Sensor Staggering in Multi-Sensor Target Tracking Systems

Ruixin Niu, Pramod Varshney, Kishan Mehrotra and Chilukuri Mohan
Department of Electrical Engineering and Computer Science
121 Link Hall
Syracuse University
Syracuse, NY 13244 USA
Email: varshney@ecs.syr.edu

Abstract — For a multi-sensor target tracking system in a cluttered environment, the effects of temporally staggered sensors on system performance are investigated and compared with those of synchronous sensors. A probabilistic data association filter (PDAF) is used to track the target. Measurements from local sensors are fused in a centralized manner for the system with synchronous sensors. The system performances are compared both in terms of track life and in-track percentage. For a wide variety of scenarios, simulation results show the superiority of temporally staggered sensors over synchronous sensors.

I. INTRODUCTION

Multi-sensor target tracking systems have generated intensive interest because of their enhanced estimation performance, surveillance coverage and robustness. Different kinds of multi-sensor tracking system architectures, are introduced and compared in [2] and [8].

With regard to the temporal issues of the system, there exists the so called “out of sequence measurement” (OOSM) problem. In this scenario, measurements from different sensors are collected at the same time, meaning they have the same time stamp. However, due to different transmission delays in the sensor network, it is possible that a measurement with time stamp $\tau$ arrives after the target state estimation has been updated to time $t > \tau$. Many authors have investigated this problem and proposed methods to deal with the OOSM [3, 4, 6, 7].

In an OOSM problem, the sensors are still assumed to operate synchronously and only the measurements arrive asynchronously. Much less attention is paid to the systems with asynchronous sensors [5, 10], because the estimation accuracy at each sensor’s sampling time is worse than the estimation accuracy at the sampling time of the system with synchronous sensors. In spite of this disadvantage, however, a system with temporally staggered sensors has much lower maximum prediction error because it is more frequently updated with new measurements.

In our previous work [9], the effects of asynchronous sensors, or temporally staggered sensors were investigated and compared with the traditional system with synchronous sensors under the assumption that neither false alarms nor missed detections exist. To make a fair comparison, a new metric of performance, the average estimation error variance (AEV), was used. Through both analytical and numerical methods, we obtained extensive results on how to optimally stagger sensors over time for different scenarios.

In this paper, we extend our research to more realistic cases, where there always exists the problem of measurement-origin uncertainty caused by false alarms, clutters and missed detections. In such cases, we simply do not know which measurement, if any, is from the target. To deal with this problem, many algorithms have been developed, such as the nearest neighbor standard filter (NNSF), the strongest neighbor standard filter (SNSF) (if the signal intensity information is available), and the probabilistic data association filter (PDAF) [2]. At each time step, they all need prior information, i.e., the prediction from the last time step, to process the new measurements. If the prediction is not accurate enough, the system will lose track of the target quickly. Thus, it is imperative that the maximum value of the prediction error be kept to a minimum. In this paper, we propose to use sensor measurement staggering as a means to keep the maximum prediction error under control. This will help maintain tracks over a longer period.

In Section II, we introduce the measurement collection model and the target dynamic model. In Section III, the realistic environment with false alarms and missed detections is discussed. In addition, we describe briefly the probabilistic data associate filter (PDAF) which is chosen to track in a cluttered environment. A criterion to judge whether a target is in track, is proposed. In Section IV, for two sensors with the same performance, we compare the performance of the system with uniformly staggered sensors to that with synchronous sensors. In Section V, the case where two sensors have different performances is studied, and the best sensor staggering schemes are obtained for various system parameters via simulation. Some concluding remarks are made in Section VI.
II. System Model

A Measurement Collection Model

In this paper, we assume that the tracking system works in a centralized manner, meaning that raw measurements are transmitted from local sensors to a fusion center for the purpose of target tracking. Because there is no pre-processing of the measurements at local sensors and no information is lost before fusion, the centralized fusion system is optimal. Two different data collection schemes for a two-sensor tracking system is illustrated in Fig. 1. Here we assume that both sensors have the same sampling interval $T$. In one system, the two sensors collect data synchronously at time $kT$; while in the other system, one sensor collects measurements at $kT$, and the other one at $kT + T_1$. The time intervals between two sensors are defined as $T_1$ and $T_2$, respectively. Obviously, the intervals satisfy the following relationship:

$$T_1 + T_2 = T$$

B Target Dynamic Model

For simplicity, we only study the case where a two-sensor system is tracking a target with one dimensional movement. A direct discrete white noise acceleration model [1] is used in this paper. We assume a target moving along a coordinate $\xi$, and the state of the target is

$$x = [\xi \ \dot{\xi}]';$$

(1)

Because the state equations for the system with synchronous sensors and the system with asynchronous sensors are quite different, we discuss them separately.

B.1 Target Dynamic Model for the System with Synchronous Sensors

The state equation for the second order (white noise acceleration) direct discrete time model[1] is

$$x(k + 1) = Fx(k) + \Gamma \nu(k)$$

(2)

where the transition matrix is

$$F = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

(3)

and the vector gain multiplying the scalar process noise is given by

$$\Gamma = \begin{bmatrix} r_1^2 \\ T \end{bmatrix}$$

(4)

The process noise covariance matrix of this direct discrete time model is

$$Q = E[\Gamma \nu(k)\nu(k)\Gamma']$$

$$= \sigma_\nu^2 \begin{bmatrix} r_1^4 & r_1^3 T \\ T & T^2 \end{bmatrix}$$

(5)

The observation model is

$$z(k) = Hx(k) + \omega(k)$$

(6)

where $z(k)$ is the measurement data, and $\omega = [\omega_1 \ \omega_2]'$ is the measurement error vector. We assume that only position measurements are available from the two sensors, meaning that

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(7)

Based on the assumption that measurement errors of the two sensors are independent, the measurement noise covariance matrix is

$$R = E[\omega(k)\omega(k)']$$

$$= \begin{bmatrix} \sigma_{\omega_1}^2 & 0 \\ 0 & \sigma_{\omega_2}^2 \end{bmatrix}$$

(8)

where $\sigma_{\omega_1}^2$ and $\sigma_{\omega_2}^2$ are the measurement error variances of sensors $S_1$ and $S_2$, respectively.

In terms of sensor $S_1$, the target maneuvering index [1] is defined as

$$\lambda = \frac{\sigma_\nu T^2}{\sigma_{\omega_1}}$$

(9)

$\lambda$ measures the degree of elusiveness of the target to be tracked: when the value of $\lambda$ is high, the target is highly maneuvering and hard to track. Notice that the definition in (9) is valid only for a system with identical sensors. For a system with sensors of different measurements noise variances, the target maneuvering index will be defined in terms of the composite sensor. This will be explained in detail later.

\[Fig. 1: Measurement pattern for two synchronous sensors vs. two staggered sensors. The sampling intervals satisfy: $T = T_1 + T_2$.\]
B.2 Target Dynamic Model for the System with Asynchronous Sensors

For asynchronous sensors, in general, \( T_1 \neq T_2 \). The state equation should be carefully adjusted to take care of the non-uniform timing. The state equation for the target dynamics is

\[
x(k + 1) = F(k)x(k) + \Gamma(k)\nu(k)
\]

where

\[
F(k) = \begin{bmatrix} 1 & T(k) \\ 0 & 1 \end{bmatrix}
\]

and

\[
\Gamma(k) = \begin{bmatrix} T^2(k) \\ \frac{T^2(k)}{2} \end{bmatrix}
\]

Note that \( T(k) \) is alternating between \( T_1 \) and \( T_2 \):

\[
T(k) = \begin{cases} T_1 & k = 2n - 1 \\ T_2 & k = 2n \end{cases}
\]

where \( n = 1, 2, \ldots \). The process noise covariance matrix is

\[
Q(k) = E[\Gamma(k)\nu(k)\nu(k)\Gamma(k)'] = \sigma^2_\nu(k) \begin{bmatrix} T^4(k) & T^3(k) \\ \frac{T^4(k)}{2} & T^2(k) \end{bmatrix}
\]

where

\[
\sigma^2_\nu(k) = \sqrt{\frac{T}{T(k)} \sigma^2_\nu}
\]

and \( \sigma_\nu \) is the process noise level corresponding to the sampling interval of \( T \). The rescaling of the state process noise variance is necessary to model the target motion uncertainty over varying sampling intervals and make the target dynamic model consistent over time [1]. Plugging (15) into (14), we have

\[
Q(k) = \sigma^2_\nu T \begin{bmatrix} T^3(k) & T^2(k) \\ \frac{T^3(k)}{2} & T(k) \end{bmatrix}
\]

It is interesting that the process noise covariance matrix \( Q(k) \) is very similar to its counterpart of a discretized continuous time model [1]: the only difference is that the \((1,1)\) element of the latter is \( \frac{T^2}{2} \). This shows that there is little difference between the two models. We have chosen the direct discrete model in our previous work because there exist closed form steady-state estimation error solutions for this model. This made our analysis much easier in [9].

The observation model is

\[
z(k) = Hx(k) + \omega(k)
\]

Because at each time, only the measurement from a single sensor is available, we have

\[
H = \begin{bmatrix} 1 & 0 \end{bmatrix}
\]

The measurement noise variance is

\[
R(k) = E[\omega^2(k)] = \begin{cases} \sigma^2_\omega & k = 2n - 1 \\ \sigma^2_\omega & k = 2n \end{cases}
\]

Unless otherwise specified, the target maneuvering index is the same as in Section B.1:

\[
\lambda = \frac{\sigma_\nu T^2}{\sigma_\omega}
\]

III. Cluttered Environments and PDAF Filter

In our previous work [9], we investigated the case where false alarms and missed detections do not exist. Under such circumstance, Kalman filter is the optimal tracker and we derived many analytical results [9].

However, in reality, there is always uncertainty associated with the detection process of a tracking system, due to the nature of hypothesis testing problems, the finite signal to noise ratio (SNR) and clutter in the environment. For a particular measurement, we do not know whether it originates from a target or is a false alarm. Given a scan of measurements, it is possible that a “true” measurement exists along with false alarms or that all the measurements are false alarms and none is generated by a target (missed detection).

Many algorithms were developed to deal with the problem of measurement-origin uncertainty caused by missed detections and false alarms, including the nearest neighbor standard filter (SSNF), the strongest neighbor standard filter (NSNF), the multi-hypothesis tracker (MHT) and N-dimensional assignment algorithms [2].

The probabilistic data association filter (PDAF) [2] is a very simple and successful one-scan target tracking algorithm in the presence of false alarms and missed detections. Assuming that the observations at each time \( k \), comprise \( n_k \) constituents, the PDAF is based on the following assumptions:

A1 The \( n_k \) observations are the locations of all threshold exceedances that have been observed.

A2 With probability \( P_f \) the true measurement generated by the target may be present along with \( n_k - 1 \) false alarms; and with probability \( 1 - P_f \) it may be absent, corresponding to a missed detection.
A3 The number of false-alarm constituents follows the Poisson distribution.

A4 False-alarm measurements are uniformly distributed within the observation region, and are independent of each other.

At each sampling time \( k \), the PDAF works as follows:

At scan \( k \), there are \( n_k + 1 \) events, of which \( n_k \) events enumerate all the possible situations where a observation originates from the target, and one event is corresponding to the case where none of the observations is a true measurement (missed detection). Based on a Gaussian prior, the PDAF assigns each event an association probability. There will be \( n_k + 1 \) updated state estimates conditioned on the \( n_k + 1 \) events. The final state estimate is obtained by an average of them weighted by their association probabilities. Equivalently, the PDAF converts its posterior target state probability density function, a mixture of Gaussian pdf’s, to a single Gaussian distribution having the same mean and variance. Therefore, at each scan, estimation is built upon a Gaussian prior, converted to a Gaussian mixture posterior, which is then approximated by a Gaussian pdf for the succeeding scan[2].

To reduce the computation load, at each time step \( k \), the PDAF uses a “gating” procedure to choose candidate measurements. The volume of the gate is proportional to \( \sqrt{|S(k)|} \), where \( S(k) \) is the innovation covariance matrix at time \( k \). The measurements outside the gate, whose association probabilities will be negligible.

In a cluttered environment, the problem of lost track always exists and hence the track life is a more proper metric. The traditional definition of “in-track” is that the number of measurements in the validation region (gate) is below a threshold \( M \) [2], i.e., the number of false alarms within the validation region is bounded. Once the number goes beyond the threshold, the filter will diverge quickly. However, this is not a very good criterion for cases where the clutter density is relatively low. Under such circumstances, even though the estimation error (and thus the \( |S(k)| \) ) is very large and the track may already be lost, the number of validated measurements within the gate may still be very small due to low false alarm density. The traditional criterion will still treat the target to be “in-track”. Therefore, we construct a new criterion which also takes into account the estimation error:

**Definition 1** A target is in-track if the number of measurements in the validated gate is below a threshold \( M \) and the updated position estimation error variance is less than \( \alpha \sigma^2_n \), where \( \alpha \) is an adjustable constant and \( \sigma^2_n \) is the measurement error covariance.

In this paper, we take \( M = 5 \) and \( \alpha = 100 \).

### IV. Sensors with Identical Performances

In this section, we investigate the simple case where the two sensors in the system have identical performances, meaning that the two sensors have the same measurement error variances \( (\sigma^2_m) \), probability of detection \( (P_d) \), and false alarm density.

Without loss of generality, we use \( T = 1 \) s. The performance metrics include average track life and in-track percentage. Track life is defined as the time interval between the time that a track is initiated (0 s) and the time that the same track is lost. In-track percentage is the percentage of simulation runs where the system keeps the target in-track for more than 100 sampling intervals (100 s). Unless otherwise specified, these performances are obtained by 3000 Monte-Carlo runs.

The average track life and in-track percentage are plotted for both the system with synchronous sensors and that with staggered sensors in Figs. 2 - 7 as functions of different system parameters.

The system performances as functions of \( P_d \) are shown in Fig. 2 and Fig. 3. We can see that the performance of uniformly staggered sensors is better than that of synchronous sensors especially when \( P_d \) is high \( (P_d > 0.8) \). When \( P_d \) is low, the performance curves of the two schemes are almost indistinguishable. This can be easily understood by analyzing the improvement in terms of probability of detection by using synchronous sensors. The equivalent probability of detection for the two synchronous sensors is

\[
PD = 1 - (1 - P_{d_1})(1 - P_{d_2})
\]

To measure the performance improvement in terms of probability of detection, we define two new metrics: detection improvement (DI) and relative detection improvement (RDI):

**Definition 2**

\[
DI = PD - P_{dm}
\]

and

\[
RDI = \frac{PD - P_{dm}}{P_{dm}}
\]

where \( P_{dm} = \min(P_{d_1}, P_{d_2}) \) and \( 0 < P_{dm} \leq 1 \).

Substituting \( P_{d_1} = P_{d_2} = P_d \) into (21), we have

\[
PD = P_d(2 - P_d)
\]

The corresponding DI and RDI are

\[
DI = (1 - P_d)P_d
\]

and

\[
RDI = 1 - P_d
\]
DI is a quadratic function of $P_d$ and it has its maximum value at $P_d = 0.5$. RDI is a monotone decreasing function of $P_d$. The higher the $P_d$, the lower is the relative improvement of the PD by using synchronous sensors. When $P_d$ tends to 1, both the relative improvement (RDI) and the absolute improvement (DI) go to 0.

Hence, when individual sensor’s $P_d$ is low or medium, by using synchronous sensors we can improve the system’s PD significantly (with high RDI). On the other hand, by using staggered sensors the system benefits from the more frequent update with new measurements. These two schemes have similar performances at low $P_d$. When $P_d$ is high, the improvement of PD by using synchronous sensors is very limited relative to the already high $P_d$. On the other hand, staggered sensors offer more frequent updates with new measurements and help keep target in track for a longer time.

The results for a high $P_d$ ($P_d = 0.9$) and varying target maneuvering index $\lambda$ are plotted in Fig. 4 and Fig. 5. The performance of synchronous sensors is better than that of uniformly staggered sensors when $\lambda$ is small. For medium or large $\lambda$, uniformly staggered sensors are almost always a better choice. With a high $\lambda$, the target is difficult to track and predict and there is more uncertainty associated with the motion prediction. The error associated with the prediction of the target state from time $k$ to time $k + 1$, and hence the volume of the gate (which is proportional to $\sqrt{|S(k)|}$) will increase very quickly with time. Therefore, it is better to use staggered sensors to reduce the update time interval and keep the prediction error low.

In Fig. 6 and Fig. 7, the performances as functions of the false alarm density are shown. With a high $P_d$ and a high $\lambda$, uniformly staggered sensors always outperform synchronous sensors regardless of the variations of the false alarm density.

From these simulation results, we can draw the conclusion that when sensors have high $P_d$ and the target has medium or high $\lambda$, uniformly staggered sensors instead of synchronous sensors should be used. Under such cases, the superiority of staggered sensors over synchronous sensors is robust to the variations of the false alarm density.

V. SENSORS WITH DIFFERENT PERFORMANCES

In practice, usually the sensors of a tracking system have different performances. Therefore, it is important to explore the best staggering schemes for such cases.

All results are obtained through simulation and shown in Figs. 8 through 11, where the optimal staggering strategy and the performance improvement of optimal staggering over uniform staggering are illustrated.

First, we assume that the two sensors have the same measurement noise level ($\sigma_{\omega_1} = \sigma_{\omega_2}$), but have different $P_d$. We fix $P_{d_1}$ as 0.9 and find optimal staggering schemes for different values of $P_{d_2}$. The results are shown in Figs. 8 and 9. From Fig. 8, when two sensors’ $P_d$ are quite different ($P_{d_2}$ is small), the best strategy is to use them synchronously to get as much information as possible at one sampling time; when their $P_d$ are close to each other ($P_{d_2}$ is high too), the optimal strategy is uniform staggering or very close to it. From Fig. 9, we can see that the degradation by using uniformly staggered sensors are tolerable.

Next we study the case where the two sensors have the same $P_{d_1}$, but have different measurement noise levels. We define $r$ as the ratio between the two sensors’ measurement
Without loss of generality, we assume that sensor $S_1$ is no worse than sensor $S_2$, i.e., $r \geq 1$. The equivalent measurement noise variance of the composite sensor (when two sensors are synchronous and measurement-origin uncertainty is absent) can be found by the following equation:

$$
\frac{1}{\sigma^2_{\omega_c}} = \frac{1}{\sigma^2_{\omega_1}} + \frac{1}{\sigma^2_{\omega_2}} = \frac{1 + r^2}{r^2} \frac{1}{\sigma^2_{\omega_1}}
$$

(28)

To make the calculation of the target maneuvering index possible for different $r$, we keep $\sigma^2_{\omega_c}$ as a constant ($\sigma^2_{\omega_c} = \frac{1}{2} m^2$). Note the composite sensor and its variance is only conceptually useful and is introduced to facilitate the calculation of the maneuvering index. The target maneuvering index is then defined, in terms of $\sigma_{\omega_c}$ as

$$
\lambda = \frac{\sigma_{\nu} T}{\sqrt{2}\sigma_{\omega_c}}
$$

(29)

The results are shown in Figs. 10 and 11. From Fig. 10, when $r$ is 1 (two sensors have the same $\sigma_{\omega}$), the optimal strategy is uniform staggering; when $r \geq 2$, the optimal is to use synchronous sensors again. From Fig. 11, we can see that the performance of uniformly staggered sensors is much worse than that of optimally staggered sensors.

From the cases we have studied and numerical results obtained, an interesting observation is that when measurements from two sensors have very dramatic difference, we should use them synchronously to make the best use of the measurements from the sensor with poor quality. On the other hand, if the two sensors are quite similar in terms of their performances, the best strategy is to stagger them uniformly over time.

VI. Summary

In a cluttered environment, we have investigated the effects of temporally staggered sensors on the target tracking performance of a two-sensor system, and compared them to those of synchronous sensors. In this paper, we used a PDAF filter to handle the measurement-origin uncertainty. Data is fused in a centralized fashion for the system with synchronous sensors. The system performances are compared in terms of track life and in-track percentage.

For a system that has two sensors with equal performances, when sensors have high probability of detection $P_d$ and the target has medium or high maneuvering index $\lambda$, uniformly staggered sensors instead of synchronous sensors should be used. Otherwise, the two kinds of systems have little difference in terms of track life. For the case where two sensors’ performances are different, we have obtained the optimal staggering schemes. If the qualities of the measurements from the two sensors are dramatically different, synchronous sensors should be chosen. On the other hand, if the two sensors have quite similar performances, the best strategy is to stagger them uniformly over time.
ACKNOWLEDGMENTS

This work was supported by the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Army Research Office under Grant DAAD19-00-1-0352.

REFERENCES

Fig. 8: Optimal staggering time $T_1$ (based on 7000 Monte Carlo runs), with false alarm density $0.01/m$, $P_{d_1} = 0.9$, $\lambda = 10$, $\sigma_{\omega_1} = \sigma_{\omega_2} = 1m$, $T=1s$.

Fig. 9: Average track life (based on 7000 Monte Carlo runs), with false alarm density $0.01/m$, $P_{d_1} = 0.9$, $\lambda = 10$, $\sigma_{\omega_1} = \sigma_{\omega_2} = 1m$, $T=1s$.

Fig. 10: Optimal staggering time $T_1$ (based on 7000 Monte Carlo runs), with false alarm density $0.01/m$, $P_{d_1} = P_{d_2} = 0.9$, $\lambda = 10$, $\frac{\sigma^2_{\omega_1} + \sigma^2_{\omega_2}}{\sigma^2_{\omega_1} + \sigma^2_{\omega_2}} = \frac{1}{2}m^2$, $T=1s$.

Fig. 11: Average track life (based on 7000 Monte Carlo runs), with false alarm density $0.01/m$, $P_{d_1} = P_{d_2} = 0.9$, $\lambda = 10$, $\frac{\sigma^2_{\omega_1} + \sigma^2_{\omega_2}}{\sigma^2_{\omega_1} + \sigma^2_{\omega_2}} = \frac{1}{2}m^2$, $T=1s$. 