Refined IV-based method for LPV partial differential equation model identification
Julien Schorsch, Vincent Laurain, Marion Gilson, Hugues Garnier

To cite this version:

HAL Id: hal-01058618
https://hal.archives-ouvertes.fr/hal-01058618
Submitted on 27 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Refined IV-based method for LPV partial differential equation model identification

Julien Schorsch\textsuperscript{1,2}, Vincent Laurain\textsuperscript{1,2}, Marion Gilson\textsuperscript{1,2} and Hugues Garnier\textsuperscript{1,2}

Abstract—This paper presents a direct identification method for linear parameter varying models described by partial differential equations in an input-output setting. The continuous space-time model is first rewritten as a multiple-input single-output model. The continuous filtering operations are reformulated as a discrete convolution product and a refined instrumental variable technique is developed to efficiently estimate the model parameters. The performance of the proposed method is then illustrated via a representative simulation example.

I. INTRODUCTION

The linear parameter varying (LPV) partial differential equation (PDE) identification case is considered in this paper where the parameters are assumed to be function of a measurable deterministic time-space varying signal. This modeling concept allows for a wide representation of physical processes.

Several identification methods for PDE models are available in the literature (see e.g. [6] for survey) and two main approaches can be distinguished. The indirect approach including two steps involves the transformation of the continuous time-space (CTS) model into a discrete time-space (DTS) model and the identification of the DTS model parameters [1]. The direct approach aims to estimate the coefficients of the CTS model directly from sampled signals ([4] and [12]). However, only a few papers deal with the PDE identification problem into the LPV framework. The use of LPV models has been applied to the identification of linear time-varying (LTV) PDE system by approximating the PDE parameters. In the ODE model case, this filter follows the same dynamic as the system. However, in the LPV-PDE case, the filter should be dependent on an external variable \( \rho \), if it is chosen to follow the same dynamics as the system. The influence of this dependency is twofold. First, this dependency is injected on the filtered noise which decreases the quality of the estimated parameters. Secondly, the classical algebraic properties (commutativity, associativity,...) are not guaranteed. The proposed solution is based on the reformulation of the single-input single-output (SISO) LPV-PDE model to a multiple-input single-output (MISO) PDE model. From this reformulation a new filter is emphasized. It is not anymore dependent on \( \rho \), keeps algebraic properties and follows a dynamic part of the considered system.

However, the resulting filter is written in terms of partial derivative operators and therefore it cannot be directly defined as a rational transfer function. Usually, the transfer functions associated with PDE models are irrational functions which involve an infinite number of poles and zeros [5]. Nevertheless, the solution developed here, is to discretize the filter by using classical numerical schemes and to write the filter operations as a convolution product.

The paper is organized as follows: in Section 2, the general class of CTS-LPV-PDE models is introduced. The reformulation of the system is provided Section 3 including a description of the proposed filter. Section 4 deals with the writing of this filter by using the discretization of the differential operator. A description of the proposed estimator is given in Section 5. Finally, an numerical example is used in Section 6 to illustrate the performance of the proposed approach.

II. PROBLEM FORMULATION

A. Deterministic model

Let \( \Omega \) a close set and consider the continuous model described by the following PDE:

\[
\begin{align*}
A(\partial, \rho)\chi(t, x) &= B(\partial, \rho)u(t, x) \quad \text{on } \Omega \times [0, +\infty[
\chi(0, x) &= \phi(x) \quad \text{on } \Omega
\end{align*}
\]

(1)

where \( x \in \Omega \) is the space variable, \( \rho \) is some parameter, \( \chi(t, x) \) is the model output, \( u(t, x) \) is the input, \( \phi(x) \) is a continuous space function which...
represents the model behaviour for $t = 0$. For $(i_t, i_x) \in \mathbb{N}^2$, $\partial^{i_t, i_x}$ denotes the partial differential operator:

$$\partial^{i_t, i_x} \chi(t, x) = \partial^{i_t} \partial^{i_x} \chi(t, x) = \frac{\partial^{i_t+i_x} \chi(t, x)}{\partial t^{i_t} \partial x^{i_x}},$$

(2)

where $\partial^{i_t} = \frac{\partial^{i_t}}{\partial t^{i_t}}$ is the $i_t$-th time partial derivative and $\partial^{i_x} = \frac{\partial^{i_x}}{\partial x^{i_x}}$ is the $i_x$-th space partial derivative. To complete Equation (1) boundary conditions needs to be specified. A large number of boundary conditions can be chosen, depending on the problem formulation, the number of variables involved and the nature of the equation.

$A(\partial, \rho)$ and $B(\partial, \rho)$ are bivariate polynomials in $\partial$ of degrees $n_t, n_x$ and $m_t, m_x$ respectively:

$$A(\partial, \rho) = p^{n_t, n_x} + \sum_{i_t, i_x=0}^{n_t, n_x} a(\rho)_{i_t, i_x} \partial^{i_t, i_x}$$

$$B(\partial, \rho) = \sum_{j_t, j_x=0}^{m_t, m_x} b(\rho)_{j_t, j_x} \partial^{j_t, j_x}$$

(3a, b)

In order to represent a larger behavior range, the model is assumed to be parameter varying. $a_{i_t, i_x}(\rho)$ and $b_{j_t, j_x}(\rho)$ are analytic functions with dependence on an external deterministic scheduling variable $\rho(t, x) = \rho$, which is called static dependence. $(n_t, n_x)$ are the orders of the system (assumed to be known). In this work, it is assumed that $n_x \geq m_x$, $m_t$; $n_t \geq m_t, m_x$; and $A(\partial, \rho)$ is a monic polynomial in $\partial_t$ variable.

B. Process model

Assuming $\rho$ a priori known, the $A(\partial, \rho)$ and $B(\partial, \rho)$ polynomials can be parametrized as:

$$A(\partial, \rho, \theta) = \partial^{n_t, n_x} + \sum_{i_t, i_x=0}^{n_t, n_x} a_{i_t, i_x}(\rho) \partial^{i_t, i_x}$$

$$\begin{align*}
    a_{i_t, i_x}(\rho) &= a^{0}_{i_t, i_x} + \sum_{i_t=1}^{n_t} a^{i_t}_{i_t, i_x} f_i(ho) \\
    &\quad i_t = 0, \ldots, n_t \quad i_x = 0, \ldots, n_x \\

    B(\partial, \rho, \theta) &= \sum_{j_t, j_x=0}^{m_t, m_x} b_{j_t, j_x}(\rho) \partial^{j_t, j_x} \\
    b_{j_t, j_x}(\rho) &= b^{0}_{j_t, j_x} + \sum_{j_t=1}^{m_t} b^{j_t}_{j_t, j_x} g_j(\rho) \\
    &\quad j_t = 0, \ldots, m_t \quad j_x = 0, \ldots, m_x.
\end{align*}$$

(4, 5)

In this parametrization, $\{f_i\}_{i=1}^{n_t}$ and $\{g_j\}_{j=1}^{m_t}$ are meromorphic functions of $\rho$, with static dependence. They can be chosen for example as linearly independent functions. The associated model parameters $\theta$ are stacked columnwise:

$$\theta = [a_{0,0} \ldots a_{n_t, n_x-1} b_{0,0} \ldots b_{m_t, m_x}]^T \in \mathbb{R}^{n_\theta}$$

(6)

where

$$a_{i_t, i_x} = [a^{0}_{i_t, i_x} a^{1}_{i_t, i_x} \ldots a^{n_t}_{i_t, i_x}] \in \mathbb{R}^{n_\theta+1}$$

(7)

and $n_\theta = ((n_t+1)(n_x+1)-1)(n_a+1) + (m_t+1)(m_x+1)$.

C. Noisy model

In this paper, stochastic PDE class is considered. These systems are a generalization of PDE taking into account a random noise term. Thus due to measurement inaccuracies, the measure $\chi(t, x)$ is not available but a disturbed version of it is obtained.

Moreover, in practical situations, such as those encountered in environmental science, the measured data are not continuous in time and in space. Given the sampled nature of the data, an obvious assumption is that the sampled output is contaminated by a discrete time-space noise.

The data-generating system takes the following Output-Error structure:

$$\begin{align*}
    A(\partial, \rho, \theta) \chi(t, x) &= B(\partial, \rho, \theta) u(t, x) \quad \text{on } \Omega \times [0, T] \\
    \chi(0, x) &= \phi(x) \quad \text{on } \Omega \\
    y(t_k, x_\ell) &= \chi(t_k, x_\ell) + e(t_k, x_\ell)
\end{align*}$$

(9)

where the observed noisy output $y(t_k, x_\ell)$ and $\chi(t_k, x_\ell)$ are the sampled signals at the time $t_k$ and the observation point $x_\ell$ for $k \in \{1, \ldots, N\}$ and $\ell \in \{1, \ldots, L\}$, $e(t_k, x_\ell)$ is a white noise, i.e. a zero mean, random spatial array, with no temporal or spatial correlation.

D. Objective

Finally, the objective is to estimate the parameter vector (6) of the model (9) based on the measured sampled input, output data and the scheduling variables $Z = \{u(t_k, x_\ell), y(t_k, x_\ell), \rho(t_k, x_\ell)\}_{k=1, k=1}^{L, N}$.

III. THE LPV-PDE REFORMULATION

A. Reformulation

The LPV-PDE model identification problem cannot be formulated directly as a prediction error minimisation because of the non-commutativity of the LPV polynomials. This problem is well-known when dealing with LPV models. In [7] (for ODE models) and [13] (for PDE models), the proposed solution is to rewrite the LPV model as a MISO LTI model for which the PEM can be formulated. The idea here is to follow the same concept and to reformulate the LPV model in order to express it as a linear regression form:

$$y^{(n_t, n_x)}(t_k, x_\ell) = \varphi(t_k, x_\ell)\theta + F(\partial, \theta)e(t_k, x_\ell)$$

(10)
Such a reformulation can be expressed as:
\[
\hat{\chi}^{(n_s,n_x)}(t_k,x_t) + \sum_{i_s,i_x=0}^{n_s,n_x} a_{i_s,i_x}^{(1)} \hat{\chi}^{(i_s,i_x)}(t_k,x_t)
\]
\[
\dot{\chi}^{(i_s,i_x)}(t_k,x_t) = F(\partial, \theta) \chi^{(i_s,i_x)}(t_k,x_t) + \sum_{i_s,i_x=0}^{n_s,n_x} a_{i_s,i_x}^{(1)} \dot{\chi}^{(i_s,i_x)}(t_k,x_t)
\]
\[
\chi^{(i_s,i_x)}(t_k,x_t) = \chi_0^{(i_s,i_x)}(t_k, x_t)
\]

where \( y_i(t_k, x_t) = \hat{\chi}^{(i_s,i_x)}(t_k,x_t) \) and \( u_i^{(j_s,j_x)}(t_k, x_t) \) represent the outputs of a filtering operation.

\[
y_i(t_k, x_t) = \frac{1}{F(\partial, \theta)} y_i(t_k, x_t)
\]

and

\[
\varphi_i(t_k, x_t) = \left[ -y_i^{(0,0)}(t_k, x_t), \ldots, -y_i^{(n_s, n_x-1)}(t_k, x_t) \right]^{T}
\]

Moreover, the proposed filter does not depend on the deterministic scheduling variable \( \rho \) but only on the LTI part of the PDE. However, since the inverse of \( \partial \) is not defined in the partial derivative context, the digital implementation of the filtering operation is discussed in the next Section.

### IV. THE DIFFERENTIAL OPERATOR DISCRETIZATION

In contrary to ODE model which can be expressed by a transfer function model between the excitation signal and output signal, it is more complicated to write PDE under this form. However, for specific linear PDE systems, several approaches are possible. As an example, a method developed in [5] consists in using the Laplace transforms with respect to the time variable \( t \) and then to apply it on the system and the boundary conditions. PDE are then written as irrational transfer functions which can be expanded into infinite dimensional rational transfer functions. A different solution is to use fractional models to describe the PDE system (see e.g. [9]). Another method consists in taking the Fourier transforms with respect to the space variable \( x \). This method is used when the space domain is assumed to be infinite and permits to obtain an analytic solution of the PDE.

The method used here, to tackle the irrational function problem, is to consider a discretization along the dimension axis, leading to a so-called parameter distributed system. This approach consists in reducing the infinite dimensional system described by PDE to a finite dimension model, using the input and output operations to construct an approximation of the solution. The continuous filter \( F(\partial, \theta) \) is then transformed into a discrete filter \( F(\bar{\partial}, \bar{\theta}) \), where \( \bar{\theta} \) is approximated by a discretized operator \( \bar{\partial} = (\bar{\partial}_t, \bar{\partial}_x) \):

\[
\begin{aligned}
\bar{\partial}_t &= \bar{\partial}_x + \mathcal{O}(T_e) \\
\bar{\partial}_x &= \bar{\partial}_x + \mathcal{O}(h)
\end{aligned}
\]

where \( T_e \) and \( h \) are the sampling periods in time and space and \( \mathcal{O}(\cdot) \) is the truncation error. This discrete filter can be expressed as a convolution product

\[
[F(\bar{\partial}, \bar{\theta}) \hat{\chi}](t_k, x_t) = (f * \hat{\chi})(t_k, x_t)
\]
which can be written under the following transfer function:

$$\dot{\chi}(t,x) = \frac{q_x^{-1} - 1}{T_s} \chi(t,x)$$

(22)

where \( q_x \) is the shift operator. In the same way, \( \partial^0.2 \dot{\chi}(t,x) \) can be approximated by the central difference approximation:

$$[\partial^0.2 \chi](t_k,x_\ell) = q_x^{-1} - 2 + q_x^{-1} \chi(t_k,x_\ell).$$

(23)

where \( q_x \) is the shift operator. Equation (21) can then be expressed into the following discrete form

$$(\partial^1.0 - \partial^0.2) \chi(t_k,x_\ell) = u(t_k,x_\ell),$$

(24)

which can be written under the following transfer function:

\[
G(q_t,q_x) = \left\{ \begin{array}{l}
q_t^{-1} + \frac{T_s}{h^2} q_x^{-1} + (1 - \frac{2T_s}{h^2}) q_t^0 q_x^0 + \frac{T_s}{h^2} q_t^0 q_x^{-1} \\
\chi(t_k,x_\ell) = G(q_t,q_x) u(t_k,x_\ell)
\end{array} \right.
\]

(25)

The stability of this transfer function can be proven by the classical methods used for studying the convergence of the numerical scheme [10]. Here the transfer function \( G(q_t,q_x) \) is stable when

$$-1 + \frac{2T_s}{h^2} \leq 0.$$  

(26)

V. ESTIMATION

Under the assumption of a priori known \( F(\bar{\sigma},\theta) \) and \( \tilde{\chi} \), the optimal estimator in the prediction error minimization sense is the least-squares (LS) estimator:

$$\hat{\theta}_{LS} = \left[ \sum_{k,\ell=1}^{N,L} \varphi_\ell(t_k,x_\ell) \varphi_\ell^T(t_k,x_\ell) \right]^{-1} \cdot \left[ \sum_{k,\ell=1}^{N,L} \varphi_\ell(t_k,x_\ell) y^{(n_x,n_x-1)}_\ell(t_k,x_\ell) \right].$$

(27)

In practice, the filters \( F(\bar{\sigma},\theta) \) and the signals \( \tilde{\chi}^{(n_x,n_x)} \) are unknown and therefore an iterative procedure is proposed in order to cope with this issue, similarly as it is achieved in the approaches developed for the ODE case in [8] and [13] for the PDE case. However, the LS method delivers biased estimates in the general practical situation where the noise structure is not known. It is then necessary to develop estimators which give unbiased estimates whatever the noise. In this paper, the proposed estimator is based on the IV method which is known to provide efficient estimates.

A. Basic IV estimator

The main idea behind the IV method is to modify the normal equations (27) so that they yield consistent estimates for arbitrary additive noise. This involves the introduction of an IV vector \( \zeta_\ell(t_\ell,x_\ell) \) whose elements are correlated with the noise-free regression vector \( \hat{\varphi}_\ell(t_\ell,x_\ell) \), where

$$\hat{\varphi}_\ell(t_\ell,x_\ell) = F(\bar{\sigma},\theta) \varphi(t_\ell,x_\ell)$$

(28)

with

$$\varphi(t_\ell,x_\ell) = (\chi^{(0,0)}(t_\ell,x_\ell), \ldots, \chi^{(m,n_x-1)}(t_\ell,x_\ell) \chi^{(0,0)}(t_\ell,x_\ell), \ldots, \chi^{(n_x,n_x-1)}(t_\ell,x_\ell)) \text{T},$$

(29)

but uncorrelated with the additive noise. The basic IV parameter estimates are then given by the following IV normal equations:

$$\hat{\theta}_{IV} = \left[ \sum_{k,\ell=1}^{N,L} \zeta_{\ell}(t_\ell,x_\ell) \varphi_{\ell}^T(t_k,x_\ell) \right]^{-1} \cdot \left[ \sum_{k,\ell=1}^{N,L} \zeta_{\ell}(t_\ell,x_\ell) y^{(n_x,n_x)}_\ell(t_k,x_\ell) \right]$$

(30)

provided that the matrix-inverse exists.

B. Consistency properties

By introducing (16) into (30) it comes:

$$\hat{\theta}_{IV} = \theta + \left[ \sum_{k,\ell=1}^{N,L} \zeta_{\ell}(t_k,x_\ell) \varphi_{\ell}^T(t_k,x_k) \right]^{-1} \cdot \left[ \sum_{k,\ell=1}^{N,L} \zeta_{\ell}(t_k,x_\ell) e_{\ell}(t_k,x_\ell) \right]$$

(31)

It can be deduced from (31) that \( \hat{\theta}_{IV} \) is a consistent estimate of \( \theta \) if

$$\left\{ \begin{array}{l}
E[\zeta_{\ell}(t_k,x_\ell) \varphi_{\ell}^T(t_k,x_k)] \text{ is not singular} \quad (32a) \\
E[\zeta_{\ell}(t_k,x_\ell) e_{\ell}(t_k,x_k)] = 0 \quad (32b)
\end{array} \right.$$

where \( E(.) \) denotes the mathematical expectation.

C. Refined Instrumental Variable algorithm

Theoretically, the IV method corrects the statistical inconsistency of the least squares solution but its practical implementation requires some method for generating an instrument that satisfies the conditions given by (32a) and (32b). Several different procedures for the ODE models have been suggested in the system and control literature. The most successful of these is the statistically optimal SRIVC algorithm which involves an iterative procedure where at each iteration, the auxiliary model used to generate the IV, as well as the associated filters are updated based on the parameter estimates at the previous iteration.

The proposed LPV-SRIVC-PDE model identification algorithm can be summarized as shown below. The algorithm consists of three first steps for the initialization and the
last five iterative steps are used to estimate the filter, the
derivatives and finally the model parameters.

- **Step 1** Initialization of $\theta^0$ for $l = 0$ only:

$$
\hat{\theta}^0 = \left[ a^0_{0,0} \cdots a^0_{n_t, n_x-1} 0 \cdots 0 \hat{\theta}^0_{m_t, m_x} \cdots 0 \right]^T
$$

using for instance the LS-PDE method proposed in [12]. In this first step the system is considered as a LTI system.

- **Step 2** Compute a first estimated filter on the basis of the estimates obtained in Step 1

$$
Q(\hat{\theta}, \hat{\theta}^0) = \frac{1}{F(\hat{\theta}, \hat{\theta}^0)}
$$

where $F(\hat{\theta}, \hat{\theta}^0)$ is given as (12).

Use the filter in order to generate the estimated derivatives

$$
\{ u_{i,t}^{(j_1,j_x)} \}_{i=0,j=0}^{m_t,m_x} = Q_c(\hat{\theta}, \hat{\theta}^0) \{ u_{i,t}^{(j_1,j_x)} \}_{i=0,j=0}^{m_t,m_x}
$$

and the regressor $\varphi_1(t_k,x_\ell)$ built up from the filtered noisy output only and the input for this initialization step

$$
\varphi_1(t_k,x_\ell) = \left[ -y_{0,0}^{(0)}(t_k,x_\ell) \cdots -y_{0,n_x}^{(n_t,n_x-1)}(t_k,x_\ell) \\
y_{1,0}^{(0)}(t_k,x_\ell) \cdots -y_{n_x,0}^{(n_t,n_x-1)}(t_k,x_\ell) \\
u_{0,0}^{(0)}(t_k,x_\ell) \cdots u_{m_x,0}^{(n_t,n_x-1)}(t_k,x_\ell) \right]^T
$$

- **Step 3** Compute an estimate $\hat{\theta}^1$:

$$
\hat{\theta}^1 = \left[ \sum_{k,t=1}^{N,L} \varphi_1(t_k,x_\ell) \varphi_1^T(t_k,x_\ell) \right]^{-1} \left[ \sum_{k,t=1}^{N,L} \varphi_1(t_k,x_\ell) y_{0,n_x}^{(n_t,n_x)}(t_k,x_\ell) \right]
$$

Set $\tau = 1$. [End of initialization part.]

- **Step 4** At any iteration $\tau+1$, compute an estimate of the noise-free output $\hat{\chi}(t_k,x_\ell)$ via numerical approximation of

$$
A(\hat{\theta}, \rho, \hat{\theta}^\tau) \hat{\chi}(t_k,x_\ell) = B(\hat{\theta}, \rho, \hat{\theta}^\tau) u(t_k,x_\ell)
$$

based on the estimates $\hat{\theta}^\tau$ at the previous iteration.

- **Step 5** Compute the estimated filters

$$
Q(\hat{\theta}, \hat{\theta}^\tau) = \frac{1}{F(\hat{\theta}, \hat{\theta}^\tau)}
$$

and use the filters as well as $\hat{\chi}(t_k,x_\ell)$ in order to generate the estimates of the derivatives:

$$
\{ \hat{\chi}_{l,t}^{(i_1,i_x)} \} = Q(\hat{\theta}, \hat{\theta}^\tau) \{ \hat{\chi}_{l,t}^{(i_1,i_x)} \}
$$

for $i_1 = 0, \ldots, n_t$; $i_x = 0, \ldots, n_x$ and $l = 1, \ldots, n_\alpha$

$$
\{ \hat{u}_{l,t}^{(j_1,j_x)} \} = Q(\hat{\theta}, \hat{\theta}^\tau) \{ \hat{u}_{l,t}^{(j_1,j_x)} \}
$$

for $j_1 = 0, \ldots, m_t$; $j_x = 0, \ldots, m_x$ and $l = 1, \ldots, m_\beta$

for $i_t = 0, \ldots, n_t$ and $i_x = 0, \ldots, n_x$.

- **Step 6** Build the estimated filtered regressor $\hat{\varphi}_1(t_k,x_\ell)$ as:

$$
\hat{\varphi}_1(t_k,x_\ell) = \left[ -\hat{\chi}_{0,0}^{(0)}(t_k,x_\ell) \cdots -\hat{\chi}_{0,n_x}^{(n_t,n_x-1)}(t_k,x_\ell) \\
-\hat{\chi}_{1,0}^{(0)}(t_k,x_\ell) \cdots -\hat{\chi}_{n_x,0}^{(n_t,n_x-1)}(t_k,x_\ell) \\
\hat{u}_{0,0}^{(0)}(t_k,x_\ell) \cdots \hat{u}_{m_x,0}^{(n_t,n_x-1)}(t_k,x_\ell) \right]^T
$$

$\hat{\varphi}_1(t_k,x_\ell)$ is composed of the filtered noisy output, the filtered noise-free estimated output and the input.

And build the filtered estimated IV $\hat{\zeta}_1(t_k,x_\ell)$ as:

$$
\hat{\zeta}_1(t_k,x_\ell) = \left[ -\hat{\chi}_{0,0}^{(0)}(t_k,x_\ell) \cdots -\hat{\chi}_{0,n_x}^{(n_t,n_x-1)}(t_k,x_\ell) \\
-\hat{\chi}_{1,0}^{(0)}(t_k,x_\ell) \cdots -\hat{\chi}_{n_x,0}^{(n_t,n_x-1)}(t_k,x_\ell) \\
\hat{u}_{0,0}^{(0)}(t_k,x_\ell) \cdots \hat{u}_{m_x,0}^{(n_t,n_x-1)}(t_k,x_\ell) \right]^T
$$

$\hat{\zeta}_1(t_k,x_\ell)$ vector is composed of the filtered noise-free estimated output and the input and represents the instruments.

- **Step 7** Compute the LPV-SRVC-PDE estimate $\hat{\theta}^\tau+1$ at the $(\tau+1)$th iteration:

$$
\hat{\theta}^\tau+1 = \left[ \sum_{k,t=1}^{N,L} \hat{\zeta}_1(t_k,x_\ell) \hat{\varphi}_1^T(t_k,x_\ell) \right]^{-1} \left[ \sum_{k,t=1}^{N,L} \hat{\zeta}_1(t_k,x_\ell) y_{0,n_x}^{(n_t,n_x)}(t_k,x_\ell) \right]
$$

- **Step 8** If $\hat{\theta}^\tau+1$ has converged or the maximum number of iterations is reached, then stop, else increase $\tau$ by 1 and go to Step 4.

VI. NUMERICAL EXAMPLE

The performance of the described algorithm is evaluated in the case of the advection-diffusion equation (ADE). This equation is often used in the water resource quality analysis to describe the transport and dispersion of a solute into a river channel [15], [3]. The ADE takes the following form:

$$
\begin{align*}
\frac{\partial \chi(t,x)}{\partial t} &= a_{02}(\rho(t)) \frac{\partial^2 \chi(t,x)}{\partial x^2} - a_{01}(\rho(t)) \frac{\partial \chi(t,x)}{\partial x} + b_{00}(\rho(t)) u(t,x) \\
\chi(t = 0, x) &= 0 \\
y(t_k, x) &= \frac{\partial \chi(t_k,x)}{\partial x} + e(t_k, x)
\end{align*}
$$

The noise-free signal $\chi(x_k,t_k)$ (Fig. 1) represents the solute propagation in the river and is obtained by discretization of the partial differential equation by a finite difference method (see e.g. [10]).

The temporal sampling is $T_s = 10s$ and $N = 8640$ temporal points are measured. The spatial sampling is $h = 10m$ for $L = 61$ space points. These 61 points can represent 61 sensors distributed in the river. $u(t,x)$ describes the source


