This is a preprint of a paper that appeared in
Procedia Computer Science (ICCS 2016), Vol. 80, Elsevier, 2016

Decentralized Dynamic Data-Driven Monitoring of Atmospheric Dispersion Processes

Tobias Ritter[1,2]*, Juliane Euler[2], Stefan Ulbrich[1,3], and Oskar von Stryk[1,2]

[1] Graduate School Computational Engineering
[2] Department of Computer Science
{ritter,euler,stryk}@sim.tu-darmstadt.de
[3] Department of Mathematics
ulbrich@mathematik.tu-darmstadt.de
Technische Universität Darmstadt, Germany

Abstract

Online state and parameter estimation of atmospheric dispersion processes using multiple mobile sensor platforms is a prominent example of Dynamic Data-Driven Application Systems (DDDAS). Based on repeated predictions of a partial differential equation (PDE) model and measurements of the sensor network, estimates are updated and sensor trajectories are adapted to obtain more informative measurements. While most of the monitoring strategies require a central supercomputer, a novel decentralized plume monitoring approach is proposed in this paper. It combines the benefits of distributed approaches like scalability and robustness with the prediction ability of PDE process models. The strategy comprises model order reduction to keep calculations computationally tractable and a joint Kalman Filter with Covariance Intersection for incorporating measurements and propagating state information in the sensor network. Moreover, a cooperative vehicle controller is employed to guide the sensor vehicles to dynamically updated target locations that are based on the current estimated error variance.

Keywords: DDDAS, Reduced Order Models, Decentralized State Estimation, Cooperative Vehicle Controller

1 Introduction

Environmental monitoring of pollutant dispersion due to chemical leaks is an important task in disaster response. It is essential to repeatedly estimate characteristic state variables and important process parameters on-line in order to be aware of the current hazardous situation. The task is usually solved using a sensor network from which measurements are incorporated into a suitable model of the underlying process. As static sensor networks are very inflexible

*The work of Tobias Ritter is supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering at Technische Universität Darmstadt

in a dynamically changing environment, the use of robotic systems is increasingly considered in this context [8, 15]. Vehicles are equipped with sensors and computing units and measure the process state at different locations of the problem domain. Instead of moving along fixed trajectories, the sensor vehicles are intended to adapt their movement on-line, depending on the current estimate and its uncertainty. This leads to the concept of Dynamic Data Driven Application Systems (DDDAS) [3, 5]. The collected data is directly integrated into the system model and the vehicle movement and further measurement process is based thereupon. In this way, a changing environment can be observed by a relatively low number of self-adapting robots moving to positions where measurements are highly profitable.

A number of DDDAS approaches in the context of monitoring atmospheric dispersion have been proposed in recent time. While on the one hand complex optimal control problems subject to process/vehicle dynamics and estimation statistics are considered for finding optimal trajectories [22, 24], more efficient suboptimal methods splitting the estimation part from the trajectory planning part [6, 19] are described on the other hand. However, all these approaches rely on a centralized structure of the sensor network. All measurements are sent to a central supercomputer, which assembles the current estimate, calculates control inputs for the vehicles and sends this information back to the sensor platforms. Obviously, such approaches are not robust due to a central point of failure and demand for vast communication ranges. Moreover, they lack scalability and modularity.

It is much more desirable to design a decentralized strategy where information is processed and exchanged locally without the utilization of a central computing node. Although several general concepts for decentralized estimation and control strategies have been developed in the last years [11, 16, 17], further research regarding amongst others the considered model types or coupling with vehicle control is required to apply these strategies. Decentralized approaches related to plume detection and information gathering were presented in [7, 23]. Due to the reduced on-board computation power and real-time requirements, all these approaches do not consider a partial differential equation (PDE) model of the underlying process since its solution requires a lot of time and high computational power. On the other hand, PDEs model physics and dynamic behavior of the process and are necessary to obtain a detailed and accurate forecast of the process. The use of PDE-models in the considered real-time context is, thus, highly advisable, but also highly challenging.

The present work is aimed to present a decentralized dynamic data-driven system for plume estimation that is based on the forecasts of a PDE-model. Pursuing, extending and distributing the methods presented in previous work [10, 20] on centralized systems, a DDDAS is proposed that is able to monitor pollutant dispersion in a distributed way with cooperating sensor vehicles and the aid of a PDE model. To meet the limited computational capacity, reduced order models are applied. Proper Orthogonal Decomposition of a snapshot set consisting of radial basis function as well as noise impulse response snapshots reduces the large dimension of the original problem drastically. Every sensor node uses and maintains such a reduced model and incorporates its own measurements into the model using a Kalman Filter. Furthermore, sensors share information (estimates and their error covariances) with other sensors located within the communication range using Covariance Intersection. Based on the error covariance of the obtained estimates, every node identifies new informative measurement locations. A Mixed Integer Linear Program is solved by each vehicle in a Model Predictive Control fashion to obtain its new control input so that the group of sensors cooperatively achieves reaching their target locations.

2 Process Model

2.1 Full Model

Atmospheric dispersion of a pollutant being emitted from unknown chemical leaks is considered in the present work. This process can be described by the advection-diffusion equation

$$\frac{\partial c}{\partial t} + \nabla \cdot (c\mathbf{v}) - \nabla \cdot (\mathbf{D}\nabla c) = s, \qquad c(\mathbf{x}, 0) = c^0(\mathbf{x}) \tag{1}$$

with nabla operator ∇, space vector $\mathbf{x} \in \Omega$ and time $t \in [0, t_f]$. While the time-independent velocity field $\mathbf{v}(\mathbf{x})$ and the diffusion matrix $\mathbf{D}(\mathbf{x})$ are assumed to be known, the concentration $c(\mathbf{x}, t)$ together with its initial condition $c^0(\mathbf{x})$ is unknown and it is the aim of the present approach to provide a repeated, precise estimate of this entity. Moreover, the location, function and shape of the source $s(\mathbf{x}, t)$ is unknown and has to be estimated time after time. It is assumed that the source $s(\mathbf{x}, t)$ can be approximated by a linear combination of n_θ Radial Basis Functions (RBF) $q_j(\mathbf{x})$ with center $\mathbf{x_j}$, i.e.

$$s(\mathbf{x}, t) \approx s_q(\mathbf{x}, t) = \sum_{j=1}^{n_\theta} \theta_j(t) q_j(\mathbf{x}), \qquad q_j(\mathbf{x}) = e^{-a_j \|\mathbf{x} - \mathbf{x_j}\|^2} \tag{2}$$

with constant parameters $a_j > 0$ representing the width of the RBF and n_θ time-dependent parameters $\theta_j(t)$ to be repeatedly determined.

To solve (1), a suitable spatial discretization scheme (e.g. Finite Element Method) can be applied yielding the state vector $\mathbf{c} \in \mathbb{R}^n$, which contains the concentration values at the locations of the underlying grid. Using a time integration scheme, the time interval $[0, t_f]$ is divided into n_t time steps with time step size Δt and the state vector $\mathbf{c^{k+1}}$ at time t^{k+1} can be predicted using a model forecast based on the state vector $\mathbf{c^k}$ at time t^k. The forecast equation has the form

$$\mathbf{c^{k+1}} = \mathbf{M^k c^k} + \mathbf{S^k} \boldsymbol{\theta^k} \tag{3}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is the vector of parameters θ_j and the matrices $\mathbf{M} \in \mathbb{R}^{n \times n}$ and $\mathbf{S} \in \mathbb{R}^{n \times n_\theta}$ depend on time and space discretization methods.

In general, the forecast equation does not perfectly match the true evolution of the process and a model error $\boldsymbol{\eta^k}$ occurs in each step. Thus, the true evolution of the state vector is governed by

$$\mathbf{c_t^{k+1}} = \mathbf{M^k c_t^k} + \mathbf{S^k} \boldsymbol{\theta_t^k} + \boldsymbol{\eta^k}. \tag{4}$$

For this work, the model error is assumed to be unbiased and Gaussian with known error covariance matrix $\mathbf{Q}$.

2.2 Reduced Order Model

Typically, the dimension of the state vector $\mathbf{c}$ is very large. This means that state forecasts and especially the estimation algorithms of Section 3 become computationally intractable. Thus, it is essential for the considered application to reduce the problem dimension.

A popular method in Model Order Reduction [2] is Proper Orthogonal Decomposition (POD) [21]. The idea of POD is to first generate an $n \times n_t n_S$ matrix $\mathbf{C}$ of snapshots $\mathbf{c_i^k}$ with $\mathbf{C} = [\mathbf{c_1^1}, \mathbf{c_1^2}, ..., \mathbf{c_{n_S}^{n_t}}]$ by n_S numerical solutions of the full system Afterwards, an orthonormal basis $\{\boldsymbol{\phi_i}\}_{i=1}^{\ell}$ with $\ell \ll n$ that represents the column vectors $\mathbf{C}_j$ in an optimal way is created

$$\min_{\{\boldsymbol{\phi_i}\}_{i=1}^{\ell}} \sum_{j=1}^{n_t n_S} \left\| \mathbf{C}_j - \sum_{i=1}^{\ell} \langle \mathbf{C}_j, \boldsymbol{\phi_i} \rangle \boldsymbol{\phi_i} \right\|^2 \tag{5}$$

$$\text{s.t.} \quad \langle \boldsymbol{\phi_i}, \boldsymbol{\phi_j} \rangle = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \neq j \end{cases} .$$

It turns out that the solution of the optimization problem (5) can be found by solving the eigenvalue problem

$$\mathbf{C}\mathbf{C}^T \boldsymbol{\phi_i} = \lambda_i \boldsymbol{\phi_i} \tag{6}$$

and choosing the ℓ leading eigenvectors as POD basis. Following this, the high-dimensional model (3) can be projected onto the POD space yielding the reduced model

$$\tilde{\mathbf{c}}^{\mathbf{k+1}} = \tilde{\mathbf{M}}^{\mathbf{k}} \tilde{\mathbf{c}}^{\mathbf{k}} + \tilde{\mathbf{S}}^{\mathbf{k}} \boldsymbol{\theta^k} \tag{7}$$

with $\tilde{\mathbf{c}} = \mathbf{\Phi}^T \mathbf{c}$, $\tilde{\mathbf{M}} = \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi}$ and $\tilde{\mathbf{S}} = \mathbf{\Phi}^T \mathbf{S}$.

A crucial point is the choice of the snapshot set so that the true system can be represented precisely. System (3) can be interpreted as a linear time-invariant system with inputs $\mathbf{S_j}\theta_j$. According to Duhamel's principle it is possible to express the solution $\mathbf{c}$ as the sum of the convolutions of the parameter θ_j with the impulse response $\mathbf{w_j}$ corresponding to an excitation $s(\mathbf{x}, t) = \mathbf{S_j}\delta_0(t)$:

$$\mathbf{c}(t) = \sum_{j=1}^{n_\theta} \int_0^t \theta_j(\tau) \mathbf{w_j}(t - \tau) d\tau. \tag{8}$$

With this motivation, the impulse response $\mathbf{w_j}$ of system (3) for all RBF $q_j(\mathbf{x})$ is computed and added to the snapshot set. It is further assumed that the initial condition c^0 can also be approximated by the RBF $q_j(\mathbf{x})$. Thus, a separate snapshot generation for the initial condition is not necessary.

Furthermore, the snapshot set is enriched with model noise impulse response snapshots so that the reduced model is able to adapt to the true process evolution. Therefore, the model error is reformulated as

$$\boldsymbol{\eta} = \sum_{j=1}^{n} \sigma_j \sqrt{\lambda_j} \boldsymbol{\varphi_j} \tag{9}$$

where λ_j and $\boldsymbol{\varphi_j}$ are eigenvalues and eigenvectors of the error covariance matrix $\mathbf{Q}$ and σ_j is a normally distributed random variable. An approximation to the noise term is obtained if the summation in (9) is only made over the most dominant eigenvectors. If this approximation is plugged into (3), the noise can be interpreted as random input and Duhamel's principle can be applied again. Thus, the snapshot set is enriched with the impulse response $\mathbf{w}_{\mathbf{j},\boldsymbol{\eta}}$ for all considered noise eigenvectors $\boldsymbol{\varphi_j}$.

3 Decentralized DDDAS

3.1 Sensor Model

A network of n_O mobile sensors is used to complement the model information with measurements of the current process state. It is assumed that the dynamics of each sensor can be described by an ordinary differential equation

$$\dot{\mathbf{r}}_{\mathbf{j}}(t) = f_j(\mathbf{r}_{\mathbf{j}}(t), \mathbf{u}_{\mathbf{j}}(t), t) \tag{10}$$

where $\mathbf{r}_{\mathbf{j}}$ comprises the sensor position $\mathbf{x}_{\mathbf{j}} \in \Omega$ and velocity $\mathbf{v}_{\mathbf{j}}$ obeying the maximum velocity $v_{\max}$ and maximum acceleration $u_{\max}$ of sensor j. The movement of the sensor can be controlled by the control vector $\mathbf{u}_{\mathbf{j}}$. Every sensor unit is also equipped with a computational device. There is no central node and every unit maintains its own model state, processes local information and communicates with the neighbors $N_j(t) = \{m \in \{1, ..., n_O\} || \mathbf{x}_{\mathbf{j}}(t) - \mathbf{x}_{\mathbf{m}}(t)| < d_j\}$, which reside in the limited communication range d_j. In every step, sensor j takes an observation $\mathbf{z}_{\mathbf{j}} \in \mathbb{R}^{n_z}$ of the form

$$\mathbf{z}_{\mathbf{j}}^{\mathbf{k}} = \mathbf{H}_{\mathbf{j}}^{\mathbf{k}} \mathbf{c}_{\mathbf{t}}^{\mathbf{k}} + \boldsymbol{\epsilon}_{\boldsymbol{j}}^{\boldsymbol{k}}. \tag{11}$$

The sensor model matrix $\mathbf{H} \in \mathbb{R}^{n_z \times n}$ maps the model state onto the observation space and is dependent on the current sensor position $\mathbf{x}_{\mathbf{j}}$. $\boldsymbol{\epsilon}_{\boldsymbol{j}}^{\boldsymbol{k}}$ represents the observation error and is assumed to be unbiased and Gaussian with covariance $\mathbf{R}_{\mathbf{j}}$.

3.2 State and Parameter Estimation

To estimate the current state vector and the process parameters, decentralized data fusion has to take place. Measurements as well as estimates of neighboring sensor nodes are taken into account to improve the individual estimate. The challenge in the present application consists of the limited communication range that leads to different estimates of the sensor nodes. Combining these estimates is difficult due to unknown cross correlations. Furthermore, the estimates should be combined based on their uncertainty as sensor nodes can be seen as experts for specific regions of the domain. This prohibits standard techniques as decentralized information filters [9] or consensus strategies [18]. Instead, Covariance Intersection (CI) [12] or CI-based diffusion strategies [4] have to be used.

In the present application, the estimation process is accomplished in three steps: First, the individual measurements are integrated into the local model. Second, estimates and error covariance matrices are exchanged with all neighbors, and, third, an updated estimate is generated based on the estimates received from the neighbors. This procedure is repeated every time a measurement becomes available.

As state and parameters are to be estimated jointly, the problem formulation has to be rewritten. The parameter vector $\boldsymbol{\theta}$ is appended to the state vector $\tilde{\mathbf{c}}$ yielding the augmented state vector $\mathbf{p}$ together with its forecast equation

$$\mathbf{p}^{\mathbf{k}} = \begin{bmatrix} \tilde{\mathbf{c}}^{\mathbf{k}} \\ \boldsymbol{\theta}^{\boldsymbol{k}} \end{bmatrix} = \mathbf{W}\mathbf{p}^{\mathbf{k-1}} = \begin{bmatrix} \tilde{\mathbf{M}}^{\mathbf{k-1}} & \tilde{\mathbf{S}}^{\mathbf{k-1}} \\ \mathbf{0} & \mathbb{1} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{c}}^{\mathbf{k-1}} \\ \boldsymbol{\theta}^{\boldsymbol{k}-\mathbf{1}} \end{bmatrix}. \tag{12}$$

With this reformulation, the standard Kalman Filter (KF) [13] can be used in the first step to combine the local model forecast with the own meausurement of the sensor node. In the first part of the KF, the old estimate is forecasted to the current time t^k using (12). Furthermore, the current error covariance matrix $\mathbf{P}$, a measure for the uncertainty of the state estimate, is computed by propagating the old covariance matrix in time

$$\mathbf{P}_{\mathbf{j}}^{\mathbf{k},-} = \mathbf{W}_{\mathbf{j}}^{\mathbf{k-1}} \mathbf{P}_{\mathbf{j}}^{\mathbf{k-1},+} {\mathbf{W}_{\mathbf{j}}^{\mathbf{k-1}}}^T + \mathbf{Q}. \tag{13}$$

Next, measurements are combined with the prior estimate using the update equations

$$\mathbf{K}_\mathbf{j}^\mathbf{k} = \mathbf{P}_\mathbf{j}^{\mathbf{k},-} {\mathbf{H}_\mathbf{j}^\mathbf{k}}^T \left(\mathbf{H}_\mathbf{j}^\mathbf{k} \mathbf{P}_\mathbf{j}^{\mathbf{k},-} {\mathbf{H}_\mathbf{j}^\mathbf{k}}^T + \mathbf{R}_\mathbf{j} \right)^{-1} \tag{14}$$

$$\mathbf{p}_\mathbf{j}^{\mathbf{k},+} = \mathbf{p}_\mathbf{j}^{\mathbf{k},-} + \mathbf{K}_\mathbf{j}^\mathbf{k} \left(\mathbf{z}_\mathbf{j}^\mathbf{k} - \mathbf{H}_\mathbf{j}^\mathbf{k} \mathbf{p}_\mathbf{j}^{\mathbf{k},-} \right) \tag{15}$$

$$\mathbf{P}_\mathbf{j}^{\mathbf{k},+} = \left(\mathbb{1} - \mathbf{K}_\mathbf{j}^\mathbf{k} \mathbf{H}_\mathbf{j}^\mathbf{k} \right) \mathbf{P}_\mathbf{j}^{\mathbf{k},-} \tag{16}$$

where the Kalman gain $\mathbf{K}$ is chosen such that the error covariance matrix $\mathbf{P}_\mathbf{j}^{\mathbf{k},+}$ is minimized.

After having computed the local state estimate, the second step consists of exchanging the individual estimates and covariance matrices with the neighboring nodes. In order to save computational time, the local information vector $\mathbf{i}_\mathbf{j}$ and the information matrix $\mathbf{I}_\mathbf{j}$ with

$$\mathbf{i}_\mathbf{j}^\mathbf{k} = {\mathbf{P}_\mathbf{j}^{\mathbf{k},+}}^{-1} \mathbf{p}_\mathbf{j}^{\mathbf{k},+}, \qquad \mathbf{I}_\mathbf{j}^\mathbf{k} = {\mathbf{P}_\mathbf{j}^{\mathbf{k},+}}^{-1}. \tag{17}$$

are exchanged instead of $\mathbf{p}_\mathbf{j}^{\mathbf{k},+}$ and $\mathbf{P}_\mathbf{j}^{\mathbf{k},+}$.

In the third step, each sensor node computes its updated estimate by combining all the estimates received from neighboring nodes. Unfortunately, a standard Kalman method cannot be used for fusion of the estimates as the corresponding errors might be correlated. To avoid the propagation of cross-covariances, CI is used for fusion. This method uses a convex combination of state vectors and error covariances:

$$\mathbf{P}_\mathbf{j}^{\mathbf{k},++} = \left(\sum_{m \in N_j^k} \omega_{j,m}^k \mathbf{I}_\mathbf{m}^\mathbf{k} \right)^{-1}, \qquad \mathbf{p}_\mathbf{j}^{\mathbf{k},++} = \mathbf{P}_\mathbf{j}^{\mathbf{k},++} \left(\sum_{m \in N_j^k} \omega_{j,m}^k \mathbf{i}_\mathbf{m}^\mathbf{k} \right). \tag{18}$$

The parameters $\omega_{j,m}^k$ are found by minimizing the trace of the resulting covariance matrix

$$\begin{aligned} \min_{\omega_{j,m}^k} \quad & \operatorname{trace} \left(\left(\sum_{m \in N_j^k} \omega_{j,m}^k \mathbf{I}_\mathbf{m}^\mathbf{k} \right)^{-1} \right) \\ \text{s.t.} \quad & \omega_{j,m}^k \geq 0, \sum_{m \in N_j^k} \omega_{j,m}^k = 1. \end{aligned} \tag{19}$$

3.3 Dynamic Data-Driven Sensor Trajectories

To complete the DDDAS, it has to be ensured that the measurement process adapts to the gathered data. In particular, the trajectories of the sensor vehicles should be dynamically adjusted so that observations can be made at locations that promise informative measurements according to the current estimates. In this way, the approach is capable of adapting to changing environments and more accurate estimations can be expected. The trajectory planning is divided into two steps: First, new feasible measurement locations with a potentially high information gain are identified. Second, suitable control inputs are computed based on an optimization problem considering vehicle dynamics, target point coverage, collision avoidance and other restrictions. Every sensor performs both parts locally on its computing node cooperating implicitly with its colleagues to get the best possible outcome. Most aspects of this subsection follows previous work described in [10, 20].

3.3.1 Generation of Target Points

The sensor vehicles are intended to get to a number of target locations that promise high information gain and that are updated from time to time. Here, the number of target points corresponds to the number of sensors and is divided into two different types of target points.

The first type is designed for a general reduction of uncertainty and considers the error covariance matrix $\mathbf{P_j}$. Large entries indicate a high uncertainty so that it is advisable to take additional measurements at positions that reduce these entries. To obtain a correlation between covariance and spatial location the reduced error covariance matrix is transformed to its original size. The diagonal of the original matrix is then examined regarding its largest entry $\mathbf{P}_{ii}$. The location belonging to max$(\mathbf{P}_{ii})$ becomes the first target point. This can be seen as an approximation to the problem of reducing the trace of the error covariance matrix where the consideration of off-diagonal entries $\mathbf{P}_{ij}$ is avoided to keep the strategy computationally tractable. To avoid clusterization, fictional observations at the current target locations are integrated into the covariance matrix before a new target location is identified in the same way. This procedure is repeated iteratively until enough target points are found. Moreover, to account for the dynamic nature of the underlying process, the target points are moved with wind velocity in each time step.

The second type of target points considers the impacts of sources. As the source output is unknown and likely to change over time, a lot of model uncertainty is emitted at the source locations. Thus, it is of great interest to take measurements at that places. For this reason, target locations are chosen at positions belonging to the largest estimated source output.

3.3.2 Vehicle Control

Each sensor vehicle maintains its personal cooperative vehicle controller that is in charge of guiding the sensor vehicles to the specified target points. Trajectories are generated that provide optimal target allocation respecting the vehicles' physical characteristics and avoiding collisions. For this purpose, a discrete-time linear Mixed Logical Dynamical (MLD) formulation of the considered multi-vehicle system is set up [1] for each sensor vehicle [14]:

$$\min_{U_{n_h,j}} \; |\mathbf{F_j}\mathbf{\hat{p}_j^{n_h}}| + \sum_{k=0}^{n_h-1} |\mathbf{G_{1,j}}\mathbf{\hat{u}_j^i}| + |\mathbf{G_{2,j}}\boldsymbol{\delta}_j^i| + |\mathbf{G_{3,j}}\boldsymbol{\gamma}_j^i| + |\mathbf{G_{4,j}}\mathbf{\hat{p}_j^i}| \tag{20a}$$

$$\text{s.t.} \quad \mathbf{\hat{p}_j^{i+1}} = \mathbf{A_j}\mathbf{\hat{p}_j^i} + \mathbf{B_{1,j}}\mathbf{\hat{u}_j^i} + \mathbf{B_{2,j}}\boldsymbol{\delta}_j^i + \mathbf{B_{3,j}}\boldsymbol{\gamma}_j^i \tag{20b}$$

$$\mathbf{E_{2,j}}\boldsymbol{\delta}_j^i + \mathbf{E_{3,j}}\boldsymbol{\gamma}_j^i \leq \mathbf{E_{1,j}}\mathbf{\hat{u}_j^i} + \mathbf{E_{4,j}}\mathbf{\hat{p}_j^i} + \mathbf{E_{5,j}} \; . \tag{20c}$$

While the system state $\mathbf{\hat{p}_j} \in \mathbb{R}^{n_y}$ comprises the state of the vehicles in the neighborhood of node j and of the personal target points, $\mathbf{\hat{u}_j} \in \mathbb{R}^{n_u}$ represents the control inputs $\mathbf{u_j}$ of sensor j and of its neighbors. $\boldsymbol{\delta_j} \in \{0,1\}^{n_\delta}$ and $\boldsymbol{\gamma} \in \mathbb{R}^{n_\omega}$ are auxiliary binary and continuous vectors, respectively, and $\mathbf{A}$, $\mathbf{B_i}$, $\mathbf{E_i}$, $\mathbf{F}$, $\mathbf{G_i}$ are corresponding matrices. The problem is considered over a fixed time horizon $[0, n_h]$ which is shifted forward each step according to a Model Predictive Control (MPC) procedure. Thus, the prediction time step t^i corresponds to the global time step t^{k+i}. As solution of problem (20), the sequence $U_{n_h,j} := \{\mathbf{\hat{u}_j^i}\}_{i=0}^{n_h-1}$ of control inputs is obtained and $\mathbf{u_j^0}$ is applied.

The objective function (20a) contains terms representing the linearly approximated distances of the vehicles to the target points (continuous part $\boldsymbol{\gamma}$), the occupation of these positions (binary part $\boldsymbol{\delta}$) by another sensor and the required control effort. It is minimized subject to the linearized vehicle dynamics (20b), movement of target points, the allocation of measurement points (20c) and restrictions due to collision avoidance (20c).

Problem (20) can be transformed into a Mixed Integer Linear Program (MILP) at each time step of the MPC procedure and can be solved by standard optimization methods.

4 Results

The presented approach is validated with a two-dimensional test case. Pollutant concentration and sources on the domain $\Omega = [-1.2, 1.2] \times [-1, 1]$ are supposed to be estimated. The velocity field consists of a vanishing y-component and a y-dependent x-component $v_x = 0.005(1 - y^2)$. It is further assumed that the diffusion matrix $\mathbf{D}$ is homogeneous and constant.

Using the finite element method, a problem with over $n = 2200$ degrees of freedom is generated. Furthermore, 3×5 RBFs located equidistantly in the square $[-0.6, 0] \times [-0.3, 0.3]$ are used to represent the initial condition and the time-varying source. It is assumed that most of the model inaccuracy is introduced in the center of the domain leading to a Gaussian kernel function for the model error covariance matrix $\mathbf{Q}$.

In an offline phase, several simulation runs are performed using the full model and calculating the described impulse response solutions for RBFs and noise impulses. Applying POD on the obtained data, a reduced model with only 60 degrees of freedom capturing over 99.99% of the total energy $\sum_{i=1}^{n} \lambda_i$ of the snapshot set is obtained.

Three sensor vehicles modeled as point masses start at the positions $\mathbf{x_1} = (-0.6, -0.3)$, $\mathbf{x_2} = (-0.3, 0)$, $\mathbf{x_3} = (0, 0.3)$ to monitor the process. Their maximum speed as well as acceleration is 0.04, their communication range is only 0.4 and the MPC horizon amounts to 20 steps where a step size of $\Delta t = 2$ is used.

A twin-experiment is performed, i.e. a reference true state is simulated along with the estimated states using (9). The true initial condition is a linear combination of three RBFs and the true source is located at $(-0.6, -0.2)$ with

$$s(\mathbf{x}, t) = 0.02 \left(1 + 0.4 \cos\left(\frac{\pi t}{55}\right)\right) \exp\left(-20(x + 0.6)^2 - 20(y + 0.2)^2\right). \tag{21}$$

In this way, measurements can be obtained using (11) and the estimation error can be evaluated after the simulation run to assess the accuracy of the applied approaches.

A comparison between true and mean estimated state (the mean is taken over all available estimates at the considered time) shows that the estimate after 110 steps represents the shape of the real pollution cloud to a large extent (Figure 1). Merely some minor differences in the concentration amount can be noticed at some locations. This might be caused by a combination of model noise and the conservative estimates of the covariance intersection.

The error plot (Figure 2) also expresses a good performance of the proposed decentralized method. After some measurements, the error is decreased and kept constantly under a certain level even though more pollution and related noise is introduced into the system. In general, the error increases slightly over a longer time before decreasing rapidly in a short amount of time. This is caused by the fact that sensors often cannot communicate with each other over a longer time and exchange all the gathered information when they are in each other's communication range again.

To assess the results, another strategy with three mobile sensor vehicles starting at the same positions is applied. The sensor vehicles are patrolling on the lines $x = -0.6$, $x = -0.3$ and $x = 0$, an intelligent choice considering that the centers of the RBFs are located on the same lines. The error of this non-adaptive approach is also kept on a relatively constant level after some steps. However, compared to the proposed dynamic data-driven strategy, the error is nearly twice as high on average.

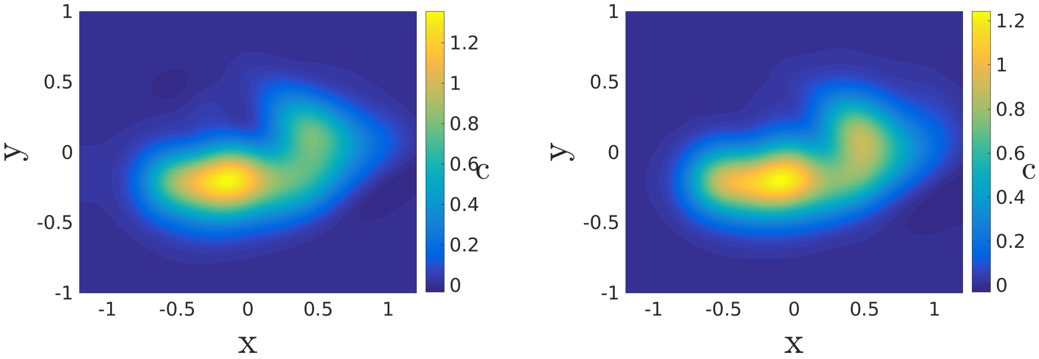

Figure 1: Average estimated (left) vs. true (right) concentration field after 110 steps

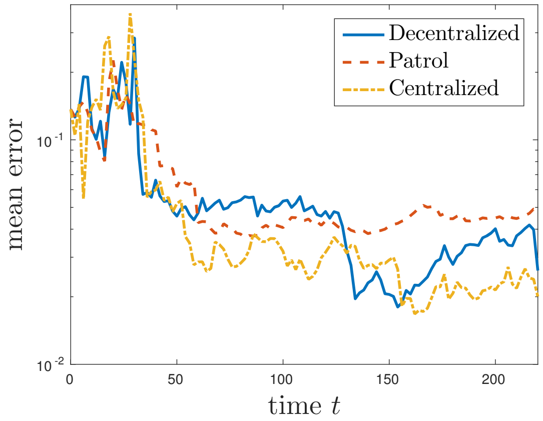

Figure 2: Error related to true solution over time

Another simulation run was made with a centralized variant of the proposed DDDAS approach to compare the behavior of the decentralized approach to its centralized counterpart. The centralized variant includes a pure KF approach for data fusion instead of Covariance Intersection and is basded on the assumption that the central node receives all measurements at the time they become available. Figure 2 shows that, on the one hand, the error of the centralized version is less than of the decentralized version, but, on the other hand, the difference is not very large so that the benefits of a decentralized approach might outweigh minor accuracy losses.

The source function at the source location $(-0.6, -0.2)$ is plotted over time in Figure 3. As expected, the centralized version performs best whereas the decentralized DDDAS outperforms the patrolling strategy. The evident phase delay of the decentralized version is mainly caused by the fact that it takes some time to spread information in the decentralized network.

Figure 4 shows an extract of the trajectory history and illustrates the functioning of the proposed DDDAS approach. While the two sensors on the right are in charge of reducing the general uncertainty (they are heading for the first type of target points), the sensor on the left approaches $(-0.6, -0.2)$, which is the estimated (and true) source location. As the sensors on the right are not located within the other's communication range, they do not share the same

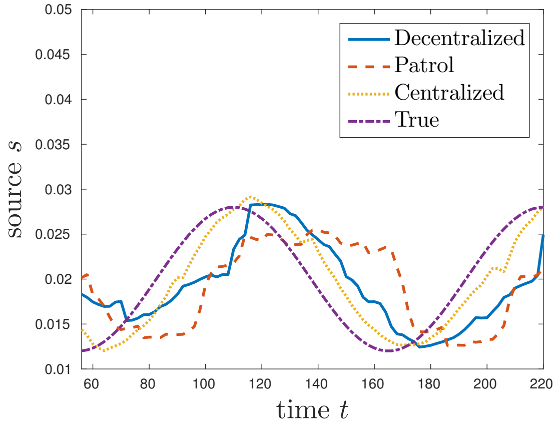

Figure 3: Source function at location of the true source over time

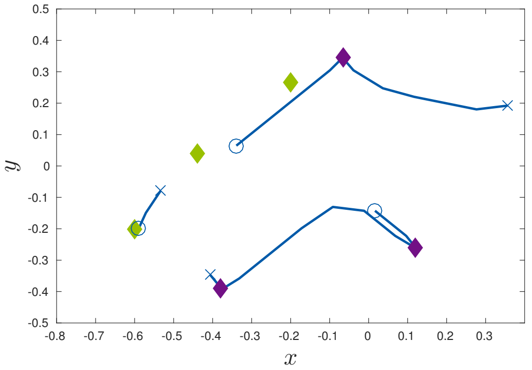

Figure 4: Extract of sensor trajectories. Sensors ○ move from starting point × to targets ◇. Purple ◇ mark visited target points, green ◇ mark currently active target points. For clarity, only a selection of target points is depicted.

target points and the lower sensor vehicle is heading for a target point located in a part recently observed by the other platform. However, before this target point will be reached, the two sensor vehicles will get into each others communication ranges and will exchange information so that the considered target point will probably become deprecated and targets will be recalculated. In this way, repeated information exchange between the sensors is accomplished implicitly.

5 Conclusion

A new decentralized DDDAS for environmental monitoring of pollutant dispersion was presented. The proposed system avoids a central computing node and backs on a decentralized mobile sensor network strategy to gain robustness, scalability, and flexibility. Each sensor node takes observations, processes local information, exchanges them with its neighbor, and computes its new control input to get to more informative measurement locations. Compared to other decentralized approaches, process characteristics available in the form of PDEs are considered in this approach. To account for the real-time requirements, the high-dimensional process model is transformed into a more tractable reduced order model via POD. Furthermore, the strategy applies a KF and Covariance Intersection for joint state and parameter estimation, a local target point generation scheme and a cooperative vehicle controller based on a MILP for dynamically attaining more information. A basic test case states that the new strategy provides better results than using patrolling robots. Accuracy losses compared to a fully centralized network are minor.

References

[1] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[2] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[3] E. Blasch, G. Seetharaman, and K. Reinhardt. Dynamic data driven applications system concept for information fusion. *Procedia Computer Science*, 18:1999–2007, 2013.

[4] F.S. Cattivelli and A.H. Sayed. Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084, 2010.

[5] F. Darema. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *International Conference on Computational Science (ICCS)*, pages 662–669. Springer, 2004.

[6] M.A. Demetriou and D. Ucinski. State estimation of spatially distributed processes using mobile sensing agents. In *Proceedings of the American Control Conference*, pages 1770–1776, 2011.

[7] M. Duckham, S. Nittel, and M. Worboys. Monitoring dynamic spatial fields using responsive geosensor networks. In *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*, pages 51–60, 2005.

[8] M. Dunbabin and L. Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics Automation Magazine*, 19(1):24–39, 2012.

[9] H.F. Durrant-Whyte, B.Y.S. Rao, and H. Hu. Toward a fully decentralized architecture for multisensor data fusion. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1331–1336 vol.2, 1990.

[10] J. Euler, T. Ritter, S. Ulbrich, and O. von Stryk. Centralized ensemble-based trajectory planning of cooperating sensors for estimating atmospheric dispersion processes. In *Dynamic Data-Driven Environmental Systems Science*, number 8964 in Lecture Notes in Computer Science, pages 322–333. Springer International Publishing, 2015.

[11] D. Hall, C.Y. Chong, J. Llinas, and M. Liggins II. *Distributed data fusion for network-centric operations.* CRC Press, 2012.

[12] S.J. Julier and J.K. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference*, volume 4, pages 2369–2373, 1997.

[13] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[14] J. Kuhn, C. Reinl, and O. von Stryk. Predictive control for multi-robot observation of multiple moving targets based on discrete-continuous linear models. In *Proceedings of the 18th IFAC World Congress*, pages 257–262, 2011.

[15] N.E. Leonard, D.A. Paley, F. Lekien, R. Sepulchre, D.M. Fratantoni, and R.E. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[16] J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: a decentralized information-theoretic approach.* New York: Ellis Horwood, 1995.

[17] A.G.O. Mutambara. *Decentralized estimation and control for multisensor systems.* CRC press, 1998.

[18] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control, European Control Conference*, pages 8179–8184, 2005.

[19] L. Peng, M. Silic, and K. Mohseni. A DDDAS plume monitoring system with reduced Kalman Filter. In *International Conference on Computational Science (ICCS)*, volume 51, pages 2533–2542. Elsevier, 2015.

[20] T. Ritter, J. Euler, S. Ulbrich, and O. von Stryk. Adaptive observation strategy for dispersion process estimation using cooperating mobile sensors. In *Proceedings of the 19th IFAC World Congress*, pages 5302 – 5308, 2014.

[21] L. Sirovich. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.

[22] Z. Song, Y.Q. Chen, J. Liang, and D. Ucinski. Optimal mobile sensor motion planning under nonholonomic constraints for parameter estimation of distributed systems. *International Journal of Intelligent Systems Technologies and Applications*, 3(3/4):277–295, 2007.

[23] R. Stranders, A. Farinelli, A. Rogers, and N.R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 299–304, 2009.

[24] D. Zhang, C. Colburn, and T. Bewley. Estimation and adaptive observation of environmental plumes. In *Proceedings of the American Control Conference*, pages 4281–4286, 2011.