2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Optimization-Based Autonomous Remote Sensing of Surface Objects Using an Unmanned Aerial Vehicle

Joakim Haugen
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
Email: joakim.haugen@ntnu.no

Lars Imsland
Department of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
Email: lars.imsland@ntnu.no

Abstract— **This manuscript presents an optimization-based approach for path planning of an aerial mobile sensor that monitors a set of moving surface objects. The purpose of the optimization problem is to obtain feasible mobile sensor trajectories with an objective to minimize the uncertainty of the objects, represented as the trace of the state estimation error covariance.**

The dynamic optimization problem is discretized into a large-scale nonlinear programming (NLP) problem using the direct transcription method known as simultaneous collocation. The numerical simulation periodically provides desired sensor trajectories and thus illustrates the approach.

I. Introduction

Remote sensing of terrestrial and atmospheric phenomena is utilized on, or considered for, a variety of applications. Urban planning [1], crop monitoring [2], polar remote sensing [3] and other environmental monitoring applications [4] all benefit from information obtained from moving sensor platforms such as satellites and manned aerial vehicles.

In many cases the required accuracy and frequency of acquisition are met by the mentioned sensor platforms. Satellites are, however, forced to follow predefined trajectories, and as a consequence limited flexibility in terms of frequency of acquisition is a reality. Of course, the number of relevant satellites may increase over time and partly mitigate this issue. Nevertheless, for demanding applications this may not be sufficient.

Manned aerial vehicles do have more flexibility than satellites in terms of following desired trajectories and may in some cases act as a valuable supplementary sensor platform. Moreover, the features are closer and it may be easier to obtain measurements of finer resolution. On the other hand, manned vehicles are bulky, expensive, and carry something invaluable, namely people.

The authors of [5] argued that unmanned aerial vehicles may become an inexpensive and efficient sensor platform in ice management operations, which include the objective of detecting, tracking, and forecasting of icebergs [6]. In ice-infested regions where marine operations are present there is a significant interest in solving this objective. The petroleum production at Grand Banks of Newfoundland is contaminated by drifting icebergs and needs to obtain reliable knowledge about the surroundings to ensure the safety of operation. "Most of the strategic and tactical iceberg detection for Grand Banks operations today is performed from fixed wing aircraft" [6].

The approach presented in this paper is motivated by the objective of tracking and forecasting icebergs trajectories using an unmanned aerial vehicle as a mobile sensor platform.

A. Previous Work

The general problem of path planning of mobile sensors to achieve the task of classification/forecast of some objects or distributed parameter system in either an open or cluttered environment has, due to its many applications, received especially much attention in literature. Reference [7] may serve as a starting point.

The development of a framework for continuous trajectory planning using a two-dimensional nonholonomic model for environmental forecasting of spatially distributed processes is presented in [8]. In [9], feasible real-time trajectory planning using direct collocation was successfully demonstrated using an aerial vehicle. The objective was to maximize the observation time of a single object, either moving or static. Klesh et al. [10] used a spatially motivated signal-to-noise ratio as a weighting of the measurement noise together with a Kalman filter in the path planning of a nonholonomic vehicle with the purpose to observe a set of static objects with corrupted visibility or uncertain object location.

B. Contributions

A framework for autonomous monitoring of moving surface objects is outlined. The path planning of a mobile sensor is accomplished by setting up an optimal control problem that minimizes weighted traces of the objects' covariance matrices. In this way the expected reduction in the uncertainty of non-spatial object state variables may be taken into consideration in the path planning.

C. Notation

The combined column vector of other vectors is denoted $\mathrm{col}(x, y, z) := [x^\mathsf{T}, y^\mathsf{T}, z^\mathsf{T}]^\mathsf{T}$. A countable finite index set of positive natural numbers is defined as $\mathbb{I}_n := \{i \in \mathbb{N}_1 : i \leq n\}$. Define the set of positive definite real matrices as $\Pi_n := \{A \in \mathbb{R}^{n\times n} : x^\mathsf{T}Ax > 0 \quad \forall n \in \mathbb{N}_1, x \in \mathbb{R}^n_{\neq 0}\}$. The planar orientation space is defined by $\mathbb{S} := [-\pi, \pi)$. The first moment of a random vector

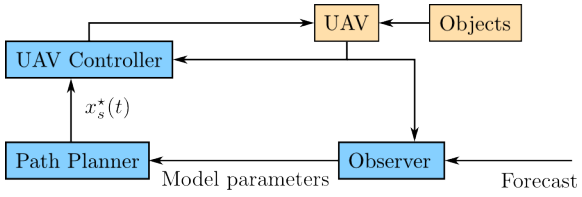

Fig. 1. Components of the Real-Time Optimization.

x is denoted by the expectation operator $\mathrm{E}(x)$. The covariance matrix of two random vectors x and y is defined as $\mathrm{cov}(x,y) := \mathrm{E}[(x-\mathrm{E}(x))(y-\mathrm{E}(y))^{\mathsf{T}}]$. A zero-mean continuous-time white noise process $w(t)$ of dimension n has the properties $\mathrm{E}(w) = 0$ and $\mathrm{cov}(w(t), w(\tau)) = Q(t)\delta(t-\tau)$, where $Q : \mathbb{R}_{\geq 0} \to \Pi_n$ is the deterministic spectral density and $\delta(t)$ is the dirac delta function. The above mentioned properties of $w(t)$ are written compactly as $w(t) \sim (0, Q(t))$. Given a column vector $v \in \mathbb{R}^n$, we define the weighted trace of a matrix $A \in \mathbb{R}^{n\times n}$ as $\mathrm{tr_w}(A,v) := \sum_{i=1}^{n} v_i a_{ii}$, where v_i is the i^{th} element of v and a_{ii} is the element at the i^{th} row and column of A.

II. Problem Formulation

A. System Overview

The system under consideration consists of five main components, shown in Fig. 1, with the following description:

UAV: Unmanned aerial vehicle that acts as a mobile remote sensor.

Objects: A set of physical processes located on a planar surface to be monitored by the UAV.

Observer: A system that processes raw measurements and other inputs to return the most likely model state and parameters of the UAV and the objects.

Path Planner: Generates guidance input $x_s^\star(t)$ of where we want the mobile sensor to go and obtain measurements.

UAV Controller: Feedback controller that achieves the desired task $x_s^\star(t)$.

The focus in this paper is the development of an optimization-based *Path Planner*. We will also briefly discuss the *Observer* component.

B. Mobile Sensor Dynamics

The sensor dynamics is intended for path-planning purposes and need not be a high-fidelity model. It is, however, important that the planned paths are feasible with respect to chosen maneuverability constraints of the mobile sensor. We assume that a constrained first-order nonlinear ordinary differential equation (ODE) with state and input constraints captures the vehicle dynamics with sufficient fidelity. Let for each $t \geq t_0$, $x_s(t) \in \mathbb{R}^{n_{x_s}}$ denote the state vector, $u_s(t) \in \mathbb{R}^{n_{u_s}}$ the control input, and $p_s \in \mathbb{R}^{n_{p_s}}$ a vector of constant parameters. The sensor model is described by the deterministic system

$$\dot{x}_s(t) = f_s(t, x_s(t), u_s(t), p_s), \tag{1a}$$
$$x_s(t_0) = x_{s,0}, \tag{1b}$$
$$x_s(t) \in \mathbb{X}_s \subseteq \mathbb{R}^{n_{x_s}}, \tag{1c}$$
$$u_s(t) \in \mathbb{U}_s \subseteq \mathbb{R}^{n_{u_s}}, \tag{1d}$$

where $f : \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_{x_s}} \times \mathbb{R}^{n_{u_s}} \times \mathbb{R}^{n_{p_s}} \to \mathbb{R}^{n_{x_s}}$ is a sufficiently smooth function, and both $\mathbb{X}_s$ and $\mathbb{U}_s$ are convex sets.

C. Object Dynamics

Let n_o be the number of objects and define the index set $\mathbb{O} := \mathbb{I}_{n_o}$. Then, $\forall m \in \mathbb{O}$, the dynamics of the objects can be described by first-order stochastic nonlinear ODEs as

$$\dot{x}_{o,m}(t) = f_{o,m}(t, x_{o,m}(t), w_{o,m}(t), p_{o,m}), \tag{2a}$$
$$x_{o,m}(t_0) = x_{o,m,0}, \tag{2b}$$

where for each $t \geq t_0$, $x_{o,m}(t) \in \mathbb{R}^{n_{x_{o,m}}}$ is the state vector, $w_{o,m}(t) \in \mathbb{R}^{n_{x_{o,m}}} \sim (0, Q_{o,m}(t))$ is process noise, $p_{o,m} \in \mathbb{R}^{n_{p_{o,m}}}$ is a vector of constant parameters, and $f_{o,m} : \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_{x_{o,m}}} \times \mathbb{R}^{n_{p_{o,m}}} \to \mathbb{R}^{n_{x_{o,m}}}$ is a sufficiently smooth function.

The output obtained from mobile sensor measurements are for each object $m \in \mathbb{O}$ defined as

$$y_{o,m}(t) = h_{o,m}(t, x_{o,m}(t), v_{o,m}(t)), \tag{2c}$$

where $x_{o,m}(t)$ is the state vector, $v_{o,m}(t) \in \mathbb{R}^{n_{y_{o,m}}} \sim (0, R_{o,m}(t))$ is measurement noise, and $h_{o,m} : \mathbb{R}_{\geq 0} \times \mathbb{R}^{n_{x_{o,m}}} \times \mathbb{R}^{n_{y_{o,m}}} \to \mathbb{R}^{n_{y_{o,m}}}$ is a sufficiently smooth function.

We assume that the mobile sensor is able to measure the states of each object in such way that the state vectors are observable in the sense that with $w_{o,m} = 0$, $v_{o,m} = 0$, $p_{o,m}$ known, and given the output $y_{o,m}(t)$ over some finite horizon $[t_0, T]$ are sufficient to uniquely determine the state vector $x_{o,m}(t_0)$ [11].

D. Object Uncertainty Measure

The states of the objects are random variables and the state estimates are equipped with an uncertainty measure. More specifically, $\forall m \in \mathbb{O}$ we define $\mathrm{E}(x_{o,m}(t)) = \hat{x}_{o,m}(t)$ as the state estimate of $x_{o,m}(t)$. For each $m \in \mathbb{O}$, the estimation error and estimation error covariance are defined, respectively, as

$$\tilde{x}_{o,m}(t) = x_{o,m}(t) - \hat{x}_{o,m}(t), \tag{3}$$
$$P_{o,m}(t) = \mathrm{cov}(\tilde{x}_{o,m}(t), \tilde{x}_{o,m}(t)). \tag{4}$$

E. Problem Statement

The objective is to devise an approach to generate feasible trajectories for the mobile sensor in such way that uncertainty measures of the objects' state estimates are minimized. This problem includes formulating and efficiently solving an optimization problem that includes both the constrained dynamics of the sensor, as well as the covariance response of the objects. The problem also involves finding a plausible way of letting the object covariance be influenced by the planar trajectory of the sensor.

III. Measurement Models for the Mobile Sensor

Define the two-dimensional Cartesian coordinates of the sensor as $q_s(t) \in \mathbb{R}^2$. Further, $\forall m \in \mathbb{O}$ we have the planar positions $q_{o,m}(t) \in \mathbb{R}^2$. We assume that the positions of the objects will remain within a subset of the constrained configuration space of the sensor. Let T denote the final time of interest. Then for all $t \in [0, T]$ we have $q_{o,m} \in \mathbb{D} \subseteq \mathbb{X}_s$. We also assume that the planar position is part of the dynamics of both the sensor and the objects, that is, $q_s \subseteq x_s$ and $\forall m \in \mathbb{O} : q_{o,m} \subseteq x_{o,m}$.

We assume that the objects are located in such a way that the mobile sensor cannot necessarily measure all the objects simultaneously. We propose to use a weighting which depends on the coordinates of the mobile sensor to reflect how the output vector is sampled by the UAV. In this context, the output vector is a vector function that depends on the state vector of an object being monitored. A mobile sensor may consist of a set of measuring devices that samples the states in different ways. These devices may not have the same measuring capabilities, so in order to keep the representation general we assume that each element of the output vector $y(t) \in \mathbb{R}^{n_y}$ is weighted with its own scalar weighting function.

Define the family of scalar weighting functions as $\mathbb{W}$ and let $\mathbb{B}$ be the codomain of this family. For all $i \in \mathbb{I}_{n_y}$ let $w_i : \mathbb{R}_{\geq 0} \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{S} \to \mathbb{B}$ and define a diagonal matrix function $W : (w_1, w_2, \cdots, w_{n_y}) \mapsto \operatorname{diag}(w_1, w_2, \cdots, w_{n_y})$ with codomain $\in \mathbb{B}^{n_y \times n_y}$. The weighted measurement vector is therefore defined as

$$y_w(t) = W\left(t, q_s(t), q(t), \psi(t)\right) y(t). \tag{5}$$

A weighted measurement vector captures the case where a measuring device has a field of view, for instance an image obtained from an optical device. It also captures the more general case of a smoothly decaying weight that depends on the distance to the measured object.

In the following we propose two different weighting functions, each having merit in different situations.

A. Non-smooth weighting function

The purpose of this model is to simulate that the measuring device has a field of view (FOV) in which it is able to obtain measurements. This includes for instance the cases of roll and pitch stabilized downward-looking optical devices and spectrometers.

Let $\Delta_{x_i} > 0$, $i \in \{1, 2\}$ and define $\Delta_x := \operatorname{col}(\Delta_{x_1}, \Delta_{x_2})$. We define the two-dimensional FOV metric as a weighted infinity norm

$$\|x\|_{\text{FOV},\Delta_x} := \max\left(\frac{|x_1|}{\Delta_{x_1}}, \frac{|x_2|}{\Delta_{x_2}}\right). \tag{6}$$

Suppose the origin of a body-fixed Cartesian coordinate system $\{b\}$ is $q_s(t)$ and its orientation relative some stationary reference frame $\{i\}$ is $\psi(t)$, following the right-hand rule. Let $\mathbb{B}_{\mathcal{C}^{-1}} := \{0, 1\}$ be the codomain of a binary weighting function $w_{\mathcal{C}^{-1}} : \mathbb{R}_{\geq 0} \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{S} \to \mathbb{B}_{\mathcal{C}^{-1}}$ such that the weighting is nonzero only if a coordinate point $q(t) \in \mathbb{R}^2$ is within the convex set formed by a FOV metric. The two-dimensional rotation matrix is

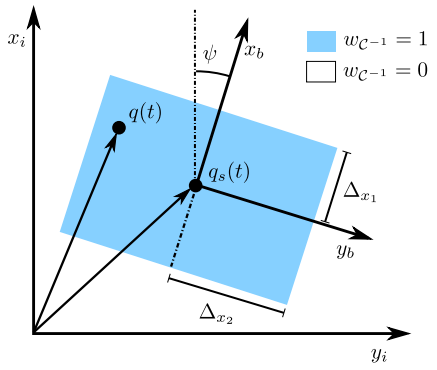

Fig. 2. The non-smooth weighting function is one inside the box and zero otherwise. x_i and y_i denote the axes of the stationary reference frame.

$$R(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}. \tag{7}$$

We can write the binary weighting function as

$$w_{\mathcal{C}^{-1}}(t, q_s, q, \psi) := \begin{cases} 1, & \left\|R^{\mathsf{T}}(\psi)(q - q_s)\right\|_{\text{FOV},\Delta_x} < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

Fig. 2 graphically illustrates the behavior of the binary weighting function. We see that Δ_{x_1} and Δ_{x_2} quantify respectively the field of view in x and y direction of the body-fixed reference frame.

B. Smooth weighting function

In some cases, for instance in an optimization problem, a continuous weighting function with positive weighting may be preferable. Let $\mathbb{B}_{\mathcal{C}^\infty} := \{w \in \mathbb{R} : 0 \leq w \leq 1\}$. Define a smooth weighting function $w_{\mathcal{C}^\infty} : \mathbb{R}_{\geq 0} \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{S} \to \mathbb{B}_{\mathcal{C}^\infty}$, which is 1 if $q_s = q$ and less than 1 otherwise. Let $\alpha \in \mathbb{B}_{\mathcal{C}^\infty}$, $K_1, K_2 \in \Pi_2$, and $\tilde{q} = q - q_s$. A possible smooth weighting function is the sum of several two-dimensional Gaussian functions, for instance

$$w_{\mathcal{C}^\infty}(t, q_s, q, \psi) := \alpha e^{-\tilde{q}^{\mathsf{T}} R(\psi) K_1 R^{\mathsf{T}}(\psi) \tilde{q}} + (1 - \alpha) e^{-\tilde{q}^{\mathsf{T}} R(\psi) K_2 R^{\mathsf{T}}(\psi) \tilde{q}}. \tag{9}$$

The purpose of having a combination of exponential functions is that the weighting surface can be shaped in such way that it approximates a non-smooth weighting surface while still having a nonzero image. Fig. 3 displays an example of such a weighting surface.

IV. The Path Planner

A. Covariance Dynamics of Object States in Optimization

The state estimation error covariance of the objects serve as a quantification of how uncertain the state variables are. We want to reduce this covariance by obtaining measurements of the objects, which is achieved by planning the path of the mobile sensor. To describe the future covariance response of the objects, we employ the continuous-time extended Kalman filter described in for instance [12]. For

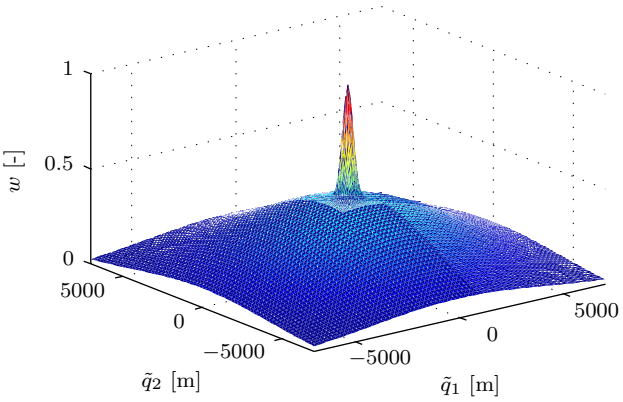

Fig. 3. A smooth weighting function. The axes represent $\tilde{q} = (q - q_s)$, which is the relative coordinate between an object and a sensor.

each object $m \in \mathbb{O}$ the system equations defined by (2) are linearized along the predicted trajectories of the objects with both measurement and process noise set to zero. Let $\hat{x}_{o,m}(t)$ be the solution to the initial value problem (IVP) (2a)-(2b) with $x_{o,m}(t_0) = \hat{x}_{o,m,0}$ and $p_{o,m} = \hat{p}_{o,m}$, which are the best estimates of the state vector at time t_0 and the constant parameter vector, respectively. We get the following partial derivatives $\forall m \in \mathbb{O}$:

$$A_{o,m}(t) = \frac{\partial f_{o,m}}{\partial x_{o,m}}(t, \hat{x}_{o,m}(t), 0, \hat{p}_{o,m}), \tag{10a}$$

$$L_{o,m}(t) = \frac{\partial f_{o,m}}{\partial w_{o,m}}(t, \hat{x}_{o,m}(t), 0, \hat{p}_{o,m}), \tag{10b}$$

$$C_{o,m}(t) = \frac{\partial h_{o,m}}{\partial x_{o,m}}(t, \hat{x}_{o,m}(t), 0), \tag{10c}$$

$$M_{o,m}(t) = \frac{\partial h_{o,m}}{\partial v_{o,m}}(t, \hat{x}_{o,m}(t), 0). \tag{10d}$$

In order to incorporate the effect of the sensor's trajectory in the covariance dynamics, we first need to employ a measurement model for the mobile sensor based on the smooth weighting function discussed in Section III-B. Specifically, we choose smooth diagonal matrix functions $W_{o,m} : \mathbb{R}_{\geq 0} \times \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{S} \to \mathbb{B}_{\mathcal{C}^\infty}^{n_{y_{o,m}} \times n_{y_{o,m}}}$, where the diagonal elements contain, possibly different, smooth weighting functions. We get $\forall m \in \mathbb{O}$

$$W_{o,m}(t, q_s, q_{o,m}, \psi) = \mathrm{diag}(w_{\mathcal{C}^\infty,1,m}, \cdots, w_{\mathcal{C}^\infty,n_{y_{o,m}},m}). \tag{11}$$

The motivation for using smooth weighting functions is that the NLP solver needs smoothness and curvature to find a solution of the optimization problem. When the mobile sensor is away from an object, it will still slightly affect the covariance dynamics of the object. This facilitates an effect of the measurements on the objects' estimation error covariance, and the mobile sensor will have an incentive to move closer to an object.

We use the matrix functions to weight the linearized measurement operators $C_{o,m}(t)$ along the predicted trajectories of all objects $m \in \mathbb{O}$ as

$$C_{o,m,w}(t) = W_{o,m}(t, q_s, \hat{q}_{o,m}, \psi) C_{o,m}(t). \tag{12}$$

Eventually, though, we want these weighted measurement operator to almost vanish far away from their respective objects, so that if the covariance is huge, the term will still be small. For the sensor to still care about distant objects, we propose to manipulate the process noise matrices $Q_{o,m}(t)$. More specifically, we use a non-vanishing weighting function to reduce the process noise when the sensor is close to an object. In this way, the mobile sensor's movement will always affect the covariance of the objects. Define for all objects $m \in \mathbb{O}$

$$Q_{o,m,w}(t, q_s, \hat{q}_{o,m}, \psi) = Q_{0,m}(t)(1 - w_{\mathcal{C}^\infty}(q_s, q_{o,m}, \psi)). \tag{13}$$

Define the shorthand expressions

$$\overline{Q}_{o,m,w}(t, q_s, q_{o,m}, \psi) = L_{o,m}(t) Q_{o,m,w}(t, q_s, q_{o,m}, \psi) L_{o,m}^{\mathsf{T}}(t), \tag{14}$$

$$\overline{R}_{o,m}(t) = M_{o,m}(t) R_{o,m}(t) M_{o,m}^{\mathsf{T}}(t). \tag{15}$$

Now, we are in position to define the covariance dynamics intended for the optimization problem. For simplicity, we omit the arguments of the expressions, and so for each $m \in \mathbb{O}$ we get the differential Riccati equation of the extended Kalman filter:

$$\dot{P}_{o,m}(t) = A_{o,m} P_{o,m} + P_{o,m} A_{o,m}^{\mathsf{T}} + \overline{Q}_{o,m,w} - P_{o,m} C_{o,m,w}^{\mathsf{T}} \overline{R}_{o,m}^{-1} C_{o,m,w} P_{o,m}, \tag{16a}$$

$$P_{o,m}(t_0) = P_{o,m,0}. \tag{16b}$$

We define a shorthand notation for each $m \in \mathbb{O}$ of (16a) as

$$\dot{P}_{o,m}(t) = f_{P,m}(t, q_s(t), \psi(t), P_{o,m}(t)). \tag{17}$$

B. Dynamic Optimization Problem

The path-planning problem is formulated as a *Bolza type* [13] dynamic optimization problem with a receding prediction horizon $t \in [t_0, t_f]$. Let the control input $u_s(t) \in \mathbb{R}^{n_{u_s}}$ form the decision variables of the optimization problem.

Define the Lagrange term as

$$\Phi_L(t, u_s) = \int_{t_0}^{t_f} \sum_{m \in \mathbb{O}} \mathrm{tr_w}(P_{o,m}(t), v_{L,m}) + \frac{du_s}{dt}^{\mathsf{T}} R_s \frac{du_s}{dt}\, dt, \tag{18}$$

where for each $m \in \mathbb{O}$ we have $v_{L,m} \in \mathbb{R}^{n_{x_{o,m}}}$ and $R_s \in \Pi_{n_{u_s}}$, which are design variables. The Mayer term is

$$\Phi_M(t_f) = \sum_{m \in \mathbb{O}} \mathrm{tr_w}(P_{o,m}(t_f), v_{M,m}), \tag{19}$$

where for each $m \in \mathbb{O}$ we have $v_{M,m} \in \mathbb{R}^{n_{x_{o,m}}}$, which also is a design vector.

The optimal control problem (OCP) is constrained by the mobile sensor system (1) and the objects' covariance

dynamics (16), and can be stated as

$$\min_{u_s} \quad \Phi_L(t, u_s) + \Phi_M(t_f) \tag{20a}$$
$$\text{s.t.} \quad \dot{x}_s(t) = f_s(t, x_s(t), u_s(t), p_s), \tag{20b}$$
$$x_s(t_0) = \hat{x}_{s,0}, \quad p_s = \hat{p}_s, \tag{20c}$$
$$x_s(t) \in \mathbb{X}_s, \quad u_s(t) \in \mathbb{U}_s, \quad \forall m \in \mathbb{O}, \tag{20d}$$
$$\dot{P}_{o,m}(t) = f_{P,m}(t, q_s(t), \psi(t), P_{o,m}(t)), \tag{20e}$$
$$P_{o,m}(t_0) = P_{o,m,0}, \tag{20f}$$
$$P_{o,m}(t) \in \Pi_{n_{x_{o,m}}}, \tag{20g}$$

where positive definiteness of $P_{o,m}$ follows from $x_{o,m}$ being a non-constant random variable. The solution to (20) provides us for each $t \in [t_0, t_f]$ with an input vector $u_s^\star(t) \in \mathbb{U}_s$. Given the variables $u_s^\star$, $\hat{x}_{s,0}$, and $\hat{p}_s$ we can solve the IVP formed by (1a)-(1b) over the optimization horizon. This results in an optimal mobile sensor state trajectory, denoted

$$x_s^\star(t) \in \mathbb{X}_s,\ t \in [t_0, t_f]. \tag{21}$$

Equation (21) serves as guidance input to a path maneuvering controller for the mobile sensor.

V. The Observer and Closing the Loop

The optimal sensor trajectory (21) found by solving (20) has an expiration date, since there are inaccuracies in the problem formulation. Specifically, the predicted trajectories of the objects may drift away from the true trajectories, the ambient conditions change, and the limited horizon of the OCP force us to periodically set up and solve new dynamic optimization problems.

In our system there is a decomposition of time scales, where the control of the mobile sensor operates at a higher frequency (milliseconds) than the optimization-based path planner (seconds to minutes). Thus, the Path Planner provides the mobile sensor with desired operating conditions, that is, a path maneuvering task, which is updated periodically. Between each update, the *Observer* continuously performs processing of the incoming plant data and other information. The model descriptions used in the Observer and the Path Planner need not be of the same fidelity. Moreover, the filter used in the former component can be more sophisticated than the extended Kalman filter used in the dynamic optimization problem.

Let $k \in \mathbb{N}_1$ denote the iteration number of the Path Planner block. Suppose the next optimization interval is denoted by $T_{k+1} := [t_{0,k+1}, t_{f,k+1}]$. A requirement of the Observer is that it must provide the following model parameters $\forall m \in \mathbb{O}$:

$$x_s(t_{0,k+1}) = \hat{x}_{s,0}, \tag{22a}$$
$$p_s = \hat{p}_{s,k+1}, \tag{22b}$$
$$x_{o,m}(t_{0,k+1}) = \hat{x}_{o,m,0}, \tag{22c}$$
$$p_{o,m} = \hat{p}_{o,m,k+1}, \tag{22d}$$
$$P_{o,m}(t_{0,k+1}) = P_{o,m,k+1}, \tag{22e}$$
$$Q_{o,m}(t), R_{o,m}(t) \text{ given } \forall t \in T_{k+1}. \tag{22f}$$

Typically, we want to employ the new operating conditions $x_{s,k+1}^\star(t)$, defined for all $t \in T_{k+1}$, starting with $t_{0,k+1}$.

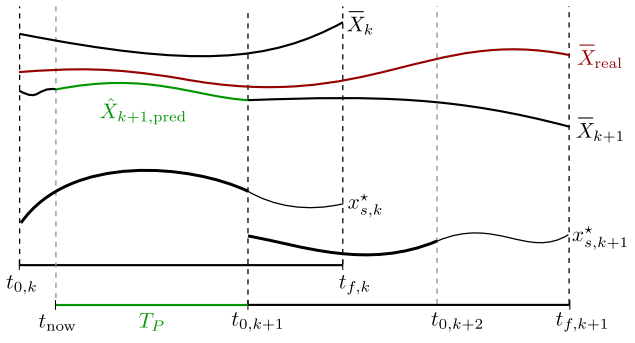

Fig. 4. The desired sensor trajectory $x_s^\star$ is updated at $t_{0,k+1}$.

Since the optimization algorithm has a nonzero computation time, the model parameters provided by the Observer are predicted values. It is important to choose $t_{0,k+1}$ sufficiently far into the future in order to ensure that $x_{s,k+1}^\star(t)$ is readily available and properly transferred to the mobile sensor in a timely manner.

To illustrate the mechanics of a re-optimization we consider a one-dimensional time-varying variable $X_{\text{real}}(t)$ that represents our plant model (UAV and objects). Let $\overline{X}_k(t|t_{0,k})$ and $\overline{X}_{k+1}(t|t_{0,k+1})$ denote the predicted trajectories for the plant at iteration k and $k+1$, respectively. Let $\hat{X}_{k+1,\text{pred}}(t|t_{\text{now}})$ denote the currently best future prediction of the plant model, which is provided by the Observer block. In a similar way we define the output trajectories at iteration k and $k+1$ as $x_{s,k}^\star(t)$ and $x_{s,k+1}^\star(t)$, which is obtained by solving (20).

Suppose t_{now} is the time instant we choose to use the current information and solve the OCP (20) for iteration $k+1$. Denote $T_P := t_{0,k+1} - t_{\text{now}}$ as the prediction interval. The time limit for an optimization is thus T_P. Fig. 4 illustrates, conceptually, what is going on.

VI. Numerical Example

A. Setup

We will investigate the case of monitoring three moving objects using an aerial mobile sensor. We assume that the initial positions of the objects are estimated from some coarse resolution intelligence (for instance satellite images). In the following, we state specific equations for the different parts of the system, with associated numerical values given in Table I.

1) Mobile Sensor Dynamics: For the mobile sensor we use a planar kinematic model (constant altitude) with the scalar bank angle $u_s(t)$ as control input. Let N_s and E_s denote the North and East position of the sensor and $q_s := \text{col}(N_s, E_s)$. The orientation of a body-fixed reference frame relative to {NED} is $\psi(t) \in \mathbb{S}$. We use the dynamics

$$\dot{N}_s(t) = p_{s,U} \cos(\psi) - p_{s,N}, \tag{23a}$$
$$\dot{E}_s(t) = p_{s,U} \sin(\psi) - p_{s,E}, \tag{23b}$$
$$\dot{\psi}(t) = \frac{p_{s,g}}{p_{s,U}} \tan(u_s(t)), \tag{23c}$$

TABLE I
PARAMETERS FOR THE NUMERICAL EXAMPLE.

Parameter	Value	Unit
$\mathrm{col}(p_{s,N}, p_{s,E})$	$\mathrm{col}(0,0)$	m/s
$p_{s,U}$	25	m/s
$p_{s,g}$	9.81	m/s^2
$[N_{s,L}, N_{s,U}]$	$[0,3]$	km
$[E_{s,L}, E_{s,U}]$	$[0,3]$	km
$[u_{s,L}, u_{s,U}]$	$\left[-\frac{5\pi}{36}, \frac{5\pi}{36}\right]$	rad
p_1	$\mathrm{col}(2.5, 0.3)$	m/s
p_2	$\mathrm{col}(1.5, -0.2)$	m/s
p_3	$\mathrm{col}(0.0, -0.3)$	m/s
$Q_{o,m\in\mathbb{I}_3}$	$50 \cdot I_2$	
$R_{o,m\in\mathbb{I}_3}$	$100 \cdot I_2$	
For $Q_{o,m,w}$:		
α	0.8	-
K_1	$1.4 \times 10^{-8} \cdot I_2$	m^{-1}
K_2	$5.2 \times 10^{-7} \cdot I_2$	m^{-1}
For $C_{o,m,w}$:		
α	1	-
K_1	$3.3 \times 10^{-5} \cdot I_2$	m^{-1}
K_2	$-$	m^{-1}
T_P	60	s
$t_{f,k} - t_{0,k}$	10	s
$v_{L,m\in\mathbb{I}_3}$	$\mathrm{col}(10, 10)$	s^2/m^2
R_s	100	rad^{-1}
$v_{M,m\in\mathbb{I}_3}$	$\mathrm{col}(50, 50)$	s^2/m^2
$Q_s(t)$	$\mathrm{diag}(1, 1, 0.1)$	
$R_{s,c}(t)$	$\mathrm{diag}(10, 10, 1)$	
Δ_x	$\mathrm{col}(300, 300)$	m

where $p_{s,N}$ and $p_{s,E}$ are wind speeds in North and East direction, $p_{s,U}$ is the surge for zero wind speeds, and $p_{s,g}$ is the standard gravity. We further constrain the planar coordinates to remain in the bounding box formed by the inequalities

$$N_{s,L} \leq N_s(t) \leq N_{s,U}, \tag{23d}$$

$$E_{s,L} \leq E_s(t) \leq E_{s,U}, \tag{23e}$$

where $N_{s,L}, N_{s,U} \in \mathbb{R}$ and $E_{s,L}, E_{s,U} \in \mathbb{R}$ are lower and upper bounds in North and East direction, respectively. Finally, the bank angle is bounded by

$$u_{s,L} \leq u_s(t) \leq u_{s,U}, \tag{23f}$$

with $u_{s,L}, u_{s,U} \in \mathbb{S}$.

2) Object Dynamics: A homogeneous set of objects are used, such that for all $m \in \mathbb{I}_3$ we define the state vector $x_{o,m} := \mathrm{col}(N_{o,m}, E_{o,m})$ and the equations

$$\dot{N}_{o,m}(t) = p_{N,m} + w_{N,m}(t), \tag{24a}$$

$$\dot{E}_{o,m}(t) = p_{E,m} + w_{E,m}(t), \tag{24b}$$

where $N_{o,m}$ and $E_{o,m}$ describe object m's North and East positions, $p_m := \mathrm{col}(p_{N,m}, p_{E,m})$ is the velocity vector, and $w_{o,m}(t) := \mathrm{col}(w_{N,m}, w_{E,m}) \sim (0, Q_{o,m}(t)$ is process noise.

We assume that when the mobile sensor is sufficiently close to an object, it is able to measure the position of the objects. For all $m \in \mathbb{I}_3$ we have

$$y_{o,m,w}(t) = W(t, q_s, x_{o,m}, \psi) \cdot I_2 \begin{bmatrix} N_{o,m} \\ E_{o,m} \end{bmatrix} + v_{o,m}(t), \tag{24c}$$

where $v_{o,m}(t) \sim (0, R_{o,m}(t))$ is a measurement noise vector for object m, and the weighting operator $W(t, q_s, x_{o,m}, \psi)$ depends on whether it is used in the Path Planner or the Observer, see definitions below.

3) Path Planner: The measurement model is the same for all the objects and employs the smooth weighting function (9), such that for all $m \in \mathbb{I}_3$ the weighting operator is

$$W_{o,m}(t, q_s, x_{o,m}, \psi) = w_{\mathcal{C}^\infty} \cdot I_2, \tag{25}$$

where I_2 is the two-dimensional identity matrix. Other relevant parameters are given in Table I.

4) Observer: We use the same vehicle and object models as in the Path Planner, with the exception that we in addition treat the sensor states as random variables (with process noise $w_s(t) \sim (0, Q_s(t))$). Moreover, we employ a hybrid extended Kalman filter [12] to process measurements and perform state estimation.

a) Mobile Sensor Estimator: We assume that the wind components $p_{s,N}$ and $p_{s,E}$ of the sensor dynamics are known parameters. Define the state estimation vector as $\hat{x}_s := \mathrm{col}(\hat{q}_s, \hat{\psi})$. We assume that all the state variables are measured linearly with $C_s := I_3$ and measurement noise $v_s(t) \sim (0, R_{s,c}(t))$. The continuous-time state and covariance dynamics of this state vector are

$$\dot{\hat{x}}_s(t) = \begin{bmatrix} p_{s,U}\cos(\hat{\psi}) - \hat{p}_{s,N} \\ p_{s,U}\sin(\hat{\psi}) - \hat{p}_{s,E} \\ \frac{p_{s,g}}{p_{s,U}}\tan(u_s(t)) \end{bmatrix} =: f_s(t, \hat{x}_s, w_s = 0), \tag{26a}$$

$$\dot{P}_s(t) = A_s P_s + P_s A_s^\mathsf{T} + Q_s - P_s C_s^\mathsf{T} R_{s,c}^{-1} C_s P_s, \tag{26b}$$

where $A_s(t)$ is the partial derivative along the currently best state estimate:

$$A_s := \frac{\partial f_s}{\partial x_s}(t, \hat{x}_s, 0). \tag{26c}$$

b) Object Estimator: We assume that the velocity vectors of the objects are constant and known. For all $m \in \mathbb{I}_3$ the object state estimation vector contains the position of the object: $\hat{x}_m := \mathrm{col}(\hat{N}_{o,m}, \hat{E}_{o,m})$. The continuous-time state and covariance dynamics of the system are $\forall m \in \mathbb{I}_3$

$$\dot{\hat{x}}_m(t) = \begin{bmatrix} p_{N,m} \\ p_{E,m} \end{bmatrix} \tag{27a}$$

$$\dot{P}_m(t) = Q_{o,m} - P_m C_{o,m,w}^\mathsf{T} R_{o,m}^{-1} C_{o,m,w} P_m, \tag{27b}$$

where the weighting operator used here is the same for all the objects. Unlike previously, we employ the more realistic non-smooth weighting function defined in (8), such that for all $m \in \mathbb{I}_3$ the measurement operator is

$$C_{o,m,w}(t) = W_{o,m}(t, q_s, x_{o,m}, \psi) C_{o,m}(t) = w_{\mathcal{C}^{-1}} \cdot I_2. \tag{27c}$$

Notice also that the process spectral densities $Q_{o,m}$ are not weighted as was the case for the path planner. Between

measurements we set $R_{o,m} = \infty \cdot I_2$. When a measurement arrives, a correction step identical to the discrete-time (Extended) Kalman filter is executed, with discrete-time measurement covariance chosen as $R_d = R_c F_s$, where R_c is the spectral density and $F_s = 1$ Hz is the sampling frequency of the measurements. We assume that the sampling frequency is the same for both the vehicle measurements and the measurements of the objects. For details on the discrete-time Kalman filter, consult for instance [12].

B. Implementation

In order to efficiently solve the optimal control problem of (20), a numerical method is required. We choose a direct transcription approach where both the state and control variables are discretized, resulting in a finite-dimensional nonlinear programming (NLP) problem. The simultaneous collocation of finite elements is used to obtain Lagrange interpolation polynomial descriptions of both the state and control input. The control input is piecewise constant, whereas the states are described using K-point Radau collocation, for details consult [13].

The resulting large-scale NLP formulation benefits from being sparse and having structure. These properties can be exploited using an efficient NLP solver. We formulate the problem in the symbolic framework CasADi [14], which provides the necessary derivative information required by both the extended Kalman filter and the NLP solver. The CasADi framework contains an interface for the primal-dual interior-point NLP solver IPOPT [15]. IPOPT is compiled with OpenBLAS [16] and the linear algebra sparse direct solver MA27 [17].

When solving initial value problems, for instance when finding predicted trajectories of the objects, we use the ODE solver CVODES, which is part of the SUNDIALS suite [18], accessible through an interface in CasADi.

C. Results

The simulation was run for a total of 15 iterations with the initial conditions $\hat{x}_s = \mathrm{col}(1500, 1500, 0)$ and $P_s = 100 \cdot I_3$ for the sensor, and for the objects we let $\hat{x}_1 = \mathrm{col}(300, 200)$, $\hat{x}_2 = \mathrm{col}(600, 2400)$, and $\hat{x}_3 = \mathrm{col}(2400, 1500)$, each with initial covariance $P_{m\in\mathbb{I}_3} = 1000 \cdot I_2$. A 2-point Radau collocation was used for the state variables, with a total of 60 collocation elements at each iteration. The control input was set to be piecewise constant with 20 collocation elements. The resulting optimization problems has 3800 variables. The solution time for each iteration was on average 5.6 s$\pm$1.8 s on a standard laptop computer. Since the realized interval of each iteration is 60 s, the current setup is applicable for real-time application. The sensor and objects start at the different markers shown in Fig. 5. The figure displays the paths traveled during the simulation. It can be seen that the mobile sensor moves between the different objects and thereby keeps the state estimation error covariances of the objects at bay. A measurement of an object is identified by a big reduction in the trace of the estimation error covariance, as shown in Fig. 6. Finally, from Fig. 7 we observe that the control input never exceeds ± 0.436 rad, which is the bounds on the bank angle. Consequently, the planned paths meets the posed maneuverability constraints.

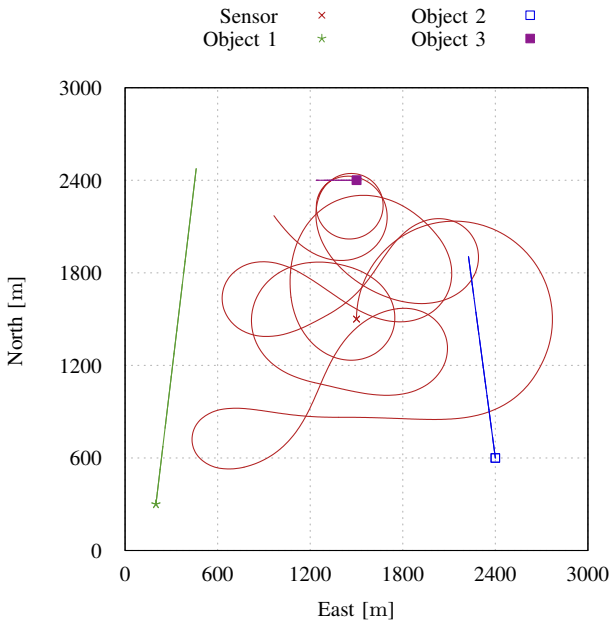

Fig. 5. The mobile sensor moves between the objects and obtains measurements.

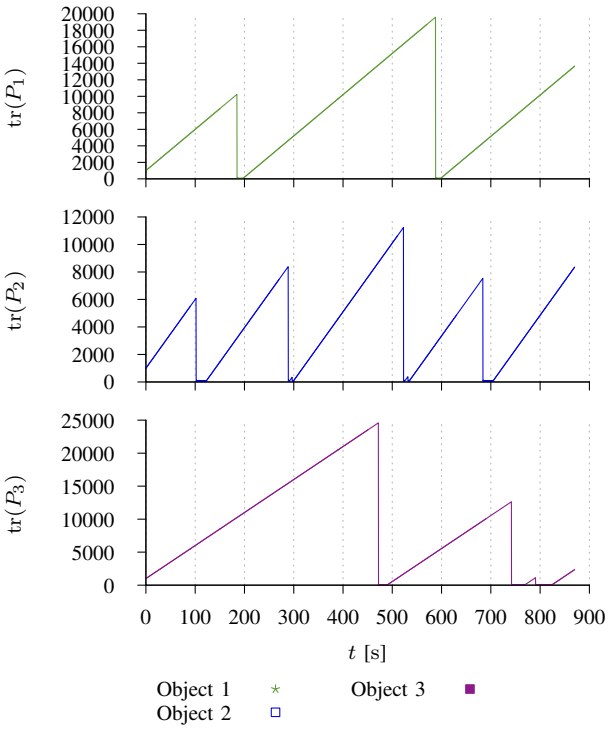

Fig. 6. The trace of the objects' covariance as a function of time. Object 1, 2, and 3 from the top, respectively.

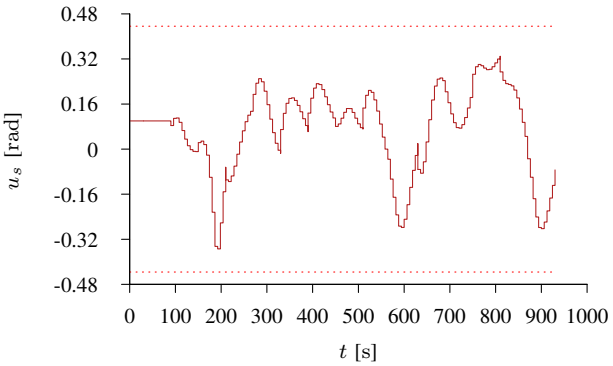

Fig. 7. The control input is always within the given upper and lower bounds of ± 0.436.

VII. Conclusion

An optimization-based mobile sensor path planner for monitoring of moving objects has been devised. The feasible planned paths promote the objective of minimizing the trace of the objects' estimation error covariance. Despite the occasional tendency of circulating a single object and the challenge of tuning, the approach may have merit with some additional work.

Future work include:

- Extending the approach to also estimate uncertain object parameters in the dynamic optimization problem.
- Analyzing convergence.
- Allow multiple mobile sensors to collaborate on achieving the objective.
- Investigate how the optimization problem scales with the number of differential states.
- Performing experiments to serve as proof of concept.

Acknowledgment

This work was supported by the Norwegian Research Council through the KMB: Arctic DP project 199567/I40 at the Norwegian University of Science and Technology.

References

[1] M. Netzband, W. L. Stefanov, and C. Redman, Eds., *Applied Remote Sensing for Urban Planning, Governance and Sustainability*. Springer, 2007.

[2] M. Moran, Y. Inoue, and E. Barnes, "Opportunities and limitations for image-based remote sensing in precision crop management," *Remote Sensing of Environment*, vol. 61, no. 3, pp. 319 – 346, 1997.

[3] D. Lubin and R. Massom, *Polar Remote Sensing, Volume I: Atmosphere and Oceans*. Springer Berlin Heidelberg, 2006.

[4] C. Elachi and J. J. van Zyl, *Introduction to the Physics and Techniques of Remote Sensing*, 2nd ed. Wiley-Interscience, 2006.

[5] J. Haugen, L. Imsland, S. Løset, and R. Skjetne, "Ice Observer System for Ice Management Operations," in *ISOPE*, Maui, Hawaii, USA, 2011, pp. 1120–1127.

[6] K. Eik, "Review of Experiences within Ice and Iceberg Management," *Journal of Navigation*, vol. 61, no. 4, p. 557, Oct. 2008.

[7] C. Cai, "Information-Driven Sensor Path Planning and The Treasure Hunt Problem," Ph.D. dissertation, Duke University, 2008.

[8] H. Choi and J. P. How, "Continuous trajectory planning of mobile sensors for informative forecasting," *Automatica*, vol. 46, no. 8, pp. 1266–1275, 2010.

[9] B. Geiger, "Unmanned Aerial Vehicle Trajectory Planning with Direct Methods," Ph.D. dissertation, The Pennsylvania State University, 2009.

[10] A. T. Klesh, P. T. Kabamba, and A. R. Girard, "Optimal path planning for uncertain exploration," in *American Control Conference*, St. Louis, MO, USA, 2009, pp. 2421–2426.

[11] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford University Press, 1998.

[12] D. Simon, *Optimal State Estimation*. Hoboken, New Jersey: John Wiley & Sons Ltd, 2006.

[13] L. T. Biegler, *Nonlinear Programming: Concepts, Algorithms and Applications to Chemical Processes*. Society for Industrial and Applied Mathematics, 2010.

[14] J. Andersson, J. Åkesson, and M. Diehl, "CasADi - A symbolic package for automatic differentiation and optimal control," in *Recent Advances in Algorithmic Differentiation*, S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, Eds. Berlin, Germany: Springer, 2012.

[15] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[16] Z. Xianyi, W. Qian, and Z. Yunquan, "Model-driven Level 3 BLAS Performance Optimization on Loongson 3A Processor," in *2012 IEEE 18th International Conference on Parallel and Distributed Systems (ICPADS)*, Singapore, 2012.

[17] HSL, "A collection of Fortran codes for large scale scientific computation." 2011. [Online]. Available: http://www.hsl.rl.ac.uk

[18] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS : Suite of Nonlinear and Differential / Algebraic Equation Solvers," *ACM Transactions on Mathematical Software*, vol. 31, no. 3, pp. 363–396, 2005.