Priority List Sensor Scheduling using Optimal Pruning

Marco F. Huber and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory
Institute of Computer Science and Engineering
Universität Karlsruhe (TH), Germany
Email: marco.huber@ieee.org, uwe.hanebeck@ieee.org

***Abstract*—State estimation and reconstruction quality of distributed phenomena that are monitored by a network of distributed sensors is strongly affected by communication failures, which is a problem in real-world sensor networks. In this paper, we propose a novel sensor scheduling approach named *priority list sensor scheduling (PLSS)*. This approach facilitates efficient distributed estimation in sensor networks, even in case of unreliable communication, by prioritizing the sensor nodes according to local sensor schedules based on the predicted estimation error. It is shown that PLSS minimizes the expected estimation error for arbitrary packet-loss or transmission probabilities. As prioritizing sensor nodes requires the calculation of several sensor schedules, a novel pruning algorithm that preserves optimal schedules is also derived in order to significantly reduce the computational demand. This is accomplished by exploiting the monotonicity of the Riccati equation and the information contribution of individual sensor nodes in combination with a branch-and-bound technique.**

Keywords: Sensor scheduling, communication failures, Kalman filtering, optimal pruning.

I. Introduction

Advances in miniaturization, wireless communication, and sensor technologies facilitate the usage of large scale sensor networks for monitoring physical phenomena including, e.g., environmental monitoring, structural monitoring of buildings, and surveillance tasks [1]. For a meaningful and detailed view on the phenomenon, an intelligent processing of the data provided by the distributed sensor nodes is essential. To increase the operational lifetime of the sensor network, the measurement rate should be as low as possible, which leads to a decrease in information gain and consequently in estimation accuracy.

Sensor scheduling, which is also referred to as sensor selection or sensor management, is a promising solution to this trade-off. A sensor schedule specifies a time sequence of sensor nodes to be allocated for performing future measurements. The problem of determining such sensor schedules can be formulated as a stochastic control problem, where the sensor identity is the control variable affecting the quality of observations and the estimation process of the monitored phenomenon. Solving the control problem involves optimization of expected scheduler costs over time, where the sensors are treated jointly for improved estimation results [2]. Typical cost functions employed for sensor scheduling are based on information theoretic measures [3], [4] or on scalar functions of the error covariance matrix of the state estimate [5], [6].

In this paper, we assume that the phenomenon monitored by the network of sensors is described by means of a linear stochastic dynamic system. For such systems and for quadratic cost functions, a separation principle holds, i.e., the sensor schedule can be determined independently of the system control policy and independently of the measurements [7]. The optimal sensor schedule results from off-line performing a tree search.

Due to the distributed nature of sensor networks, the previously mentioned scheduling algorithms are less applicable: Scheduling has to be performed in a distributed way to ensure scalability, measurements and estimates have to be transmitted using wireless communication, which is to some degree unreliable. Thus, the communication network has to be taken into account when calculating sensor schedules. Concerning this matter, some results exist for information-directed routing [8] as well as for greedy scheduling heuristics in terms of time-delays [9] and deterministic systems [10]. The effect that sensors are currently not available or reachable due to transmission failures is not considered so far. Such failures directly affect the feasibility of the calculated sensor schedule; adequate rescheduling strategies become inevitable.

The proposed sensor scheduling approach, named priority list sensor scheduling (PLSS), extends classical approaches as it considers the unreliable communication network when selecting sensors for measurement. For selection purposes, a priority list is constructed based on the optimal local sensor schedules of the individual sensor nodes. Determining optimal schedules for prioritization is accomplished by minimizing a scalar cost function that is based on the error covariance of the system state. With a given prioritization, selecting valuable sensors is possible even if some nodes are currently unavailable due to communication failures. Within PLSS, prioritization and state estimation is accomplished in a distributed way. Thus, costly transmissions to a fusion center are avoided and additional robustness is achieved.

One key contribution of this paper is the analysis of the proposed scheduling scheme in consideration of the underlying communication network. Assuming a random graph, it is shown that selecting sensors according to the priority list minimizes the expectation over the predicted estimation error.

In order to prioritize the sensor nodes, several optimal sensor schedules have to be determined. Since sensor scheduling can be interpreted as resource allocation problem, determining optimal schedules over finite time-horizons is NP-hard. To deal with this complexity, many approximate algorithms with reduced computational demand have been proposed. Greedy or myopic scheduling algorithms only calculate the one-step-ahead solution [11], [12]. Another way is to employ pruning techniques for reducing the size of the search tree. Existing pruning techniques range from suboptimal methods, where conserving the optimal schedule is not guaranteed (see e.g. [13], [14]), but drastic savings in computational demand are possible, to optimal methods, where potentially many complete schedules have to be computed [3], but deleting the optimal schedule is impossible.

To avoid the high computational load of determining optimal schedules for prioritization purposes, a novel optimal pruning algorithm is also introduced in this paper. In contrast to existing methods typically based on branch-and-bound techniques, the so-called sensor information matrix is exploited. This matrix represents the information contribution of a sensor and implies a partial order on the sensors. Together with the monotonicity of the Riccati equation, complete schedules can be pruned early, while preserving the optimal schedule is guaranteed. It is also shown that the proposed pruning algorithm can be used in combination with existing branch-and-bound techniques.

In the next section, the problem of sensor scheduling for linear systems is formulated and a short review of the Riccati equation is given. The remainder of the paper is structured as follows. In Section III, the priority list sensor scheduling scheme is described, while its theoretical discussion is part of Section IV. The optimal pruning algorithm is derived in Section V. An example application from the field of vehicle tracking is used in Section VI to demonstrate the effectiveness of PLSS and optimal pruning by means of numerical simulations. The paper closes with conclusions and an outlook to future work.

II. Problem Formulation

This paper focuses on efficiently estimating the state of a physical phenomenon via a sensor network in discrete time within a finite state space. Generally, the phenomenon can be described by means of a set of (stochastic) partial differential equations. Thus, throughout the paper it is assumed that a spatial and temporal discretization of the model of the phenomenon is already given (for details see e.g. [15]).

A. System and Sensor Model

The temporal behavior of the monitored phenomenon is described by the linear discrete-time stochastic system equation

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}_k \underline{\boldsymbol{x}}_k + \mathbf{B}_k \underline{\boldsymbol{w}}_k \ .$$

Here, the state vector $\underline{\boldsymbol{x}}_k$ comprises the state variables to be estimated at discrete time steps $k = 0, 1, \ldots, N-1$, where N is the estimation time horizon. $\mathbf{A}_k \in \mathbb{R}^{(n \times n)}$ and $\mathbf{B}_k \in \mathbb{R}^{(n \times m)}$ are real-valued matrices, $\underline{\boldsymbol{w}}_k$ is white Gaussian noise with covariance matrix $\mathbf{C}_k^w$, and the initial state vector $\underline{\boldsymbol{x}}_0$ is also Gaussian with mean $\hat{\underline{x}}_0$ and covariance matrix $\mathbf{C}_0^x$.

For updating the state estimate, measurements obtained by S sensors are used. Each sensor $i \in \{1, \ldots, S\}$ is described by the linear discrete-time stochastic measurement equation

$$\hat{\underline{y}}_k^i = \mathbf{H}_k^i \underline{\boldsymbol{x}}_k + \underline{\boldsymbol{v}}_k^i \ ,$$

where $\hat{\underline{y}}_k^i \in \mathbb{R}^s$ is the current measurement, $\mathbf{H}_k^i \in \mathbb{R}^{(s \times n)}$ is the real-valued time-variant measurement matrix, and $\underline{\boldsymbol{v}}_k^i$ is zero-mean white Gaussian noise with positive definite covariance matrix $\mathbf{C}_k^{v,i}$ affecting sensor i. It is assumed that for $i \neq j$, $\underline{\boldsymbol{v}}_k^i$ and $\underline{\boldsymbol{v}}_k^j$ are uncorrelated. Furthermore, $\mathbf{A}_k$, $\mathbf{B}_k$, $\mathbf{C}_k^w$, $\mathbf{H}_k^i$ and $\mathbf{C}_k^{v,i}$ are known to all sensors.

If sensor i takes a measurement at time step k, the covariance of the observed system evolves according to the recursive algebraic Riccati equation

$$\mathbf{C}_{k+1}^x(i) = \mathbf{A}_k \mathbf{C}_k^x \mathbf{A}_k^{\mathrm{T}} + \mathbf{B}_k \mathbf{C}_k^w \mathbf{B}_k^{\mathrm{T}} - \mathbf{A}_k \mathbf{K}_k^i \mathbf{H}_k^i \mathbf{C}_k^x \mathbf{A}_k^{\mathrm{T}} \ , \quad (1)$$

with gain $\mathbf{K}_k^i = \mathbf{C}_k^x (\mathbf{H}_k^i)^{\mathrm{T}} \Big(\mathbf{H}_k^i \mathbf{C}_k^x (\mathbf{H}_k^i)^{\mathrm{T}} + \mathbf{C}_k^{v,i} \Big)^{-1}$, as in the well-known Kalman filter [16].

B. Sensor Scheduling

Given the covariance matrix $\mathbf{C}_0^x$ of the initial system state $\underline{\boldsymbol{x}}_0$, the optimal sensor sequence

$$u_{0:N-1}^* = \arg\min_{u_{0:N-1}} V(u_{0:N-1}) \ , \quad (2)$$

when selecting one sensor per time step, results from minimizing the function of the *predicted estimation error*[1]

$$V(u_{0:N-1}) = \sum_{k=0}^{N-1} g\left(\mathbf{C}_{k+1}^x(u_{0:k})\right) \ , \quad (3)$$

with $\mathbf{C}_{k+1}^x(u_{0:k})$ according to (1) when applying the sensor sequence $u_{0:k} = [u_0, u_1, \ldots, u_k]$ to $\mathbf{C}_0^x$, where u_n is the n-th element of $u_{0:k}$ indexing the sensor to be selected for measurement at time step n. The scalar function $g(\,\cdot\,)$ in (3) is used to quantify the size of the error covariance, where the trace or the determinant of $\mathbf{C}_{k+1}^x(u_{0:k})$ are typical choices. The extension to multiple measurements per time step is straightforward [6].

With (2), a discrete optimization problem is given, where the function of the predicted estimation error is optimized over a finite set of possible sensor schedules and over a finite time horizon. It is worth mentioning that (3) can be evaluated without knowledge of the actual measurement $\hat{\underline{y}}_k^i$ of sensor i. However, the covariance matrix is a function of the sensor schedule. So, with the assumption that the current estimate $\hat{\underline{x}}_k$ with covariance matrix $\mathbf{C}_k^x$ of $\underline{\boldsymbol{x}}_k$ can be successfully transmitted over the sensor network, the optimal sensor sequence can be determined off-line by an exhaustive tree search [7], since all possible sensor schedules form a path in a tree with depth N.

[1]The terms predicted estimation error and estimation error are used interchangeably throughout the paper.

Distributed sensor scheduling can easily be carried out in case of no communication failures. Initially, at time step $k = 0$, the so-called *leader node* s, i.e., the sensor node responsible for calculating the current estimate, can be determined directly in a distributed manner, since any sensor node knows the measurement matrix and noise of any other sensor. After performing the measurement, the leader node can hand over the leader role and thus, the updated state estimate to the next sensor of the sensor schedule $u^*_{0:N-1}$ and so forth.

In sensor networks, communication is typically carried out over a wireless medium. Thus, the assumption of always successfully transmitting the estimates is no longer valid. The communication link between two sensors is unreliable, i.e., the packet containing the current estimate may be dropped. In literature, this effect has not been considered so far when scheduling sensors for measurement.

III. Priority List Sensor Scheduling (PLSS)

For the typical case of a wireless communication between the sensors, it is possible that some sensors of the optimal sensor schedule $u^*_{0:N-1}$ are currently unavailable. To deal with this fact, several options arise: The measurement update for the current time step can be omitted or another sensor can be selected for measurement. In the following, we present a scheduling scheme that gives a practical solution to this problem, while theoretical investigations are the content of Section IV.

A. Network Model

The communication network of the sensors is modeled by means of a random directed graph G with S vertices (see e.g. [17]). An edge, i.e., a communication link from sensor i to sensor j, is established randomly and independently of other edges with probability $c_k^{i,j} \in [0, 1]$. In the following, $c_k^{i,j}$ is denoted as *probability of successful transmission* (short: *transmission probability*). These probabilities are not known to the sensor nodes and do not change in-between two consecutive time steps k and $k + 1$.

In some cases, the current estimate of the system state $\underline{\boldsymbol{x}}_k$ is processed without communication. It remains at the currently selected sensor. Such self-loops in G have probability $c_k^{i,i} = 1$.

B. Scheduling Scheme

The key idea of the proposed distributed sensor scheduling approach is to provide a prioritization of the sensors. The sensor with the highest priority at time step $k + 1$ is the first sensor of the sensor schedule with the overall minimum estimation error during time horizon N. The sensor with the second highest priority is the first sensor of the sensor schedule with the second lowest estimation error and so on. As illustrated in Fig. 1, at time step k the priority list for $S = 2$ sensors is calculated (framed by rounded box) by determining the optimal schedules $u^{*,1}_{k+1:N-1}$ and $u^{*,2}_{k+1:N-1}$ beginning with sensor $u_{k+1} = 1$ and $u_{k+1} = 2$, respectively. If the sensor schedule starting with $u_{k+1} = 1$ has the lowest cost, then sensor 1 is the sensor with highest priority and the priority

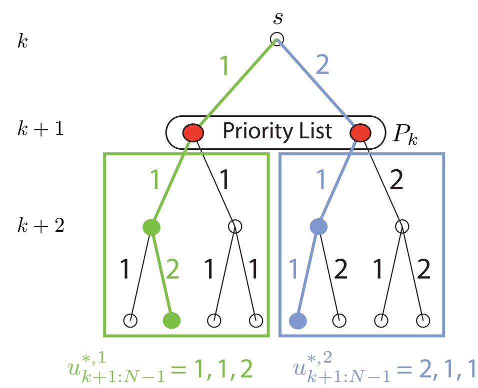

Figure 1. Priority list determination for two sensors.

list is $P_k = [1, 2]$. Otherwise, sensor 2 is the sensor with the highest priority and the priority list is $P_k = [2, 1]$.

In the proposed *priority list sensor scheduling (PLSS)* algorithm for any time step k, three operations have to be performed:

1. Priority List Calculation For each sensor i its so-called *optimal local sensor schedule* $u^{*,i}_{k+1:N-1}$ with $u_{k+1} = i$ is calculated according to (3). Ranking the sensors in ascending order with respect to the function of the estimation error $V^i_{k+1} := V(u^{*,i}_{k+1:N-1})$ yields the priority list

$$P_k = [p_k^1, p_k^2, \ldots, p_k^S] \ , \tag{4}$$

where $p_k^j = \arg\min_i \{V^i_{k+1} | i \in \{1, \ldots, S\} \backslash \{p_k^1, \ldots, p_k^{j-1}\}\}$. All these calculations take place at leader node s, which was selected at time step $k - 1$ for performing the measurement.

2. Reachability Check Leader node s requests one sensor at a time according to its rank in the priority list P_k and waits for its response. Requesting the sensors stops once the currently requested sensor responds.

3. Sensor Selection Sensor s hands over the leader role and thus the current state estimate $\hat{\underline{x}}_k$ as well as the state covariance $\mathbf{C}^x_k$ to this sensor.

At time step $k + 1$, the operations described above are repeated until the end of the time horizon is reached. It is obvious that in case of a transmission without packet losses, the sensor sequence resulting from the priority list approach is identical to the well-known solution neglecting communication constraints.

Remark 1 *Alternatively to requesting one sensor at a time during the reachability check, the current leader node could broadcast the request into the network and wait for response.*

IV. Minimizing the Expected Estimation Error

The intention of PLSS is to indirectly include information about the communication network when performing the reachability check rather than utilizing the transmission probabilities. This is motivated by the fact that existing methods for determining transmission probabilities are inadequate. While estimating the probabilities by using models of the communication channel or sensor movement is fairly complex [18], measuring the probabilities, e.g., by counting successfully transmitted packets leads to imprecise results.

A. Expected Estimation Error

By requesting the sensors according to the priority list (4), the first sensor of the *currently best reachable* local sensor schedule is selected for measurement. This procedure minimizes the *expected cost* or *expected estimation error*

$$\mathrm{E}\{V(u^i_{k+1:N-1})\}\ , \tag{5}$$

where $\mathrm{E}\{\,\cdot\,\}$ is the expectation with respect to the one-step transmission probabilities $c_k^{s,i}$. The following theorem summarizes this finding.

Theorem 1 (PLSS minimizes** (5)**)
PLSS minimizes the expected estimation error without being aware of the transmission probabilities $c_k^{s,i}$.

PROOF. The idea for proving this result is to show that any arbitrary ranking of sensor nodes can be reordered into the ranking of the priority list. This reordering is performed by first shifting the sensor with the highest cost to the end of the sensor list. Then, the sensor with the second highest cost is shifted to the second to last position and so. All shift operations correspond to a successive and monotonic reduction of the expected estimation error until the minimum of the priority list ranking is reached.

Without loss of generality, the arbitrary ranking of sensor nodes is given by the list $[1, 2, \ldots, S]$, where the leader node can be an arbitrary node. Furthermore, the time index k and the index s of the leader node are omitted for improved readability. With the transmission probabilities $c^i := c^{s,i}$, the expected estimation error is

$$\mathrm{E}\{V(u^i_{k+1:N-1})\} = \sum_{i=1}^{S} \left(\prod_{j=1}^{i-1} (1 - c^j) \right) \cdot c^i \cdot V^i\ . \tag{6}$$

When shifting the sensor i with the highest estimation error V^i toward its position in the priority list, sensor i's position has to be interchanged with its right neighbor $i+1$. The resulting list will be $[1, 2, \ldots, i-1, i+1, i, i+2, \ldots, S]$. In the next step, the position of sensor i is interchanged with sensor $i+2$ and so on. We will now show, that interchanging positions between sensor i and $i+1$, where $V^i \geq V^{i+1}$, successively reduces the expected estimation error. Here, the fact is used that interchanging two neighboring sensors only affects the summands i and $i+1$ in (6), i.e., we can disregard all other summands. Before interchanging the sensors, the variant part in (6) is

$$\prod_{j=1}^{i-1} (1 - c^j) \cdot \left(c^i \cdot V^i + (1 - c^i) \cdot c^{i+1} \cdot V^{i+1} \right)\ , \tag{7}$$

and after interchanging we have

$$\prod_{j=1}^{i-1} (1 - c^j) \cdot \left(c^{i+1} \cdot V^{i+1} + (1 - c^{i+1}) \cdot c^i \cdot V^i \right)\ . \tag{8}$$

Subtracting (8) from (7) leads to

$$\prod_{j=1}^{i-1} (1 - c^j) \cdot \left(c^i \cdot c^{i+1} \cdot V^i - c^i \cdot c^{i+1} \cdot V^{i+1} \right)\ ,$$

which is non-negative since $V^i \geq V^{i+1}$ and $c^j \in [0, 1]$. This holds for any initial sensor ranking. The minimization of (5) only depends on the cost values V^i and not on specific transmission probabilities. □

The result of Theorem 1 can be interpreted as follows: As long as the reachability of the sensors can be checked, the sensor nodes should by requested or selected according to their estimation error. This maximizes the chance to continue with the best local sensor schedule. Thus, PLSS provides a practical method for minimizing the estimation error under unreliable communication without utilizing computationally expensive but often imprecise methods for determining transmission probabilities. Furthermore, PLSS is generally applicable, since other effects like sensor breakdowns or topology changes can be directly handled.

B. Optimal Schedule

Applying PLSS and thus minimizing the expected estimation error allows for the randomness of communication failures. However, there is a gap between the resulting schedule of PLSS and the best possible, i.e., optimal schedule. If at most at one time step within the time horizon communication failures occur or only one sensor per time step is reachable, PLSS generates sensor schedules with minimum estimation error. In all other cases, it is not guaranteed that the schedule of PLSS is optimal. Optimal schedules in cases of random communication failures can only be calculated, if the occurrence of all communication failures and thus the reachability of each sensor at any time step is known a priori. Thus, an algorithm calculating optimal schedules is *acausal* and thus impractical, since it depends on future knowledge of the communication network. In Example 1, the difference between acausal optimal scheduling and PLSS is demonstrated.

Example 1 (Acausal Scheduling vs. PLSS)
In this example, a two-dimensional time-invariant system characterized by the matrices $\mathbf{A} = 1.5 \cdot \mathbf{I}$, $\mathbf{B} = \mathbf{I}$ and $\mathbf{C}^w = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix, is observed for $N = 2$ time steps. The initial system state $\underline{x}_0$ has the mean $\hat{\underline{x}}_0 = [0, 0]^{\mathrm{T}}$ and covariance matrix $\mathbf{C}_0^x = \mathbf{I}$. Three sensors are used for observation, with measurement matrices

$$\mathbf{H}^1 = \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix}\ ,\ \mathbf{H}^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\ ,\ \mathbf{H}^3 = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}\ ,$$

and noise covariances $\mathbf{C}^{v,1} = 0.5 \cdot \mathbf{I}$, $\mathbf{C}^{v,2} = 1.5 \cdot \mathbf{I}$ and $\mathbf{C}^{v,3} = 0.1 \cdot \mathbf{I}$. The trace is used for $g(\,\cdot\,)$ in (3).

Assuming that at the first time step only sensor two and three and in the second time step only sensor one and three are reachable, it can be seen in Fig. 2 that PLSS results in an estimation error of 13.8, while the acausal schedule has a value of 12.1. At the first time step PLSS constructs the priority list $P_0 = [3, 1, 2]$, since the schedule $3, 2$ is optimal in case of no communication errors, the next best schedule is $1, 2$. Since sensor two is not reachable at the next time step, the schedule $2, 3$ becomes optimal.

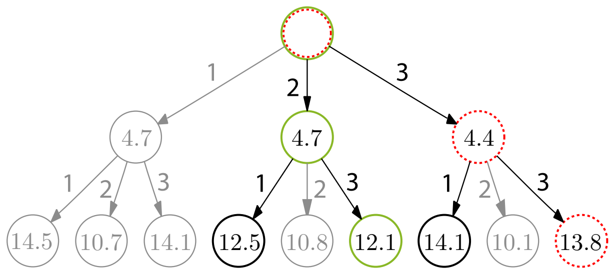

Figure 2. Difference between optimal schedule (green, sequence $2, 3$) and PLSS (red, dotted, sequence $3, 3$).

V. Optimal Pruning

When calculating the priority list, searching each tree of a sensor for the optimal local sensor scheduling is required. If all possible sensor sequences are considered to yield the optimal schedule, the complexity of searching grows exponentially with the length of the time horizon N. The increase in complexity can be reduced by employing pruning techniques. Here, sub-trees are disregarded from the search, if they do not contain the optimal schedule. The computational demand can be drastically reduced if the identification of sub-trees that contain only suboptimal schedules occurs as early as possible.

In the following, a novel optimal pruning technique by employing the so-called *sensor information matrix* and the monotonic character of the Riccati equation (1) is introduced, where sub-trees are pruned without explicitly evaluating the Riccati equation. Furthermore, branch-and-bound techniques can be easily included, which leads to further savings in computation and improvement in pruning performance.

A. Sensor Information Matrix

For the sake of clarity and brevity it is assumed that $\mathbf{A}_k = \mathbf{I}$ and $\mathbf{B}_k = \mathbf{0}$[2]. Hence, the Riccati equation corresponds exactly to the measurement update step of the Kalman filter and the information form is given by [19]

$$\left(\mathbf{C}^x_{k+1}\right)^{-1} = \left(\mathbf{C}^x_k\right)^{-1} + \mathbf{H}^{\mathrm{T}}_k \left(\mathbf{C}^v_k\right)^{-1} \mathbf{H}_k \ ,$$

where $\left(\mathbf{C}^x_{k+1}\right)^{-1}$ is the Fisher information matrix. This equation can be interpreted as gain of information over $\underline{\boldsymbol{x}}_k$ when performing a measurement. In this context the matrix $\mathbf{M}_k := \mathbf{H}^{\mathrm{T}}_k \left(\mathbf{C}^v_k\right)^{-1} \mathbf{H}_k$ is denoted as *sensor information matrix.*

Theorem 2 (Order of Sensor Information Matrices)
Given the covariance matrix $\mathbf{C}^x_k$ and the sensor information matrices $\mathbf{M}^i_k$ and $\mathbf{M}^j_k$ of the two sensors i and j such that

$$\mathbf{M}^i_k \succeq \mathbf{M}^j_k \ , \tag{9}$$

i.e., $\mathbf{M}^i_k - \mathbf{M}^j_k$ is positive semi-definite, then $\mathbf{C}^x_{k+1}(i) \preceq \mathbf{C}^x_{k+1}(j)$.

[2]This assumption corresponds to a static system.

Proof. Multiplying both sides of (9) from left with $\mathbf{C}^x_k$, adding the identity matrix $\mathbf{I}$, inverting both sides, applying the matrix inversion lemma [19]

$$\left(\mathbf{A} + \mathbf{BC}\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{I} + \mathbf{CA}^{-1}\mathbf{B}\right)^{-1}\mathbf{CA}^{-1}$$

and multiplying both sides from right with $\mathbf{C}^x_k$ yields $\mathbf{C}^x_{k+1}(i) \preceq \mathbf{C}^x_{k+1}(j)$. □

So far, we know that if it is possible to determine the order (9) between two sensors i and j, selecting sensor i provides a smaller covariance at time step $k+1$. Due to the monotonic character of the Riccati equation this result is also true for all time steps $n \geq k$.

Theorem 3 (Monotonicity of Riccati Equation)
Given two covariance matrices $\mathbf{C}^x_k$ and $\tilde{\mathbf{C}}^x_k$ with $\mathbf{C}^x_k \succeq \tilde{\mathbf{C}}^x_k$, applying the Riccati equation (1) for sensor $u_k \in \{1, \ldots, S\}$ results in

$$\mathbf{C}^x_{k+1} \succeq \tilde{\mathbf{C}}^x_{k+1} \ .$$

Proof. A proof can be found in [6], Lemma 2. □

Thus, selecting arbitrary sensors does not effect an existing order of covariance matrices. This holds also for arbitrary sensor sequences.

Corollary 1
Given two covariance matrices $\mathbf{C}^x_k$ and $\tilde{\mathbf{C}}^x_k$ with $\mathbf{C}^x_k \succeq \tilde{\mathbf{C}}^x_k$, $\forall u_{k:n}$, $n \geq k$ exists a sensor sequence $\tilde{u}_{k:n}$ such that

$$\mathbf{C}^x_{n+1} \succeq \tilde{\mathbf{C}}^x_{n+1} \ .$$

Proof. Due to Theorem 3, at least the sensor sequence $u_{k:n} = \tilde{u}_{k:n}$ yields $\mathbf{C}^x_{n+1} \succeq \tilde{\mathbf{C}}^x_{n+1}$. □

Together with the following lemma the optimal pruning technique based on the sensor information matrix can be formulated. For a proof of the lemma see [20].

Lemma 1 *Suppose that $\mathbf{C}^x_k \succeq \tilde{\mathbf{C}}^x_k$, then $\mathrm{trace}(\mathbf{C}^x_k) \geq \mathrm{trace}(\tilde{\mathbf{C}}^x_k)$ and $|\mathbf{C}^x_k| \geq |\tilde{\mathbf{C}}^x_k|$.*

Corollary 2 (Information-Based Pruning (IBP))
Suppose that the covariance matrix $\mathbf{C}^x_k(u_{0:k-1})$ for the sensor sequence $u_{0:k-1}$ is given at time step k. If for two sensors u_k and $\tilde{u}_k$

$$\mathbf{M}^{u_k}_k \preceq \mathbf{M}^{\tilde{u}_k}_k \ ,$$

then $V \geq \tilde{V}$ for any sequence $u_{k+1:N-1}$, where $V := V(u_{0:N-1})$ and $\tilde{V} := V([u_{0:k-1}, \tilde{u}_k, u_{k+1:N-1}])$.

Proof. Follows directly from Theorem 2 and Corollary 1. □

Thus, without evaluating the Riccati equation and only by comparing the sensor information matrices it can be decided that only sensor $\tilde{u}_k$ needs to expanded for determining the optimal schedule, while the sub-tree of sensor u_k can be pruned. In the following example, the effectiveness of the proposed optimal scheduling scheme is illustrated.

Example 2 (Pruning based on Sensor Information Matrices) We consider the system from Example 1. The sensor information matrices of the three sensors are

$$\mathbf{M}^1 = \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \ , \ \mathbf{M}^2 = \begin{bmatrix} 2/3 & 0 \\ 0 & 2/3 \end{bmatrix} \ , \ \mathbf{M}^3 = \begin{bmatrix} 10 & 10 \\ 10 & 10 \end{bmatrix} \ .$$

It follows that $\mathbf{M}^3 \succeq \mathbf{M}^1$, while all other comparisons do not result in positive semi-definite differences. Due to the time-invariance of the system, sensor 1 never needs to be considered for determining the optimal schedule. The tree in Fig. 2 is reduced to 7 nodes.

B. Order of Sensors

When comparing sensor information matrices (or equivalently covariance matrices), the difference is in some cases indefinite, i.e., it is not determinable, if one sensor information matrix is “larger” than another (see $\mathbf{M}^1$ and $\mathbf{M}^2$ in Example 2). This is due to the fact that the order relation $\preceq$ of positive semi-definite matrices leads to partial orders. Thus, in general, not all sensors can be pruned at a specific time step.

There is one exception in case of scalar systems. Here, the partial order becomes a total order. Per time step, it is now possible to prune all sensors except of one, which is equivalent to selecting the sensor that minimizes the covariance at each time step. This greedy strategy leads automatically to the optimal sensor schedule. Similar results can be found in [21] for the case of multiple scalar systems observed by one sensor.

C. Branch & Bound

Applying the proposed pruning technique leads to a significant reduction of possible sensor schedules. However, due to the partial order, the remaining number of nodes in the search tree may still be large. To further prune the tree, the proposed technique is combined with branch-and-bound (B&B) pruning algorithms. B&B pruning is common for classical problems like traveling-salesman or resource allocation. Here, we use the B&B algorithm proposed in [3], where sub-trees not yet evaluated are bounded from below by the estimation error of their root node. This bound is also valid here, since (3) is cumulative with non-negative summands.

Combining B&B and the sensor information matrix based pruning yields the *information-based pruning* (IBP*)* algorithm (see Algorithm 1). Here, a node i is pruned, if its sensor information matrix is smaller than the sensor information matrix of an other node j or its estimation error value $V(i)$ (calculated using (3)) is larger than the error value V_B of the currently best, completely evaluated schedule. Otherwise, the node i is expanded (lines 6–14). If the algorithm completes a schedule whose estimation error is less than V_B, then V_B is set to this error value (line 1–2). For accelerating the decrease of the bounding value V_B and thus for improving the pruning performance, the sensors are sorted in ascending order according to the error values $V(i)$ (line 10).

Algorithm 1 IBP*(u), where u is the currently expanded sensor node. Initially, the best error bound is set to $V_B = \infty$.

```
1: if k = N and V(u) ≤ V_B then
2:     V_B ← V(u)
3: else
4:     U ← child(u)                  // Children of sensor u
5:     L ← ∅                         // List of sensors to expand
6:     for i, j ∈ U do
7:         if M^i ⪰ M^j then L ← L ∪ {i}
8:         end if
9:     end for
10:    L ← sort(L)
11:    for all sensors i ∈ L do
12:        if V(i) < V_B then IBP*(i)
13:        end if
14:    end for
15: end if
```

VI. Simulation Results

To illustrate the effectiveness of the proposed priority list sensor scheduling scheme and the optimal pruning algorithm, numerical simulations from the field of vehicle tracking are conducted. The state $\underline{\boldsymbol{x}}_k = [\boldsymbol{x}_k, \dot{\boldsymbol{x}}_k, \boldsymbol{y}_k, \dot{\boldsymbol{y}}_k]^{\mathrm{T}}$ of the vehicle comprises the two-dimensional position $[\boldsymbol{x}_k, \boldsymbol{y}_k]^{\mathrm{T}}$ and the velocities $[\dot{\boldsymbol{x}}_k, \dot{\boldsymbol{y}}_k]^{\mathrm{T}}$ in x and y direction, while the system model is given by

$$\underline{\boldsymbol{x}}_{k+1} = \mathbf{A}\underline{\boldsymbol{x}}_k + \underline{\boldsymbol{w}}_k \ ,$$

where system matrix and noise covariance matrix of $\underline{\boldsymbol{w}}_k$ are

$$\mathbf{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \ , \ \mathbf{C}_k^w = q \cdot \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix} \ ,$$

resulting from converting the stochastic ordinary differential equation of the vehicle into a discrete-time model [5], [22]. Here, $T = 1\,\mathrm{s}$ is the sampling interval and $q = 0.1$ is the scalar diffusion strength. The mean vector and the covariance matrix of the initial state $\underline{\boldsymbol{x}}_0$ are $\underline{\hat{x}}_0 = [0, 1, 0, 1]^{\mathrm{T}}$ and $\mathbf{C}_0^x = \mathbf{I}$, respectively.

A sensor network consisting of 6 sensors with measurement matrices

$$\mathbf{H}^1 = \mathbf{H}^3 = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \ , \ \mathbf{H}^2 = \mathbf{H}^5 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \ ,$$
$$\mathbf{H}^4 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \ , \ \mathbf{H}^6 = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}$$

and noise covariances $\mathbf{C}^{v,1} = 0.2 \cdot \mathbf{I}$, $\mathbf{C}^{v,2} = \mathbf{C}^{v,3} = 0.1 \cdot \mathbf{I}$, and $\mathbf{C}^{v,4} = \mathbf{C}^{v,5} = \mathbf{C}^{v,6} = 0.05 \cdot \mathbf{I}$ is used. The sensors are placed in a square $[0, 20]\ \mathrm{m} \times [0, 20]\ \mathrm{m}$ according to Fig. 3 (a).

A. Optimal Pruning

At first, the effectiveness of the proposed pruning scheme is demonstrated. Three scheduling algorithms are utilized for determining the optimal sensor schedule:

Full: Scheduling by means of an exhaustive tree search without any pruning.

Information-based pruning (IBP): Scheduling utilizing pruning according to Corollary 2.

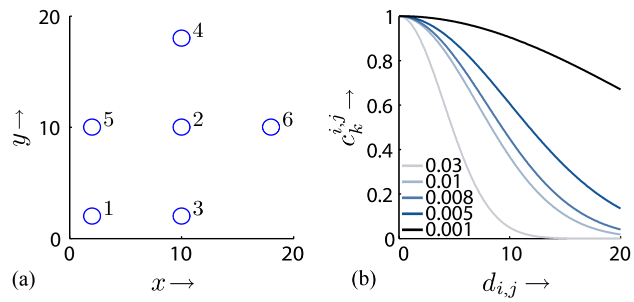

Figure 3. (a) Sensor placement. (b) Transmission probabilities.

Information-based pruning (IBP*):* Combines IBP with B&B techniques according to Algorithm 1.

Table I
NUMBER OF EXPANDED NODES AND COMPUTATION TIME.

	Full	IBP	IBP*
# expanded nodes	55986	5460	144
time in s	185.38	3.94	0.37

The time horizon is set to $N = 6$ and no communication failures are assumed. As shown in Table I, by using the sensor information matrix for pruning, the number of expanded nodes during the calculation of the optimal schedule is reduced by a factor of 10 compared to an exhaustive tree search. The savings in computation time are greater, which is essential for sensor networks consisting of less capable sensor nodes. A further reduction in the number of expanded nodes and computation time is achieved by additionally employing B&B techniques. Here, the IBP can be interpreted as a preselection on candidate schedules, since it always prunes schedules containing sensor 1 and 2, while B&B further thins out this candidate set. Finally, the optimal sensor sequence is given by $u^*_{0:N-1} = [4, 6, 5, 3, 5, 3]$.

Although the presented pruning scheme finds the optimal schedule very early, it is not aware of it. Thus, all sensor sequences that cannot be pruned have to be evaluated for determining the optimal schedule with certainty. This fact leaves enough space for further improving the pruning performance. For example finding better bounds than the admittedly conservative bound used here is one way.

B. Priority List Sensor Scheduling

In this section, the effect of communication failures on the estimation performance is demonstrated. Again, three methods are used for comparison. Besides the proposed PLSS, these methods are:

OPT: The acausal optimal scheduling algorithm described in Section IV-B.

Greedy: If the first sensor of the optimal schedule is not reachable, the leader again performs a measurement.

The transmission probabilities $c_k^{i,j}$ are established using the communication model proposed in [23]. Here, one-hop communication is assumed, where the transmission probability between two sensor nodes i and j is

$$c_k^{i,j} = \exp\{-\tfrac{R}{C} d_{i,j}^2\} \ , \qquad (10)$$

where $C = \frac{P_t \cdot G_t \cdot G_r \cdot \lambda^2}{(4\pi)^2 L}$ is the receiving signal power, $d_{i,j}$ is the Euclidean distance between the sensor nodes, and R is the receiving threshold. It is obvious that the function in (10) is monotonically decreasing with respect to $d_{i,j}$ and for $d_{i,j} = 0$, $c_k^{i,j} = 1$. For simulation purposes several values

$$\tfrac{R}{C} \in \{0.001, 0.005, 0.008, 0.01, 0.03\}$$

for the *decay factor* $\frac{R}{C}$ in (10) are used. The resulting probabilities for these factors are shown in Fig. 3 (b). For each decay factor 20 Monte Carlo (MC) simulation runs are performed with an estimation horizon $N = 10$.

In Fig. 4 (a), the average estimation error over all MC runs for each scheduling method and each decay factor is depicted. Except for $\frac{R}{C} = 0.03$, there is almost no difference between the estimation error of PLSS and the optimal scheduling method. Only for very adverse communication conditions with $\frac{R}{C} = 0.03$, the discrepancy becomes obvious. Here, a transmission probability of 50% and less arises, if the distance between the sensors is larger than $5\,\mathrm{m}$. Since the minimum distance between the sensors in the simulation setup is $8\,\mathrm{m}$, the chance of not reaching a sensor is quite high. However, by employing PLSS the estimation error is drastically improved compared to the greedy strategy, which diverges strongly even for relatively good transmission conditions as communication effects are not or only marginally considered in scheduling.

Consequently, the same effect can be seen for the concrete estimates of the state vector $\underline{\boldsymbol{x}}$. In Fig. 4 (b)–(c), the average root mean square error (rmse) over all simulation runs with respect to the position x and the velocity $\dot{x}$ is illustrated. Similar results are obtained for the position and velocity in y direction. PLSS provides estimates close to the optimal method, if the communication failures are not too strong. It is important to note that particularly better estimates of the greedy method compared to PLSS or OPT can occur. However, the estimation error and thus the covariance of OPT and PLSS is smaller (as depicted in Fig. 4 (a)), which corresponds to a larger reliability of the estimate.

VII. Conclusions and Future Work

The approach introduced in this paper treats the problem of sensor scheduling in the presence of unreliable communication, as it is the case in real-world sensor networks. Besides efficiently calculating sensor schedules that minimize the estimation error, which is the goal of many established sensor scheduling approaches, *priority list sensor scheduling (PLSS)* also takes transmission failures into account. As a result of a prioritization among the sensors, the best currently reachable sensor is selected for measurement, which leads to a minimization of the expected estimation error without explicit knowledge about packet-loss or transmission probabilities. For realistic communication conditions, PLSS leads to an estimation performance that is comparable to those of the

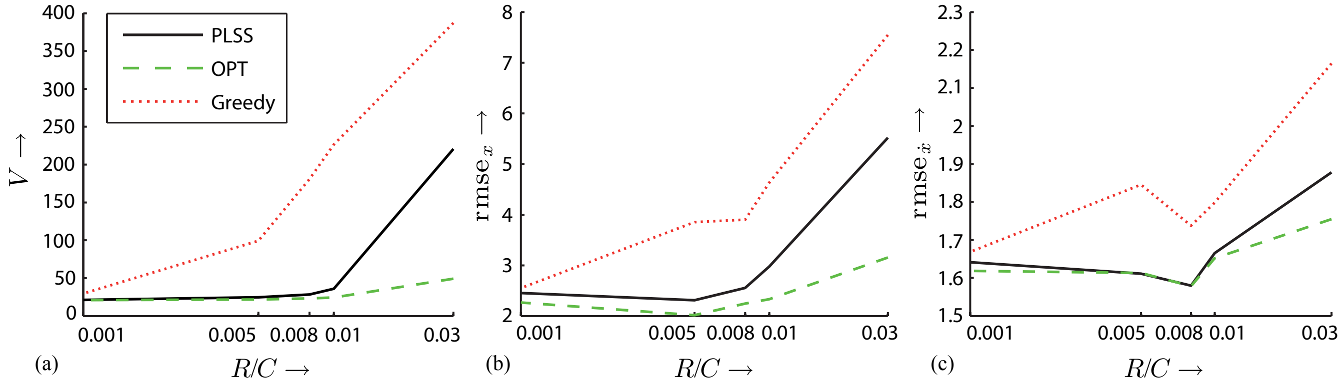

Figure 4. (a) Average predicted estimation error for PLSS (black, solid), acausal scheduling (green, dashed), and greedy scheduling (red, dotted). (b) Average root mean square error (rmse) for estimating the position x. (c) Average rmse for estimating the velocity $\dot{x}$.

optimal but impractical scheduling algorithm. Additionally, other situations leading to unavailable sensors, e.g., sensor breakdown due to low battery power or topology changes due to sensor movements can easily be handled with PLSS.

Since all calculations, especially state estimation and prioritization are carried out locally in the sensor nodes, the computational demand for the priority list calculation has to be as low as possible. Thus, a novel optimal pruning approach is introduced, which guarantees preserving optimal sensor schedules. Compared to existing methods, suboptimal schedules can be pruned without explicitly evaluating the function of the estimation error.

To improve the practical applicability of the proposed approach, especially for a better scalability in large sensor networks, it is worth to investigate clustering strategies. Furthermore, the performance of the pruning algorithm can be increased by determining less conservative bounds.

VIII. Acknowledgements

This work was partially supported by the German Research Foundation (DFG) within the Research Training Group GRK 1194 "Self-organizing Sensor-Actuator-Networks".

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] S. I. Roumeliotis and G. A. Bekey, "Distributed Multirobot Localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, 2002.

[3] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Nonmyopic Sensor Scheduling and its Efficient Implementation for Target Tracking Applications," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–18, 2006.

[4] E. Ertin, J. W. Fisher, and L. C. Potter, "Maximum Mutual Information Principle for Dynamic Sensor Query Problems," in *The Proceedings of Information Processing in Sensor Networks (IPSN)*. Springer-Verlag, Apr. 2003, pp. 405–416.

[5] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "Scheduling Multiple Sensors using Particle Filters in Target Tracking," in *IEEE Workshop on Statistical Signal Processing*, 2003, pp. 549–552.

[6] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "On a Stochastic Sensor Selection Algorithm with Applications in Sensor Scheduling and Sensor Coverage," *Automatica*, vol. 42, no. 2, pp. 251–260, Feb. 2006.

[7] L. Meier, J. Peschon, and R. M. Dressler, "Optimal Control of Measurement Subsystems," *IEEE Transactions on Automatic Control*, vol. AC-12, no. 5, pp. 528–536, Oct. 1967.

[8] J. Liu, F. Zhao, and D. Petrovic, "Information-Directed Routing in Ad Hoc Sensor Networks," in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, San Diego, CA, 2003, pp. 88–97.

[9] S. Kagami and M. Ishikawa, "A Sensor Selection Method Considering Communication Delays," in *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, New Orleans, LA, Apr. 2004.

[10] Z. Quan, W. J. Kaiser, and A. H. Sayed, "A Spatial Sampling Scheme Based on Innovations Diffusion in Sensor Networks," in *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, Cambridge, MA, Apr. 2007, pp. 323–330.

[11] Y. Oshman, "Optimal Sensor Selection Strategy for Discrete-Time State Estimators," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 307–314, Apr. 1994.

[12] F. Zhao, J. Shin, and J. Reich, "Information-driven dynamic sensor collaboration for tracking applications," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 61–72, 2002.

[13] P. Alriksson and A. Rantzer, "Sub-Optimal Sensor Scheduling with Error Bounds," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, Jul. 2005.

[14] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, "Sensor Scheduling Algorithms Requiring Limited Computations," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2004.

[15] F. Sawo, K. Roberts, and U. D. Hanebeck, "Bayesian Estimation of Distributed Phenomena using Discretized Representations of Partial Differential Equations," in *3rd International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Aug. 2006, pp. 16–23.

[16] R. E. Kalman, "A new Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME, Journal of Basic Engineering*, no. 82, pp. 35–45, 1960.

[17] B. Bollobás, *Random Graphs*, 2nd ed. Cambridge University Press, 2001.

[18] C. Lott and D. Teneketzis, "Stochastic Routing in Ad Hoc Wireless Networks," Univ. Michigan Control Group, Tech. Rep. CGR 01-01, Feb. 2001.

[19] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, 2nd ed. Prentice Hall, 2000.

[20] K. M. Abadir and J. R. Magnus, *Matrix Algebra*. Cambridge University, 2005.

[21] S. Howard, S. Suvorova, and B. Moran, "Optimal Policy for Scheduling of Gauss-Markov Systems," in *Proceedings of the Seventh International Conference on Information Fusion*, Stockholm, Sweden, 2004, pp. 888–892.

[22] P. Maybeck, *Stochastic Models, Estimation, and Control*. Academic Press, 1979.

[23] M. Torrent-Moreno, F. Schmidt-Eisenlohr, H. Füßler, and H. Hartenstein, "Effects of a Realistic Channel Model On Packet Forwarding in Vehicular Ad Hoc Networks," in *Proceedings of IEEE Wireless Communication and Networking Conference (WCNC)*, vol. 1, Las Vegas, NV, Apr. 2006, pp. 385–391.