Technical report 17-018

Identification of distributed-parameter systems from sparse measurements*

Z. Hidayat, R. Babuška, A. Núñez, and B. De Schutter

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/17_018.html

Identification of Distributed-Parameter Systems from Sparse Measurements

Z. Hidayat[a,*], R. Babuška[a], A. Núñez[b], B. De Schutter[a]

[a]*Delft Center for Systems and Control, Delft University of Technology,*
Mekelweg 2, 2628 CD, Delft, The Netherlands
[b]*Section of Railway Engineering, Delft University of Technology,*
Stevinweg 1, 2628 CN, Delft, The Netherlands

Abstract

In this paper, a methodology for the identification of distributed-parameter systems is proposed, based on finite-difference discretization on a grid in space and time. It is considered the case when the partial differential equation describing the system is not known. The sensor locations are given and fixed, but not all grid points contain sensors. Per grid point, a model is constructed by means of lumped-parameter system identification, using measurements at neighboring grid points as inputs. As the resulting model might become overly complex due to the involvement of neighboring measurements along with their time lags, the Lasso method is used to select the most relevant measurements and so to simplify the model. Two examples are reported to illustrate the effectiveness of the methodology, a simulated two-dimensional heat conduction process and the construction of a greenhouse climate model from real measurements.

Keywords: system identification, finite-difference method, input selection, indoor climate modeling, greenhouse climate model

1. Introduction

Many real-life processes are distributed-parameter systems. Examples include chip manufacturing plants [1]; process control systems such as: packed-bed reactors [2], reverse flow reactors [3], and waste water treatment plants [4]; flexible structures in atomic force microscopes [5], ultraviolet disinfection installations in food industry [6], electrochemical process [7], or drying installations [8].

Distributed-parameter systems are typically modeled using partial differential equations. However, developing such models from first principles is a tedious and time-consuming process [9]. If input-output measurements are available, a model can be constructed by using system identification methods. However, due to the large number of spatially interdependent state variables, the identification of distributed-parameter systems is considerably more complex than the identification of lumped-parameter systems, and known as ill-posed inverse problems [10] because the solution is not unique [11]. There are three main approaches to the identification of distributed-parameter systems [12]: (i) direct identification, (ii) reduction to a lumped-parameter system, and (iii) reduction to an algebraic equation. While the direct identification approach uses the infinite-dimensional system model, the reduction-based approaches involve spatial discretization to create a set of ordinary differential equations in time to which identification methods for lumped-parameter systems can be applied. This approach, also called time-space separation [9], is the subject of this paper.

Two recent books related to the modeling of partial differential equations, and consequently to the proposed method, namely [13] by Cressie and Wikle and [14] by Billings. Cressie and Wikle [13] extensively treat statistical modeling and analysis of spatial, temporal, and spatio-temporal data. The spatio-temporal

---

*Corresponding author, phone: +62-21-5601378

Email addresses: `z.hidayat@tudelft.nl` (Z. Hidayat), `r.babuska@tudelft.nl` (R. Babuška), `a.a.nunezvicencio@tudelft.nl` (A. Núñez), `b.deschutter@tudelft.nl` (B. De Schutter)

data are in the form of time series of discrete-time and spatially distributed measurements. The text covers methods to interpolate values at locations that are not measured, as well as model simplification via spectral decomposition. Billings [14] addresses identification problems for nonlinear systems, including those based on spatio-temporal discretized partial differential equations with cellular automata and coupled map lattice structures. The models use polynomial basis function and are simplified by using orthogonal forward regression.

In this paper, a methodology for the identification of finite-dimensional models for distributed-parameter systems with a *small number of fixed sensors* is proposed. Compared to other finite-difference identification methods in the literature [15, 16, 17, 18, 19, 20, 14], this methodology:

- does not assume a dense set of measurement locations in space,
- uses an input selection method to reduce the complexity of the model.

The methodology also allows the use of external inputs in the model, a problem not addressed by Cressie and Wikle [13]. In addition, an application that, to our knowledge, has not yet been described in the literature is presented, namely the identification of a model for temperature dynamics in a greenhouse.

The remainder of the paper is organized as follows: Section 2 presents the problem formulation for which the methodology is proposed. Section 3 gives the details of the methodology. In Section 4, two examples are presented to show the effectiveness of the methodology: identification using data from a simulation of a 2D heat conduction equation and identification using temperature measurements of a real-life greenhouse setup. Section 5 concludes the paper.

2. Problem Formulation

Consider a distributed-parameter system described by a partial differential equation, with the associated boundary and initial conditions. For the ease of notation and without loss of generality, a system that is first-order in time and second-order in a two-dimensional space is presented:

$$\frac{\partial g(\boldsymbol{z},t)}{\partial t} = f\bigg(\boldsymbol{z}, t, g(\boldsymbol{z},t), \frac{\partial g(\boldsymbol{z},t)}{\partial z_1}, \frac{\partial g(\boldsymbol{z},t)}{\partial z_2}, \frac{\partial g(\boldsymbol{z},t)}{\partial z_1 z_2},$$
$$\frac{\partial^2 g(\boldsymbol{z},t)}{\partial z_1^2}, \frac{\partial^2 g(\boldsymbol{z},t)}{\partial z_2^2}, u(\boldsymbol{z},t), w(\boldsymbol{z},t) \bigg), \forall \boldsymbol{z} \in \mathscr{Z} \setminus \mathscr{Z}_{\mathrm{b}}, \forall t \tag{1a}$$

$$0 = \hbar\bigg(\boldsymbol{z}, t, g(\boldsymbol{z},t), \frac{\partial g(\boldsymbol{z},t)}{\partial z_1}, \frac{\partial g(\boldsymbol{z},t)}{\partial z_2}, u(\boldsymbol{z},t), w(\boldsymbol{z},t) \bigg), \forall \boldsymbol{z} \in \mathscr{Z}_{\mathrm{b}}, \forall t \tag{1b}$$

$$g(\boldsymbol{z},t_0) = g_0(\boldsymbol{z}), \forall \boldsymbol{z} \in \mathscr{Z} \tag{1c}$$

Here $g(\cdot,\cdot)$ is the variable of interest, $f(\cdot)$ is the system function, $\hbar(\cdot)$ is the boundary value function, $\boldsymbol{z} = (z_1, z_2) \in \mathscr{Z} \subset \mathbb{R}^2$ is the spatial coordinate,[1] $t \in \mathbb{R}^+ \cup \{0\}$ is the continuous-time variable, $u(\cdot,\cdot)$ is the input function, $w(\cdot,\cdot)$ is the process noise, and $\mathscr{Z}_{\mathrm{b}}$ is the set of spatial boundaries of the system. Higher-order and multi-variable systems can be defined analogously.

Assume that a set of input-output measurements are available from the distributed-parameter system (1) with unknown functions $f(\cdot)$ and $\hbar(\cdot)$. The sensors are located at specified points to measure $g(\cdot,\cdot)$, and there are also actuators that generate inputs $u(\cdot,\cdot)$ to the system. Since the actuators and the sensors are placed at known and fixed locations, the space is discretized with a set of grid points $\mathscr{M}_{\mathrm{g}}$ such that the actuator locations $\mathscr{M}_{\mathrm{u}}$ and the sensor locations $\mathscr{M}_{\mathrm{s}}$ are in $\mathscr{M}_{\mathrm{g}}$, i.e., $\mathscr{M}_{\mathrm{u}} \subset \mathscr{M}_{\mathrm{g}}$ and $\mathscr{M}_{\mathrm{s}} \subset \mathscr{M}_{\mathrm{g}}$. Assume that the measurements, concatenated in a vector $\boldsymbol{y}(\cdot)$, are affected by additive Gaussian noise $v(z_i, t) \sim \mathcal{N}(0, \sigma_{\mathrm{v}_i}^2)$. The input and measurement vectors are defined as:

$$\boldsymbol{u}(t) = \begin{bmatrix} u(\boldsymbol{z}_{\mathrm{u},1}, t) & \dots & u(\boldsymbol{z}_{\mathrm{u},N_{\mathrm{u}}}, t) \end{bmatrix}^\top \tag{2a}$$

---

[1]Vectors are denoted by boldface symbols.

$$\boldsymbol{y}(t) = \begin{bmatrix} g(\boldsymbol{z}_{\mathrm{g},1},t) + v(\boldsymbol{z}_{\mathrm{g},1},t) & \dots & g(\boldsymbol{z}_{\mathrm{g},N_{\mathrm{s}}},t) + v(\boldsymbol{z}_{\mathrm{g},N_{\mathrm{s}}},t) \end{bmatrix}^{\top} \tag{2b}$$

where N_{u} is the number of actuators, N_{s} the number of sensors, the coordinates of the inputs are denoted by $\boldsymbol{z}_{\mathrm{u},j} \in \mathcal{M}_{\mathrm{u}}$, the measurement coordinates by $\boldsymbol{z}_{\mathrm{g},i} \in \mathcal{M}_{\mathrm{g}}$, and the superscript $\top$ denotes the transpose of a matrix or vector. Note that not every grid point is associated with a sensor or actuator.

The measurements are collected at discrete time steps $t_k = k \cdot T_{\mathrm{s}}$ with $k \in \mathbb{N} \cup \{0\}$, where T_{s} is the sampling period. To simplify the notation, the discrete time instant t_k is subsequently written as k. The notation is further simplified by using an integer subscript assigned to the given sensor or actuator location:

$$u_j(k) = u(\boldsymbol{z}_{\mathrm{u},j},t)\big|_{t=k \cdot T_{\mathrm{s}}}, \quad j = 1, \dots, N_{\mathrm{u}} \tag{3}$$

for the inputs and

$$y_i(k) = \Big(g(\boldsymbol{z}_{\mathrm{g},i},t) + v(\boldsymbol{z}_{\mathrm{g},i},t)\Big)\Big|_{t=k \cdot T_{\mathrm{s}}}, \quad i = 1, \dots, N_{\mathrm{s}} \tag{4}$$

for the outputs. The input and output data (3) and (4) are the only available information to construct a distributed finite-order model of (1).

3. Identification Methodology for Distributed-Parameter Systems

The main idea of the methodology proposed in this paper is to identify at each sensor location a lumped-parameter system, described by a dynamic model. To take into account the spatial dynamics of the system, measurements from the neighboring locations are included as inputs.

Given the set of input-output measurements from an unknown distributed-parameter system, the identification procedure is the following:

1. Create a spatial grid for the system so that each sensor and each actuator is associated with a grid point. The grid may have a uniform or a nonuniform spacing, depending on the actuator and sensor locations. Recall that not all grid points are occupied by sensors or actuators. The sensors and actuators are numbered consecutively: $i = 1, \dots, N_{\mathrm{s}}$ for the sensors and $j = 1, \dots, N_{\mathrm{u}}$ for the actuators. An illustration of a 2D system, with spatial grid points and labels for the sensors and actuators, is shown in Figure 1.
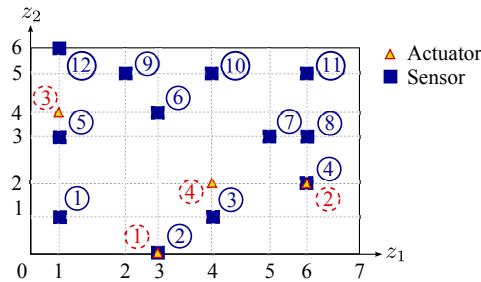

Figure 1: An illustration of a 2D system with a nonuniform spatial grid. Sensors and actuators are indicated by solid and dashed circles, respectively.

2. For each sensor i in the grid:
   (a) Determine the dynamic model structure, using one of the available structures for lumped-parameter systems, such as auto-regressive with exogenous input (ARX), output error (OE), Box-Jenkins (BJ), etc.
   (b) Define the set of neighboring sensors and actuators, i.e., those that are located in a defined neighborhood (details on the notion of neighborhood are given in the next section). The neighboring measurements and inputs from neighboring actuators become inputs to the dynamic model of sensor i. Determine the (temporal) system order and construct the regressors.

3

(c) When the number of regressors is large, optimize the model structure in order to simplify the model.

(d) Estimate the parameters of the dynamic model for sensor i.

(e) Validate the dynamic model. If the model is rejected, return to step 2a to use a different system structure or to 2b to change the set of neighbors.

The sequence of the steps and decisions of the methodology is shown in Figure 2 and the steps are detailed next. More specifically, it is discussed:

- How to construct coupled discrete-time dynamic models in Section 3.1.
- How to identify and estimate the parameters of the models in Section 3.2.
- How to simplify the identified models to obtain simpler models in Section 3.3.
- Sensor placement and interpolation for locations where measurements are not available in Section 3.4.
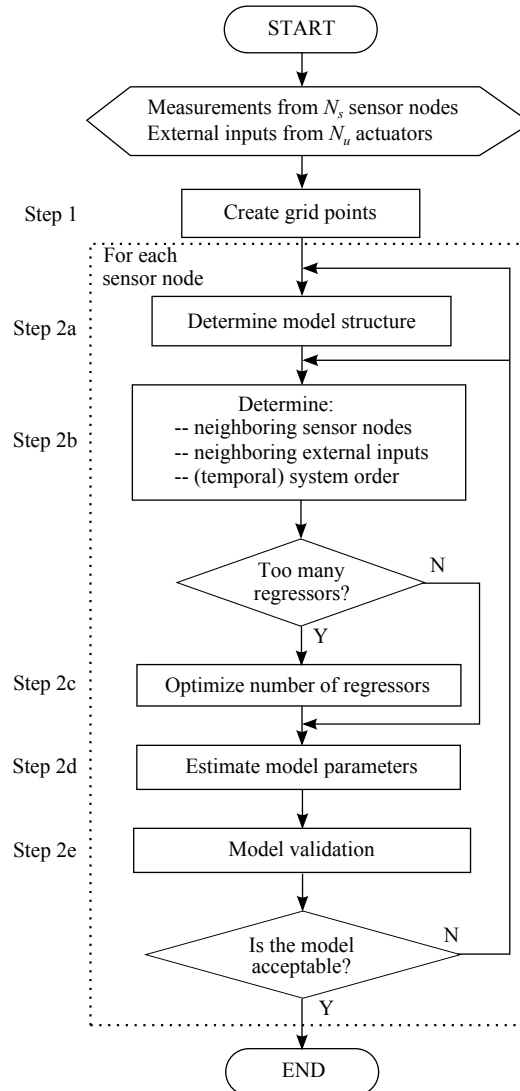

Figure 2: Flow chart of the proposed methodology.

4

Remarks:

- The proposed framework performs off-line identification for distributed-parameter systems, however, the method can be extended directly to recursive identification for the ARX structure.
- For structures that require the predicted output to compute the parameter, extension to recursive parameter estimation is possible in condition that the measurements are updated synchronously.
- The convergence analysis for the recursive implementation of the framework follows [21].

3.1. Construction of coupled discrete-time dynamic models

The discretization of a partial differential equation in space by using the finite-difference method results in a set of coupled ordinary differential equations. At time instant t, the coupling spatially relates the value of the variable of interest at node i, $g_i(t)$, to values of the same variable at the neighboring nodes. The influence of more distant neighbors may be delayed due to the finite speed of spatial propagation of the quantity of interest. As an example, consider the following simplification of (1a) to an autonomous one-dimensional case:

$$\frac{\partial g(z,t)}{\partial t} = m\left(\frac{\partial^2 g(z,t)}{\partial z^2}\right) \tag{5}$$

where $g(z,t) \in \mathbb{R}$ is the variable of interest, $z \in \mathbb{R}$ is the spatial coordinate, and $m(\cdot)$ is a nonlinear function. The system is spatially discretized using the finite-difference method by creating grid points, which, for the sake of simplicity, are uniformly spaced at distance Δ_z. Denote $g_i(t)$ for $g(z,t)$ at grid point $z = i \cdot \Delta_z$, called node i for short. The central approximation [22] of the second-order derivative in space is:

$$\left.\frac{\partial^2 g(z,t)}{\partial z^{2^2}}\right|_{z=i} \approx \frac{g_{i+1}(t) - 2g_i(t) + g_{i-1}(t)}{(\Delta_z)^2} \tag{6}$$

which results in:

$$\frac{\mathrm{d}g_i(t)}{\mathrm{d}t} = m\left(\frac{g_{i+1}(t) - 2g_i(t) + g_{i-1}(t)}{(\Delta_z)^2}\right) \tag{7}$$

Then, by using the forward-difference approximation of the time derivative:

$$\left.\frac{\mathrm{d}g_i(t)}{\mathrm{d}t}\right|_{t=k} \approx \frac{g_i(k+1) - g_i(k)}{T_{\mathrm{s}}}$$

to discretize the left-hand side of (7), which gives:

$$g_i(k+1) = g_i(k) + T_{\mathrm{s}} \cdot m\left(\frac{g_{i+1}(k) - 2g_i(k) + g_{i-1}(k)}{(\Delta_z)^2}\right) \tag{8}$$

or in a slightly more general form:

$$g_i(k+1) = q\Big(g_i(k), g_{i-1}(k), g_{i+1}(k), T_{\mathrm{s}}, \Delta_z\Big) \tag{9}$$

Note that in this example $g_i(k)$ is influenced only by its immediate neighbors. For systems with a higher spatial order and with exogenous inputs (9) can be written as:

$$g_i(k+1) = q(g_{\mathcal{N}_{\mathrm{s},i}}(k), u_{\mathcal{N}_{\mathrm{u},i}}(k), T_{\mathrm{s}}, \Delta_z) \tag{10}$$

where $g_{\mathcal{N}_{\mathrm{s},i}}(k) = \{g_j(k)|j \in \mathcal{N}_{\mathrm{s},i}\}$ is the set of neighboring variables of interest, including $g_i(k)$ itself and $u_{\mathcal{N}_{\mathrm{u},i}}(k) = \{u_l(k)|l \in \mathcal{N}_{\mathrm{u},i}\}$ is the set of neighboring inputs including $u_i(k)$ itself.

In the system identification setting, Δ_z and T_{s} are known and fixed and instead of $g_i(k)$ the measurement $y_i(k)$ is used (which includes the effect of measurement noise $v_i(k)$). Thus the following model is obtained:

$$y_i(k+1) = \mathrm{w}(y_{\mathcal{N}_{\mathrm{s},i}}(k), u_{\mathcal{N}_{\mathrm{u},i}}(k), v_{\mathcal{N}_{\mathrm{s},i}}(k)) \tag{11}$$
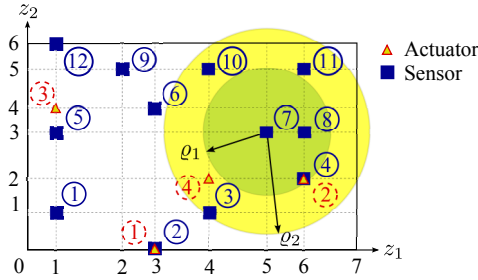
Figure 3: An illustration of the neighboring measurements and inputs set with two possible neighborhoods of sensor 7 using a Euclidean distance criterion. The first set of neighbors is defined using distance ϱ_1 from sensor 7 and the second set using distance ϱ_2.

where $y_{\mathcal{N}_{\mathrm{s},i}}(k)$ is the set of neighboring measurements at node i, including $y_i(k)$. The neighbors of node i can be simply the nodes that are within a specified distance ϱ, i.e., $y_{\mathcal{N}_{\mathrm{s},i}}(k) = \left\{ y(z,k) | \, \|z - z_i\| \leq \varrho, z \in \mathcal{M}_{\mathrm{u}} \cup \mathcal{M}_{\mathrm{s}} \right\}$ for measurements and $u_{\mathcal{N}_{\mathrm{s},i}}(k) = \left\{ u(z,k) | \, \|z - z_i\| \leq \varrho, z \in \mathcal{M}_{\mathrm{u}} \cup \mathcal{M}_{\mathrm{s}} \right\}$ for inputs, see Figure 3. A priori knowledge can be used to obtain a suitable value of ϱ.

An inappropriate choice of ϱ may, however, yield a large number of neighbors that are included in the model. In order to reduce the model complexity, an input or regressor selection method is applied. This topic is discussed later on in Section 3.3.

When $w(\cdot)$ in (11) is not known, an approximation can be designed using the available input-output data and linear or nonlinear system identification. Assuming that the system can be approximated by a linear model, linear system identification methods can be applied (11), as described in the following section.

3.2. System identification and parameter estimation

Identification methods for linear systems (including linear-in-parameters nonlinear systems) use the following model representation:

$$\hat{y}_i(k+1) = \boldsymbol{\phi}_i(k)\boldsymbol{\theta}_i^T \tag{12}$$

where $\hat{y}_i(k)$ is the predicted $y_i(k)$, $\boldsymbol{\phi}_i(k)$ is the regressor vector at time step k, and $\boldsymbol{\theta}_i$ is the vector of parameters. Note that the subscript index i corresponds to sensor i as in the previous section. The regressor vector contains lagged input-output measurements, including those of neighboring sensors and inputs. The parameter vector $\hat{\boldsymbol{\theta}}_i$ can be estimated by using least-squares methods [23], so that the following prediction error is minimized:

$$\hat{\boldsymbol{\theta}}_i = \arg\min_{\boldsymbol{\theta}_i} \sum_{k=1}^{N} \left\| y_i(k+1) - \boldsymbol{\phi}_i(k)\boldsymbol{\theta}_i^T \right\|_2^2$$

$$= \arg\min_{\boldsymbol{\theta}_i} \sum_{k=1}^{N} \left\| \epsilon_i(k) \right\|_2^2$$

with $\epsilon_i(k) = y_i(k) - \hat{y}_i(k)$ the prediction error. The use of neighboring measurements as inputs to the model may lead a situation where the regressors are corrupted by noise. This requires an error-in-variables identification approach, solved, e.g., by using total least squares [24]. For a thorough discussion of the total-least squares method refer to [25]. When noiseless input variables to the actuators are among the regressors, a mixed ordinary-total least-squares method must be used [25].

In nonlinear system identification, the problem is more difficult as there is no unique way to represent the nonlinear relation between the regressors and the output, and different methods are available to represent the nonlinearity. For instance, Wiener systems [26] and Hammerstein systems [27] use nonlinear functions cascaded with a linear system, Takagi-Sugeno fuzzy models combine local linear models by weighting them via membership functions [28], while neural networks use global nonlinear basis functions [29].

3.3. Model reduction by using regressor selection

Including neighboring measurements as inputs will increase the size of the regressor vector $\boldsymbol{\phi}_i(k)$. This size is determined by the number of neighboring inputs and the number of components of each neighboring regressor vector. For several reasons, it is desired to have a simpler model by removing inputs that do not contribute to the output to reduce computational load, especially when the models are used in on-line control design.

Three methods are commonly used in standard linear regression [30]: stepwise regression, backward elimination, and exhaustive search. With these methods, the inclusion or exclusion of a regressor is decided based on statistical tests, such as the F-test. One of more recent methods is Lasso [31], which stands for the *least absolute shrinkage and selection operator*. Assumed is the following regression model:

$$\hat{y}(k) = \theta_0 + \boldsymbol{\phi}^\top \boldsymbol{\theta}(k)$$

with $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \dots & \theta_{n_{\mathrm{r}}} \end{bmatrix}^\top$ and θ_0 the parameters of the model and $\boldsymbol{\phi}$ the vector of regressors. Lasso computes the parameters so that the parameters of regressors that have the least importance are made zero by using a regularization parameter. Lasso solves the following optimization problem [31]:

$$\begin{bmatrix} \hat{\theta}_0 & \hat{\boldsymbol{\theta}}^\top \end{bmatrix}^\top = \arg\min_{\theta_0, \boldsymbol{\theta}} \sum_{i=1}^{N} \left(y_i - \theta_0 - \boldsymbol{\phi}_i^\top \boldsymbol{\theta} \right)^2, \quad \text{s.t.} \sum_{j=1}^{n_{\mathrm{r}}} |\theta_j| \leq \tau \tag{13}$$

where τ is the tuning parameter, and for the sake of simplicity the scalar case is considered (extension to the vector case is straightforward). This problem can also be written as:

$$\begin{bmatrix} \hat{\theta}_0 & \hat{\boldsymbol{\theta}}^\top \end{bmatrix}^\top = \arg\min_{\theta_0, \boldsymbol{\theta}} \left(\frac{1}{2N} \sum_{i=1}^{N} \left(y_i - \theta_0 - \boldsymbol{\phi}_i^\top \boldsymbol{\theta} \right)^2 + \lambda \sum_{j=1}^{n_{\mathrm{r}}} |\theta_j| \right) \tag{14}$$

where λ is the nonnegative regularization parameter. Note that the two formulations are equivalent in the sense that for any $\tau \geq 0$ in (13), there exists a $\lambda \in [0, \infty)$ in (14) such that the both formulations have the same solution, and vice versa.

As for nonlinear systems there is no unique representation, regressor selection is more complex. The simplest method, but computationally inefficient, is by directly searching the most optimal set of regressors using exhaustive search. Regarding model-specific methods, forward regression has been used for polynomial models [32, 33], neural networks [34], and for adaptive network fuzzy inference systems [35]. For an example of model-independent regressor selection method, one may refer to, e.g., [36], which uses fuzzy clustering.

3.4. Sensor and actuator locations and interpolation

Measurements and actuations in distributed-parameter systems are commonly performed at spatially sampled locations. This practice raises two related problems in control and estimation of distributed-parameter systems:

1. How many sensors and/or actuators are required and where they should be placed to obtain good output observations? For the identification of distributed-parameter systems, the problem is about using the smallest number sensors possible and placing them in certain locations such that the experiment data can be used to obtain a valid model. A short introduction to this topic is given in a survey by Kubrusly and Malebranche [37] and a more recent and thorough treatment on the optimal sensor placement is given by Ucińsky [38]. In case the underlying partial differential model is known, the locations of the sensors will influence the identifiability of the distributed-parameter system [38]. In this paper the underlying partial differential model is unknown and the sensor and actuator locations are assumed fixed and given; therefore, the sensor and actuator location problem is not considered further here.

   Beside the actuator and sensor locations, the input signals applied to the system are also crucial to get a useful model. A general requirement for the excitation is that the signals should be persistently

exciting so that the measurement data contain the important dynamics of the system, from which the system can be identified. The pseudo-random binary signal is a commonly used input signal for system identification because it has the desired properties [21].

2. How to interpolate outputs at locations that are not measured? This problem naturally arises because the sensors give information only at their locations [13].

   For the interpolation problem, kriging and splines are commonly used methods [13]. However, only kriging, more specifically ordinary kriging, is used and briefly presented in this paper following [13]. Kriging was initially developed to solve estimation-related problems in geology and it able to interpolate in time and space. Because temporal interpolation is not required in our setting, only spatial kriging is given in this section.

Given a spatial random process, also called random field:

$$Y(\boldsymbol{z}) = G(\boldsymbol{z}) + V(\boldsymbol{z}), \quad \boldsymbol{z} \in \mathscr{X}$$

where $Y(\cdot)$, $G(\cdot)$, and $V(\cdot)$, are respectively the measured random field, the true but unknown random field, and the random measurement noise, $\boldsymbol{z}$ is the spatial coordinate, and $\mathscr{X}$ is the spatial domain. As the spatial domain $\mathscr{X}$ has been discretized using the finite-difference method, the measurements of the random field realizations can be written as $y_i = g_i + v_i$, where the subscript i is defined similarly to that of (7), from which the measurement vector $\boldsymbol{y}_{\mathrm{z}}$ is defined as the stacked measurements from $N_{\boldsymbol{y}_{\mathrm{z}}}$ sensor locations.

Remarks:

1. Depending on the purpose, a spatio-temporal random process

$$Y(\boldsymbol{z}, t) = G(\boldsymbol{z}, t) + V(\boldsymbol{z}, t), \quad \forall \boldsymbol{z} \in \mathscr{X}, \forall t \tag{15}$$

   for a certain fixed time t can be viewed as a random field $Y(z)$ or as a dynamic random process $Y(t)$ [39, 40]:

$$Y(\boldsymbol{z}) = G(\boldsymbol{z}) + V(\boldsymbol{z}), \quad \forall \boldsymbol{z} \in \mathscr{X} \tag{16a}$$

$$Y(t) = G(t) + V(t), \quad \forall t \tag{16b}$$

2. In the case of the proposed methodology, (2b) is the discrete-time realization of (16b) at sensor location $\boldsymbol{z}_i \in \mathscr{M}_{\mathrm{s}}$.

Kriging [13] a linear estimation method to obtain the optimal spatial estimate of the second order stationary process $G(\boldsymbol{z})$ at a coordinate location that is not measured $\boldsymbol{z}_{\mathrm{o}} \notin \mathscr{M}_{\mathrm{s}}$, such that the mean square estimation error (MSE):

$$\mathrm{MSE} = \mathbb{E}\left\{ \left(g_{\boldsymbol{z}_{\mathrm{o}}} - \hat{G}(\boldsymbol{y}_{\mathrm{z}}) \right)^2 \right\} \tag{17}$$

is minimized, where $g_{\boldsymbol{z}_{\mathrm{o}}}$ is the true but unknown value of the process $G(\boldsymbol{z})$, $\hat{G}(\boldsymbol{y}_{\mathrm{z}})$ is the estimator, and $\mathbb{E}\{\cdot\}$ is the expectation operator.

In ordinary kriging, the mean of $G(\boldsymbol{z})$ is assumed constant, i.e., $\mathbb{E}\{G(\boldsymbol{z})\} = \mu_G, \boldsymbol{z} \in \mathscr{X}$, the covariance function $\mathrm{Cov}(g_i, g_j)$ and the zero mean measurement error variance σ_V^2 are assumed to be known. The estimator has the following form:

$$\hat{G}_{\mathrm{O}}(\boldsymbol{y}_{\mathrm{z}}) = \boldsymbol{\gamma}^\top \boldsymbol{y}_{\mathrm{z}} \tag{18}$$

for the column vector $\boldsymbol{\gamma} \in \mathbb{R}^{N_{\boldsymbol{y}_{\mathrm{z}}}}$ is the estimator parameter and $\top$ denotes the transpose operation of a vector and a matrix. The problem of kriging is to find $\boldsymbol{\gamma}$ to minimize (17). To impose unbiasedness, $\boldsymbol{\gamma}^\top \mathbf{1} = 1$ has to be fulfilled, where $\mathbf{1}$ is a column vector with 1 as the elements. By using the Lagrange multiplier ζ, the parameter vector $\boldsymbol{\gamma}$ is computed by solving the following optimization problem:

$$\arg\min_{\boldsymbol{\gamma}} \left(\mathbb{E}\left\{ \left(g_{\boldsymbol{z}_{\mathrm{o}}} - \boldsymbol{\gamma}^\top \boldsymbol{y}_{\mathrm{z}} \right)^2 \right\} - 2\zeta \cdot \left(\boldsymbol{\gamma}^\top \mathbf{1} - 1 \right) \right) \tag{19}$$

The solution of the above optimization problem is:

$$\boldsymbol{\gamma}^* = \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\Big(\mathrm{Cov}\big(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}}\big) + \zeta^*\boldsymbol{1}\Big) \tag{20}$$

$$\zeta^* = \frac{1 - \boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\,\mathrm{Cov}\big(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}}\big)}{\boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\boldsymbol{1}} \tag{21}$$

where $\boldsymbol{\gamma}^*$ and ζ^* are respectively the optimal parameter vector and Lagrange multiplier, $\mathrm{Var}(\cdot)$ is the variance function, and $\boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}$ is the covariance matrix of measurement vector $\boldsymbol{y}_{\mathrm{z}}$ defined as:

$$\boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}} = \begin{cases} \mathrm{Var}(y_i) + \sigma_V^2 & i = j \\ \mathrm{Cov}(y_i, y_j) & i \neq j \end{cases}$$

Substituting ζ^* in (20) and $\boldsymbol{\gamma}^*$ into (18) gives:

$$\hat{G}_{\mathrm{O}}(\boldsymbol{y}_{\mathrm{z}}) = \left(\mathrm{Cov}(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}}) + \boldsymbol{1}\frac{1 - \boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\,\mathrm{Cov}(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}})}{\boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\boldsymbol{1}}\right)^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\boldsymbol{y}_{\mathrm{z}} \tag{22}$$

with the corresponding mean square error:

$$\mathrm{MSE} = \mathrm{Var}(g_{\boldsymbol{z}_\circ}) - \mathrm{Cov}(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}})^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\mathrm{Cov}(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}}) + \frac{1 - \boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\,\mathrm{Cov}(g_{\boldsymbol{z}_\circ}\boldsymbol{y}_{\mathrm{z}})}{\boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\boldsymbol{1}} \tag{23}$$

Equation (22) can be rewritten as:

$$\hat{G}_{\mathrm{O}}(\boldsymbol{y}_{\mathrm{z}}) = \hat{\mu}_G + \mathrm{Var}(g_{\boldsymbol{z}_\circ})^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1} \cdot (\boldsymbol{y}_{\mathrm{z}} - \hat{\mu}_G \boldsymbol{1}) \tag{24}$$

with $\hat{\mu}_G$ the generalized least-squares estimator of μ_G [41]:

$$\hat{\mu}_G = \frac{\boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\boldsymbol{y}_{\mathrm{z}}}{\boldsymbol{1}^\top \boldsymbol{C}_{\boldsymbol{y}_{\mathrm{z}}}^{-1}\boldsymbol{1}} \tag{25}$$

Another variant of kriging is universal kriging, which assumes $\mu_G(\boldsymbol{z})$ to be a linear model instead of a constant as in ordinary kriging. An interesting application of this kriging variant is the Kalman filter method for distributed motion coordination strategy of mobile robot positioning at critical locations [42].

4. Simulations and Applications

To illustrate the effectiveness of the proposed identification approach, two examples are considered, based on synthetic and real data, respectively. The synthetic data are generated from a linear two-dimensional heat conduction equation. The real-life data are temperature measurements from a small-scale real greenhouse.

4.1. Heat conduction process

Consider the following two-dimensional heated plate conduction process:

$$\frac{\partial T(\boldsymbol{z},t)}{\partial t} = \frac{\kappa}{\rho C_{\mathrm{p}}}\left[\frac{\partial^2 T(\boldsymbol{z},t)}{\partial z_1^2} + \frac{\partial^2 T(\boldsymbol{z},t)}{\partial z_2^2}\right], \forall \boldsymbol{z} \in \mathscr{X} \setminus \mathscr{X}_{\mathrm{b}}, \forall t \tag{26a}$$

$$T(\boldsymbol{z},t) = T_{\mathrm{b}}(t), \quad \forall \boldsymbol{z} \in \mathscr{X}_{\mathrm{b}}, \forall t \tag{26b}$$

$$T(\boldsymbol{z},0) = T_0, \quad \forall \boldsymbol{z} \in \mathscr{X} \tag{26c}$$

where $T(\boldsymbol{z},t)$ is the temperature of the plate at location $\boldsymbol{z}$ and at time t, ρ the density of the plate material, T_0 the initial temperature, C_{p} the heat capacity, κ the thermal conductivity, and $\boldsymbol{z} = (z_1, z_2)$ the spatial

Table 1: The plate parameters for the 2D heat conduction equation example

Parameter	Symbol	Value	Unit
Material density	ρ	4700	$\mathrm{kg\,m^{-3}}$
Thermal conductivity	κ	700	$\mathrm{W\,m^{-1}\,K^{-1}}$
Heat capacity	C_p	383	$\mathrm{J\,kg^{-1}\,K^{-1}}$
Plate length	L	0.7	m
Plate width	W	0.5	m
Initial temperature	T_0	35	$^\circ\mathrm{C}$
Sampling period	T_s	1	s
Grid size	$\Delta_{z_1}, \Delta_{z_2}$	0.05	m

coordinate on the plate. Equations (26b) and (26c) are the boundary and initial conditions, respectively. The plate's parameters are listed in Table 1. The values of the material properties are adopted from [43] and modified to speed up the simulation.

For this example, a set of identification data is obtained by simulating the discretized (26). The central approximation of the finite-difference method is used to discretized the space to create a grid of 14 by 10 cells and the zero-order hold method is used to discretize the time. The resulting discretized equation is simulated by letting the boundary values $T_\mathrm{b}(\cdot)$ follow pseudo-random binary signals with levels of $25\,^\circ\mathrm{C}$ and $80\,^\circ\mathrm{C}$ where each boundary B-1 through B-4 (as defined in Figure 4) is excited by a different signal u_1 through u_4. It is assumed that the excitation is uniformly distributed along the boundary. The duration of the steps is randomly selected from the set $\{80, 120, \ldots, 200\}$ seconds. The maximum value of the step duration was determined based on the largest time constant of the nodes responses, i.e., $180\,\mathrm{s}$.
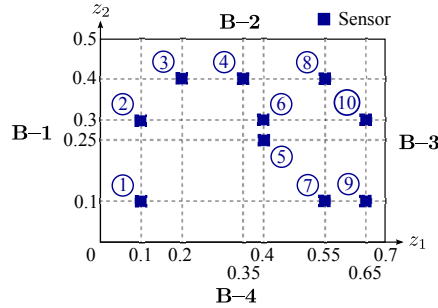

Figure 4: Illustration of sensor node locations for the 2D heat conduction example.

Ten sensor nodes are placed to measure the temperature of the plate as illustrated in Figure 4, with the exact sensor locations given in Table 2. The measurements are sampled with the period $T_\mathrm{s} = 1\,\mathrm{s}$ and Gaussian noise with zero mean and variance $0.1\,^\circ\mathrm{C}^2$ is added to the measurements. The data set is divided into an identification set and a validation set, consisting of 1500 and 740 samples, respectively.

Table 2: Coordinates of the sensor node locations for the 2D heat conduction equation example

Sensor #	(z_1, z_2)	Sensor #	(z_1, z_2)
Sensor 1	(0.10, 0.10)	Sensor 6	(0.40, 0.30)
Sensor 2	(0.10, 0.25)	Sensor 7	(0.55, 0.10)
Sensor 3	(0.20, 0.40)	Sensor 8	(0.55, 0.40)
Sensor 4	(0.35, 0.40)	Sensor 9	(0.65, 0.10)
Sensor 5	(0.40, 0.25)	Sensor 10	(0.65, 0.30)

The neighboring nodes are defined to be the nodes that lie within the distance $\varrho = 0.35\,\mathrm{m}$ from a given node. The value of this neighborhood radius is set sufficiently large compared to the physical dimensions so that there are sufficient neighboring sensors included in the model. Typically, prior knowledge about the

process can be used to determine a suitable value for the radius ϱ.

Results from two representative sensors are presented: 1 and 5. Sensor 1 is relatively close to the boundaries; it has three neighboring sensors. Since boundaries B-1 and B-4 are inside the radius ϱ, the values at boundaries B-1 and B-4 are included as inputs to the model of sensor 1. Sensor 5 is near the middle of the plate; it has 9 neighboring sensors and it also uses the values of boundaries B-2, B-3, and B-4 as inputs. The boundary inputs are assumed spatially continuous and each of them is spatially constant for each discrete-time k. In case a sensor node has a boundary in the neighborhood, it is taken as one input to the model.

Subsequently, it is necessary to determine the order of the system. Considering that the system is slow, 10th-order systems are used with an ARX structure for the models. Thus, sensor 1 has initially 61 regressors for the model and sensor 5 has 131 regressors including the bias. Lasso is applied to reduce the number of parameters in the model, using the `lasso` function in the Statistics Toolbox of Matlab. The function requires the maximum number of parameters in the model as additional input and returns a set of models with the number of parameters varying from one to the maximum number specified. The function returns a set of reduced models for different values of regularization parameter λ and the corresponding MSE values. Then, one of those models is selected, based on the smallest MSE obtained from the validation data set.

After input selection, a model with 11 parameters is obtained for sensor 1 and a model with 26 parameters for sensor 5. The reduced models are the following:

$$
\begin{aligned}
y_1(k+1) = {} & 0.0155\, y_1(k-1) + 0.0540\, y_3(k-1) + 0.0467\, y_5(k-1) + \\
& + 0.4118\, u_1(k-1) + 0.0173\, u_1(k-2) + 0.0026\, u_1(k-3) \\
& + 0.4134\, u_4(k-1) + 0.0244\, u_4(k-2) + 0.0034\, u_4(k-3) \\
& - 0.5318 \\
y_5(k+1) = {} & 0.0089\, y_5(k-1) + 0.0093\, y_8(k-1) + 0.0037\, y_{10}(k-2) \\
& + 0.0145\, y_2(k-1) + 0.0050\, y_2(k-2) + 0.0035\, y_2(k-3) \\
& + 0.1352\, y_1(k-1) + 0.0241\, y_1(k-2) + 0.0073\, y_1(k-3) \\
& + 0.1640\, u_2(k-1) + 0.1074\, u_2(k-2) + 0.0350\, u_2(k-3) \\
& + 0.0131\, u_2(k-4) + 0.0016\, u_2(k-5) + 0.0002\, u_2(k-6) \\
& + 0.0831\, u_3(k-1) + 0.0627\, u_3(k-2) + 0.0221\, u_3(k-3) \\
& + 0.0063\, u_3(k-4) + 0.0006\, u_3(k-5) + 0.1646\, u_4(k-1) \\
& + 0.0588\, u_4(k-2) + 0.0245\, u_4(k-3) + 0.0094\, u_4(k-4) \\
& + 0.0039 u_4(k-5) - 0.8707
\end{aligned}
$$

where $y_i(k)$ is the measurement from sensor i, and $u_j(k)$ is the input from boundary j. From the above models, it can be seen that the model for sensor 5 uses more parameters with larger lags of inputs and neighboring measurements; this indicates that more time is needed to propagate those inputs and neighboring measurements to influence sensor 5. This is different in the case of sensor 1, which is closer to the boundaries and for which the resulting model is mainly influenced by the inputs, which yields a simpler model. The models also have constant/bias terms which can be interpreted as heat transferred between the adjacent nodes.

Figure 5 and 6 show the one-step ahead predictions, the free-run[2] simulation predictions and their corresponding errors in comparison with validation part of the data. As one can expect, for the validation data set the one-step-ahead prediction error is much lower than the error from the free-run simulation. In addition, it can also be seen that the free-run prediction errors are smaller for the reduced input models than those of the full input models. This is more obvious for the model of sensor 5. As one expects that the full models would deliver smaller errors, this means the full models are overfit. In general, the proposed identification approach works well in this case and delivers sufficiently good models.

---

[2]Free-run simulations mean that the model outputs are predicted based on the inputs and the past predicted outputs. Output measurements are not used to generate the predictions. This is a very stringent test of the model prediction accuracy.

(a) Sensor 1: measurements and pre-
dictions

(b) Sensor 5: measurements and pre-
dictions

(c) Sensor 1: prediction error
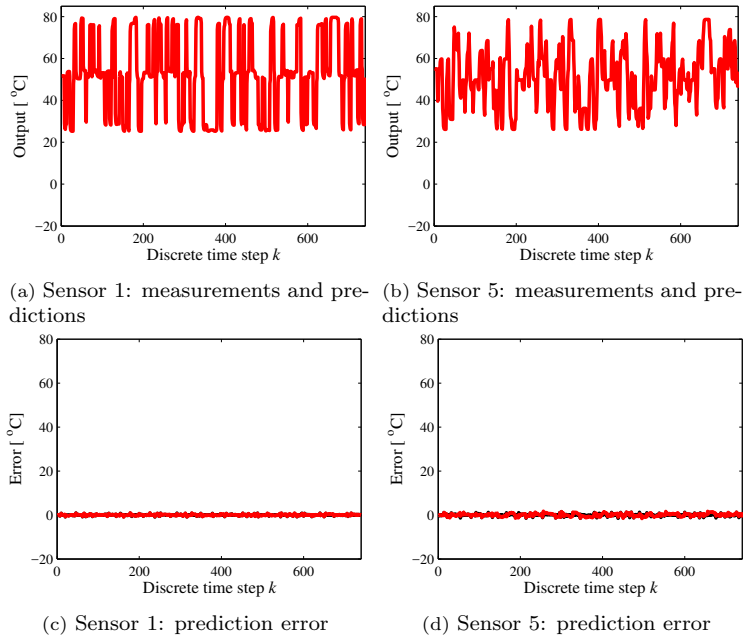
(d) Sensor 5: prediction error

Figure 5: Measurements (blue, invisible due to the overlap) and one-step ahead prediction for the models with full inputs (black curves) and the ones with reduced inputs (red curves) using the validation data set for the two-dimensional heat conduction example. Note that the prediction error of the full and the reduced input models are overlapping.
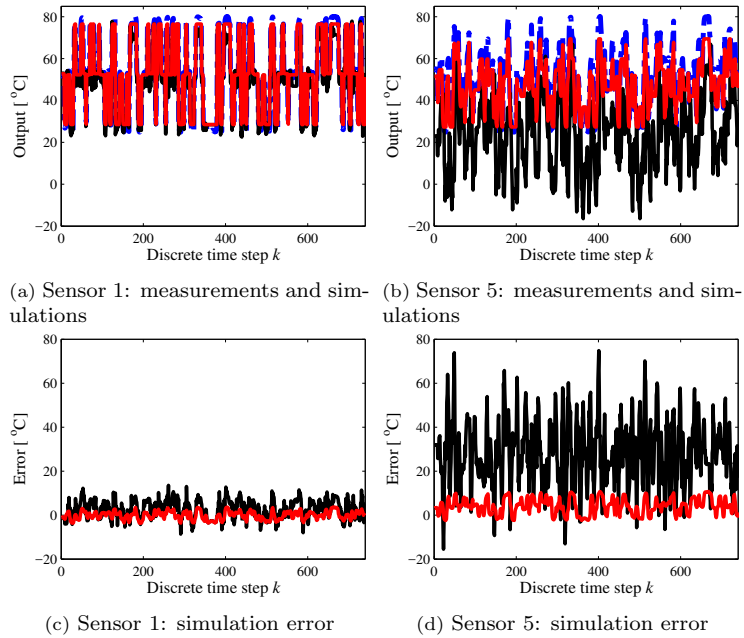

(a) Sensor 1: measurements and sim-
ulations

(b) Sensor 5: measurements and sim-
ulations

(c) Sensor 1: simulation error

(d) Sensor 5: simulation error

Figure 6: Measurements (blue) and free-run simulation predictions for the models with full inputs (black) and the ones with reduced inputs (red) using the validation data set for the two-dimensional heat conduction example. Note that the output error of the full and the reduced input models are overlapping.

The figures also show that the output error of the model using measurements from sensor 1 is generally smaller than that of sensor 5. This can be explained as follows: Figure 4 shows that sensor 5 has more neighboring sensors than sensor 1. This means the identification for measurements of sensor 5 involves more noise from measurements of neighboring sensors than in the case of sensor 1.

Figure 7 shows contour plots of the temperature distribution based on the validation data and their one-step-ahead and free-run simulation predictions at discrete-time step $k = 90$; the time step value is picked without any preference. The sensor locations are marked with black square boxes where sensor numbers are placed at the left-hand side of the markers. It can be seen the contour of the one-step ahead prediction is very similar to that from the validation data. This is confirmed by the error contour, which is almost uniformly colored around the zero value. Note that the contours look relatively coarse because they are plotted based on sparse measurement locations using the ordinary kriging, implemented in the ooDACE toolbox [44, 45], to interpolate temperature at locations that are not measured.

(a) Validation data   (b) One-step ahead prediction   (c) Reduced identified models

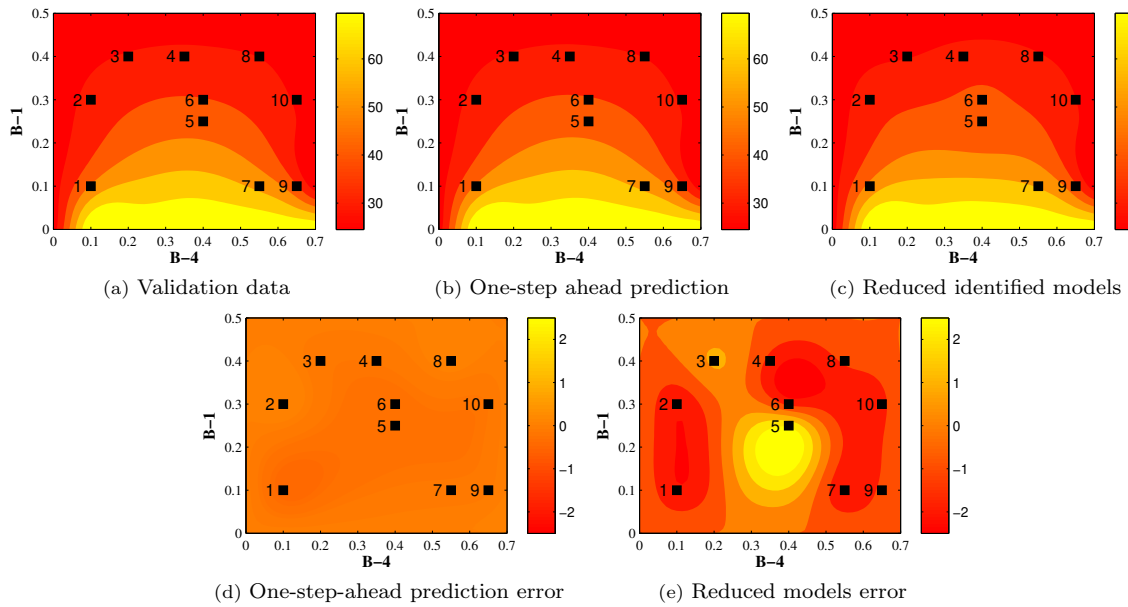(d) One-step-ahead prediction error   (e) Reduced models error

Figure 7: Contours of the heated plate model at discrete-time step $k = 90$ of the validation data. The black square markers are the sensor locations with their corresponding sensor numbers left of the markers.

The R^2 fit for the full models and reduced models of sensor 1 and sensor 5 is shown in Table 3. The table shows that the R^2 fit of the identified models is accurate. It can also be seen for the free-run simulation prediction, the reduced input models have a better R^2 fit that the full ones. This shows that in this case the full models are over-parameterized and that an input reduction results in better models.

Table 3: The R^2 fit of the full and reduced models for one-step ahead and free-run simulation predictions of the heat equation example.

Sensor #	One-step ahead		Free-run simulation	
	Full model	Reduced model	Full model	Reduced model
1	99.9807%	99.9784%	94.9872%	99.0116%
5	99.9168%	99.8016%	-15.5446%	93.2703%

Figure 8 shows the change of the mean squared one-step-ahead prediction error. It can be seen that the decrease of the signal-to-noise (SNR) ratio increases the prediction error. The figures also show that the full models have better prediction performance than the reduced ones, but the difference decreases the SNR decreases (increase of the noise level). For the full models, the error increases exponentially while for the reduced models it is relatively constant for larger SNR values and almost linearly increases for smaller ones.

It can also be seen that for a relatively narrow range of low noise level, the reduced models have better robustness than the full ones.

(a) Sensor 1: increasing noise
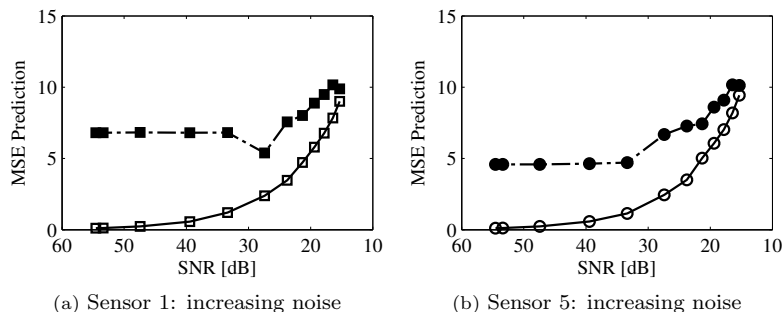
(b) Sensor 5: increasing noise

Figure 8: One-step ahead prediction error of sensors 1 and 5 for the increasing noise variance. The solid lines are the full model and the dashed lines the reduced models.

4.2. Greenhouse temperature model identification

The proposed approach is also used to identify a model based on data from a small-scale greenhouse setup shown in Figure 9. The setup was built at TNO in the Netherlands. Its length is 4.6 m, its width 2.4 m, while the height of the wall and the roof are 2.4 m and 2.9 m, respectively. Six 400 W convection heaters, each of 0.6×0.6 m, are placed on the floor of the setup. This gives an average of 200 W m^{-2} irradiance. The heaters are meant to mimic the convective effect as the absorption of solar energy by the floor during the day [46]. The coordinates of the centers of the heaters are shown in Table 4.
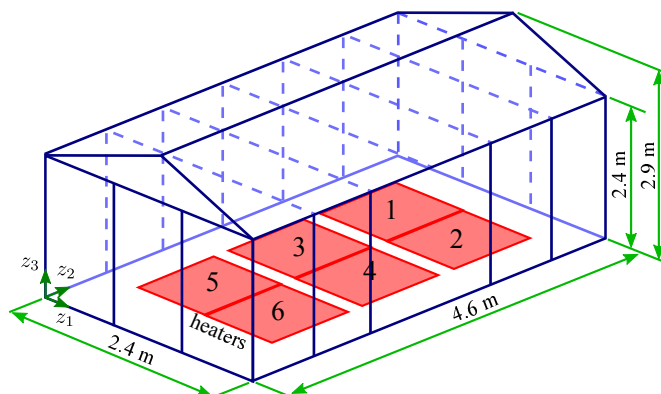
Figure 9: A schematic of the greenhouse with its physical dimensions.

The temperature measurements are collected using wireless sensors, which is a promising technology, with some applications in production greenhouses already reported [47]. For the experiments, a total of 68 sensor nodes have been installed to measure the temperature inside the greenhouse. Out of these, 45 sensor nodes are arranged on a grid with the spacing in z_1, z_2, and z_3 equal to 0.3000 m, 0.7667 m, and 0.5500 m, respectively. Additional 5 sensor nodes are placed below the roof, 6 sensor nodes are right at the center of the heaters, and 12 sensors are attached on the four walls of the greenhouse. The schematic of the sensor locations is given in Figure 10 and the photo of the setup is in Figure 11.

Throughout the identification experiments, the heaters were turned on and off in pairs: heater 1 paired with heater 4, heater 2 with heater 5, and heater 3 with heater 6 so that there are three different input signals. In total 3179 data samples have been acquired of which 2149 samples are used for identification and 1030 samples for validation. The data sets are centered by subtracting their means before the identification and model reduction with lasso are applied.
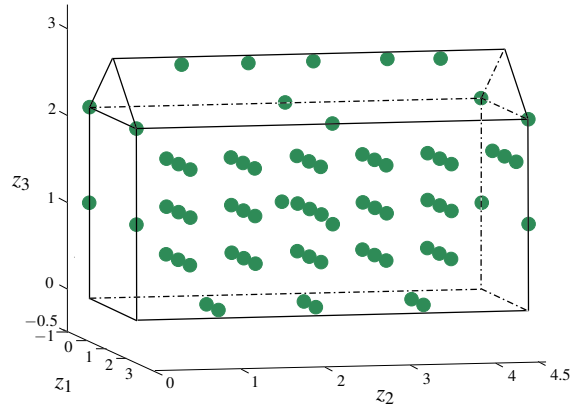
14

Figure 10: A schematic of the sensor locations in the greenhouse.

Figure 11: A photograph of the greenhouse setup used in the case study.

Table 4: The center coordinates of the convection heaters in the greenhouse

Heater #	(z_1, z_2, z_3)	Heater #	(z_1, z_2, z_3)
Heater 1	$(0.90, 3.45, 0.00)$	Heater 4	$(1.50, 3.45, 0.00)$
Heater 2	$(0.90, 2.30, 0.00)$	Heater 5	$(1.50, 2.30, 0.00)$
Heater 3	$(0.90, 1.15, 0.00)$	Heater 6	$(1.50, 1.15, 0.00)$

Table 5: Mean square error and variance of error of the full and reduced models for the greenhouse data example

	MSE		R^2 fit	
Sensor #	Full model	Reduced model	Full model	Reduced model
215	0.0310	0.0261	99.8772%	98.9155%
257	0.0229	0.0243	99.6821%	99.6515%

Among all sensors, identification results from two sensor nodes are presented: sensor node 215, located at position (1.80, 3.83, 1.10) and sensor node 257 located at (0.00, 0.00, 2.20). The neighborhood radius selected is $\varrho = 1.25$ m, which gives 19 neighbors for sensor node 215 and 7 neighbors for sensor node 257.

(a) Sensor node 215   (b) Sensor node 257

(c) Sensor node 215 error   (d) Sensor node 257 error
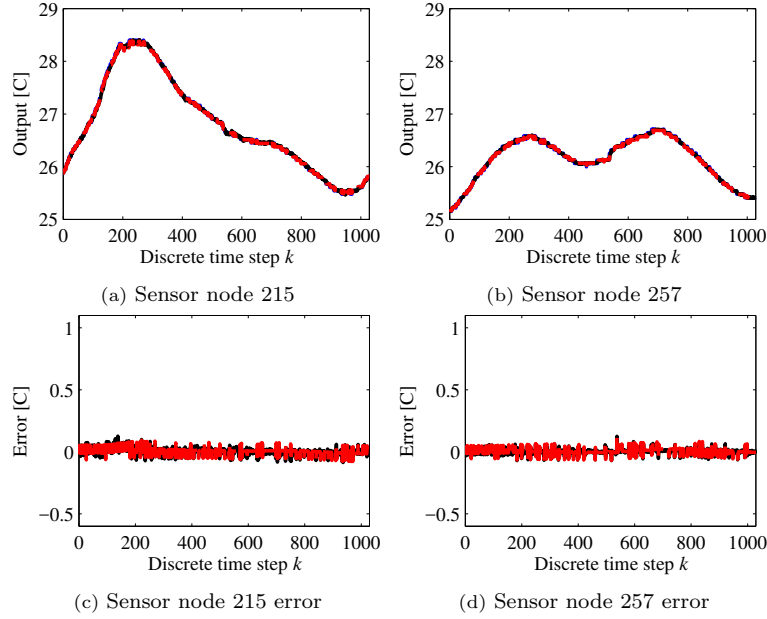
Figure 12: Greenhouse setup measurements (blue) and one-step-ahead predictions (red) for the model with full set of inputs and reduced inputs using the centered validation data set and their corresponding estimation error, i.e., error from the full model (black) and from the reduced model (red).

A 30th-order linear ARX structure is selected for the model so that initially there are 570 parameters and 280 parameters for respectively sensor node 215 and 257. The measurements, full input model, and reduced input model simulation output, and the corresponding one-step estimation errors of the validation data are shown in Figure 12. Setting the maximum number of parameters to 10, the following models are obtained:

$$
\begin{aligned}
y_{215}(k+1) = {} & 0.8241\, y_{215}(k-1) + 0.1332\, y_{215}(k-2) + 0.0065\, y_{215}(k-3) \\
& + 0.0037\, y_{215}(k-5) + 0.0041\, y_{215}(k-7) + 0.0043\, y_{215}(k-8) \\
& + 0.0113\, y_{206}(k-1) + 0.0048\, y_{218}(k-1) + 0.0027\, y_{218}(k-2) \\
& + 0.0043\, y_{218}(k-3) \\
y_{257}(k+1) = {} & 0.6206\, y_{257}(k-1) + 0.2410\, y_{257}(k-2) + 0.0093\, y_{257}(k-3) \\
& + 0.0368\, y_{257}(k-4) + 0.0092\, y_8(k-1) + 0.0480\, y_{220}(k-1) \\
& + 0.0064\, y_{234}(k-1) + 0.0198\, y_{264}(k-1) + 0.0003\, y_{20}(k-1) \\
& + 0.0036\, y_{20}(k-2)
\end{aligned}
$$

where the same as the example above, $y_i(k)$ is the measurement at sensor node i. It can be seen that the neighboring measurements contribute to the identified model. The MSE and the R^2 fit for the validation

data are shown in Table 5. For sensor 215, it can be seen that the MSE is smaller and the R^2 fit is larger for the reduced input model compared to the full model; while for sensor 257, the MSE increases slightly and the R^2 fit decreases slightly. For the case of sensor 215, the reduction of the R^2 fit suggests the full model is over-parameterized. Generally, it can be said that reducing the number of inputs in the models does not significantly decrease the performance of the models. This also indicates that the proposed identification framework works well in this example.

A set of simulations were performed to see how good the performance, in term of the one-step ahead prediction MSE, the models for different number of neighbors for sensor 238. This sensor is located about the middle of the setup and has 8 neighbors with the same height z_3. For neighbor visualization ease, the labeled sensors are shown in Figure 13. The identification is performed for 2, 4, 6, and 8 neighbors excluding sensor 238 itself. The performance of the full and the reduced models are compared for the validation data. The neighbors and the performance comparison are shown in Table 6. From the table, it can be seen that the one-step ahead prediction errors differ insignificantly for different number of neighbors. This shows the proposed framework does not sensitive to the number of neighboring sensors.
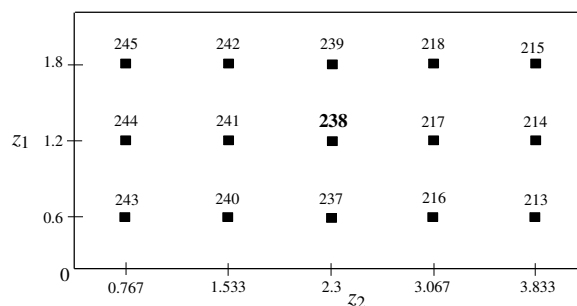

Figure 13: Sensors at $z_3 = 1.1$ with sensor id labels.

Table 6: The coordinates of sensor 238, its neighbors, and its performance for different numbers of neighboring sensors. The X symbol means the sensor is used as neighbors

Sensor #	Number of neighbors							
	2		4		6		8	
213		X		X	X		X	
214		X			X		X	X
215				X		X		X
216							X	
217				X			X	X
218					X			X
237					X			
239					X			
240	X		X		X			X
241	X		X	X	X		X	X
242				X	X		X	
243			X			X		X
244			X		X		X	
245							X	X
MSE full	0.0215	0.0215	0.0216	0.0211	0.0220	0.0208	0.0220	0.0222
MSE red	0.0236	0.0230	0.0238	0.0233	0.0235	0.0239	0.0235	0.0239

An experiment to estimate values at locations that are not measured is also performed for sensors shown in Figure 13. In this experiment, data from sensor 217, 238, and 241 are not identified and their estimates for the validation data are obtained by using the ordinary kriging. The experiment is performed for both full and reduced models. The kriging models are developed by using the estimates of the validation data of

the rest of the sensors. The results are shown in Figure 14 for the estimates and their corresponding error respectively. The experiment is repeated by omitting sensor 216, 217, 218, 240, 241, and 242. The estimates are shown in Figure 15 and their corresponding error in Figure 16.

The figures show that the ordinary kriging estimates the values at locations that are not measured sufficiently well. Furthermore, estimate differences between the full and the reduced models are not significant. For the second experiment, it can be seen that the kriging estimates for sensor 216, 217, and 218 look similar; and so are those for sensor 240, 241, and 242. This is can be explained by looking at the validation data from sensor 216, 217, and 218 plotted as a group in Figure 17a and those from sensor 249, 241, and 242 as the other group in Section 4.2. From the figure, it can be seen that the temperature difference within a group is small and this creates kriging estimates with insignificant difference among them.

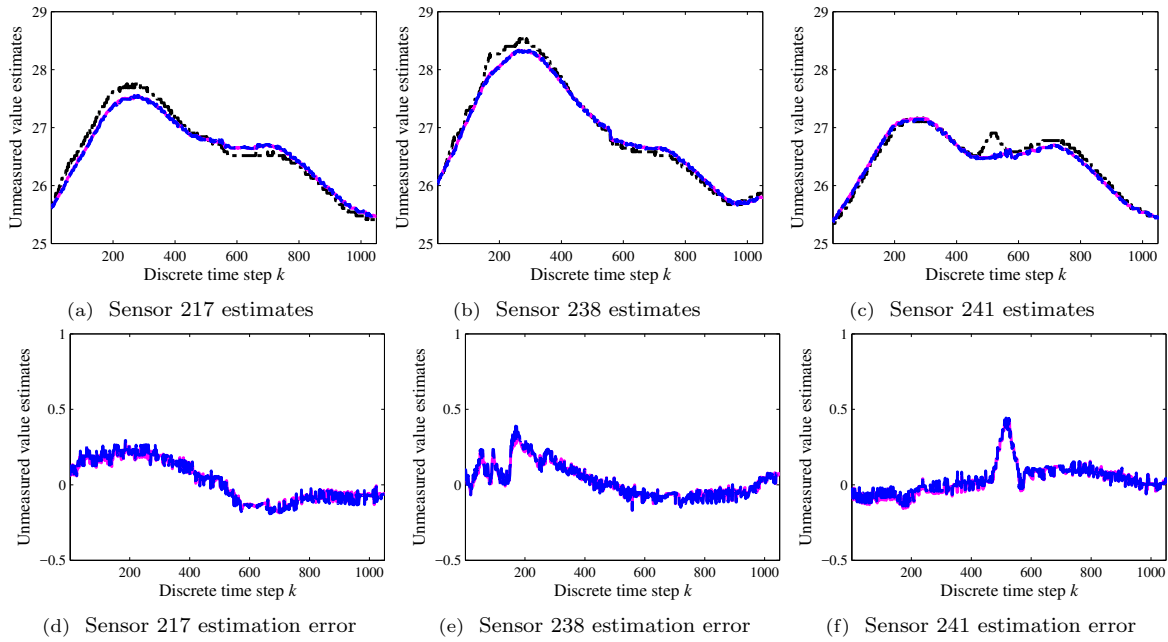| (a) Sensor 217 estimates	(b) Sensor 238 estimates	(c) Sensor 241 estimates
(d) Sensor 217 estimation error	(e) Sensor 238 estimation error	(f) Sensor 241 estimation error

Figure 14: Validation data estimates for sensor 216, 238, and 241 by using the ordinary kriging and their corresponding error. For (a), (b), and (c), black plots are the validation data, magenta plots are estimates from the full models, and blue plots are estimates from the reduced models. For (d), (e), and (f), magenta plots are errors from the full models and blue plots are errors from the reduced models.

Contour plots of the greenhouse temperature for $0.6 \geq z_1 \geq 1.8$, $0.767 \geq z_2 \geq 3.833$, and fixed $z_3 = 1.1$ are shown in Figure 18. The plots are in 2D because the ooDACE toolbox is only able to build kriging model from 2D data. The same as in the heated plate example, the plots show the contour of the validation data, the one-step ahead prediction of the full and reduced models. It can be seen that the interpolation is larger with the reduced models than that of the full model.

5. Conclusions and future research

In this paper, a methodology for the identification of distributed-parameter systems was presented. The methodology is a finite-difference based method that takes into account inputs from neighboring measurements and actuators into the model. The methodology assumes that the underlying partial differential equation is not known. Although a finite-difference based method is proposed, the methodology does not require dense measurement locations in the system. This feature allows the applicability of the methodology to real-life systems, which generally have a limited number of measurements. Model reduction methods may be applied to reduce the complexity of the model in case a large number of inputs are involved in the model.
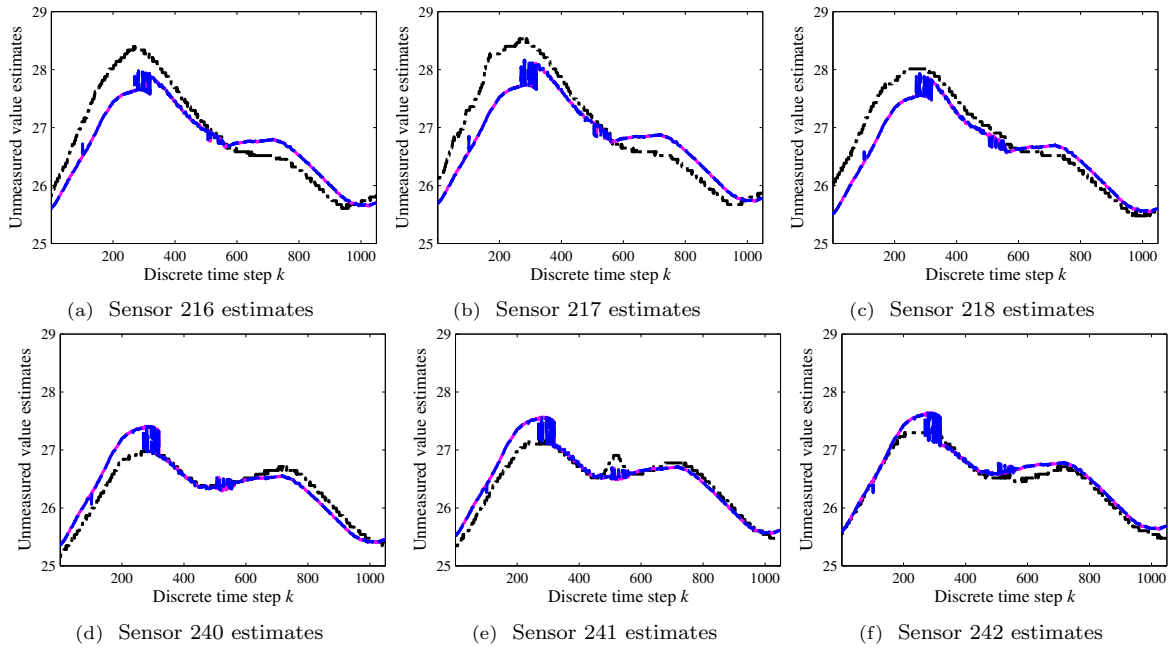
18

Figure 15: Validation data estimates for sensor 216, 217, 218, 240, 241, and 242 by using the ordinary kriging. Black plots are the validation data, magenta plots are estimates from the full models, and blue plots are estimates from the reduced models.
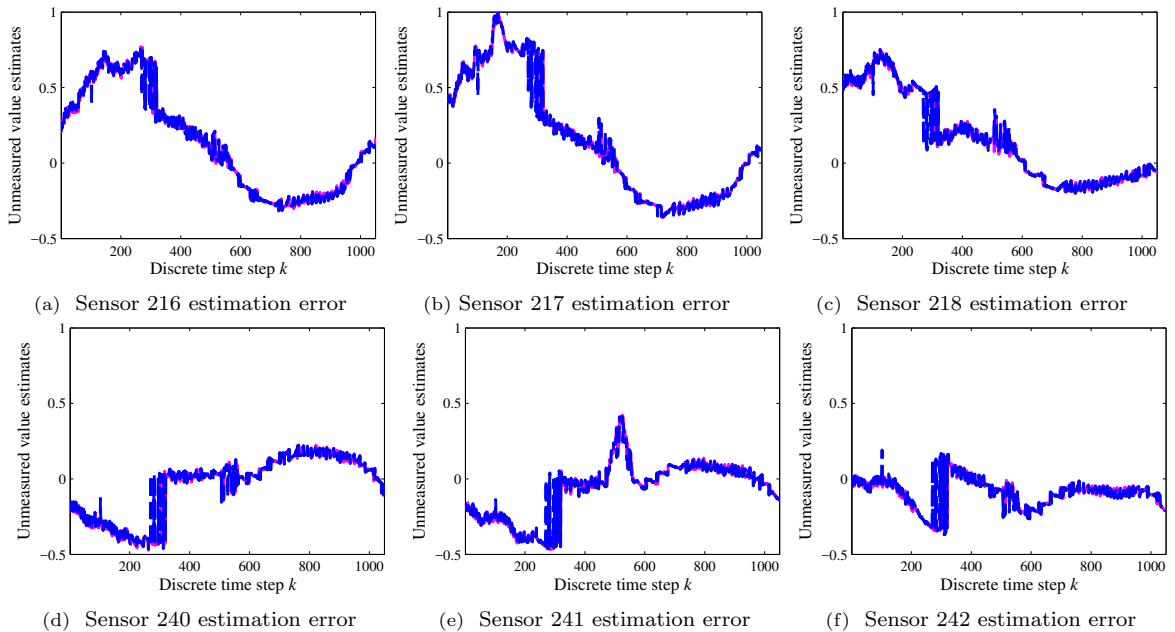

Figure 16: Validation data estimation error for sensor 216, 217, 218, 240, 241, and 242 by using the ordinary kriging. Magenta plots are errors from the full models and blue plots are errors from the reduced models.
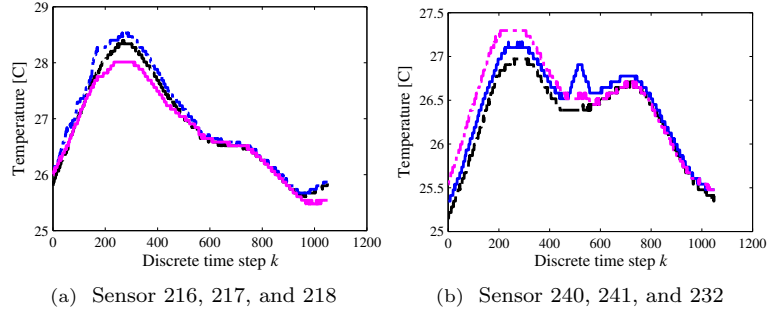
(a) Sensor 216, 217, and 218

(b) Sensor 240, 241, and 232

Figure 17: Validation data plot from sensor: (a) 216 in black, 217 in blue), 218 in magenta (b) 240 in black, 241 in blue, and 243 in magenta
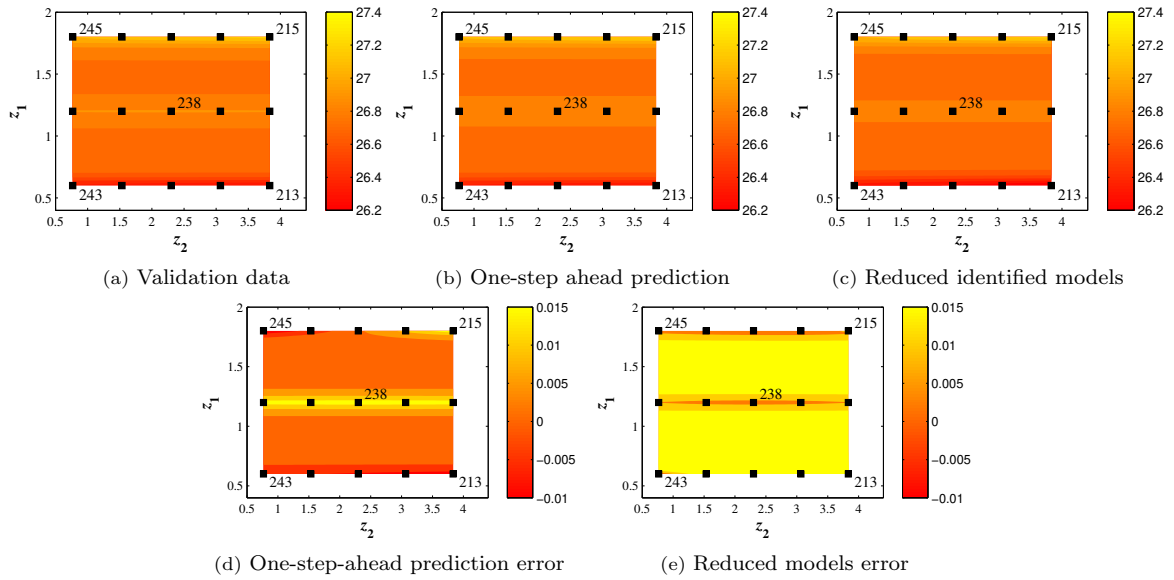

(a) Validation data

(b) One-step ahead prediction

(c) Reduced identified models

(d) One-step-ahead prediction error

(e) Reduced models error

Figure 18: Contours of the greenhouse temperature model at discrete-time step $k = 400$ of the validation data for $0.6 \geq z_1 \geq 1.8, 0.767 \geq z_2 \geq 3.833$ and fixed $z_3 = 1.1$. The black square markers are the sensor locations and the labeled sensors are used to build the kriging model.

The effectiveness of the methodology has been shown with the help of two examples, a simulated heated plate and a real greenhouse.

There are several open problems related to the proposed methodology. The first one is how to use the identified model to design a controller or an observer. Models from each sensor can be stacked to form a state space representation, where the measurements at sensor locations represent the states of the system. From the fact that the states are coupled across different measurement locations, the question is how straightforward it is to apply the available control design methods for the identified model. The second open problem is about optimal sensor location. In the literature, techniques have been proposed to place sensors for a distributed-parameter system given a certain partial differential model [38]. An extension to handle an unknown or partially known model structure may increase the applicability of the proposed methodology. The third open problem is the choice of the neighbors. Selecting the right neighbors helps to reduce the computational effort to solve the identification problem. For example, for the greenhouse the neighbor selection is important in case the influence of air flow dynamics inside the modeled chamber cannot be neglected. This leads to the fourth open problem, namely, how to apply the methodology online in case dynamic neighbor selection is required to handle the air flow dynamics. Finally, further research will be focused on the extension of the methodology to nonlinear distributed-parameter systems.

Acknowledgment

References

[1] C. Doumanidis, N. Fourligkas, Temperature distribution control in scanned thermal processing of thin circular parts, IEEE Transactions on Control Systems Technology 9 (5) (2001) 708–717.

[2] P. D. Christofides, Robust control of parabolic PDE systems, Chemical Engineering Science 53 (16) (1998) 2949–2965.

[3] D. Edouard, D. Schweich, H. Hammouri, Observer design for reverse flow reactor, AIChE Journal 50 (9) (2004) 2155–2166.

[4] A. Vande Wouwer, C. Renotte, I. Queinnec, P. Bogaerts, Transient analysis of a wastewater treatment biofilter: Distributed parameter modelling and state estimation, Mathematical and Computer Modelling of Dynamical Systems: Methods, Tools and Applications in Engineering and Related Sciences 12 (5) (2006) 423–440.

[5] M. Krstić, R. Vazquez, A. A. Siranosian, M. Bement, Sensing schemes for state estimation in turbulent flows and flexible structures, Smart Structures and Materials 2006: Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 6174 (1) (2006) 61740U.

[6] D. Vries, K. J. Keesman, H. Zwart, A Luenberger observer for an infinite dimensional bilinear system: A UV disinfection example, in: Proceedings of the 3rd IFAC Symposium on System, Structure and Control, Foz do Iguassu, Brazil, 2007, pp. –.

[7] R. Klein, N. Chaturvedi, J. Christensen, J. Ahmed, R. Findeisen, A. Kojic, State estimation of a reduced electrochemical model of a lithium-ion battery, in: Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 2010, pp. 6618–6623.

[8] P. Dufour, P. Laurent, C. Xu, Observer based model predictive control of the water based painting drying using a humidity profile soft sensor and a temperature measurement, in: Proceedings of the 1st International Workshop and Symposium on Industrial Drying (IWSID), Mumbay, India, 2004, pp. –, paper SY152.

[9] H.-X. Li, C. Qi, Modeling of distributed parameter systems for applications – a synthesized review from time-space separation, Journal of Process Control 20 (8) (2010) 891–901.

[10] C. Kravaris, J. H. Seinfeld, Identification of spatially varying parameters in distributed parameter systems by discrete regularization, Journal of Mathematical Analysis and Applications 119 (1-2) (1986) 128–152.

[11] J.-L. Lions, Some aspects of modelling problems in distributed parameter systems, in: P. D. A. Ruberti (Ed.), Distributed Parameter Systems: Modelling and Identification, no. 1 in Lecture Notes in Control and Information Sciences, Springer Berlin Heidelberg, 1978, pp. 11–41.

[12] C. S. Kubrusly, Distributed parameter system identification: A survey, International Journal of Control 26 (4) (1977) 509–535.

[13] N. Cressie, C. K. Wikle, Statistics for Spatio-Temporal Data, John Wiley & Sons, USA, 2011.

[14] S. A. Billings., Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, John Wiley & Sons, 2013.

[15] H. Park, D. Cho, The use of the Karhunen-Loève decomposition for the modeling of distributed parameter systems, Chemical Engineering Science 51 (1) (1996) 81–98.

[16] X.-G. Zhou, L.-H. Liu, Y.-C. Dai, W.-K. Yuan, J. Hudson, Modeling of a fixed-bed reactor using the K-L expansion and neural networks, Chemical Engineering Science 51 (10) (1996) 2179–2188.

[17] D. Coca, S. A. Billings, Direct parameter identification of distributed parameter systems, International Journal of Systems Science 31 (1) (2000) 11–17.

[18] S. Mandelj, I. Grabec, E. Govekar, Statistical approach to modeling of spatiotemporal dynamics, International Journal of Bifurcation and Chaos 11 (11) (2001) 2731–2738.

[19] D. Zheng, K. A. Hoo, M. J. Piovoso, Low-order model identification of distributed parameter systems by a combination of singular value decomposition and the Karhunen-Loève expansion, Industrial & Engineering Chemistry Research 41 (6) (2002) 1545–1556.

[20] Y. Pan, S. A. Billings, The identification of complex spatiotemporal patterns using coupled map lattice models, International Journal of Bifurcation and Chaos (IJBC) in Applied Sciences and Engineering 18 (4) (2008) 997–1013.

[21] L. Ljung, System Identification: Theory for the User, 2nd Edition, Prentice Hall, USA, 1999.

[22] R. J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady State and Time Dependent Problems, SIAM, Philadelphia, PA, USA, 2007.

[23] Å. Björck, Numerical Methods for Least Squares Problems, SIAM, USA, 1996.

[24] T. Söderström, Errors-in-variables methods in system identification, Automatica 43 (6) (2007) 939–958.

[25] S. Van Huffel, J. Vandewalle, The Total Least Squares Problem: Computational Aspects and Analysis, SIAM, Philadelphia, PA, USA, 1991.

[26] L. Zhou, X. Li, F. Pan, Gradient based iterative parameter identification for Wiener nonlinear systems, Applied Mathematical Modelling 37 (16–17) (2013) 8203–8209.

[27] F. Ding, Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling, Applied Mathematical Modelling 37 (4) (2013) 1694–1704.

[28] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics 15 (1) (1985) 116–132.

[29] K. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions on Neural Networks 1 (1) (1990) 4–27.

[30] N. R. Draper, H. Smith, Applied Regression Analysis, 3rd Edition, Wiley-Interscience, USA, 1998.

[31] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society. Series B (Methodological) 58 (1) (1996) 267–288.

[32] S. Chen, S. A. Billings, W. Luo, Orthogonal least squares methods and their application to non-linear system identification, International Journal of Control 50 (5) (1989) 1873–1896.

[33] E. Mendez, S. Billings, An alternative solution to the model structure selection problem, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 31 (6) (2001) 597–608.

[34] G. P. Liu, V. Kadirkamanathan, S. A. Billings, Neural network-based variable structure control for nonlinear discrete systems, International Journal of Systems Science 30 (10) (1999) 1153–1160.

[35] D. Erdirencelebi, S. Yalpir, Adaptive network fuzzy inference system modeling for the input selection and prediction of anaerobic digestion effluent quality, Applied Mathematical Modelling 35 (8) (2011) 3821–3832.

[36] R. Šindelář, R. Babuška, Input selection for nonlinear regression models, IEEE Transactions on Fuzzy Systems 12 (5) (2004) 688–696.

[37] C. Kubrusly, H. Malebranche, Sensors and controllers location in distributed systems – a survey, Automatica 21 (2) (1985) 117–128.

[38] D. Uciński, Optimal Measurement Methods for Distributed Parameter System Identification, CRC Press, USA, 2004.

[39] J. A. Duan, A. E. Gelfand, C. F. Sirmans, Modeling space-time data using stochastic differential equations, Bayesian Analysis 4 (4) (2009) 733–758.

[40] C. K. Wikle, M. B. Hooten, A general science-based framework for dynamical spatio-temporal models, TEST 19 (3) (2010) 417–451.

[41] A. S. Goldberger, Best linear unbiased prediction in the generalized linear regression model, Journal of the American Statistical Association 57 (298) (1962) 369–375.

[42] J. Cortes, Distributed kriged kalman filter for spatial estimation, IEEE Transactions on Automatic Control 54 (12) (2009) 2816–2827.

[43] J. P. Holman, Heat Transfer, 10th Edition, McGraw-Hill Higher Education, New York, NY, USA, 2009.

[44] I. Couckuyt, F. Declercq, T. Dhaene, H. Rogier, L. Knockaert, Surrogate-based infill optimization applied to electromagnetic problems, International Journal of RF and Microwave Computer-Aided Engineering 20 (5) (2010) 492–501.

[45] I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, T. Dhaene, Blind kriging: Implementation and performance analysis, Advances in Engineering Software 49 (2012) 1–13.

[46] T. Boulard, G. Papadakis, C. Kittas, M. Mermier, Air flow and associated sensible heat exchanges in a naturally ventilated greenhouse, Agricultural and Forest Meteorology 88 (1–4) (1997) 111–119.

[47] N. Wang, N. Zhang, M. Wang, Wireless sensors in agriculture and food industry–recent development and future perspective, Computers and Electronics in Agriculture 50 (1) (2006) 1–14.