Multi-Step Sensor Selection with Position Uncertainty Constraints

Multi-Step Sensor Selection with Position Uncertainty Constraints

Markus Fröhle, Ali A. Zaidi, Erik Ström, and Henk Wymeersch
Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden
E-mail: {frohle,aliabb,erik.strom,henkw}@chalmers.se

***Abstract*—Research on localization systems has shifted from focusing mainly on accuracy towards a more cognitive design, accounting for communication constraints, energy limitations, and delay. This leads to a variety of sensor selection optimization problems that are solved using techniques from convex optimization. We provide a novel formulation of the sensor selection problem over an extended time horizon, aiming to minimize the sensing cost of an entire path while guaranteeing a certain position accuracy. We state algorithms for determining lower and upper bounds on the sensing cost and utilize these in a path selection problem for autonomous agents. Simulation results confirm the usefulness of our approach, where we observe a benefit of optimizing over longer time horizons in low to medium noise scenarios compared to a myopic sensor selection scheme.**

I. Introduction

Increasingly sophisticated algorithms along with massive computational capabilities have enabled autonomous agents/robots to solve complex tasks in uncertain environments, such as mapping of disaster areas and search-and-rescue operations. For agents to explore and interact with the environment, it is imperative for them to have a coherent view of this environment and their positions within it [1], [2]. Such situational awareness is generally achieved through simultaneous localization and mapping (SLAM) [3] and localization using heterogeneous sensor fusion [4]. While research on situational awareness has traditionally focused on improving the localization accuracy, the focus has now shifted to localization methods that are cognitive with respect to energy and communication constraints as well as the agent's higher-level task [5], [6]. For instance, a higher-level task may be the navigation/manipulation of the robot in the environment form its current position to its intended final position.

To aid positioning, a robot is typically equipped with multiple sensors providing position information. Since the usage of each of these sensors consumes energy, careful selection of when to use which sensor is important. This problem is known as *sensor selection*, where the agent chooses a subset of available sensors out of the full sensor suite, in order to optimize an objective. The objective is generally a scalarized version (e.g., trace or determinant) of the state estimation error covariance [7], [8], an expected utility defined by Bayes risk [9], or a measure of information such as conditional entropy [10]. These objectives can be optimized (i) over a single time step [7]–[9], [11] or (ii) over a prediction window [10], [12]–[14]. In the first class, the objectives and constraints are made temporally separable, leading to efficient, low-complexity solutions. In particular, considering either linear Gaussian [7], [8] or general nonlinear [11] models, the optimization is cast as a mixed integer program, and solved with relaxation techniques. In [9], a hidden Markov model is considered, and a greedy objective function is introduced, combining expected utility and instantaneous cost. These greedy approaches can be considered simplifications of the more general problem of sensor selection over a time window. In that more general case, the objective and constraints are temporally inseparable, generally leading to problems that grow in complexity exponentially in the prediction horizon. This complexity is reduced through relaxation techniques [12], [14], pruning of the search tree [13], or considering dynamic programming formulations [10]. We point out that the above works generally focus on minimizing a function of the state error covariance. In contrast, for battery-constrained devices a more relevant problem is to minimize energy consumption while placing the requirement on the state error covariance as a constraint.

In this paper, we assign to each sensor usage a cost. The objective of the sensor selection problem is then to minimize the accumulated cost over a prediction horizon, while ensuring the state error is within a user-defined threshold. The inclusion of convex communication constraints can be easily accommodated, but is not considered explicitly here. This problem belongs to the class of temporally inseparable problems. We formally introduce this problem and derive low-complexity strategies (i.e., polynomial in the time horizon) for finding upper and lower bounds on the optimal cost. In addition, we provide numerical results comparing the proposed strategies with a greedy policy.

Notation: Vectors are denoted in boldface (e.g., $\mathbf{x}$) and $[\mathbf{x}]_i$ denotes its i-th entry. The matrix trace of a matrix F is denoted by $\mathrm{tr}(F)$. The identity matrix of proper size is denoted by $\mathbb{I}$ (usually of size 2×2). $\lfloor \cdot \rfloor$ denotes the floor operator, $S \succ 0$ denotes a positive definite matrix, and $R \succeq 0$ denotes a positive semi-definite matrix.

II. Problem Formulation

A. Scenario

Consider the scenario shown in Fig. 1, where a mobile agent with estimated start position $\mathbf{p}_0$ wishes to reach a goal position $\mathbf{p}_{\mathrm{goal}} \in \mathbb{R}^2$ in discrete time steps. The agent is equipped with M sensors providing location information, the quality of which may be location-dependent (e.g., a GPS signal is usually stronger closer to a window than in the middle of a building).

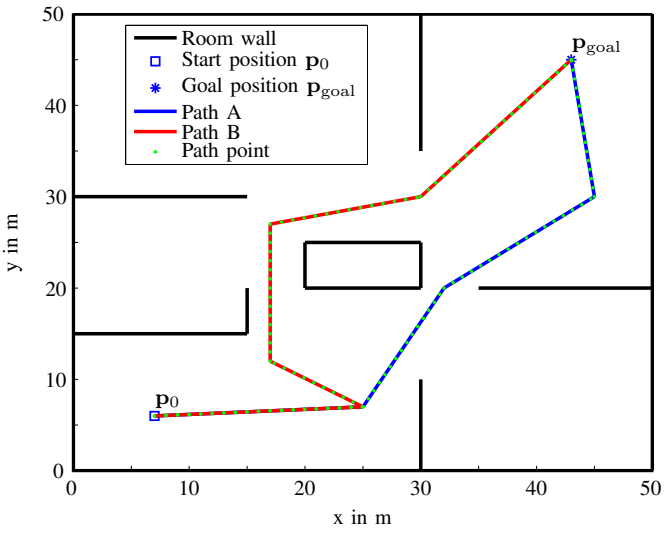

Fig. 1. Example scenario with two possible paths (A and B) from the start position $\mathbf{p}_0 = [7, 6]^{\mathrm{T}}$ to the goal position $\mathbf{p}_{\text{goal}} = [43, 45]^{\mathrm{T}}$. Path A has a path length of $N = 68$ positions and path B has a length of $N = 80$ positions (separated by black dots). The objective of the agent is to determine the path with the lowest cost, while at all times ensuring a given positioning accuracy.

The usage of a sensor system $m \in \{1, 2, \ldots, M\}$ has an associated cost (e.g., related to battery usage or maintenance) $c^{(m)} \geq 0$. The agent can follow multiple paths to move to the goal position (e.g., 2 paths in Fig. 1). While navigating along any path, the agent is allowed to utilize information from one sensor at every time step. Hence, the choice of a certain path and the selection of sensors along that path has an associated cost. Our goal is for the agent to determine the least costly path to move towards the destination, while at all times ensuring a certain accuracy in the position information.

B. System Model

Without loss of generality, we consider the agent's state as its position. The initial state $\mathbf{x}_0$ is unknown and modeled as a Gaussian random variable $\mathbf{x}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, P_0)$. The goal position $\mathbf{p}_{\text{goal}}$ is known exactly, as is the floor plan of the environment and the measurement quality of each sensor across the environment. The agent is assumed to have the ability to generate J paths from an estimated starting position $\boldsymbol{\mu}_0$ to the goal position $\mathbf{p}_{\text{goal}}$ [15]. A path $j \in \{1, 2, \ldots, J\}$ is comprised of $N_j + 1$ positions, $\mathbf{p}_{j,0}, \mathbf{p}_{j,1}, \ldots, \mathbf{p}_{j,N_j}$, where $\mathbf{p}_{j,0} = \boldsymbol{\mu}_0$ and $\mathbf{p}_{j,N_j} = \mathbf{p}_{\text{goal}}$. We now drop the path index j for notational convenience. In order to move along the path, the agent will apply a sequence of controls $\mathbf{u}_k$, $k = 0, 1, \ldots, N-1$, associated with the linear state (position) update equation

$$\mathbf{x}_k = F_{k-1}\mathbf{x}_{k-1} + G_{k-1}\mathbf{u}_{k-1} + \mathbf{n}_{k-1}, \tag{1}$$

where F_{k-1} and G_{k-1} are known matrices, and $\mathbf{n}_{k-1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, Q_{k-1})$ is the process noise in the state space model with error covariance matrix Q_{k-1}. At any time step k, the measurement model of sensor m is given by

$$\mathbf{y}_k^{(m)} = H_k^{(m)}\mathbf{x}_k + \mathbf{v}_k^{(m)}, \tag{2}$$

where $H_k^{(m)}$ is a known matrix and $\mathbf{v}_k^{(m)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, R_k^{(m)})$ is the measurement noise[1] associated with sensor m when used in location $\mathbf{x}_k$. Since the state and measurement models are linear and the noise variables are Gaussian, the optimal state estimator is the Kalman filter [16]. The agent now proceeds as follows:

1) For each path, the agent determines the predicted control sequence $\mathbf{u}_{0:N-1}$ with the help of the path positions $\mathbf{p}_k$;
2) For a given path and a given sequence of selected sensing systems, the agent can compute (i) the expected accuracy of its predicted position and (ii) the associated cost.

Based on this information, the agent is in principle able to asses the expected cost of any path, and thus to select the least costly path. In the next section, we will describe how to determine the minimal cost of a given path $j \in \{1, 2, \ldots, J\}$. The globally cheapest path is then found as the path that minimizes the path cost.

III. Optimization Formulation

The agent can determine the minimum total cost of a path by solving the following optimization problem:

$$\begin{aligned}
\underset{A}{\text{minimize}} \quad & \mathbf{c}^{\mathrm{T}} A \mathbf{1} && \text{(3a)}\\
\text{subject to} \quad & A^{\mathrm{T}}\mathbf{1} = \mathbf{1} && \text{(3b)}\\
& A \in \{0,1\}^{M \times N} && \text{(3c)}\\
& \operatorname{tr}\left((I_k^+)^{-1}\right) \leq \Delta^2,\ k \in \{1, 2, \ldots, N\}, && \text{(3d)}
\end{aligned}$$

where $\mathbf{c} = [c^{(1)}, c^{(2)}, \ldots, c^{(M)}]^{\mathrm{T}}$, $\mathbf{1}$ is a column vector of proper size containing all ones, I_k^+ is the posterior information matrix, and A denotes the optimization variable structured as

$$A = \begin{bmatrix} a_1^{(1)} & \ldots & a_N^{(1)} \\ \vdots & \ddots & \vdots \\ a_1^{(M)} & \ldots & a_N^{(M)} \end{bmatrix}, \tag{4}$$

where a value of $a_k^{(m)} = 1$ activates the m-th sensor system in the k-th time step. Constraint (3b) ensures that only one out of M measurement systems is active at a time. Constraint (3d) forces the trace of the inverse of the posterior information matrix, i.e. $(I_k^+)^{-1} = P_k$, to remain below a user-defined threshold Δ^2, expressed in m^2. This ensures that the expected[2] root mean square position error (RMSE) is not more than Δ meter.

[1] As a special case, not using any sensor can be modeled by having a virtual sensor with $R_k^{(m)} = \alpha\mathbb{I}$, for $\alpha > 0$ extremely large.

[2] Note that the agent only aims to determine the path with the lowest *expected* cost. This cost is computed assuming the agent measures in the nominal positions $\mathbf{p}_k$. However, once a path and sensor activation schedule has been selected, the actual position of the agent will be different from $\mathbf{p}_k$. Hence, the actual root mean position error may exceed Δ. For small values of Δ, this impact will be negligible.

We use the information form of the Kalman filter [16, Sec. 6.2] to describe the evolution of the information matrix. It can be shown that I_k^+ is given by the following recursion

$$I_k^+ = M_k + Q_{k-1}^{-1} - Q_{k-1}^{-1}F_{k-1}(I_{k-1}^+ + F_{k-1}^{\mathrm{T}}Q_{k-1}^{-1}F_{k-1})^{-1}F_{k-1}^{\mathrm{T}}Q_{k-1}^{-1}, \tag{5}$$

where

$$M_k = \sum_{m=1}^{M} a_k^{(m)} {H_k^{(m)}}^{\mathrm{T}} \left(R_k^{(m)}\right)^{-1} H_k^{(m)}, \tag{6}$$

and initialized by $I_0^+ = P_0^{-1}$.

The optimization problem (3) is a combinatorial problem, where the search space for the optimal solution grows exponentially with horizon length N. This makes it impractical to solve even when N is relatively small. Also, due to the matrix inversion in (5), the problem involves nonlinear equality constraints. Rather than solving (3) directly, we provide lower and upper bounds on the cost per path.

A. Upper Bound: Dynamic Programming

By quantizing the information matrix to S levels, we can reduce the complexity from $\mathcal{O}(M^N)$ [13] to $\mathcal{O}(SMN)$, though we are no longer guaranteed optimality, even for large S (due to the loss of dimensions). Our solution is based on dynamic programming (DP) [17] over a trellis where the states correspond to the quantized information at each time.

We define a finite state space $\mathcal{S} = \{s_1, s_2, \ldots, s_S\}$ and an operator $s \triangleq q(I_k^+)$. The operator works two-fold: (i) it scalarizes its argument (e.g., by computing the trace or determinant); (ii) it maps the scalar value to $s \in \mathcal{S}$. The state space is constructed by partitioning the interval $[0, \Delta^2]$ into S bins. As an example, we will define the operator $q(\cdot)$ as[3]

$$q(I_k^+) = \begin{cases} s_l, & \mathrm{tr}\left((I_k^+)^{-1}\right) \in [(l-1), l)\,\frac{\Delta^2}{S}, \\ \text{undefined} & \text{otherwise.} \end{cases} \tag{7}$$

Note that when constraint (3d) is violated, $q(I_k^+)$ is not defined and thus does not correspond to any state. An illustration of this is provided in Fig. 2, showing a trellis diagram of the system. The trellis consists of S different states $s_1, s_2, \ldots, s_S$ for each of the $N+1$ time steps. Note, at $k=0$ we start the algorithm in one of the S states (assuming we have a feasible starting state with $\mathrm{RMSE}(\mathbf{x}_0) \leq \Delta$), given by $q(I_0^+)$. In this example, the initial state is s_1 (solid black circle).

We further define a cost $c_k(s)$ as the accumulated cost up to time k when we are in state $s \in \mathcal{S}$. The costs are initialized to $+\infty$, except for time $k=0$ and state $q(I_0^+)$, for which the cost is set to zero (in the example, we set $c_0(s_1) = 0$). To reach the next time step $k=1$, the algorithm tests all of the M sensor systems: for system i (corresponding to $a_1^{(i)} = 1$), we compute I_1^+ with the help of (5), the corresponding state $q(I_1^+)$, and cost $c^{(i)}$. In the example only action i puts the system in state s_1, so that best path to end in state s_1 at time $k=1$ has an accumulated cost $c_1(s_1) = c^{(i)}$. In the trellis, we also maintain a pointer to the previous state (i.e., from s_1 at $k=1$ to s_1 at $k=0$). In case a certain sensor, say m, gives rise to a I_1^+ that violates (3d), $q(I_1^+)$ is undefined and the transition is ignored. Moving from time $k=1$ to $k=2$, we see in Fig. 2 that applying sensor j from state s_2 and sensor i from state s_1 end up in the same state s_2 at time $k=2$. In that case, the cheapest action ending in state s_2 is chosen as the one that minimizes $[c_1(s_1) + c^{(i)}, c_1(s_2) + c^{(j)}]$, say action j. In that case, the accumulated cost for s_2 at time $k=2$ is set to $c_2(s_2) = c_1(s_2) + c^{(j)}$ and a pointer to s_2 at time $k=1$ is maintained. Proceeding in this manner allows us to compute the resulting states at the subsequent steps, gradually revealing the structure of the trellis. At time $k=N$ we select the terminal state s with lowest accumulated state cost $\arg\min_s c_N(s)$. Since we have stored the action to this state and a pointer to the state in the previous time step $k=N-1$, we can read out the whole sequence of actions $a_{1:N}^*$ from the last time step N to the first time step 1.

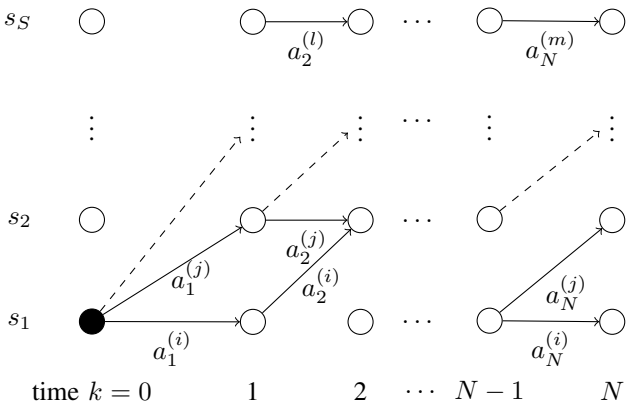

Fig. 2. Trellis diagram demonstrating the DP algorithm. It is comprised of S different states $s_1, s_2, \ldots, s_S$ for all $N+1$ time steps. Here, the initial state at time step $k=0$ is s_1 (solid black circle). The usage of sensor i (with $a_1^{(i)} = 1$) puts the agent in state s_1 at time step $k=1$, and with sensor j (i.e., with $a_1^{(j)} = 1, j \neq i$) the agent is in s_2 at time step $k=1$.

B. Lower Bound: Semi-definite Programming

Theorem 1 (Lower bound)**.** Problem (3) can be relaxed to

$$\begin{aligned}
\underset{A, S_k}{\text{minimize}} \quad & \mathbf{c}^{\mathrm{T}} A \mathbf{1} && \text{(8a)}\\
\text{subject to} \quad & A^{\mathrm{T}}\mathbf{1} = \mathbf{1} && \text{(8b)}\\
& A \in \{0,1\}^{M\times N} && \text{(8c)}\\
& \mathrm{tr}(S_k) \leq \Delta^2 && \text{(8d)}\\
& \begin{bmatrix} S_k & \mathbb{I} \\ \mathbb{I} & I_k^+ \end{bmatrix} \succeq 0 && \text{(8e)}\\
& \begin{bmatrix} Q_{k-1}^{-1} + M_k - I_k^+ & Q_{k-1}^{-1}F_{k-1} \\ F_{k-1}^{\mathrm{T}}Q_{k-1}^{-1} & I_{k-1}^+ + F_{k-1}^{\mathrm{T}}Q_{k-1}^{-1}F_{k-1} \end{bmatrix} \succeq 0. && \text{(8f)}
\end{aligned}$$

The proof is provided in the Appendix. Note that constraints (8d)–(8f) should be met for all k, and that M_k is a function of A (see (6)). Relaxing the integer constraint (8c) to the box constraint $A \in [0,1]^{M\times N}$ leads to a standard semi-definite

[3] An alternative quantization could be based on $\log \mathrm{tr}\left((I_k^+)^{-1}\right)$.

program (SDP), which can be solved efficiently and optimally using standard tools. The solution will provide a lower bound to the cost of (3). Since we relaxed the integer constraint (8c) to obtain a standard SDP problem, we cannot directly state the selected sensor for every time step. To unrelax the solution of the SDP, we can use heuristics such as selecting the sensor m at time step k which has highest value of $a_k^{(m)}$ and check whether all the constraints are still fulfilled. A feasible solution from the solution of the SDP problem can also be obtained by using the approaches in [7], [18], [19]. In this paper, we state the solution of the SDP problem directly. Hence the unrelax step is not performed and the SDP lower bound may contain partial sensor usages.

IV. Performance Evaluation

A. Setup

The scenario is outlined in Fig. 1. The mobile agent's prior is given by $\boldsymbol{\mu}_0 = [7, 6]^{\mathrm{T}}$ and $P_0 = 0.05\ \mathbb{I}$. The goal position is $\mathbf{p}_{\mathrm{goal}} = [43, 45]^{\mathrm{T}}$. To generate paths, we utilized a geometric path planner by first spreading 18 routing points randomly in the environment. Using a depth-first search [15], we generated possible paths between $\mathbf{p}_0 = \boldsymbol{\mu}_0$ and $\mathbf{p}_{\mathrm{goal}}$ that do not go through walls. This procedure leads to 169 distinct paths (of which 2 are shown in Fig. 1) with path lengths in the range $N \in [58, 172]$. We set $\Delta = 1$, $F_k = \mathbb{I}$, $H_k^{(m)} = \mathbb{I}$, $Q_k = \sigma_Q^2\ \mathbb{I}$, $\forall k, m$. We considered different process noise levels $\sigma_Q^2 \in \{0.01, 0.1, 0.2\}$. We used $M = 4$ sensors, defined as follows. Sensor 1 corresponds to not sensing, with $R_k^{(1)} = 10^6\ \mathbb{I}$. Sensor 2 corresponds to a GPS-like sensor which is accurate near the windows, but poor deeper inside the environment. This is modeled through $R_k^{(2)} = 20\ \mathbb{I}$ for $15 \leq [\mathbf{x}]_1, [\mathbf{x}]_2 \leq 35$ and $R_k^{(2)} = 2\ \mathbb{I}$ elsewhere. Sensor 3 is an RFID-like sensor with 4 tags placed near the corners. This is modeled through

$$R_k^{(3)} = \begin{cases} \sigma_{\text{low}}^{2,(3)}\ \mathbb{I}, & [\mathbf{x}]_1 \leq 15, [\mathbf{x}]_2 \leq 15, \\ \sigma_{\text{low}}^{2,(3)}\ \mathbb{I}, & [\mathbf{x}]_1 \geq 35, [\mathbf{x}]_2 \leq 15, \\ \sigma_{\text{low}}^{2,(3)}\ \mathbb{I}, & [\mathbf{x}]_1 \leq 15, [\mathbf{x}]_2 \geq 35, \\ \sigma_{\text{low}}^{2,(3)}\ \mathbb{I}, & [\mathbf{x}]_1 \geq 35, [\mathbf{x}]_2 \geq 35, \\ \sigma_{\text{high}}^{2,(3)}\ \mathbb{I}, & \text{otherwise.} \end{cases} \tag{9}$$

We set $\sigma_{\text{low}}^{2,(3)} = 0.2$ and $\sigma_{\text{high}}^{2,(3)} = 30$. Sensor 4 is an ultra wideband-like (UWB) sensor with 4 UWB reference nodes in the corners. Such a system would exhibit high sensor measurement quality in the middle of the environment, but not near the corners. This is modeled through a similar expression as sensor 3, but where we replace $\sigma_{\text{low}}^{2,(3)}$ with $\sigma_{\text{high}}^{2,(4)}$ and $\sigma_{\text{high}}^{2,(3)}$ with $\sigma_{\text{low}}^{2,(4)}$. We set $\sigma_{\text{low}}^{2,(4)} = 0.1$ and $\sigma_{\text{high}}^{2,(4)} = 10$. The costs of the sensors are set as follows: $c^{(1)} = 1$ comprising the cost of movement, and $c^{(2:4)} = [3, 2, 4]^{\mathrm{T}}$. Note that costs $c^{(2:4)}$ are higher than $c^{(1)}$ and can thus be considered to comprise two parts: a cost for utilizing the specific sensor, and the cost for movement (which is 1 in our example).

In our implementation of the DP we used $S = 10$ states. For the SDP lower bound (8), we used the software package CVX [20]. As a reference method we considered a greedy approach to solve (3): for each time step we select the cheapest sensor such that the accuracy constraint (3d) is satisfied.

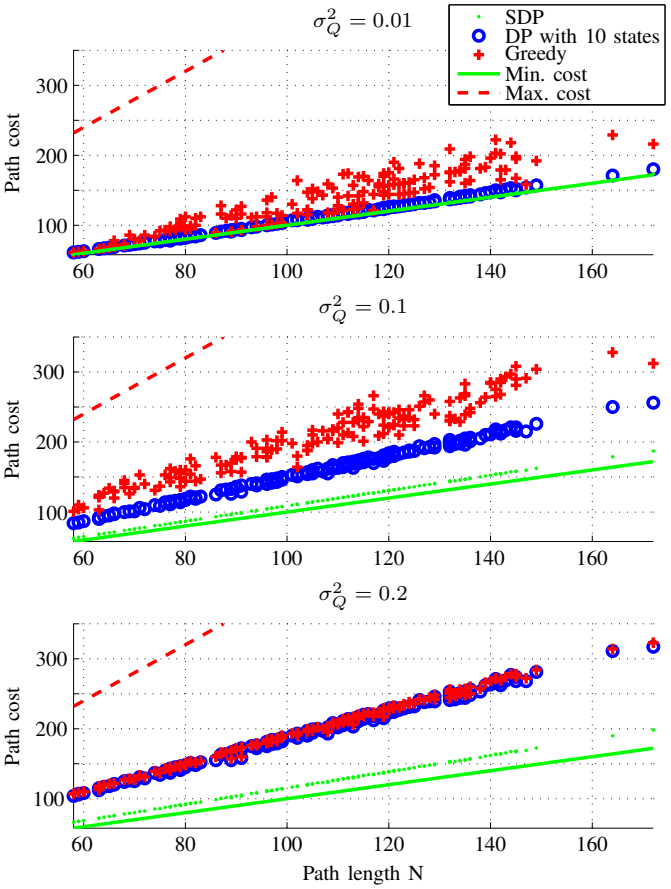

Fig. 3. Path costs for the different paths sorted by path length. Top to bottom plot correspond to different values of the process noise variance $\sigma_Q^2 \in \{0.01, 0.1, 0.2\}$. The minimum and the maximum path costs are obtained by using the sensor with lowest and highest cost along the whole path. The lower bound (SDP) and upper bound (DP) are compared to the greedy solution.

B. Results and Discussion

We will first evaluate the path cost with the three different methods for the 169 paths between the initial agent position $\boldsymbol{\mu}_0$ and the goal position $\mathbf{p}_{\mathrm{goal}}$ (Fig. 3), and then focus on a fixed path (Figs. 4–5). For the purpose of visualization, the paths are sorted by path length (note that certain paths may have the same length). Fig. 3 shows the results of the DP, SDP, and greedy approach along with trivial lower and upper bounds (obtained by always using the cheapest and most expensive sensor without considering whether constraint (3d) is fulfilled), for different values of the process noise level σ_Q^2.

We observe in Fig. 3 that for low and medium noise levels σ_Q^2 the gap between the greedy method and the DP method is significant (top and middle plot). This means that for these cases there is a benefit of optimizing over a longer time horizon (multiple steps along the trajectory), and hence the DP method provides us with a tighter upper bound on the optimal

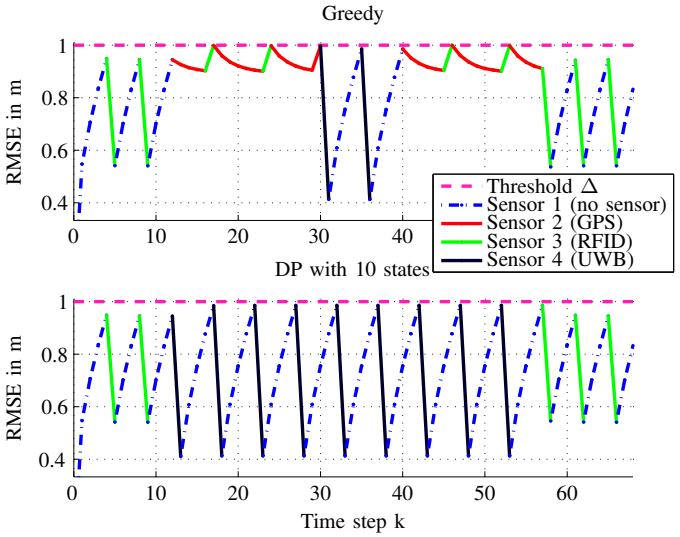

Fig. 4. RMSE plot along path A (cf. Fig. 1). The sensor used is highlighted. Top: greedy solution (cost 145), Bottom: DP solution (cost 100). The user-defined RMSE threshold is $\Delta = 1$. The noise level of sensor 4 is set to $\sigma^{2,(4)}_{\text{low}} = 0.1$.

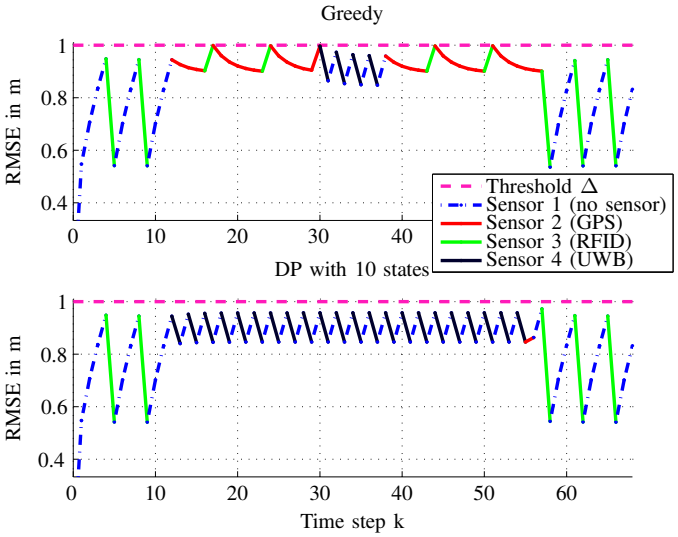

Fig. 5. RMSE plot along path A (cf. Fig. 1). The sensor used is highlighted. Top: greedy solution (cost 155), Bottom: DP solution (cost 141). The user-defined RMSE threshold is $\Delta = 1$. The noise level of sensor 4 is set to $\sigma^{2,(4)}_{\text{low}} = 1$.

path cost compared to the greedy method. If the process noise level σ_Q^2 is increased further the system needs to utilize sensors $2-4$ more frequently to maintain the RMSE below the desired threshold Δ. In this case, we observe that the solution of the greedy approach and the solution of the DP approach deliver similar path costs (bottom plot of Fig. 3). For the case of a high process noise, we can conclude that there is no clear benefit of optimizing over a longer time horizon. Furthermore, we observe that the path costs of the SDP method are for all three different noise levels $\sigma_Q^2 \in \{0.01, 0.1, 0.2\}$ close to the trivial lower bound (top to bottom plot of Fig. 3). This is incurred by the relaxation of (8c) to a box constraint, where partial sensor usages become possible, so that the RMSE can be kept below the desired threshold Δ with a low path cost.

We now fix the process noise level $\sigma_Q^2 = 0.1$ and focus on a particular path (path A from Fig. 1). In Fig. 4–5, we plot the RMSE of the greedy and DP methods as a function of the time step k, for the cases $\sigma^{2,(4)}_{\text{low}} = 0.1$ and $\sigma^{2,(4)}_{\text{low}} = 1$, respectively. Note that the sensor quality of sensor 3 is good within each $15\ \text{m} \times 15\ \text{m}$ corner of the room and its error covariance matrix takes a value of $R_k^{(3)} = \sigma^{2,(3)}_{\text{low}}\ \mathbb{I}$ (see Sec. IV-A). Since the start and goal positions are inside this area (cf. Fig. 1) and sensor 3 has a lower cost compared to sensor 2 and 4, it is utilized. In the case of path A, this is true for both the greedy and the DP method for time step $k \leq 10$ and for $k \geq 55$ (see Fig. 4). In this part of the trajectory the reduction of the RMSE due to the usage of sensor 3 is significant. Once the trajectory of path A leaves the $15\ \text{m} \times 15\ \text{m}$ square area the error covariance matrix of sensor 3 switches to $R_k^{(3)} = \sigma^{2,(3)}_{\text{high}}\ \mathbb{I}$. Then the utilization of this sensor has only a marginal impact on the reduction of the state RMSE. Despite this fact, we observe in the top plot of Fig. 4 that the greedy method continues to make use of the cost efficient, but poor, sensor 3 (time step $10 < k < 55$). Although a measurement from this sensor is utilized the RMSE increases from one time step to the other. This is caused by the poor sensor quality in combination with the high process noise level of $\sigma_Q^2 = 0.1$. Hence the greedy method also needs to utilize the more expensive sensor 2 multiple times in a row in order to keep the RMSE below the threshold Δ. For regions deep inside the room, where sensor 2 does not have a sufficiently high measurement quality, sensor 4, which is the most expensive sensor, needs to be utilized in order to meet the RMSE constraint (3d). In contrast to the greedy approach, the DP approach considers the accumulated path cost over the entire path. The sensor leading to the lowest total path cost (while maintaining (3d)) is selected instead (see bottom plot of Fig. 4). This is achieved by utilizing sensor 3 whenever its measurements are of good quality (for time step $k \leq 10$ and for $k \geq 55$), and sensor 4 when this is not the case. Note that the usage of sensor 4 coincides with it having a high measurement quality, i.e., at positions $\mathbf{p}_k$ along the trajectory where $R_k^{(4)} = \sigma^{2,(4)}_{\text{low}}\mathbb{I}$. The solution of the greedy method results in a total path cost of 145 compared to 100 for the DP method.

In Fig. 5 the RMSE of the state $\mathbf{x}_k$ is plot for $\sigma^{2,(4)}_{\text{low}} = 1$. The reduction of the RMSE by utilizing sensor 4 is lower compared to the previous case (cf. Fig. 4). Both the greedy and the DP approach now need to make use of the expensive sensor 4 more frequently. Similar as before, the greedy approach uses sensor 4 whenever the usage of the remaining sensors would violate the RMSE threshold constraint (time step $30 \leq k \leq 38$). The DP approach uses sensor 4 whenever sensor 3 does not have a high measurement quality. Still with this increased number of usages of sensor 4, this sensor selection scheme leads to lowest accumulated path cost. The greedy method has a path cost of 155 compared to 141 for the DP method. Due to the high

noise (in the state space model and the sensor measurements) the gap between these two methods has become smaller and hence the benefit of optimizing over a longer time horizon has decreased.

In contrast to the greedy and DP methods, the SDP method allows partial sensor usages through the relaxation which have been made. Hence, the RMSE of $\mathbf{x}_k$ attains the threshold Δ arbitrarily close whenever this is beneficial to reduce the total path costs.

V. Conclusion

We have stated the sensor selection problem over an extended time horizon, aiming to minimize the sensing cost of an entire path while guaranteeing a certain position accuracy. Since the complexity of this sensor selection problem increases exponentially with time horizon, we provided upper and lower bounds on the total path cost that can be computed efficiently. An upper bound on the original problem is obtained through quantization and dynamic programming. The reformulation of the sensor selection problem to a standard semi-definite programming problem and relaxation of the integer constraints provides us with a lower bound. These bounds have been analyzed and compared to a greedy solution method. In scenarios with low to medium noise (in the state space and measurement models), we observed a benefit of optimizing over a longer time horizon in comparison with a myopic sensor selection scheme.

Appendix

Proof of Theorem 1. Introduce a positive semi-definite slack matrix $S_k \in \mathbb{R}^{2\times 2}$, $S_k \succeq 0$, where

$$S_k - (I_k^+)^{-1} \succeq 0. \tag{10}$$

We recall the Schur complement, given by [18]

$$\begin{bmatrix} Q & S \\ S^{\mathrm{T}} & R \end{bmatrix} \succeq 0 \Leftrightarrow R \succ 0 \text{ and } Q - SR^{-1}S^{\mathrm{T}} \succeq 0. \tag{11}$$

We now apply the Schur complement to (10) to get the linear matrix inequality

$$\begin{bmatrix} S_k & \mathbb{I} \\ \mathbb{I} & I_k^+ \end{bmatrix} \succeq 0. \tag{12}$$

This allows us to rewrite constraint (3d) as constraint (8d)–(8e) where we have ensured that $\mathrm{tr}((I_k^+)^{-1}) \leq \mathrm{tr}(S_k) \leq \Delta^2$ holds. We now relax (5) by

$$\begin{aligned} I_k^+ \preceq\ & M_k + Q_{k-1}^{-1} \\ & - Q_{k-1}^{-1}F_{k-1}(I_{k-1}^+ + F_{k-1}^{\mathrm{T}}Q_{k-1}^{-1}F_{k-1})^{-1}F_{k-1}^{\mathrm{T}}Q_{k-1}^{-1}, \end{aligned} \tag{13}$$

meaning that the predicted information matrix together with the obtained information through measuring will always provide *more* information than what is strictly needed. Using the Schur complement, and the fact that Q_{k-1} is a covariance matrix and hence symmetric, we can express (13) as (8f). □

Acknowledgment

This work is supported, in part, by the European Research Council under Grant No. 258418 (COOPNET), and by EU FP7 Marie Curie Initial Training Network MULTI-POS (Multi-technology Positioning Professionals) under Grant No. 316528.

References

[1] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The Future of Human-in-the-Loop Cyber-Physical Systems," *Computer*, vol. 46, no. 1, pp. 36–45, 2013.

[2] G. Kruijff, M. Janíček, S. Keshavdas, B. Larochelle, H. Zender, N. Smets, T. Mioch, M. Neerincx, J. Diggelen, F. Colas *et al.*, "Experience in System Design for Human-Robot Teaming in Urban Search and Rescue," in *Field and Service Robotics*. Springer, 2014, pp. 111–125.

[3] S. Thrun and Y. Liu, "Multi-robot SLAM with Sparse Extended Information Filers," in *Robotics Research*. Springer, 2005, pp. 254–266.

[4] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative Localization in Wireless Networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[5] J. Fink, "Communication for Teams of Networked Robots," Ph.D. dissertation, Elect. Syst. Eng., Univ. Pennsylvania, Philadelphia, PA, Aug 2011.

[6] G. Garcia, L. Muppirisetty, E. Schiller, and H. Wymeersch, "On the Trade-Off Between Accuracy and Delay in Cooperative UWB Localization: Performance Bounds and Scaling Laws," *Wireless Communications, IEEE Transactions on*, vol. 13, no. 8, pp. 4574–4585, Aug 2014.

[7] S. Joshi and S. Boyd, "Sensor Selection via Convex Optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[8] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy Sensor Selection: Leveraging Submodularity," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 2572–2577.

[9] D. Cohen, D. L. Jones, and S. Narayanan, "Expected-utility-based Sensor Selection for State Estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 2685–2688.

[10] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Approximate Dynamic Programming for Communication-Constrained Sensor Network Management," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4300–4311, 2007.

[11] A. S. Chhetri, D. Morrell, and A. Papandreou-Suppappola, "On the Use of Binary Programming for Sensor Scheduling," *IEEE Transactions on Signal Processing*, vol. 55, no. 6, pp. 2826–2839, 2007.

[12] X. Shen and P. Varshney, "Sensor Selection Based on Generalized Information Gain for Target Tracking in Large Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 363–375, Jan 2014.

[13] M. P. Vitus, W. Zhang, A. Abate, J. Hu, and C. J. Tomlin, "On Efficient Sensor Scheduling for Linear Dynamical Systems," *Automatica*, vol. 48, no. 10, pp. 2482–2493, 2012.

[14] M. Huber, "On Multi-Step Sensor Scheduling via Convex Optimization," in *2nd International Workshop on Cognitive Information Processing (CIP)*, June 2010, pp. 376–381.

[15] S. M. LaValle, *Planning Algorithms*. Cambridge university press, 2006.

[16] D. Simon, *Optimal State Estimation: Kalman, H infinity, and Nonlinear Approaches*. John Wiley & Sons, 2006.

[17] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific Belmont, MA, 1995, vol. 1, no. 2.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge university press, 2009.

[19] Z.-Q. Luo and W. Yu, "An Introduction to Convex Optimization for Communications and Signal Processing," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1426–1438, 2006.

[20] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming," http://cvxr.com/cvx, March 2014.