TIME-SPACE-SEQUENTIAL ALGORITHMS FOR DISTRIBUTED BAYESIAN STATE ESTIMATION IN SERIAL SENSOR NETWORKS

Ondrej Hlinka and Franz Hlawatsch

Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Vienna, Austria; e-mail: ondrej.hlinka@nt.tuwien.ac.at

ABSTRACT

We consider distributed estimation of a time-dependent, random state vector based on a generally nonlinear/non-Gaussian state-space model. The current state is sensed by a serial sensor network without a fusion center. We present an optimal distributed Bayesian estimation algorithm that is sequential both in time and in space (i.e., across sensors) and requires only local communication between neighboring sensors. For the linear/Gaussian case, the algorithm reduces to a time-space-sequential, distributed form of the Kalman filter. We also demonstrate the application of our state estimator to a target tracking problem, using a dynamically defined “local sensor chain” around the current target position.

Index Terms—Parameter estimation, state estimation, sequential Bayesian filtering, distributed inference, sensor networks, Kalman filter, target tracking.

1. INTRODUCTION

Distributed state estimation in wireless sensor networks has received some attention recently (e.g., [1–8]). It is often desirable to limit inter-sensor communications to nearby sensors [1]. Estimation schemes and network topologies using a fusion center may be less suited in this respect because the sensor data have to be communicated to a (possibly distant) central location where the final processing takes place. In contrast, fully decentralized schemes without a fusion center such as [4–7] use extensive in-network processing and neighbor-to-neighbor communication to achieve low energy consumption and high robustness to node failure.

Here, we propose a fully distributed, optimal, Bayesian algorithm for estimating a time-dependent state vector whose current value is sensed by a serial sensor network without a fusion center. The algorithm requires only local communication between neighboring sensors. It is space-sequential in that a “partial” posterior density is propagated from one sensor to the next and updated using the local measurement. Repeating this process for successive state values, we obtain a time-space-sequential, distributed form of the Kalman filter.

The proposed algorithm generalizes the space-sequential estimation method presented in [6] in that it allows the state to evolve temporally according to a state-transition model. It also generalizes the distributed method for estimating a time-dependent state presented in [7] in that the state is sensed by several sensors simultaneously.

This paper is organized as follows. In Section 2, we describe the sensor network and the state-space model. The time-space-sequential estimation algorithm is developed in Section 3. For the linear/Gaussian case, a time-space-sequential distributed form of the Kalman filter is presented in Section 4. In Section 5, the general estimation algorithm is applied to a target tracking problem and its performance is demonstrated via simulation results.

2. SYSTEM MODEL

We consider a random state (or parameter) vector \( x_n \) that evolves with time \( n \) according to the state-transition equation

\[
x_n = g_n(x_{n-1}, u_n), \quad n = 1, 2, \ldots.
\]

Here, \( g_n(\cdot, \cdot) \) is a known, generally time-dependent function and \( u_n \) is white driving noise (i.e., \( u_n \) and \( u_{n'} \) are independent for \( n \neq n' \)) with known probability density function (pdf) \( f(u_n) \). At time \( n \), state \( x_n \) is sensed by \( K_n \) sensors as depicted in Fig. 1. The measurement equations of the individual sensors are

\[
z_{n,k} = h_{n,k}(x_n, w_{n,k}), \quad k = 1, 2, \ldots, K_n, \quad n = 1, 2, \ldots,
\]

where \( z_{n,k} \) is the measurement vector at time \( n \) and sensor \( k \), \( h_{n,k}(\cdot, \cdot) \) is a known, generally time- and sensor-dependent function, and \( w_{n,k} \) is measurement noise with known pdf \( f(w_{n,k}) \) and \( f(w_{n,k}^{(1)}, w_{n,k}^{(2)}) \) independent unless both \( n = n' \) and \( k = k' \). The set of sensors may change with time \( n \). The state sequence \( x_n \) is initialized by the random vector \( x_0 \) whose pdf \( f(x_0) \) is known. The initial state \( x_0 \) and the sequences \( u_n \) and \( w_{n,k} \) are all independent.

The \( K_n \) measurement equations (2) can be combined as \( z_n = h_n(x_n, w_n) = (z_{n,1}^{(1)}, \ldots, z_{n,K_n}^{(1)})^T \) and \( w_n = (w_{n,1}^{(1)}, \ldots, w_{n,K_n}^{(1)})^T \). We also define the total measurement vector up to time \( n \), \( z_{1:n} = (z_1^{(1)}, \ldots, z_n^{(1)})^T \), where \( z_{1:0} \) is the empty vector.

Equations (1) and (2) together with our statistical assumptions determine the state-transition pdf \( f(x_n|x_{n-1}) \) and the likelihood functions \( f(z_n|x_n) \) and \( f(z_n|x_0) \). Thus, these pdf’s will be assumed known. Furthermore, it follows from our assumptions that the current state \( x_n \) is conditionally independent of all previous measurements \( z_{1:n-1} \) given the previous state \( x_{n-1} \), i.e.,

\[
f(x_n|x_{n-1}, z_{1:n-1}) = f(x_n|x_{n-1}),
\]

This work was supported by the FWF project “Statistical Inference” (S10603-N13) within the National Research Network SISE.
and the current measurement \( z_n \) is conditionally independent of all previous measurements \( z_{1:n-1} \) given the current state \( x_n \), i.e.,

\[
 f(x_n | x_1, z_{1:n-1}) = f(z_n | x_n) .
\]  

(4)

3. TIME-SPACE-SEQUENTIAL DISTRIBUTED BAYESIAN STATE ESTIMATION

Estimating the state \( x_n \) from \( z_{1:n} \), i.e., all measurements up to the current time \( n \), essentially amounts to calculating the posterior pdf \( f(x_n | z_{1:n}) \). Indeed, from \( f(x_n | z_{1:n}) \), optimal Bayesian estimators like the minimum mean-square error (MMSE) and maximum a posteriori (MAP) estimators, as well as quantities characterizing estimation performance, can be derived [9]. We will now develop a time-space sequential, distributed algorithm for calculating \( f(x_n | z_{1:n}) \).

3.1. Time recursion

Using \( z_{1:n} = (z_{1:n-1}^T, z_n^T)^T \), Bayes’ rule, and (3) and (4), it is shown e.g. in [10] that the current posterior \( f(x_n | z_{1:n}) \) can be calculated time-sequentially from the previous posterior \( f(x_n | z_{1:n-1}) \) and the current measurement \( z_n \). This recursion consists of two steps. In the prediction step, the “predicted posterior” \( f(x_n | z_{1:n-1}) \) is calculated from the previous posterior \( f(x_{n-1} | z_{1:n-1}) \) according to

\[
 f(x_n | z_{1:n-1}) = \int_{x_{n-1}} f(x_n | x_{n-1}) f(x_{n-1} | z_{1:n-1}) dx_{n-1} .
\]

(5)

In the update step, the current measurement \( z_n \) is used to convert \( f(x_n | z_{1:n-1}) \) into the current posterior \( f(x_n | z_{1:n}) \) according to

\[
 f(x_n | z_{1:n}) = \frac{f(x_n | z_n) f(x_{n-1} | z_{1:n-1})}{f(z_n | x_{n-1} | z_{1:n-1})} ,
\]

(6)

with

\[
 f(z_n | x_{n-1} | z_{1:n-1}) = \int_{x_n} f(z_n | x_n) f(x_n | z_{1:n-1}) dx_n .
\]

3.2. Space recursion

The time update (6) requires centralized processing of the total (all sensors) measurement vector \( z_{1:n} \). However, we will now show that (6) can be reformulated in a space-sequential, distributed manner in which sensor \( k \) receives a “partial posterior” from sensor \( k-1 \), updates it using the local measurement \( z_{1:n}^{(k)} \), and transmits the updated partial posterior to sensor \( k+1 \). This space-sequential update is analogous to that presented in [6], with the difference that [6] does not consider a temporal state transition.

Let \( z_{1:n}^{(k)} = (z_{1:n}, z_{1:n}^{(k)})^T \) denote the current measurements of all sensors up to sensor \( k \), where \( z_{1:n}^{(k)} \) and \( z_{1:n}^{(1:0)} \) is the empty vector. Because of (2) and our statistical assumptions, \( z_{1:n}^{(k)} \) is conditionally independent of the current measurements of all spatially previous sensors \( z_{1:n}^{(k-1)} \) and all (temporally) previous measurements \( z_{1:n-1} \) given the current state \( x_n \), i.e.,

\[
 f(x_n | z_{1:n}, z_{1:n-1}^{(k-1)}, z_{1:n-1}) = f(x_n | z_n^{(k)}) .
\]

(7)

Note that this is a refinement of (4). Using (7), we are able to apply the recursion of [6], which we will here rederive in our setting. Suppose that at time \( n \), sensor \( k \) receives from sensor \( k-1 \) the “partial posterior” \( f(x_n | z_{1:n-1}^{(k-1)}, z_{1:n-1}) \) that incorporates all previous measurements \( z_{1:n-1}^{(k-1)} \) and all current measurements up to sensor \( k-1 \), \( z_{1:n}^{(k-1)} \). Sensor \( k \) now has to calculate the updated partial posterior \( f(x_n | z_{1:n}^{(k-1)}, z_{1:n-1}) \) that incorporates also the local measurement \( z_{1:n}^{(k)} \). Using \( z_{1:n}^{(k)} = (z_{1:n}^{(k-1)} T, z_n^{(k)})^T \) and Bayes’ rule (with background knowledge \( z_{1:n}^{(k-1)} | z_{1:n-1} \)), we have

\[
 f(x_n | z_{1:n}^{(k-1)}, z_{1:n-1}) = \frac{f(x_n | z_n^{(k)}, z_{1:n}^{(k-1)}, z_{1:n-1})}{f(z_n^{(k)} | x_{n-1}^{(k-1)}, z_{1:n}^{(k-1)}, z_{1:n-1})} \cdot
\]

(8)

where (7) was used in the last step. The denominator is calculated as

\[
 f(z_n^{(k)} | x_{n-1}^{(k-1)}, z_{1:n-1}) = \int_{x_n^{(k-1)}} f(z_n^{(k)} | x_n^{(k-1)}) f(x_n^{(k-1)} | z_{1:n-1}) dx_n^{(k-1)} \cdot
\]

(9)

Equation (8) provides the desired space recursion that converts \( f(x_n | x_{1:n}^{(k-1)}, z_{1:n-1}) \) into \( f(x_n | x_{1:n}^{(k)}, z_{1:n-1}) \), using the local measurement \( z_{1:n}^{(k)} \). At the last sensor \((k = K_n)\), we obtain the “total posterior” \( f(x_n | z_{1:n}) \) since \( f(x_n | z_{1:n}^{(K_n)}, z_{1:n-1}) = f(x_n | z_{1:n}) \). From \( f(x_n | z_{1:n}) \), an estimate of \( x_n \) and quantities characterizing its performance can be calculated.

The space recursion is initialized by \( f(x_n | x_{1:n}^{(1:0)}, z_{1:n-1}) = f(x_n | z_{1:n-1}) \). This is the predicted posterior, which is calculated according to (5), using the known \( f(x_n | z_{1:n-1}) \) and the previous posterior \( f(x_{n-1} | z_{1:n-1}) \).

3.3. Time-space sequential algorithm

The space-sequential formulation (8) of the time update (6) yields a time-space-sequential algorithm for calculating the sequence of posteriors \( f(x_n | z_{1:n}) \), \( n = 1, 2, \ldots \). This algorithm is fully distributed and uses only communication between neighboring sensors. Its operation at time \( n \) can be summarized as follows (cf. Fig. 1).

1) Preparation. The last sensor \((k = K_n)\) of the previous time recursion \((n-1)\) transmits the previously calculated posterior \( f(x_n | z_{1:n-1}) \) to the current first sensor \((k = 1)\). (It is assumed that these two sensors are spatially close; in many cases, they will be identical, so that no transmission is required.) Initialization: This preparation step is not performed for \( n = 1 \).

2) Prediction. The first sensor \((k = 1)\) converts \( f(x_{n-1} | z_{1:n-1}) \) into the predicted posterior \( f(x_n | z_{1:n-1}) \) according to (5). Note that this prediction step does not require the current measurement \( x_n \). Initialization: For \( n = 1 \), \( z_{1:n-1}^{(1:0)} = z_{1:0} \) is empty, and thus (5) degenerates to the following relation involving the initial statistics \( f(x_0) \):

\[
 f(x_1) = \int_{x_0} f(x_1 | x_0) f(x_0) dx_0 .
\]

(10)

3) Space updates. The sensors \( k = 1, \ldots, K_n \) sequentially carry out the space update (8), using the local measurements \( z_{1:n}^{(k)} \) and communication between neighboring sensors as described in Section 3.2. At the last sensor \((k = K_n)\), the total posterior \( f(x_n | z_{1:n}) \) is obtained. Initialization: The space updates are initialized at the first sensor \((k = 1)\) by the predicted posterior \( f(x_n | z_{1:n-1}) \), which was calculated by the first sensor in Step 2. For \( n = 1 \), the space update (8) takes on the following degenerate form:

\[
 f(x_1 | z_{1:n}^{(1)}) = \frac{f(x_1 | z_1^{(1)}) f(z_1^{(1)} | x_1^{(1)})}{f(z_1^{(1)})} .
\]
with (see (9))
\[ f(z_{k}^{(k)}|z_{1:k-1}) = \int_{x_{1}} f(z_{k}^{(k)}|x_{1}) f(x_{1}|z_{1:k-1}) \, dx_{1}, \]

for \( k = 1, \ldots, K_{1} \). This is initialized by \( f(x_{1}|z_{1:k-1}) = f(x_{1}), \) which was calculated in Step 2 (see (10)).

4. TIME-SPACE-SEQUENTIAL DISTRIBUTED KALMAN FILTER

Next, we study the special case of a linear/Gaussian state-space model. Consider a linear state-transition equation (1)

\[ x_{n} = g_{n}(x_{n-1}, u_{n}) = A_{n}x_{n-1} + u_{n} \]

and linear measurement equations (2)

\[ z_{n}^{(k)} = h_{n}^{(k)}(x_{n}, w_{n}^{(k)}) = H_{n}^{(k)}x_{n} + w_{n}^{(k)}, \quad k = 1, 2, \ldots, K_{n} \]

with Gaussian driving and measurement noises \( u_{n} \sim \mathcal{N}(0, C_{u_{n}}), w_{n}^{(k)} \sim \mathcal{N}(0, C_{w_{n}^{(k)}}) \) and Gaussian initial state \( x_{0} \sim \mathcal{N}(\mu_{x_{0}}, C_{x_{0}}) \).

As before, all random vectors \( u_{n}, w_{n}^{(k)}, x_{0} \) are independent. The model parameters \( A_{n}, H_{n}^{(k)}, C_{u_{n}}, C_{w_{n}^{(k)}}, \) and \( C_{x_{0}} \) are known.

The measurement equations (11) can be combined as z_{n} = H_{n}x_{n} + w_{n}^{(k)}, with \( H_{n} \triangleq H_{n}^{(1)}, \ldots, H_{n}^{(K_{n})} \) and the block-diagonal covariance matrix \( C_{w_{n}} \triangleq \text{diag}(C_{w_{n}^{(1)}}, \ldots, C_{w_{n}^{(K_{n})}}) \).

The pdf’s \( f(x_{n}|z_{1:n}), f(z_{n}|x_{n}), f(x_{n}|z_{1:n-1}) \) and \( f(x_{n}|x_{n-1}) \) are now Gaussian and hence fully described by their means and covariances. In particular, let \( x_{n|n} \triangleq \mathbb{E}[x_{n}|z_{1:n}] \) the MMSE estimate of \( x_{n} \) from \( z_{1:n} \), \( C_{n|n} \triangleq \mathbb{E}[x_{n}x_{n}^{T}|z_{1:n}] - x_{n|n}x_{n|n}^{T} \) and predicted covariance \( C_{n|n-1} \) according to (12), (13). Initialization: For \( n = 1 \), the prediction step (12), (13) is initialized by the initial statistics, i.e., \( x_{n|1} = x_{0} = \mu_{x_{0}} \) and \( C_{n|1} = C_{x_{0}} \).

Space updates. Each sensor \( k \) receives from sensor \( k-1 \) the partial mean and covariance \( \hat{x}_{n|n-1}^{(k-1)}, C_{n|n-1}^{(k-1)} \), uses (16), (17) to calculate the updated partial mean and covariance \( x_{n|n-1}^{(k)}, C_{n|n-1}^{(k)} \) incorporating the local measurement \( z_{n}^{(k)} \), and transmits the updated quantities to sensor \( k+1 \). At the last sensor \( (k = K_{n}) \), the total mean (MMSE estimate) \( x_{n|n} \) and total covariance \( C_{n|n} \) are obtained. Initialization: These space updates are initialized at the first sensor \( (k = 1) \) by the predicted mean \( x_{n|n-1}^{(1)} \) and predicted covariance \( C_{n|n-1}^{(1)} \), which were calculated by the first sensor in Step 2.

5. APPLICATION TO TARGET TRACKING

We now apply the distributed estimation algorithm of Section 3 to the tracking of a target moving in a plane. The target’s 2-D Cartesian coordinate vector \( x_{n} \) is to be estimated from the sound emitted by the target. We assume that the target’s speed and heading are uniformly distributed in \([0, v_{\text{max}}]\) and in \([0, 2\pi]\), respectively. It follows that the state-transition pdf \( f(x_{n}|x_{n-1}) \) is uniform inside a disk of radius \( T_{\text{max}} \) centered at \( x_{n-1} \) (here, \( T \) is the sampling period, i.e., the time duration of the discrete-time interval \([n-1, n]\)).

The target is sensed by acoustic amplitude sensors \([1]\) deployed in the plane. At time \( n \), \( K_{n} \) sensors that are close to the current target position \( x_{n} \) form a “local sensor chain” (LSC) emulating the sequential processing was previously derived (however, not in the context of distributed sensor processing) in [12]; it is enabled by the block-diagonal structure of \( C_{w_{n}} \).

The space-sequential formulation (16), (17) of the time update (14), (15) yields a "time-space-sequential", distributed form of the Kalman filter that uses only communication between neighboring sensors. The operation of this algorithm at time \( n \) can be summarized as follows.

1) Preparation. The last sensor of the previous time recursion (time \( n-1 \)) transmits \( x_{n-1|n-1} \) and \( C_{n-1|n-1} \) to the current first sensor \((k = 1)\). Initialization: This step is not performed for \( n = 1 \).

2) Prediction. The first sensor \((k = 1)\) converts \( x_{n-1|n-1} \) and \( C_{n-1|n-1} \) into the predicted mean \( x_{n|n-1}^{(1)} \) and predicted covariance \( C_{n|n-1}^{(1)} \).

3) Space updates. Each sensor \( k \) receives from sensor \( k-1 \) the partial mean and covariance \( x_{n|n-1}^{(k-1)}, C_{n|n-1}^{(k-1)} \), uses (16), (17) to calculate the updated partial mean and covariance \( x_{n|n-1}^{(k)}, C_{n|n-1}^{(k)} \) incorporating the local measurement \( z_{n}^{(k)} \), and transmits the updated quantities to sensor \( k+1 \). At the last sensor \((k = K_{n})\), the total mean (MMSE estimate) \( x_{n|n} \) and total covariance \( C_{n|n} \) are obtained. Initialization: These space updates are initialized at the first sensor \((k = 1)\) by the predicted mean \( x_{n|n-1}^{(1)} \) and predicted covariance \( C_{n|n-1}^{(1)} \), which were calculated by the first sensor in Step 2.

2059
with $\Delta a \triangleq a_2 - a_1$ and $\Phi(y) \triangleq \frac{1}{\sqrt{2\pi}} \int_0^\infty e^{-u^2/2} \, du$.

Within the LSC, the space-sequential update of the posterior proceeds as described in Section 3. The total posterior is available at the last ($K_n$,th) sensor, which is also the first sensor of the LSC at the next time instant $n + 1$ (this new LSC is located around target position $x_{n+1}$, see Fig. 2). This first sensor performs the prediction step, and the predicted posterior initializes the space-sequential update procedure performed by the new LSC.

At any given time, the current LSC is formed dynamically using the following decentralized scheduling algorithm. The sensor currently active selects as the next sensor of the LSC the neighboring sensor with minimum distance to (the local MMSE estimate of) the target position. (The local MMSE estimate of the target position is the mean of the local partial posterior. We assume that each sensor knows the positions of its neighbors.) From (18), it then follows that sensors with a higher signal-to-noise ratio are more likely to be included in the LSC. While a sensor may appear only once in a given LSC, it may be reused in a subsequent LSC.

We simulated this target tracking method with 64 sensors randomly deployed in a rectangular region of size $1000 \text{m} \times 1000 \text{m}$, using target parameters $v_{\text{max}} = 180 \text{m}/\text{T}$, $a_1 = 90$, and $a_2 = 130$, noise variance $\sigma^2 = 0.0042$, and constant LSC size $K_n \equiv 10$. The posteriors and likelihoods were discretized using a 2-D sampling grid of $100 \times 100$ samples. (This simple discretization results in high-rate communications and has to be improved in practical applications—see, e.g., [7].) In Fig. 3, we show the empirical root mean-square error (RMSE) $\sqrt{E\{[\hat{x}^{(1:k)}_{n+1} - x_n]^2\}}$ of the local MMSE state estimate $\hat{x}^{(1:k)}_{n+1}$ (i.e., of the mean of the partial posterior $f(\hat{x}_n | x_n^{(1:k)}, z_{1:n-1})$) as a function of time $n$ and sensor index $k$. The RMSE was estimated by averaging $\|\hat{x}^{(1:k)}_{n+1} - x_n\|^2$ over 1000 simulation runs. It is seen that at fixed time $n$, the RMSE decreases with the number $k$ of sensor measurements included in the state estimate. However, when passing on to the next time, the RMSE again increases because the state has evolved to a new value. For one realization, we show in Fig. 4 the discretized partial posterior $f(\hat{x}_n | z^{(1:k)}_{n}, z_{1:n-1})$ and estimated target position for three consecutive sensor indices $k$ of an LSC. It is seen that the posterior becomes narrower for increasing $k$. (This behavior is not observed for all realizations, even though it is true on average, cf. Fig. 3).

6. CONCLUSION

We presented a distributed algorithm for nonlinear/non-Gaussian state estimation for the case where the current state is sensed by a serial sensor network. Our algorithm is sequential both temporally and across sensors, and it requires only local communication between neighboring sensors. In the linear/Gaussian case, it reduces to a time-space-sequential, distributed form of the Kalman filter. As a simple application, we considered target tracking by means of acoustic amplitude sensors, using a decentralized, dynamic choice of a “local sensor chain” around the current target position.

7. REFERENCES


