

Informative Path Planning for Active Tracking of Agile Targets

Per Boström-Rost, Daniel Axehill and Gustaf Hendeby

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-155035>

N.B.: When citing this work, cite the original publication.

Boström-Rost, P., Axehill, D., Hendeby, G., (2019), Informative Path Planning for Active Tracking of Agile Targets, *2019 IEEE Aerospace Conference*, , 1-11. <https://doi.org/10.1109/AERO.2019.8741840>

Original publication available at:

<https://doi.org/10.1109/AERO.2019.8741840>

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

<http://www.ieee.org/>

©2019 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.



Informative Path Planning for Active Tracking of Agile Targets

Per Boström-Rost
Linköping University
581 83 Linköping
+46 732 48 58 59
per.bostrom-rost@liu.se

Daniel Axehill
Linköping University
581 83 Linköping
+46 13 28 40 42
daniel.axehill@liu.se

Gustaf Hendeby
Linköping University
581 83 Linköping
+46 13 28 58 15
gustaf.hendeby@liu.se

Abstract—This paper proposes a method to generate informative trajectories for a mobile sensor that tracks agile targets. The goal is to generate a sensor trajectory that maximizes the tracking performance, captured by a measure of the covariance matrix of the target state estimate. The considered problem is a combination of estimation and control, and is often referred to as informative path planning (IPP). When using nonlinear sensors, the tracking performance depends on the actual measurements, which are naturally unavailable in the planning stage. The planning problem hence becomes a stochastic optimization problem, where the expected tracking performance is used in the objective function. The main contribution of this work is an approximation of the problem based on deterministic sampling of the predicted target distribution. This is in contrast to prior work, where only the most likely target trajectory is considered. It is shown that the proposed method greatly improves the ability to track agile targets, compared to a baseline approach.

TABLE OF CONTENTS

1. INTRODUCTION.....	1
2. PROBLEM FORMULATION.....	2
3. MOTION DISCRETIZATION.....	2
4. OBJECTIVE FUNCTION APPROXIMATIONS.....	3
5. GRAPH SEARCH ALGORITHM.....	4
6. SIMULATION STUDY.....	6
7. CONCLUSIONS.....	10
ACKNOWLEDGMENTS.....	10
REFERENCES.....	10

1. INTRODUCTION

This paper addresses the problem of optimizing the trajectory for a mobile sensor such that the target tracking performance is maximized. The considered problem is often referred to as informative path planning (IPP) [1] and is closely related to the field of sensor management [2] as it is a combination of estimation and control. IPP problems are typically formulated as optimization problems where the objective is to maximize the information and have been widely studied. Applications include environmental monitoring [3], area exploration [4], and active target tracking [5].

IPP problems in general are stochastic optimization problems, as they involve optimizing over uncertain quantities. When using sensors with limited field of view or other nonlinearities, the tracking performance depends on the actual measurements obtained, as well as the trajectory of the target. This is a complicating factor for IPP, as the objective function of the optimization problem thus depends on future measure-

ments and the true target trajectory, both which are unavailable in the planning stage. A problem that involves planning under uncertainty can formally be posed as a partially observable Markov decision process (POMDP) [6], which provides a general framework for this type of problems. The computational complexity of solving POMDP problems is considerable, and in many cases approximations are needed to at all solve the problem in reasonable time.

A popular approximation is to proceed with a receding horizon approach and represent the target state with Gaussian distributions. By neglecting the uncertainty in future target trajectories, the resulting deterministic problem can be solved using local numerical optimization [7], sampling-based techniques [8], and graph search methods [9], [10]. This paper extends the work of [9] by incorporating the knowledge of uncertainty of the future target trajectory in the planning process.

Most IPP approaches, including the one considered in this paper, assume that the state of the sensor is perfectly known and that the uncertainty comes from the unknown state of the targets. A related problem is considered in motion planning under uncertainty, where knowledge of the uncertainty of the vehicle state is incorporated into motion planning problems [11], [12], [13], [14]. For classical motion planning problems, where the vehicle state is considered to be perfectly known and the objective is to find a path between two locations, solutions can be obtained efficiently using, *e.g.*, a state-lattice approach [15]. This work applies the state-lattice approach in an IPP setting.

In this paper, the problem of tracking a moving target by controlling the trajectory of a mobile sensor with limited field of view and sensing range is considered. The problem is formulated as a stochastic IPP problem and a method to solve this problem using approximations based on sampling of the predicted target distribution is proposed. An extended Kalman filter (EKF) is used to estimate the state of the target and by sampling the predicted target distribution, a number of plausible trajectories are obtained. These candidate target trajectories are then used to approximate the expectation with respect to the target state. This is in contrast to prior work, where only the most likely target trajectory is considered. The proposed method is evaluated using Monte Carlo simulations of several tracking scenarios. A conventional IPP method is used as baseline. It is shown how the proposed method significantly improves the ability to track agile targets, with retained tracking accuracy of the tracked targets, compared to the baseline approach.

2. PROBLEM FORMULATION

The problem considered in this paper is tracking of a moving target using a mobile sensor with limited field of view. This boils down to an optimization problem with a measure of the expected tracking accuracy as objective function, and where models of the measurements, as well as the platform and the target become constraints, as outlined below.

Motion and Measurement Models

Consider a sensor mounted on a vehicle, a *sensor platform*, where the dynamics are governed by the model

$$s_{k+1} = g(s_k, u_k), \quad (1)$$

where s_k is the state of the sensor platform and u_k is the control input, both at time k . The set of feasible states is denoted by \mathcal{S} and the set of admissible controls is denoted by \mathcal{U} .

The motion of the target is assumed random, and it is modeled by the stochastic equation

$$x_{k+1} = f(x_k) + v_k, \quad v_k \sim \mathcal{N}(0, Q_k), \quad (2)$$

where x_k represents the state of the target and v_k is the process noise, which is assumed to be a white and Gaussian, with covariance Q_k .

The sensor is assumed to have limited field of view and sensing range. If the target is visible to the sensor at time k , an imperfect observation of the target is obtained according to

$$y_k = h(x_k, s_k) + e_k, \quad e_k \sim \mathcal{N}(0, R_k), \quad (3)$$

where y_k is the measurement obtained by the sensor and e_k is additive white measurement noise, which is assumed to be Gaussian with covariance R_k .

Information Representation

An *extended Kalman filter* (EKF) [16] is employed to track the state x_k of the target based on available measurements y_k . The output of the EKF is a state estimate \hat{x}_k with corresponding covariance matrix P_k . A natural representation of the target tracking accuracy is a measure of the covariance matrix.

The EKF works in a two stage process. In each time step, a prediction step is performed according to

$$\hat{x}_{k+1|k} = f(\hat{x}_{k|k}), \quad (4a)$$

$$P_{k+1|k} = F_k P_{k|k} F_k^\top + Q_k, \quad (4b)$$

where

$$F_k \triangleq \frac{\partial f(\hat{x}_{k|k})}{\partial x}.$$

If a measurement is obtained at time k , *i.e.*, if the target is within the field of view of the sensor at time k , the state estimate and covariance matrix are updated in a measurement update step

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (y_k - h(\hat{x}_{k|k-1}, s_k)), \quad (5a)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}, \quad (5b)$$

where

$$K_k = P_{k|k-1} H_k^\top (H_k P_{k|k-1} H_k^\top + R_k)^{-1},$$

$$H_k \triangleq \frac{\partial h(\hat{x}_{k|k-1}, s_k)}{\partial x}.$$

If no measurement is obtained at time k , *i.e.*, $y_k = \emptyset$, no measurement update is performed at that time step and the time indices are simply updated: $\hat{x}_{k|k} = \hat{x}_{k|k-1}$, $P_{k|k} = P_{k|k-1}$.

For notational convenience, let $\hat{x}_k = \hat{x}_{k|k}$ and $P_k = P_{k|k}$ and abbreviate the update steps of the EKF as

$$\hat{x}_{k+1} = \rho_x(\hat{x}_k, P_k, s_{k+1}, y_{k+1}), \quad (6a)$$

$$P_{k+1} = \rho_P(\hat{x}_k, P_k, s_{k+1}). \quad (6b)$$

Informative Path Planning Problem

The goal of the considered planning problem is to generate a sensor trajectory that maximizes the tracking performance. Since an EKF is used to track the target, an indication of the tracking performance can be obtained by a measure of the resulting covariance matrix of the target state estimate. By denoting this measure $l_k(\cdot)$, the objective function can be written as

$$J_N = \sum_{k=1}^N l_k(P_k), \quad (7)$$

where N is the planning horizon. The authors of [17] provide several suggestions for the choice of $l_k(\cdot)$, *e.g.*, the volume of the confidence ellipsoid $l_k(P) = \det(WP)$ or the overall expected mean squared error for the state estimator, given by $l_k(P) = \text{tr}(WP)$, where W is a positive semidefinite weighting matrix.

With the objective function and models outlined above, the planning problem can be formally stated as follows. Given a finite planning horizon N , an initial state of the mobile sensor \bar{s}_0 , and an initial distribution of the target state $x_0 \sim \mathcal{N}(\hat{x}_{0|0}, P_{0|0})$, find a sequence of control inputs by solving

$$\text{minimize} \quad \mathbf{E} \left[\sum_{k=1}^N l_k(P_k) \right] \quad (8)$$

$$\begin{aligned} \text{subject to} \quad & s_{k+1} = g(s_k, u_k), \\ & x_{k+1} = f(x_k) + v_k, \quad v_k \sim \mathcal{N}(0, Q_k), \\ & y_k = h(x_k, s_k) + e_k, \quad e_k \sim \mathcal{N}(0, R_k), \\ & \hat{x}_{k+1} = \rho_x(\hat{x}_k, P_k, s_{k+1}, y_{k+1}), \\ & P_{k+1} = \rho_P(\hat{x}_k, P_k, s_{k+1}), \\ & x_0 = \hat{x}_{0|0}, \quad (\hat{x}_0, P_0) = (\hat{x}_{0|0}, P_{0|0}), \\ & s_0 = \bar{s}_0, \quad s_k \in \mathcal{S}, \quad u_k \in \mathcal{U}, \end{aligned}$$

where the expectation is taken with respect to the stochastic target trajectory and measurement noise.

Since the target maneuvers, it is necessary to re-plan the sensor trajectory online as new measurements are obtained. This is done in standard receding horizon fashion [18].

The following sections propose methods to solve the stochastic planning problem using approximations based on sampling of the predicted target distribution in different ways.

3. MOTION DISCRETIZATION

The problem (8) is challenging to solve. To make it computationally tractable, the numerical optimization problem can be

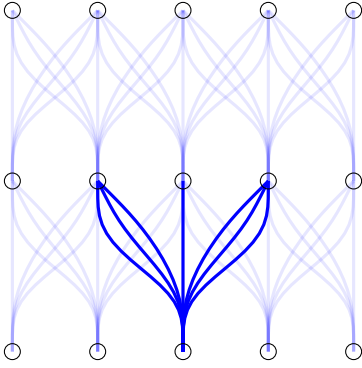


Figure 1: Example of a state-lattice. The motion primitives (blue) are position-invariant and can be applied to all grid positions of the lattice.

approximated by a graph search problem by applying a state-lattice motion planning approach [15]. This means that the continuum of control inputs is discretized and that the motion of the sensor platform instead is computed by combining feasible motions from a finite set of pre-computed motion primitives.

The motion primitives are generated using numerical optimal control [19], which works for both discrete-time and continuous-time descriptions of the platform dynamics. Constraints on inputs and states can be handled to guarantee that the motion primitives are dynamically feasible for the sensor platform. It is also possible to use different objective functions depending on the application, *e.g.*, minimum energy or minimum jerk.

Using the state-lattice motion planning approach [15], the continuous state space \mathcal{S}_c of the sensor platform is sampled in a regular form to obtain a discrete set of states $s_k \in \mathcal{S} \subset \mathcal{S}_c$, referred to as lattice states. The lattice states of \mathcal{S} define the states that the sensor platform can reach using a finite set of motion primitives \mathcal{P} . By utilizing position-invariance, each motion primitive can be used to connect any two equivalent lattice states, as shown in Figure 1.

With some abuse of notation, the discrete motion model of the sensor platform, that will be used in this paper, is given by

$$s_{k+1} = g(s_k, p_k), \quad (9)$$

where $s_k \in \mathcal{S}$ is the state of the sensor platform, \mathcal{S} is the set of lattice states, and $p_k \in \mathcal{P}$ is the applied motion primitive.

4. OBJECTIVE FUNCTION APPROXIMATIONS

The main challenge with (8) is that it is a stochastic optimization problem. For linear Gaussian systems, the covariance matrix recursion of the Kalman filter is independent of the state of the target and the informative path planning problem (8) becomes a deterministic problem. This is not the case for nonlinear systems, since (6b) involves a linearization around the current state estimate, which depends on the actual measurements which are naturally unavailable in the planning stage.

This section develops approximations of the stochastic optimization problem based on sampling of the predicted target distribution in different ways. First, a commonly used approach that will be used as baseline in this paper is presented.

Following this, the approximation that is developed for this paper is discussed.

Maximum Likelihood Observations

One way to handle stochastic systems in receding horizon control is to approximate them by deterministic systems. A straightforward approach is to replace stochastic quantities with their maximum likelihood values. For the informative path planning problem (8), there are two sources of stochasticity: the target state and the observations. Thus, a reasonable approximation is to replace the stochastic target state by its maximum likelihood value $x_k^{\text{ml}} = \arg \max p(x_k)$ and the stochastic observation y_k by $y_k^{\text{ml}} = \arg \max_y p(y|x_k^{\text{ml}})$. With these assumptions, the stochasticity in (6a) is eliminated and by that also the stochasticity in (6b), and the following deterministic problem is obtained:

$$\begin{aligned} & \text{minimize} && \sum_{k=1}^N l_k(P_k) && (10) \\ & \text{subject to} && s_{k+1} = g(s_k, p_k), \\ & && \hat{x}_{k+1} = f(\hat{x}_k), \\ & && P_{k+1} = \rho_P(P_k, \hat{x}_k, s_{k+1}), \\ & && (\hat{x}_0, P_0) = (\hat{x}_{0|0}, P_{0|0}), \\ & && s_0 = \bar{s}_0, \quad s_k \in \mathcal{S}, \quad p_k \in \mathcal{P}. \end{aligned}$$

The assumption of obtaining maximum likelihood observations corresponds to linearizing the measurement model around the predicted target trajectory during the planning phase and is commonly used in informative path planning [9], [10] and motion planning under uncertainty [14], [20]. This approximation will be used as a baseline approach in this paper, and will thus be referred to as *the baseline approach* for the remainder of this paper.

Expectation of the Target State

A more advanced approximation than the baseline approach is to explicitly consider the stochasticity in the target state and use the maximum likelihood assumption only for the measurements. This approximation gives another stochastic optimization problem, where the expectation is taken with respect to the target trajectory. This is still a challenging problem, and finding a solution would require solving a complicated integral in each time step. However, numerical integration can be used as an approximation. By sampling the distribution of the predicted target state, a number of plausible target trajectories are obtained. These candidate target trajectories are then used to approximate the expectation with respect to the target state. This approximation yields the following deterministic optimization problem:

$$\begin{aligned} & \text{minimize} && \sum_{i=1}^L \sum_{k=1}^N w^{(i)} l_k(P_k^{(i)}) && (11) \\ & \text{subject to} && s_{k+1} = g(s_k, p_k), \\ & && y_k^{(i)} = h(x_k^{(i)}, s_k), \\ & && \hat{x}_{k+1}^{(i)} = \rho_x(\hat{x}_k^{(i)}, P_k^{(i)}, s_{k+1}, y_{k+1}^{(i)}), \\ & && P_{k+1}^{(i)} = \rho_P(\hat{x}_k^{(i)}, P_k^{(i)}, s_{k+1}), \\ & && (\hat{x}_0^{(i)}, P_0^{(i)}) = (\hat{x}_{0|0}, P_{0|0}), \\ & && s_0 = \bar{s}_0, \quad s_k \in \mathcal{S}, \quad p_k \in \mathcal{P}, \quad x_k^{(i)} \in \mathcal{X}_k, \end{aligned}$$

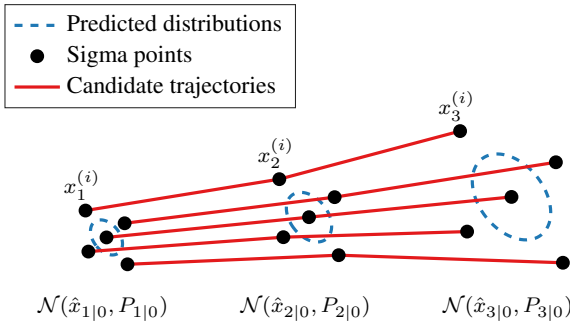


Figure 2: Conceptual illustration of how the candidate target trajectories are generated.

where i is used for entities related to the i^{th} candidate target trajectory, L is the number of candidate target trajectories, $w^{(i)}$ is the weight of candidate trajectory i , and \mathcal{X}_k is the set of sampled target states at time k .

The number of candidate target trajectories and the method for selecting them affect the accuracy of the approximation. If a large number of candidate trajectories are selected at random, the expectation with respect to the target trajectory is approximated using Monte Carlo integration. If instead a few deterministically sampled trajectories are used to represent the distribution, the approach resembles the unscented transform. If just a single candidate target trajectory is used and selected to be the most likely target trajectory, the method coincides with the baseline approach from [9].

Several alternatives for selecting the candidate trajectories exist. This paper uses a method inspired by the unscented transform [21] to select the candidate trajectories by deterministic sampling of the predicted target state distribution. In the standard form of the unscented transform, which is used in this work, $L = 2n_x + 1$ carefully chosen samples, called sigma points, are used to represent the distribution. The parameter n_x is the dimension of the state, and the representing sigma points $x^{(i)}$ and their corresponding weights $w^{(i)}$ are chosen as

$$x^{(0)} = \mathbf{E}[x], \quad w^{(0)} \text{ is a parameter,} \quad (12a)$$

$$x^{(\pm i)} = x^{(0)} \pm \sqrt{\frac{n_x}{1 - w^{(0)}}} z_i, \quad w^{(\pm i)} = \frac{1 - w^{(0)}}{2n_x}, \quad (12b)$$

where z_i is the i^{th} column in the square root of $\text{cov}(x)$; B is the a square root of A if $A = BB^T$.

The proposed method to generate candidate target trajectories consists of three steps: (i) prediction, (ii) sampling, and (iii) connection. The first step is to predict the *a priori* target state distribution $\mathcal{N}(\hat{x}_{k|0}, P_{k|0})$ for each time step k within the planning horizon N . In the sampling step, the predicted distributions are sampled according to (12) in order to create sets \mathcal{X}_k of representing sigma points and weights. In the third step, the candidate target trajectories are created by connecting the matching sigma points. The process is illustrated in Figure 2.

5. GRAPH SEARCH ALGORITHM

The solution to IPP problems with discrete sets of motion primitives can be obtained by enumerating all possible sensor

Algorithm 1 $\text{expand}(\mathcal{T}_{k-1})$

```

 $\mathcal{T}_k = \emptyset$ 
for all  $(s_{k-1}, J_{k-1}, \{\hat{x}_{k-1}^{(i)}, P_{k-1}^{(i)}\}_{i=1}^L) \in \mathcal{T}_{k-1}$  do
  for all feasible  $p_{k-1} \in \mathcal{P}$  do
     $s_k = g(s_{k-1}, p_{k-1})$ 
    for  $i = 1 : L$  do
       $y_k^{(i)} = h(x_k^{(i)}, s_k) \triangleright x_k^{(i)}$  is a parameter
       $\hat{x}_k^{(i)} = \rho_x(\hat{x}_{k-1}^{(i)}, P_{k-1}^{(i)}, s_k, y_k^{(i)})$ 
       $P_k^{(i)} = \rho_P(\hat{x}_{k-1}^{(i)}, P_{k-1}^{(i)}, s_k)$ 
    end for
     $J_k = J_{k-1} + \sum_{i=1}^L w^{(i)} l_k(P_k^{(i)})$ 
     $\mathcal{T}_k = \mathcal{T}_k \cup (s_k, J_k, \{\hat{x}_k^{(i)}, P_k^{(i)}\}_{i=1}^L)$ 
  end for
end for
return  $\mathcal{T}_k$ .

```

trajectories and finding the one that results in the smallest objective value. However, the number of possible trajectories increases exponentially with the planning horizon N and the number of motion primitives, which makes the exhaustive search computationally intractable. This section presents methods that decrease the computational complexity of the search, without giving up guarantees on global optimality of the solution.

Graph Generation

Classical motion planning using state-lattices involves a systematic search over a directed graph, where the nodes are the discrete states of the lattice $s_k \in \mathcal{S}$ and the edges are the motion primitives of \mathcal{P} [15]. In this setting, the aim is not to find a path to a given goal but rather to find a path from where the sensor can obtain maximally informative measurements. Thus, each node also contains the accumulated objective value J_k as well as the target state estimate $\hat{x}_k^{(i)}$ and the corresponding covariance matrix $P_k^{(i)}$ for each of the candidate target trajectories. The graph, or search tree, is constructed using a forward search, and we let $\text{expand}(\mathcal{T}_k)$ denote the process of generating the set of nodes \mathcal{T}_{k+1} by expanding the tree from level \mathcal{T}_k , *i.e.*, applying every feasible control input $p_k \in \mathcal{P}$ to each node $(s_k, J_k, \{\hat{x}_k^{(i)}, P_k^{(i)}\}_{i=1}^L) \in \mathcal{T}_k$ in order to generate new nodes. The process is outlined in Algorithm 1.

By constructing the search tree, traditional systematic graph search methods can be applied to find the solution of the planning problem. The most straightforward approach is to perform an exhaustive search and form the search tree by sequentially applying $\text{expand}(\cdot)$ to its outermost level. The exhaustive search is guaranteed to find the optimal trajectory but suffers from exponential complexity since the number of nodes in the search tree corresponds to the number of trajectories of length N . Thus, this approach quickly gets intractable.

Traditional graph search methods typically use heuristic estimates of the remaining cost to increase the efficiency of the search [22]. For classical motion planning, where a goal state is specified, an intuitive heuristic is the remaining path length, *e.g.*, approximated by the Euclidean distance to the goal. When the heuristic is defined, algorithms like A* [23] can be used to avoid searching in parts of the tree that are unlikely to contain the optimal solution.

Algorithm 2 greedy_search

```
 $\mathcal{T}_0 = \{(s_0, 0, \{\hat{x}_0^{(i)}, P_0^{(i)}\}_{i=1}^L)\}$ 
for  $k = 1 : N$  do
   $\mathcal{T}'_k = \text{expand}(\mathcal{T}_{k-1})$ 
   $\bar{J}_k = \infty$ 
  for all  $(s_k, J_k, \{\hat{x}_k^{(i)}, P_k^{(i)}\}_{i=1}^L) \in \mathcal{T}'_k$  do
    if  $J_k < \bar{J}_k$  then
       $\mathcal{T}_k = \{(s_k, J_k, \{\hat{x}_k^{(i)}, P_k^{(i)}\}_{i=1}^L)\}$ 
       $\bar{J}_k = J_k$ 
    end if
  end for
end for
return  $\bar{J}_N$ .
```

For informative path planning it is more difficult to come up with a suitable heuristic, since there is no well-defined goal state to which the distance could estimate the remaining cost. Instead, the heuristic function would be used to approximate the amount of information that is possible to collect from the current state until the end of the planning horizon. An alternative to focusing the search using heuristics is to apply search tree reductions that decrease the width of the search and thereby lead to computational savings. The search algorithm that is used in this work uses two types of pruning strategies, that are described in the following subsections.

Branch-and-Bound

The first pruning strategy has similarities to the branch-and-bound algorithm [24] in the sense that it is based on upper and lower bounds on the optimal objective value. During the execution of the search algorithm, an upper bound on the globally optimal objective value is maintained. As a new node is created, a lower bound on the objective value for a trajectory that contains the newly created node is computed. If the lower bound at a certain node is greater than the upper bound on the optimal objective value, the subtree rooted at that particular node does not need to be explored further. The optimal trajectory will not contain the node, since there must exist a better node in another part of the search tree, and the node can be pruned.

The first upper bound on the optimal objective value is obtained using a greedy search method that is outlined in Algorithm 2. Each level of the search tree created with this greedy method contains just one single node — all nodes except the one with lowest objective value are discarded when the level is created.

As the full search is conducted and new nodes are added, the upper bound may be improved. When a new node is created, an upper bound on the objective value for a trajectory that contains that particular node is computed. If the current upper bound on the optimal objective value is greater than the new bound, it is replaced by the new bound. There are several ways to compute the upper bound, *e.g.*, by performing a greedy search from the new node to the end of the planning horizon. To save computations, we compute the upper bound using a worst case scenario by evaluating the objective function as if no more measurements were to be obtained.

The lower bound in branch-and-bound algorithms can be found in various ways, *e.g.* by convex relaxations of the original problem. Generally, there is a trade-off between how tight a lower bound is and the effort needed to compute it. The

algorithm in this paper uses a simple method to find the lower bound: a lower bound on the objective value for a trajectory that contains a certain node is the accumulated cost at that particular node. This corresponds to assuming that the state of the target will be perfectly known for the remaining time steps.

Pruning Based on Algebraic Redundancy

The second pruning strategy is based on the ideas in [25]. If there are several trajectories that end up in the same state of the sensor platform at the same time but yield different covariance matrices, the trajectories that result in clearly less informative measurements are considered redundant and can be discarded. This pruning strategy is the main result of [25], where the idea of redundancy is formalized using the following definition and theorem:

Definition 1: (Algebraic redundancy) [25].

A node $(s, J, \{\hat{x}^{(i)}, P^{(i)}\}_{i=1}^L)$ is algebraically redundant with respect to the set $\{(s_j, J_j, \{\hat{x}_j^{(i)}, P_j^{(i)}\}_{i=1}^L)\}_{j=1}^K$ if there, for all $i \in [1, L]$, exist nonnegative constants $\alpha_j^{(i)}$ such that

$$\sum_{j \text{ s.t. } s_j=s} \alpha_j^{(i)} = 1, \quad \text{and} \quad \begin{bmatrix} P & 0 \\ 0 & J \end{bmatrix} \succeq \sum_{j \text{ s.t. } s=s_j} \alpha_j^{(i)} \begin{bmatrix} P_j^{(i)} & 0 \\ 0 & J_j \end{bmatrix},$$

where \succeq denotes the generalized inequality associated with the positive semidefinite cone.

Theorem 1 (Algebraic redundancy pruning [25]) Let \mathcal{T} be a set of nodes at the same level of the search tree. If the node $A \in \mathcal{T}$ is algebraically redundant with respect to $\mathcal{T} \setminus A$, then the node A and all of its descendants can be pruned without eliminating the optimal solution from the search tree.

By applying Theorem 1, algebraically redundant nodes can be pruned from the search tree without the risk of eliminating the optimal trajectory.

Resulting Algorithm

The resulting algorithm for constructing the search tree is outlined in Algorithm 3. By combining the two pruning strategies presented in the previous subsections it is possible to reduce the number of nodes that need to be explored in order to find the optimal solution.

Algorithm 3 Graph search

```
 $\mathcal{T}_0 = \{(s_0, 0, \{\hat{x}_0^{(i)}, P_0^{(i)}\}_{i=1}^L)\}$ 
 $\bar{J}_N = \text{greedy\_search}(\mathcal{T}_0)$ 
for  $k = 1 : N$  do
   $\mathcal{T}'_k = \text{expand}(\mathcal{T}_{k-1})$ 
   $\mathcal{T}_k = \emptyset$ 
  for all  $A_k = (s_k, J_k, \{\hat{x}_k^{(i)}, P_k^{(i)}\}_{i=1}^L) \in \mathcal{T}'_k$  do
    if  $J_k \leq \bar{J}_N$  then  $\triangleright$  Pruning based on upper bound
      if  $A_k$  not alg. red. w.r.t.  $\mathcal{T}_k$  then
         $\mathcal{T}_k = \mathcal{T}_k \cup A_k$ 
         $\bar{J}_N = \min\{\bar{J}_N, \text{upper\_bound}(A_k)\}$ 
      end if
    end if
  end for
end for
return best node in  $\mathcal{T}_N$ .
```

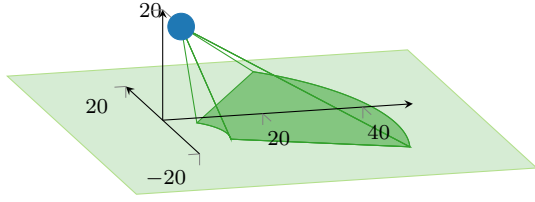


Figure 3: Illustration of the sensor’s limited field of view. Targets located in the darker area are visible to the sensor.

6. SIMULATION STUDY

In this section, the planning approaches are evaluated on a number of target tracking scenarios. The aim of the simulation study is to evaluate the effects of varying target agility, planning horizon, and robustness to different measurement noise levels.

Simulation Setup

The true target moves on the ground according to a differential drive model with angular rate ω and constant speed of five meters per seconds. For planning and estimation, the target state is modeled by its two dimensional position and velocity $x = [x_1, x_2, v_1, v_2]^T$ and a near-constant velocity motion model driven by Gaussian white noise, discretized with a sampling period $\tau = 0.5$ s. That is

$$x_{k+1} = \begin{bmatrix} I & \tau I \\ 0 & I \end{bmatrix} x_k + v_k,$$

$$v_k \sim \mathcal{N}\left(0, q \begin{bmatrix} \tau^3/3I & \tau^2/2I \\ \tau^2/2I & \tau I \end{bmatrix}\right).$$

The sensor platform moves according to differential drive dynamics at a constant altitude of 20 meters, and the state consists of its two dimensional position $[s_1, s_2]^T$, heading angle θ and speed v . When applying the state-lattice approach, the state space needs to be discretized. For the experiments, the state space discretization is made such that the discretization accuracy of the position is 0.5 meter for both s_1 and s_2 , the heading angle is discretized into 16 different angles and $v \in \{4, 6\}$ m/s. The motion primitives are generated using ACADO Toolkit [26] and fulfills the dynamic constraints of the motion model, which makes it possible to guarantee that the planned trajectory will be dynamically feasible for the sensor platform.

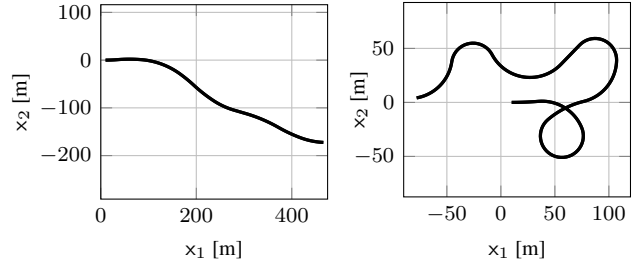
The sensor platform carries a lidar-like sensor with limited field of view and limited sensing range, as illustrated in Figure 3. If the target is visible to the sensor, the obtained measurement consists of the horizontal range and bearing from the sensor to the target:

$$y_k = h(x_k, s_k) + e_k, \quad e_k \sim \mathcal{N}(0, R_k),$$

$$h(x, s) = \begin{bmatrix} r(x, s) \\ \alpha(x, s) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2} \\ \arctan((x_2 - s_2)/(x_1 - s_1)) \end{bmatrix}.$$

Note that even though the magnitude of the measurement noise, given by the constant covariance matrix R_k , is not affected by the current state x_k of the sensor, the Jacobian of the measurement equation is given by

$$H = \frac{1}{r(x, s)} \begin{bmatrix} (x_1 - s_1) & (x_2 - s_2) & 0 & 0 \\ -\sin(\alpha(x, s)) & \cos(\alpha(x, s)) & 0 & 0 \end{bmatrix},$$



(a) Slowly maneuvering target (b) Agile target

Figure 4: Target trajectories considered in the experiments.

which means that the information contained in a measurement increases rapidly with decreasing distance between sensor and target. Note that the information cannot increase indefinitely due to the sensor footprint, as illustrated in Figure 3. Since the magnitude of the covariance of the target state estimate is tightly connected to the amount of information collected and the aim is to reduce the covariance as much as possible, the sensor should position itself such that the target is as close as possible, but still within field of view.

Simulation Results

The proposed method is evaluated and compared to the baseline approach using Monte Carlo simulations of different relevant tracking scenarios. The statistics are generated from 1000 simulations for each experiment. In all simulations, the stage cost is the trace of the covariance matrix: $l_k(P) = \text{tr}(P)$, and the sampling period is $\tau = 0.5$ s. The performance metrics are: (i) the ability to maintain track of the target, (ii) the root mean square error (RMSE) of the estimated target position, and (iii) the computational complexity of the planning algorithms.

Effects of Varying the Planning Horizon—The first set of experiments investigates how the planning horizon affects the performance. Two different scenarios are considered, both illustrated in Figure 4, in order to determine the impact of the planning horizon for different target motions. For all simulations, the parameter q is set to 0.1 and the sensor parameters are set to model a high-quality lidar: the standard deviation of the range and bearing measurements are $\sigma_r = 0.05$ m and $\sigma_\alpha = 0.5^\circ$, respectively.

In the first scenario, the target is slowly maneuvering and follows an almost straight path, which means that the near-constant velocity model is a fairly accurate description of target motion. This means that the maximum-likelihood prediction of the target trajectory often coincides with the actual target trajectory. Thus, the baseline approach should perform well in this scenario.

The results from the simulations of this scenario are presented in Figure 5. As expected, the baseline approach performs well and the sensor loses track of the target in only about two percent of the simulations. The performance of the baseline approach is also increased with longer planning horizon, which is expected since the target behaves as the model predicts. However, it is still outperformed by the proposed method, which manages to maintain track of the target in almost all simulations. In the second scenario, an agile target is performing sharp turns and is thus constantly deviating from the trajectory predicted by the constant veloc-

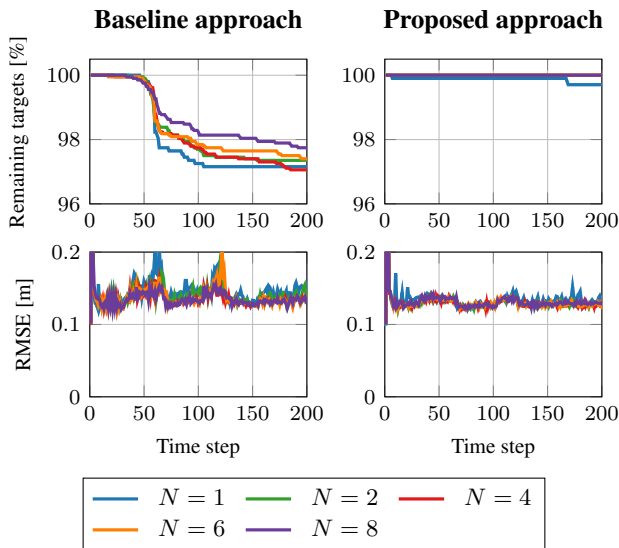


Figure 5: Results for active tracking of a slowly maneuvering target with trajectory as in Figure 4a. In the first column, the baseline approach is used to plan the trajectory for the sensor platform. In the second column, the approach proposed in this paper is used. Note that the y -axis is cropped in the upper row.

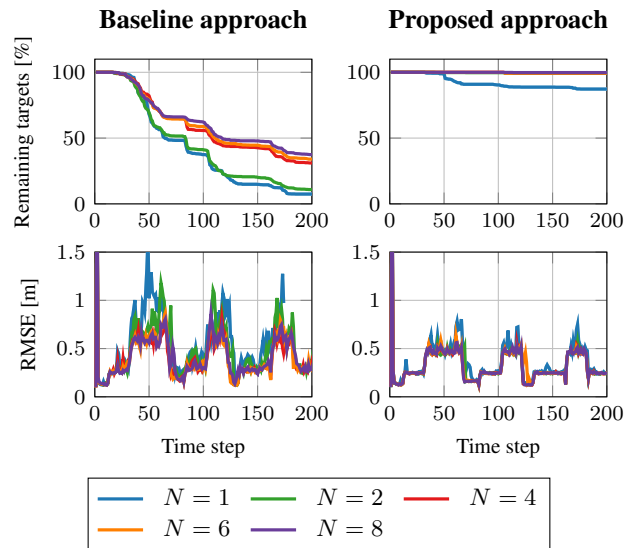


Figure 6: Results for active tracking of an agile target with trajectory as in Figure 4b. In the first column, the baseline approach is used to plan the trajectory for the sensor platform. In the second column, the approach proposed in this paper is used.

ity motion model. Since the baseline approach does not take uncertainty in the target trajectory into account, it is expected to perform worse than it did in the previous scenario. The proposed approach on the other hand takes this uncertainty into account, and should be able to handle also agile targets.

These predictions are confirmed by the simulations, of which the results are presented in Figure 6. The proposed approach is able to maintain track of the target in almost every simulation, provided that the planning horizon is longer than just a single time step. The baseline approach performs poorly for this scenario and even though the performance is increased by increasing the planning horizon, the agile target is still lost in more than 50 % of the simulations after 200 time steps.

Overall, the proposed approach is significantly less sensitive to the size of the planning horizon N . Even though a planning horizon of two steps would be enough for the proposed approach, $N = 6$ will be used for the remainder of the experiments since it gives reasonable performance also for the baseline approach.

Sensitivity to Target Agility—The next set of experiments investigates how well the planning algorithms are able to handle target trajectories of varying agility. We let the angular rate ω reflect the target’s agility level and test the performance for the target trajectories that are illustrated in Figure 7. The trajectory with largest angular rate, $\omega = \pi/12.5$, corresponds to the minimum turning radius for the sensor platform. For all simulations, the planning horizon is $N = 6$ and the standard deviation of the range and bearing measurements are $\sigma_r = 0.05$ m and $\sigma_\alpha = 0.5^\circ$, respectively.

The tuning parameter q reflects the expected agility of the target. A high value of q allows for sharp turns of the target. To evaluate how sensitive the algorithms are to the tuning of this parameter, the simulations are run for three different values of q : (i) a low value ($q = 0.01$), representative for

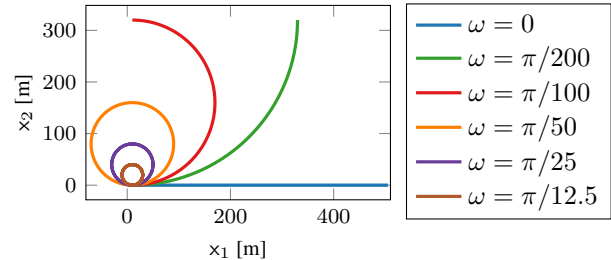


Figure 7: Target trajectories for different angular rates ω .

target trajectories where the magnitude of the angular rate ω is less than $\pi/100$; (ii) a “normal” value ($q = 0.1$), that could be considered as a reasonable tuning for tracking of targets with trajectories as in Figure 7; and (iii) a high value ($q = 1$), which would be appropriate to use for targets that turn more quickly than in the considered trajectories.

The sensor’s ability to maintain track of the target is shown in Figure 8. The baseline approach is able to handle slowly maneuvering targets, *i.e.*, trajectories with small angular rate ω , if the parameter q is tuned appropriately. In other cases, it generates sensor trajectories that eventually lead to losing track of the target. If the parameter q is set to a high value, the latest measurement greatly affects the current target state estimate. Since the baseline approach plans the sensor trajectory based only on the most likely target trajectory given the current estimate of the target state, a noisy measurement could send the sensor in a direction that leads to a lost track.

By taking the uncertainty in the target trajectory into account, the approach proposed in this paper greatly improves the ability to maintain track of the target. As shown in the second row of Figure 8, the target is essentially not lost in any of the simulations except for the case when the tuning of q is way

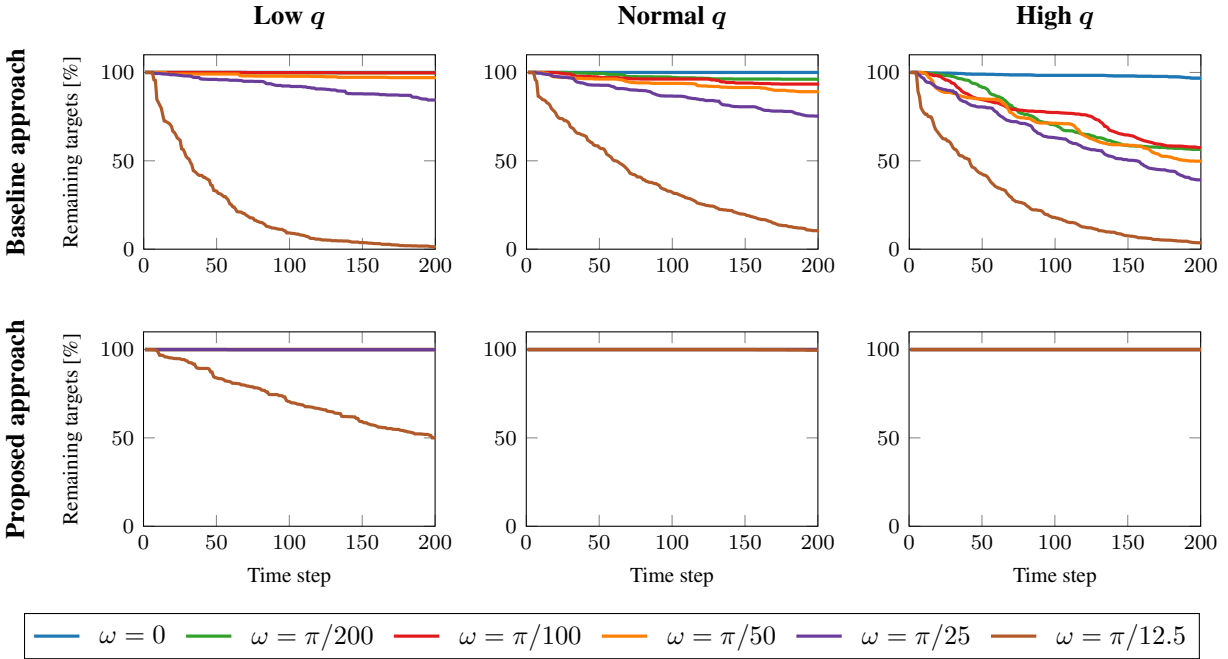


Figure 8: Percentage of the simulations that the sensor is able to keep track of the target for different target turning rates. In the first row, the baseline approach is used to plan the trajectory for the sensor platform. In the second row, the approach proposed in this paper is used.

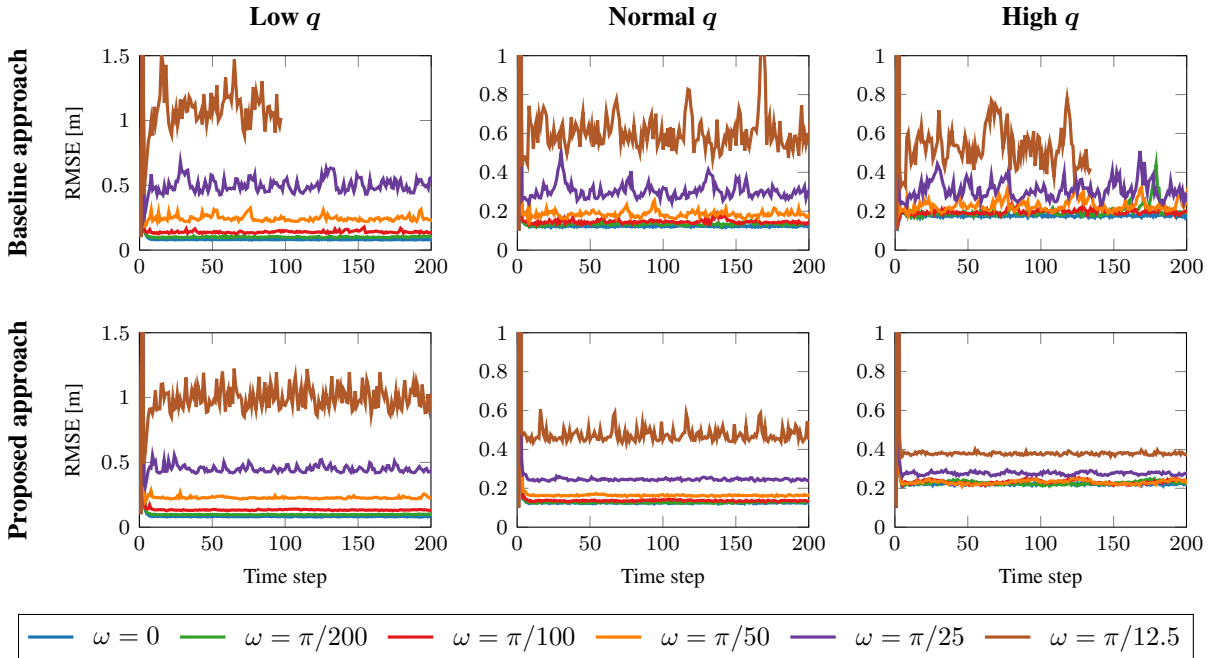


Figure 9: RMSE of the estimated target position for different target turning rates. In the first row, the baseline approach is used to plan the trajectory for the sensor platform. In the second row, the approach proposed in this paper is used.

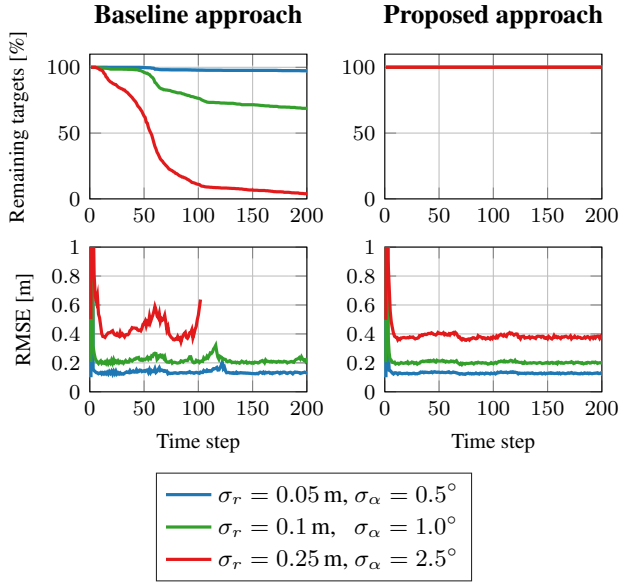


Figure 10: Results for active tracking of a slowly maneuvering target with trajectory as in Figure 4a. In the first column, the baseline approach is used to plan the trajectory for the sensor platform. In the second column, the approach proposed in this paper is used.

too low. Compared to the baseline approach, the proposed approach is significantly more robust to the tuning of the parameter q and avoids losing track of the target. In particular, the method copes with higher agility than expected much better than the baseline approach. This allows for tuning for the nominal case but still handling extreme cases.

Figure 9 shows the tracking accuracy of the tracked targets. Note that only targets that are tracked are taken into account when calculating the RMSE. By comparing the first row to the second row of Figure 9, it is evident that the position RMSE for the remaining targets are approximately the same for both approaches. Thus, the increased robustness of the proposed approach does not come at the price of reduced tracking accuracy in terms of RMSE.

Sensitivity to Measurement Noise—This set of experiments investigates how the magnitude of the measurement noise affects the performance of the planning algorithms. This is tested by simulating the scenarios illustrated in Figure 4 using different values for the standard deviation of the measurement noise. The noise levels are reasonable for lidars, and not greater than what the baseline approach can handle. For all simulations, the planning horizon is $N = 6$ and the parameter q is set to 0.1. The results are presented in Figures 10 and 11.

The ability to keep track of the target degrades with increasing level of measurement noise, but the proposed approach is more robust than the baseline approach. The baseline approach is dependent on having an accurate estimate of the target’s velocity, since it does not take uncertainty into account. The velocity is not measured directly, but estimated through measurements of the target’s position, which means that the accuracy of this estimate rapidly decreases as the magnitude of the measurement noise increases. Furthermore, for both scenarios and planning approaches, the performance in terms of RMSE degrades with increasing level of measurement noise, as might be expected.

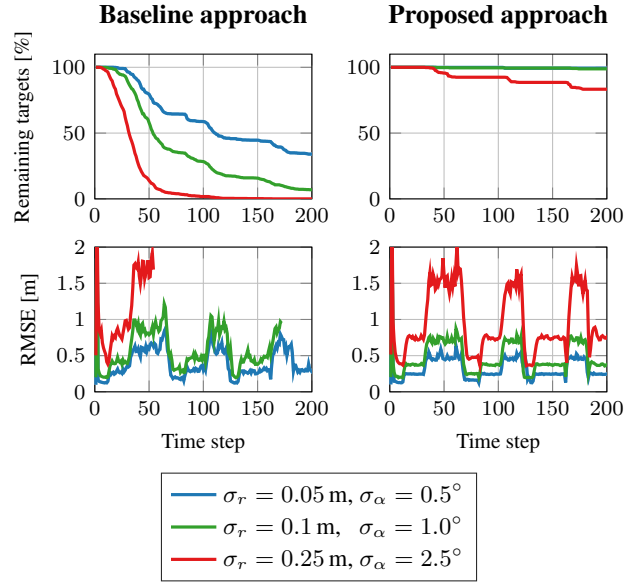


Figure 11: Results for active tracking of an agile target with trajectory as in Figure 4b. In the first column, the baseline approach is used to plan the trajectory for the sensor platform. In the second column, the approach proposed in this paper is used.

Computational Complexity—As new measurements are obtained, it is necessary to re-plan the sensor trajectory online to adapt to the newly acquired information. Hence, the time needed to find a solution to the graph search problem is of interest if the planning algorithm is to be used in real-time applications. To investigate this, the algorithms have been naively implemented in C++, with little effort to optimize the code, and simulations run on a laptop with a 2.6 GHz Intel i7 processor with 16 GB of RAM running Ubuntu 16.04. Figure 12 reveals that the computational complexity of both approaches grow rapidly with the planning horizon and that the proposed approach is, not surprisingly, computationally heavier than the baseline approach. However, recall that proposed approach performs significantly better than the baseline approach even when using a shorter planning horizon. Furthermore, when applying a receding horizon approach, the next plan can be computed during execution of the first motion primitive of the current plan. In this case, the duration of all motion primitives are 0.5s, which means that even this rudimentary implementation of the proposed approach is fast enough for use in real-time applications when using a planning horizon of less than eight time steps.

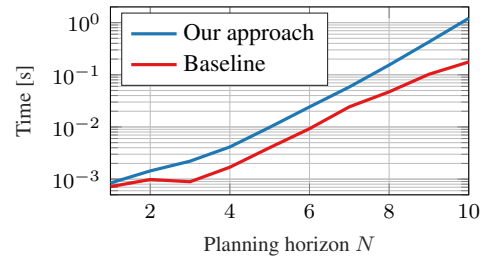


Figure 12: Average computation time needed to find the solution to the graph search problem for different planning horizons.

7. CONCLUSIONS

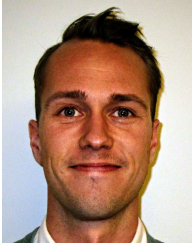
An approximate method for active target tracking problems where the objective function is dependent on the future target trajectory and thus stochastic has been proposed. The method takes the target uncertainty into account by deterministic sampling of the predicted target trajectory. The samples are selected to represent different target maneuvers and robustness is achieved by including all candidate trajectories in the objective function. Compared to a conventional approach that neglects the uncertainty in the target trajectory, the proposed approach was shown to increase robustness. In particular, the proposed approach decreases the need for a long planning horizon and is less sensitive to tuning of the parameters related to target agility and measurement noise. The ability to track agile targets is greatly improved, with retained tracking accuracy of the tracked targets. The benefits have been verified in a numerical study highlighting relevant properties. The proposed methodology opens up for taking other aspects of the uncertainty into account, *e.g.*, by more advanced schemes for selecting the candidate target trajectories.

ACKNOWLEDGMENTS

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

- [1] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *J. Art. Int. Res.*, vol. 34, pp. 707–755, 2009.
- [2] G. W. Ng and K. H. Ng, "Sensor management – what, why and how," *Information Fusion*, vol. 1, no. 2, pp. 67–75, 2000.
- [3] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, 2012.
- [4] F. Zhang and N. E. Leonard, "Cooperative filters and control for cooperative exploration," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 650–663, 2010.
- [5] T. H. Chung, J. W. Burdick, and R. M. Murray, "A decentralized motion coordination strategy for dynamic target tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, 2006, pp. 2416–2422.
- [6] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
- [7] C. Leung, S. Huang, N. Kwok, and G. Dissanayake, "Planning under uncertainty using model predictive control for information gathering," *Robot. Auton. Syst.*, vol. 54, no. 11, pp. 898–910, 2006.
- [8] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [9] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 6447–6454.
- [10] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1025–1032, 2018.
- [11] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, no. 11-12, pp. 1448–1465, 2009.
- [12] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [13] S. Patil, Y. Duan, J. Schulman, K. Goldberg, and P. Abbeel, "Gaussian belief space planning with discontinuities in sensing domains," in *Proc. IEEE Int. Conf. Robot. Autom.*, Hong Kong, China, 2014, pp. 6483–6490.
- [14] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel, "Scaling up Gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation," in *Algorithmic Foundations of Robotics XI*. Cham, Switzerland: Springer, 2015, pp. 515–533.
- [15] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field. Robot.*, vol. 26, no. 3, pp. 308–333, 2009.
- [16] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY, USA: Academic Press, 1970.
- [17] C. Yang, L. Kaplan, and E. Blasch, "Performance measures of covariance and information matrices in resource management for target state estimation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 3, pp. 2594–2613, 2012.
- [18] J. M. Maciejowski, *Predictive control: With constraints*. Englewood Cliffs, NJ, USA: Prentice Hall, 2002.
- [19] J. T. Betts, *Practical methods for optimal control and estimation using nonlinear programming*. Philadelphia, PA, USA: SIAM, 2010.
- [20] R. Platt Jr, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Proc. Robot.: Sci. Syst. VI*, Zaragoza, Spain, 2010.
- [21] S. J. Julier, "The scaled unscented transformation," in *Proc. Amer. Control Conf.*, vol. 6, Anchorage, AK, USA, 2002, pp. 4555–4559.
- [22] S. M. LaValle, *Planning Algorithms*. Cambridge, UK: Cambridge University Press, 2006.
- [23] J. Pearl, *Heuristics: Intelligent search strategies for computer problem solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.
- [24] C. A. Floudas, *Nonlinear and mixed-integer optimization: Fundamentals and applications*. London, UK: Oxford University Press, 1995.
- [25] 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.
- [26] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO Toolkit – An open source framework for automatic control and dynamic optimization," *Opti. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.



***Per Boström-Rost** is a PhD student in the Division of Automatic Control, Department of Electrical Engineering, Linköping University. He received his MSc degree in Electrical Engineering from Linköping University in 2013 and is since then a systems engineer at Saab Aeronautics. His research interests include sensor fusion and sensor control.*



***Daniel Axehill** received his M.Sc. degree in Applied Physics and Electrical Engineering in 2003. Furthermore, he received the degree of Lic.Eng. in Automatic Control in 2005 and the Ph.D. degree in Automatic Control in 2008. All three degrees are from Linköping University, Linköping, in Sweden. In year 2006 he spent three months at UCLA in Los Angeles. From January 2009*

and until November 2010 he worked as a post-doc at the Automatic Control Laboratory at ETH Zurich. He is currently employed as an Associate Professor at the Division of Automatic Control at Linköping University. His research interests are related to optimization, optimal control, motion planning, hybrid systems, and applications of control.



***Gustaf Hendeby** is Associate Professor in the Division of Automatic Control, Department of Electrical Engineering, Linköping University. He received his MSc in Applied Physics and Electrical Engineering in 2002 and his PhD in Automatic Control in 2008, both from Linköping University. He worked as Senior Researcher at the German Research Center for Artificial Intelligence (DFKI)*

2009–2011, and Senior Scientist at Swedish Defense Research Agency (FOI) and held an adjunct Associate Professor position at Linköping University 2011–2015. Dr. Hendeby's main research interests are stochastic signal processing and sensor fusion with applications to nonlinear problems, target tracking, and simultaneous localization and mapping (SLAM). He has experience of both theoretical analysis as well as implementation aspects.