Distributed Target Tracking with Propagation Delayed Measurements

Umut Orguner
Division of Automatic Control
Department of Electrical Engineering
Linköping University
581 83, Linköping, Sweden.
umut@isy.liu.se

Fredrik Gustafsson
Division of Automatic Control
Department of Electrical Engineering
Linköping University
581 83, Linköping, Sweden.
fredrik@isy.liu.se

Abstract – This paper presents a framework for making distributed target tracking under significant signal propagation delays between the target and the sensors. Each sensor considered makes estimation using its own measurements compensating for the involved signal propagation delay using a deterministic sampling based algorithm proposed previously. Since the individual sensor readings might not be enough to localize the target, the sensors have to share their estimates with each other at specific time instants and correct their individual estimates. This work is mainly related to how this estimate correction and fusion should be carried out. An internal covariance approximation which keeps consistency but at the same time bypasses the track correlation problem is proposed. The results are illustrated on a challenging two-sensor bearings-only tracking scenario.

Keywords: Distributed estimation, multiple sensors, propagation delay, state estimation, target tracking, track fusion, covariance intersection, largest ellipsoid.

1 Introduction

Sensors used in target tracking all observe emitted (passive sensors) or reflected (active sensors) energy from the target. The classical standard sensor is the active radar, and common examples of passive sensors include vision (EOR, IR) and radar warning systems. The observation delay in these sensors is negligible since the speed of light is much larger than the speed of the target.

However, one trend in sensor networks is to use standard low cost sensors as microphones and geophones on land, and sonar in water. The assumption of negligible target speed compared to the speed of the media cannot always be made here. Consider for instance an array of sensors that by coherent signal processing techniques gives a bearing measurement to a previous position of the target. This time delay depends on the distance between the sensor and the target at that unknown time instant. This gives a further dimension to target tracking, where both the target state and the observation delays are unknown. With many (arrays of) sensors in the network, the observations are completely asynchronized over space.

Recently the delayed measurements problem has been investigated under a novel framework in [1]. In this paper we generalize the solution of [1], which we call propagation delayed measurement filter (PDMF or PDM filter) from here on, to multiple sensor observations. We follow a distributed estimation strategy where the sensors can communicate their local estimates with each other. Each sensor processes the local measurements using PDMFs and makes intermittent corrections to the local estimates using the received estimates from other sensors. This property makes the current work directly related to the track fusion concept.

The area of track fusion is mainly concerned about the correlation between the estimates to be fused. Even if the sensors used in a network collects measurements independent of each other, local processing of the measurements in the presence of common process noise in the target dynamics makes the local estimation errors correlated [2]. Moreover, the existence of data feedback loops can cause rumor propagation all over the network that would cause inconsistencies, overconfidence and in turn even filter divergence. The proposed solutions for the track correlation problem range from the ones requiring extra information transmission (e.g. Kalman filter gains [3]) or extra processing (e.g. information decorrelation [4, 5]) in order to compensate for the correlation to correlation independent methodologies like the covariance intersection [6, 7] and the largest ellipsoid algorithm [8, 9]. A very interesting survey and comparison of the possible approaches was presented recently in [10]. In this study, we adapt the largest ellipsoid algorithm in order to make the fusion of the local sensor estimates with the received estimates of the other sensors in the form of a Kalman filter measurement update.

The remaining parts of the paper are organized as follows. We illustrate the state estimation problem involved in this paper via a simplified example in Sec-
tion 2. Section 3 makes a formal problem definition. The single sensor solution of the problem, i.e., the PDM filter of [1], is summarized in Section 4. Section 5 gives the main results of the paper generalizing PDMF to multiple sensors. In Section 6, we present the results of a simulation study on a two-sensor bearings-only tracking scenario. Conclusions are drawn in Section 7.

2 Simplified Example

Leaving the general and formal problem formulation to Section 3, we here make a simplified introduction to the problem. Consider a case where we have perfect knowledge of the target state vector $x_{t_k-1}$ of a target at time $t_{k-1}$ and that the target position $p_{t_{k-1}}$ is a subset of the state vector. One sensor shown as s gets an observation related to $x_{t_k-\Delta t_k}$ at time $t_k$. The delay $\Delta t_k$ can be described as a function of the position (and hence the state) of the target at time $t_k-\Delta t_k$ using the physics rules of signal propagation in the medium as

$$\Delta t_k = d_{tk}(x_{t_k-\Delta t_k}).$$

On the other hand, using an assumed or known target dynamics, we can obtain a prediction of $x_{t_k-\Delta t_k}$ from perfectly known $x_{t_{k-1}}$ as

$$x_{t_k-\Delta t_k} = f(t_k-x_{t_k-\Delta t_k}(x_{t_{k-1}})).$$

Consider, for instance, the case in Figure 1 where we observe a target whose scalar state $z_i$ is the position on the x-axis. The target has the known state $x_0 = 0$ at time $t_{k-1} = 0$ and moves with a known constant speed $v_x$ along the x-axis. At time $t_k$, a sound sensor $s_1$ positioned on the y-axis value $y_{s_1}$ collects the bearing $\phi_{s_1}$ of the target (corresponding to time $t_k - \Delta t_k^s$). Then, the specific models corresponding to (1) and (2) would be

$$\Delta t_k = \frac{1}{v_x} \sqrt{y_{s_1}^2 + x_{t_k-\Delta t_k}^2},$$

$$x_{t_k-\Delta t_k}^s = v_x(t_k - \Delta t_k),$$

respectively, where $v_x$ is the speed of sound. These two equations have a solution $\Delta t_k^s > 0$ satisfying

$$v_x^2(t_k - \Delta t_k^s)^2 + y_{s_1}^2 = (v_x\Delta t_k^s)^2$$

as shown in Figure 1. Now, instead of solving the parabolic equation for $\Delta t_k^s$, one can define a recursion for $\Delta t_k^s$ with the initial value e.g. $\Delta t_k^s(0) = 0$ by substituting (4) into (3) to get

$$\Delta t_k^s(m + 1) = \frac{1}{v_x} \sqrt{y_{s_1}^2 + [v_x(t_k - \Delta t_k^s(m))]^2}$$

which can be shown to converge to the positive root of if $v_x < v_s$.

Just as in the case of this simple example, in a more general setting, (1) and (2) together define an implicit equation for $\Delta t_k^s$ (like (2)) whose solution can be obtained with iterative techniques similar to (6). This type of implicit and in general nonlinear constraints and their inclusion into the estimation process were examined in the work [1]. In this simplified scenario, we have neglected three sources of uncertainty:

- The initial state $x_{t_{k-1}}$ is actually random.
- A process noise term must be added into the simplified description (2).
- The propagation time itself through (1) might be uncertain due to possible reasons such as the uncertain position of the sensor.

The PDMF filter of [1] covers all the uncertainties mentioned above for the single sensor case and is based on including the delay $\Delta t_k$ in the state vector while processing the observation taken at time $t_k$. Since the measurements observe the delayed state values $x_{t_k-\Delta t_k}$, the augmented state is formed as $[x_{t_k}^T, \Delta t_k^s]$. In the multiple sensor case even if the sensors are synchronized to acquire measurements at a specific time $t_k$, due to the different sensor positions, each of them observes the target at a different time instant $t_k - \Delta t_k^s$ (see the case of two sensors illustrated in Figure 1). While making distributed estimation, the form of the states $[x_{t_k}^T, \Delta t_k^s]T$ to be updated makes the problem much more involved than the standard track fusion. This is because the state element $x_{t_k-\Delta t_k}$ in each local sensor estimate has an ambiguous time stamp that depends on the density of the $\Delta t_k^s$ component of the state in the same sensor.

3 Problem Definition

We consider the following discrete-time nonlinear state space model

$$x_{t_{k+1}} = f(x_{t_{k+1}, t_k} + w_{t_{k+1}, t_k})$$

Figure 1: Simplified example where a constant speed target on the x-axis is observed with sound sensors acquiring bearing information.
where \( \{x_{tk} \in \mathbb{R}^{n_s}\} \) is the state sequence with initial distribution \( x_{t_0} \sim p_0(x_{t_0}) \). We here adopt an implicit simplified notation such that the system state dynamics given by (7) is a discretized version of a corresponding continuous time dynamics
\[
\dot{x}_t = f(x_t) + w_t.
\] (8)

In (7), \( t_k \in \mathbb{R} \) is an arbitrary time value and \( f_{x_{tk+1},t_k}(.) \) is the state transition function transforming \( x_{tk} \) to \( x_{tk+1} \) according to continuous time dynamics \( f(.) \). We also assume that the time sequence \( \{t_k\}_{k=0}^{\infty} \) is non-decreasing and therefore the transformation involved in (7) is not necessarily invertible. \( \{w_{tk} \in \mathbb{R}^{n_w}\} \) is a white process noise sequence with distribution \( w \sim \mathcal{N}(0, Q(t_k)) \). Here it is important to emphasize that \( w_{tk}, t_k \in \mathbb{R} \) models the lumped effects of a continuous independent increment process noise \( w_t \) between the time instants \( t_k \) and \( t_{k+1} \).

The discrete delayed measurements \( \{y_{tk} \in \mathbb{R}^{n_y}\} \) of this system are collected by the sensors \( s=1, \ldots, S \) as
\[
y_{tk} = h_{tk}(x_{tk} - \Delta_s^k) + v_{tk}^s
\] (9)
where \( \Delta_s^k \) is the amount of delay in the measurement \( y_{tk}^s \) and \( \{v_{tk}^s \in \mathbb{R}^{n_v}\} \) is a white measurement noise sequence independent from the process noise with distribution \( v_{tk}^s \sim \mathcal{N}(0, R(t_k)) \). We here assume that we know about the time delay \( \Delta_s^k \) in the form of an implicit equation (which we call \( c_k^s \)) as follows
\[
c_k^s : \quad \Delta_s^k = d_{tk}^s(x_{tk} - \Delta_s^k) + \tau_k^s
\] (10)
where \( d_{tk}^s(.) \) is in general a nonlinear function of the delayed state value \( x_{tk} - \Delta_s^k \) and \( \{\tau_k^s \in \mathbb{R}\} \) is a white noise sequence independent from the process and measurement noise with distribution \( \tau_k^s \sim \mathcal{N}(0, \bar{R}^s(t_k)) \). Our main motivation for selecting such an expression for the time delay sequence \( \Delta_s^k \) is the case of a passive sound sensor whose delay expression is given as
\[
\Delta_s^k = \frac{\|p(t_k^s - \Delta_s^k) - p_{ts}^s\|}{v_s} + \tau_k^s
\] (11)
where \( p(t_k^s - \Delta_s^k) \) is the position of the target at time \( t_k^s - \Delta_s^k \) (which is a function of the delayed state \( x_{tk} - \Delta_s^k \) and hence the form of (10)), \( p_{ts}^s \) is the position of the sensor \( s \) at time \( t_k^s \) and \( v_s \) is the speed of sound. The noise term \( \tau_k^s \) then represents unpredictable effects in the transmission of the sound (pressure) wave in the environmental conditions or the possible uncertainty in the position of the sensor. In this work, we consider each constraint \( c_k^s \) of (10) as a piece of information to include into the estimation process and we show the cumulative information of constraints up to and including time \( n \) as \( c_n \triangleq \{c_k^s\}_{k=0}^{n} \). Although the constraints themselves are not random variables, in the following, we are going to use them as the arguments of the probability density functions in given conditions and Bayes rules and these must be interpreted information-wise.

**Problem Definition:** Given the state dynamics (7) and measurement relation (9), find a possibly approximate expression for the density \( p(x_{tk}|y_{0:k}^{0:S}, c_{0:k}^{1:S}) \).

## 4 PDM Filter

The defined problem above was originally proposed and analyzed in [1] where a deterministic sampling based solution, PDM filter, is proposed for the single sensor case. The method, in a Bayesian framework, completes a recursion for the density \( p(x_{tk} - \Delta_s^k, \Delta_k^s|y_{0,k}^0, c_{0:k}^0) \). \( x_{tk} - \Delta_s^k \) and \( \Delta_k^s \) are assumed to form the state \( \xi_k^s \) of the problem as
\[
\xi_k^s \triangleq \left| x_{tk}^s - \Delta_s^k \Delta_k^s \right|^T.
\](12)
It is assumed that the density \( p(x_{tk} - \Delta_s^k, \Delta_k^s|y_{0,k}^0, c_{0:k}^0) \) is Gaussian as
\[
p(x_{tk} - \Delta_s^k, \Delta_k^s|y_{0,k}^0, c_{0:k}^0) = \mathcal{N}(\xi_k^s; \xi_k^s|k,k^s, \Sigma_k|k,k^s)
\](13)
where the notation \( \mathcal{N}(\xi; \mu, \Sigma) \) stands for a Gaussian probability density function which has a mean \( \mu \) and covariance \( \Sigma \) evaluated at the dummy variable \( \xi \). Notice that, in Step-1, the constraint information \( c_k^s \) as well as state time update has to be incorporated into the density \( p(x_{tk} - \Delta_s^k, \Delta_k^s|y_{0,k}^0, c_{0:k}^0) \). This is done using a recursion (similar to (6)) given in a theorem in [1] on the \( \sigma \)-points generated from the sufficient statistics \( \xi_k^s|k-1|k-1, k \) and \( \Sigma_k^s|k-1|k-1 \). In other words, we find \( \xi_k^s|k|k \) and \( \Sigma_k^s|k|k \) from the transformed \( \sigma \)-points generated from \( \xi_k^s|k-1|k-1, \Sigma_k^s|k-1|k-1 \). Due to the space constraints in this paper, the reader is referred to [1] for further details of this procedure.
If the function $h_{k}^{s}(\cdot)$ is linear, then the measurement update in Step-2 can be done using the Kalman filter measurement update optimally. If a nonlinearity is involved, suboptimal approaches like EKF or UKF measurement updates can be applied.

We have noted above that the estimate of $x_{t_{1}^{s}-\Delta_{k}^{s}}$ is meaningless without the density of $\Delta_{k}^{s}$ that would specify the uncertainty in the time stamp. Hence in addition to the main loop of the algorithm which is formed by Step-1 and Step-2, we also have Step-3 that projects $x_{t_{1}^{s}-\Delta_{k}^{s}}$ to a determined future time relying on the uncertainty description of $\Delta_{k}^{s}$. Even if the original system dynamics $f(\cdot)$ is linear, the overall transformation from $\xi_{k}^{s}$ to $x_{t_{1}^{s}}$ in (14) becomes nonlinear due to the delay term. Suboptimal approaches like EKF or UKF transformation strategies can be used. The schematic diagram of the delay compensation algorithm for a single sensor is illustrated in Figure 2 in terms of the densities involved in the process.

5 Distributed Multiple Sensor Generalization

In this section, we are going to generalize the PDM filter to multiple sensors. When multiple sensors track a common target, we assume that each sensor makes its own delay compensated estimation using PDM filter. We also assume that the sensors are able to communicate their time stamped estimates (and covariances) with each other. Having obtained another sensor’s transmitted time stamped state estimate (and covariance), the receiving sensor must be able to incorporate this information into its PDMF estimation loop. This necessitates the update of the sufficient statistics (i.e., the state $\xi_{k}^{s|k,k}$ and the covariance $\Xi_{k}^{s|k,k}$) of the receiving sensor with the transmitted state estimate in a consistent manner.

In the following parts of this section, without loss of generality, we are going to concentrate on the case where the receiving and the transmitting sensors are denoted with sensor indices $s_{1}$ and $s_{2}$ respectively. We are going to show the transmitted quantities as $t^{s_{2}}$, $\hat{x}^{s_{2}}$ and $P^{s_{2}}$ which are the time stamp, state estimate and the covariance at the time stamp respectively. Notice here that the time stamp information $t^{s_{2}}$ relates to the target time and not the sensor time. In other words, $t^{s_{2}}$ is the delay compensated time value specifying that $x_{t^{s_{2}}} \sim N(x_{t^{s_{2}}}; \hat{x}^{s_{2}}, P^{s_{2}})$. The receiving sensor’s current PDMF state estimate and covariance are shown as $\xi_{k}^{s_{1}}$ and $\Xi_{k}^{s_{1}}$ respectively where we shortened the subscripts like $k|k$, $k$ for simplicity. The aim in this section is therefore updating $\xi_{k}^{s_{1}}$ and $\Xi_{k}^{s_{1}}$ using $t^{s_{2}}, \hat{x}^{s_{2}}$ and $P^{s_{2}}$.

The methodology that we are going to apply to this problem in this section is to devise a method that requires no additional information (like the correlation degree etc.) than the ones that are mentioned above. The sensors, at any time, are able to transmit information to other available sensor sites without taking care of the cross-correlation that might have been formed during the previous communications. On the other hand, the updating algorithm at the receiving side should still be able to fight inconsistency in order not to be overconfident. To achieve this aim, we here propose a type of internal ellipsoid method.

We are going to consider the transmitted (and received) quantities $t^{s_{2}}, \hat{x}^{s_{2}}$ and $P^{s_{2}}$ (or some possibly complicated transform of them) as measurements and then update the estimate $\hat{\xi}^{s_{2}}$ and the covariance $\Xi_{k}^{s_{2}}$ using this hypothetical measurement. In order to evade the much more complicated out of sequence measurement problem, we assume that the receiving sensor first projects the measurements to the current time $t > t_{1}^{s_{2}} \geq t_{1}^{s_{1}} - \Delta_{k}^{s_{1}}$ using the model (7). Standard EKF and UKF based approaches can be used here to get the resulting mean and covariance we call as $\hat{x}^{s_{2}}$ and $P^{s_{2}}$ respectively. Now instead of the transmitted quantities $t^{s_{2}}, \hat{x}^{s_{2}}$ and $P^{s_{2}}$, we consider the projected (predicted) mean $\hat{x}^{s_{1}}$ and covariance $P^{s_{1}}$ as a measurement of the state $\xi^{s_{1}}_{k}$. However, since the quantities $\hat{x}^{s_{2}}$, $P^{s_{2}}$ are correlated with the receiving sensor quantities because of

1. the common process noise

2. possible previous communication (and hence the fusion of the local estimates) between the sensors, using $\hat{x}^{s_{2}}$ and $P^{s_{2}}$ directly is not possible without the knowledge of the cross correlation. In the following sub-sections, we are going to make this measurement update interpretation of the fusion process possible by deriving consistent measurements using a correlation independent methodology.

5.1 Modeling Fusion Process as a Measurement Update

The standard track fusion solutions require one to transmit extra information (like Kalman gains [3]) or to keep track of previous communications [5]. Therefore, the algorithms that can avoid inconsistencies in spite of unknown correlations between the estimates are of great importance. Two such algorithms were already proposed in the literature with names covariance intersection [6, 7] and largest ellipsoid algorithm [8] (also called as internal ellipsoid approximation [9]). Defining the unit level set (ULS) of a covariance $\Sigma$ as the set $\{z \in \mathbb{R}^{n} | z^{T}\Sigma^{-1}z \leq 1\}$, both of these approaches rely on the fact that the ULS of the fused covariance of two correlated estimates lies inside the intersection of the corresponding ULSs of the covariances of the local estimates for all possible correlations, i.e.,

$$\{z | z^{T}P^{-1}z \leq 1\} \subset \{z | z^{T}P_{1}^{-1}z \leq 1\} \cap \{z | z^{T}P_{2}^{-1}z \leq 1\}$$

where we consider two arbitrarily correlated local estimates $\hat{x}_{1}$, $\hat{x}_{2}$ with corresponding local covariances $P_{1}$, $P_{2}$ and we show the optimally fused estimate and covariance (via ML criterion) as $\hat{x}$ and $P$ respectively. In
\[ x_{n} - \Delta_{n} = f_{n} - \Delta_{n}^{2} - \Delta_{n}^{1} (x_{n} - \Delta_{n}^{1}) + w_{n} - \Delta_{n}^{2} - \Delta_{n}^{1} \]
\[ \Delta_{k}^{*} = d_{k} (x_{n} - \Delta_{n}^{1}) + \tau_{k}^{*} \]
\[ y_{k}^{*} = h_{k} (x_{n} - \Delta_{n}^{1}) + v_{k}^{*} \]
\[ p(x_{n} - \Delta_{n}^{1}, \Delta_{k}^{*} | y_{k}^{*}, c_{0,k}^{*}) \]
\[ \text{Prediction} \]
\[ p(x_{n} - \Delta_{n}^{1}, \Delta_{k}^{*} | y_{k}^{*}, c_{0,k}^{*}) \]
\[ \text{Update} \]
\[ p(x_{n} - \Delta_{n}^{1}, \Delta_{k}^{*} | y_{k}^{*}, c_{0,k}^{*}) \]
\[ \text{Marginalization} \]
\[ p(x_{n} - \Delta_{n}^{1}, \Delta_{k}^{*} | y_{k}^{*}, c_{0,k}^{*}) \]

**Figure 2:** Deterministic sampling based delay compensation algorithm: Single sensor case.

**Figure 3:** Covariance intersection and largest ellipse algorithm in two dimensional case \((n_{x} = 2)\). Parts of this figure was taken from [8]. Similar figures also appear in [6, 7].

the two dimensional case \((n_{x} = 2)\), this identity can be illustrated as in Figure 3 where the dotted ellipses represent the boundaries of some of the possible fused covariances for different cross correlations. The covariance intersection (CI) and the largest ellipsoid algorithm (LEA) calculate the fused covariance as

- the smallest covariance whose ULS contains the intersection of the corresponding ULSs of the local covariances
- the largest covariance whose ULS is inside the intersection of the ULSs of the corresponding local covariances respectively.

In this part, we would like to interpret the update from \(\hat{x}_{1}, P_{1}\) to \(\hat{x}, P\) using the quantities \(\hat{x}_{2}, P_{2}\) as a Kalman filter measurement update. Note that, for this purpose, covariance intersection would prove to be useless because in some directions, the fused covariance \(P_{CI}\) might be larger than the initial covariance \(P_{1}\). This would therefore result in an indefinite (or possibly negative definite) measurement covariance. On the other hand, the fused covariance of the largest ellipsoid algorithm is always smaller than or equal to the initial covariance \(P_{1}\) in all directions. Although this is better than the CI case, we can still have a problem of unbounded covariance (zero information) for some directions in the state space. The solution to this problem is to discard the uninformative directions from the measurement and its covariance by deriving a lower dimensional measurement. This is investigated in the following subsection in the specific context of LEA.

### 5.2 Largest Ellipsoid Algorithm

Suppose the singular value decomposition of \(P_{1}\) is given as \(P_{1} = U_{1} \Lambda_{1} U_{1}^{T}\). Here, we define the transformation \(T_{1} = \Lambda_{1}^{-1/2} U_{1}^{T}\). We then form the matrix \(P_{3} = T_{1} P_{2} T_{1}^{T}\) and we suppose that the singular value decomposition of \(P_{3}\) is given as \(P_{3} = U_{3} \Lambda_{3} U_{3}^{T}\). We now define the transformation \(T_{2} = U_{3}^{T} T_{1}\). It is now easy to show that if we make a coordinate change such that \(x' = T_{2} x\), then under this transformation, we have the equivalent covariances

\[
P_1' = T_2 P_2 T_2^T = I_{n_x} \quad \text{and} \quad P_2' = T_2 P_2 T_2^T = \Lambda_3. \tag{15}
\]

Below, for expressing this algorithm mathematically, we show the \(i\)th element of a vector \(x\) by \([x]_i\) and \(ij\)th element of a matrix \(P\) by \([P]_{ij}\). We also define the set of indices \(I \triangleq \{ i | \Lambda_3_{ii} < 1 \}\). Then, the fused estimate \(\hat{x}'\) and the covariance \(P'\) under the \(T_2\) transform can be calculated as

\[
[x']_i \triangleq \begin{cases} [x'_1]_i & i \notin I \\ [x'_2]_i & i \in I \\ \end{cases} \quad [P']_{ij} \triangleq \begin{cases} [P'_1]_{ij} & i = j, i \notin I \\ [P'_2]_{ij} & i = j, i \in I \\ 0 & i \notin j \end{cases}
\]

This estimate and covariance must be back transformed into the original coordinate axes of the problem to obtain the final fused estimate \(\hat{x}_{LEA}\) and the covariance.

---

1This specific form is due to the fact that \(P_{1}\) is symmetric.
\[ P_{LEA} \text{ as follows.} \]
\[
\dot{x}_{LEA} = T_2^{-1} \dot{x}' \quad P_{LEA} = T_2^{-1} P'T_2^{-T}. \quad (16)
\]
Notice that
\[
P_{LEA}^{-1} = P_{1}^{-1} + T_2^T (P')^{-1} T_2 - P_{1}^{-1} P_{1}^{-1} T_2 = T_2^{-1} \left[ (P')^{-1} - (P_1')^{-1} \right] T_2.
\quad (17)
\]
Now by analogy to the Kalman filter measurement update in the information form, we can interpret \( P_{LEA} \) as the measured updated version of \( P_1 \) with a measurement that has the error covariance \( R \) and the measurement matrix \( T_2 \). As mentioned in sub-section 5.1, the problem with this interpretation is that the matrix \( \left[ (P')^{-1} - (P_1')^{-1} \right] \) might be singular and therefore \( R \) may become undefined. In that case, we have to find a measurement which is lower dimensional than \( n_x \) which has a well-defined covariance. We know that the zero diagonal elements of the matrix \( \left[ (P')^{-1} - (P_1')^{-1} \right] \) correspond to the indices \( i \) that are not in the set \( I \). Hence, calling the cardinality of \( I \) as \( n_s \), we define

\[ \text{new diagonal matrices } P_r, P'_s \in \mathbb{R}^{n_s \times n_s} \text{ which have only the diagonal elements of } P' \text{ and } P'_1, \text{ respectively, corresponding to indices } i \in I; \]

\[ \text{new measurement matrix } T_r \in \mathbb{R}^{n_s \times n_s} \text{ which contains the rows of } T_2 \text{ corresponding to indices } i \in I; \]

\[ \text{new vectors } \hat{x}'_r, \hat{x}'_1 \in \mathbb{R}^{n_s} \text{ which have only the elements of } \hat{x}' \text{ and } \hat{x}'_1, \text{ respectively, corresponding to indices } i \in I. \]

Now, we can form our measurement model as
\[ y_{LEA} = T_r x_t + v_{LEA} \quad (19) \]
where \( v_{LEA} \sim \mathcal{N}(v_{LEA}, 0, R_{LEA}) \). The measurement \( y_{LEA} \) and the measurement covariance \( R_{LEA} \) are given as
\[ R_{LEA} = \left[ (P_r')^{-1} - (P'_s)^{-1} \right]^{-1}, \quad (20) \]
\[ y_{LEA} = R_{LEA} \left[ (P'_r)^{-1} \hat{x}'_r - (P'_r)^{-1} \hat{x}'_1 \right]. \quad (21) \]

Note that this time \( R_{LEA} \) is positive definite (and diagonal) by definition and both \( P'_r \) and \( P'_r \) are identity matrices. Also it is important to emphasize that in the case that \( I = \emptyset \) then this means that \( \hat{x}_{LEA} = \hat{x}_1 \) and \( P_{LEA} = P_1 \). Therefore one does not have to do any measurement update which is consistent with the fact that \( y_{LEA} \) would also be empty.

The measurement \( y_{LEA} \), which is related to the state via the measurement matrix \( T_r \) in the presence of the noise \( v_{LEA} \), is an independent (consistent) measurement that should update \( \hat{x}_1, P_1 \) in order to obtain the same result as that would be obtained from the consistent fusion of \( \hat{x}_1, P_1 \) and \( \hat{x}_2, P_2 \). We can substitute this measurement into an estimation process with complex relations to get simple and still consistent solutions which is basically what is done in the next sub-section.

### 5.3 Application of the Results to the Original Problem

The above results are now going to be applied to the problem of consistently updating the mean \( \hat{x}_k \) and the covariance \( \Xi_k \) with the quantities \( \hat{x}'_k \) and the covariance \( P'_k \).

**Algorithm 2** Suppose we are given the mean \( \hat{x}'_k \) and the covariance \( \Xi_k \), the predicted received mean \( \hat{x}'_2 \) and the predicted received covariance \( P'_2 \).

1. **Instrumental Prediction:** Predict the mean \( \hat{x}'_k \) and the covariance \( \Xi_k \) using
\[
\hat{x}_t = \hat{x}'_k - \Delta_k \left( x_{k-1}' - \Delta_k \right) + w_t \quad (22)
\]
to obtain \( \hat{x}'_s \) and \( P'_s \) which are to be used only for calculating the consistent measurement noise parameters.

2. **LEA:** Set \( \hat{x}_1 = \hat{x}'_s \), \( P_1 = P'_s \), \( \hat{x}_2 = \hat{x}'_2 \) and \( P_2 = P'_2 \) in the LEA algorithm given above to obtain \( \hat{x}'_{LEA} \) and \( P_{LEA} \). Using the related quantities calculate the consistent measurement \( y_{LEA} \), measurement matrix \( T_r \) and measurement noise covariance \( R_{LEA} \).

3. **Measurement Update:** Make the measurement update of the mean \( \hat{x}'_k \) and the covariance \( \Xi_k \) using the measurement relation
\[
y_{LEA} = T_r x_t + v = T_r f_{t, \hat{x}'_s} - \Delta_k \left( x_{k-1}' - \Delta_k \right) + T_r w_{t, \hat{x}'_s} + v \quad (23)
\]
where \( v \sim \mathcal{N}(v, 0, R_{LEA}) \).

Notice that for the steps 1 and 3, one can use either EKF or UKF based approaches. It is also interesting that very similar equations can serve for different interpretations in prediction and measurement updates.

### 6 Simulation Study

In this section, the performance of PDM filter equipped with LEA measurement update is going to be compared on a simulated target tracking scenario with other possible approaches which include:

- An EKF and LEA combination which totally neglects that there is a delay in sensing.
- An EKF and LEA combination which tries to compensate for the delay by using its state estimate.

We consider a two-dimensional bearings-only tracking problem with two stationary sensors. The single target in the scenario makes a clockwise coordinated turn of radius 500m with a speed about 200km/h beginning in y-direction with the initial position \([-500m, 800m]\) for 45s. The tracking sensors called \( s_1 \) and \( s_2 \) acquire bearing data of the target corrupted by i.i.d.
Gaussian measurement noises with zero mean and standard deviation of 0.05 radians ≈ 3 degrees with sampling period $T = 1\text{ sec}$ both beginning at $t_0 = 4\text{ secs}$. Each sensor, therefore, gets a total number of 42 measurements in the interval [4secs, 45secs]. The sensors communicate their estimates $\hat{x}^s_k$, $\hat{v}^s_k$ and corresponding covariances $P^s_k$ every $T_s = 4\text{ secs}$ starting at $t = 7\text{ secs}$. This communication is assumed to be lossless and instantaneous.

The true target trajectory and the sensor positions used in the example are illustrated in Figure 4. The target motion is modeled in the filters with a discretized coordinated turn model with unknown constant turn rate (i.e., the turn rate is also a state variable) and with Cartesian velocity. Therefore, the state of the target is given as $x_k = [p^k, v^k, \omega_k]^T$, where $p$, $v$ and $\omega$ variables denote the position, velocity and turn rate respectively. In all simulations, we selected the standard deviations for the turn rate and speed as $\sigma_\omega = 0.01\text{ rad/sec}^2$ and $\sigma_v = 1\text{ m/sec}^2$ respectively. The delay expression used in the simulations is given by (11) with $v_k \approx 344\text{ m/sec}$ and the mean and the variance of $\tau_k$ have been taken as zero, i.e., $\tau_k \equiv 0$.

Three different algorithms are tested with the following brief descriptions (and abbreviations).

- **EKF+LEA**: Standard EKF algorithm which uses the measurement equation

$$y_k = h_k^e(x^t_k) + v_k^e.$$ (23)

This algorithm ignores the delays completely. The estimates of the local EKFs are corrected with the other sensor’s estimate at fusion times using LEA.

- **PDMF+LEA**: Local PDM filters’ estimates are corrected by the other sensor’s estimate at fusion times using the LEA based measurement update in Algorithm 2. The parameters of the PDM filters are taken the same as the ones used in the example of [1].

- **EKFD+LEA**: The local EKFs know that what they estimate is $x^t_k - \Delta^t_k$. Therefore, they calculate a delay estimate $\hat{\Delta}^t_k$ from (11) using their last state estimate and then extrapolate the state estimate by $\Delta^t_k$ to form their output estimates. Local estimates are corrected by the other sensor’s estimate at fusion times using the LEA based measurement update similar to PDMF+LEA.

All the filters to be run have been initialized with the initial state values $\hat{x}^{s_1}_0 = [-400, 900, 10, 10, -0.1]^T$, $\hat{x}^{s_2}_0 = [-600, 1000, 10, 10, -0.1]^T$, initial covariances $P^{s_1}_0 = \text{diag}(100^2, 200^2, 50^2, 50^2, 0.1^2)$, $P^{s_2}_0 = \text{diag}(200^2, 100^2, 50^2, 50^2, 0.1^2)$ for the sensors $s_1$ and $s_2$ respectively. The initial position estimates and covariances are illustrated in Figure 4 with asterisk signs and dotted curves respectively.

A total number of 1000 Monte-Carlo runs have been performed by changing the realization of the measurement noise in each one. The RMS position and velocity errors for the (fused) local estimates of $s_1$ and $s_2$ are shown along with the corresponding clairvoyant parametric Cramer-Rao lower bounds (PCRLBs) (calculated with known delays) in Figures 5 and 6 respectively. The fusion (correction) times of the local estimates are immediately recognizable from the figures.
EKF algorithm neglecting all delay information gets the worst results. PDMF based algorithm clearly has better results than the heuristic EKFD based algorithm especially after $t = 25$ secs. Towards the end of the scenario, the local estimation errors of PDMF+LEA are lowest and very close to each other similar to the local PCRLB curves. The other algorithms’ local estimation errors, on the other hand, still remain quite separated from each other which is a manifestation of poor delay compensation.

7 Conclusions

This paper has presented the multi sensor generalization of a new approach to handle non-uniform and dynamically changing delays in measurements called PDM filter. In a distributed estimation framework, the fusion of the local PDMF states is achieved in the form of a measurement update. The consistent measurements and its parameters used in the update are derived using the largest ellipsoid algorithm.

The simulation study on a two-sensor bearings-only target tracking scenario proves that the robust estimation properties of PDMF observed in [1] for the single sensor case are carried also to the case of multiple sensors.

Acknowledgments

The authors gratefully acknowledge fundings from SSF Strategic Research Center MOVIII and the Vinnova/FMV TAIS project ARCUS. The strategic motivation and practical relevance for this contribution stem from the Vinnova, SSF (Swedish Foundation for Strategic Research) and KKS Institute Excellence Centre for Advanced Sensors, Multisensors and Sensor Networks (FOCUS).

References


