 1994. He is currently pursuing the Ph.D. degree in computer architecture at SUT. His current research interests include data fusion in distributed sensor networks.

A. M. Afshin Hemmatyar was born in Tehran, Iran, in 1965. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Sharif University of Technology (SUT), Tehran, Iran, in 1988, 1991, and 2007, respectively. Since 1991, he has been with the Department of Computer Engineering, SUT. His current research interests include the wireless sensor networks, cognitive radio networks, mobile ad-hoc networks, radio frequency identification systems, software radio systems, and RF and microwave communication systems.