International Journal of Systems Engineering
2024; 8(2): 22-39
http://www.sciencepublishinggroup.com/j/ijse
doi: 10.11648/j.ijse.20240802.11
ISSN: 2640-4222 (Print); ISSN: 2640-4230 (Online)

Identification of Physical Dynamical Processes Via Linear Structure Models (Part 2)

Oleg Yu. Kopysov

Cybernetics Department, DesCartes Science Center, Varna, Bulgaria

Email address:
info@cartesiancenter.eu (Oleg Yu. Kopysov)

To cite this article:
Oleg Yu. Kopysov. (2024). Identification of Physical Dynamical Processes Via Linear Structure Models (Part 2). *International Journal of Systems Engineering, 8*(2), 22-39. https://doi.org/10.11648/j.ijse.20240802.11

Received: 26 July 2024; **Accepted:** 24 September 2024; **Published:** 31 October 2024

Abstract: Well-known methods of joint estimation of the state and parameters (quasilinearization, invariant imbedding, extended Kalman filter and others like them) expand the vector of the state of the system by including equations for parameters in the model. Such a task of joint estimation of the state and parameter is nonlinear even for linear systems. For Linear Structure Models (LSModels), an analytical method is proposed for the transition to an auxiliary model in which the parameter vector is expanded by initial states and the task of identifying parameter and initial states becomes linear. With the help of an auxiliary state vector, the initial dynamic model is reduced to an auxiliary model with residual. In this case, the auxiliary model does not contain derivatives of the measured elements of the initial dynamic model, but contains filtered measured elements. The proof of the identity of solutions according to the initial and auxiliary models is given. An Iterative algorithm of identification of order, parameters and state estimation is proposed. An analytical example of solving the problem of joint estimation of parameters and state for the heat equation is given and its software implementation in the MATLAB is discussed in detail. Next, another auxiliary model is proposed. If the first implies that the order of the differential equation is unknown but only limited by a certain value, then the second model has a given order. Now there can be two types of auxiliary models to it. An example of a nonlinear initial model is given.

Keywords: Identification Iterative Algorithm, Linear Structure Models, Auxiliary LSModel, State Estimation

1. Introduction

The article proposes a method for simultaneous identification of parameters, determination of the model equation order and evaluation of the state of nonlinear dynamical systems.

Well-known methods of joint estimation of the state and parameter (quasilinearization [1, 2], invariant imbedding [3-6], extended Kalman filter [7, 8] and others like them [9, 10]) expand the vector of the state of the system by including equations for parameters in the model. Such a task of joint estimation of the state and parameter is nonlinear even for linear systems.

The idea of the auxiliary model is very simple. Let the initial LSModel have the form:
$a_1\dot{x}(t) + a_2 x(t) + a_3 g(t) = 0$:
Integrating the equation of the model, we obtain the auxiliary model:

$$a_1 x(t) - a_1 x(t_0) + a_2 \int_{t_0}^{t} x(\tau) d\tau + a_3 \int_{t_0}^{t} g(\tau) d\tau = 0.$$

Thus, we got another parameter $a_1 x(t_0)$ that contains the initial conditions, and the problem remains linear. This article is devoted to generalizing this simple idea.

For Linear Structure Models (LSModels) [11, 12], an analytical method is proposed for the transition to an auxiliary model in which the parameter vector is expanded by initial states and the task of identifying parameter and initial states becomes linear.

In the first part of the article, for estimating the state of the system was used numerical differentiation. Hereinafter we will show how to replace differentiation with integration with measurement filtering elements.

The method of projection onto the plane of guarantors [13] from the first part of the article has also gained traction. Now the projection is performed onto the span of filtered guarantors.

2. Auxiliary Linear Structure Model

Let's consider the following Linear Structure Model (LSModel) with residual

$$\sum_{k=1}^{K}\sum_{s=1}^{S} a_{ks} f_k^{(S-s)}(t,y,g) = \varepsilon(t), \tag{1}$$

where

$\varepsilon(t)$ – residual;

a_{ks} – LSModel parameters;

$f_k(t,y,g)$ – LSModel elements, depending on input $g(t)$, output $y(t)$ and on an independent variable $t \in \boldsymbol{t}$;

$f_k^{(S-s)}(t,y,g) = \dfrac{d^{S-s}}{dt^{S-s}} f_k(t,y,g)$ – corresponding derivatives of $f_k(t,y,g)$ with respect to independent variable $t \in \boldsymbol{t}$;

$\boldsymbol{t}$ – is an observation interval.

I'd like to note that the order of LSModel equation (1) is not fixed, because any of the parameters a_{ks} can be zero. The order is only limited by the value S, which we can assign at our discretion.

Following [11-15], let's introduce *an auxiliary vector* $\boldsymbol{z}(t)$ with the following components:

$$\begin{aligned}
&z_1(t) = e(t),\\
&z_2(t) = z_1^{(1)}(t) - c_1 z_1(t) - \sum_{k=1}^{K} a_{k1} f_k(t,y,g),\\
&\vdots\\
&z_S(t) = z_{S-1}^{(1)}(t) - c_{S-1} z_1(t) - \sum_{k=1}^{K} a_{k,S-1} f_k(t,y,g).
\end{aligned} \tag{2}$$

Let's show that $\boldsymbol{z}(t)$ satisfies the equation of *auxiliary LSModel in vector form*

$$\frac{d}{dt}\boldsymbol{z}(t) = C\boldsymbol{z}(t) + \sum_{k=1}^{K} \boldsymbol{a}_k \cdot f_k(t,y,g), \tag{3}$$

in this case, the residual $e(t)$ satisfies the equation

$$e^{(S)}(t) - \sum_{s=1}^{S} c_s e^{(S-s)}(t) = \varepsilon(t). \tag{4}$$

Where C is a matrix with arbitrary coefficients c_s, looking as follows:

$$C = \begin{bmatrix} c_1 & 1 & 0 & \cdots & 0\\ c_2 & 0 & 1 & \cdots & 0\\ \vdots & \vdots & & \ddots & \\ c_{S-1} & 0 & 0 & \cdots & 1\\ c_S & 0 & 0 & \cdots & 0 \end{bmatrix};$$

$\boldsymbol{z}(t) = [\,z_1(t)...z_S(t)\,]'$ is the introduced auxiliary column-vector;
$\boldsymbol{a}_k = [\,a_{k1}...a_{kS}\,]'$ is a column-vector of parameters at f_k and its derivatives.

Having substituted the first component $z_1(t)=e(t)$ into equation (2) for the second component $z_2(t)$ we get

$$z_2(t) = e^{(1)}(t) - c_1 e(t) - \sum_{k=1}^{K} a_{k1} f_k(t),$$

and now the obtained component $z_2(t)$ into the third one

$$z_3(t) = \Big(e^{(1)}(t) - c_1 e(t) - \sum_{k=1}^{K} a_{k1} f_k(t)\Big)^{(1)} ... - c_2 e(t) - \sum_{k=1}^{K} a_{k2} f_k(t) = e^{(2)}(t) - c_1 e^{(1)}(t) - c_2 e(t) - \sum_{k=1}^{K}\sum_{s=1}^{2} a_{ks} f_k^{(2-s)}(t)$$

and so on and the obtained $(S-1)$-th into the S-th.

As a result, we get

$$z_S(t) = e^{(S-1)}(t) - c_1 e^{(S-2)}(t) - ... - c_{S-1} e(t) ... - \sum_{k=1}^{K}\sum_{s=1}^{S-1} a_{ks} f_k^{(S-1-s)}(t).$$

Let's differentiate the obtained expression

$$\frac{d}{dt}z_S(t)=e^{(S)}(t)-c_1e^{(S-1)}(t)-...-c_{S-1}e(t)^{(1)}\ ...-\sum_{k=1}^{K}\sum_{s=1}^{S-1}a_{ks}f_k^{(S-s)}(t).$$

From (4) it follows that

$$e^{(S)}(t)-c_1e^{(S-1)}(t)-...-c_{S-1}e(t)^{(1)}=\varepsilon(t)+c_Se(t)$$

and from (3) it follows that

$$\sum_{k=1}^{K}\sum_{s=1}^{S-1}a_{ks}f_k^{(S-s)}(t)=\varepsilon(t)-\sum_{k=1}^{K}a_{kS}f_k^{(S-S)}(t).$$

As a result

$$\frac{d}{dt}z_S(t)=\varepsilon(t)+c_Se(t)-\varepsilon(t)+\sum_{k=1}^{K}a_{kS}f_k^{(S-S)}(t)=c_Se(t)+\sum_{k=1}^{K}a_{kS}f_k(t),$$

what proves our statement.

By direct substitution it is easy to make sure that a Cauchy problem solution for the LSModel in vector form (3) with initial conditions

$$\boldsymbol{z}(0)=[\,z_{10}...z_{S0}\,]'$$

can be written as $an\ auxiliary\ LSModel\ in\ matrix\ form$:

$$\boldsymbol{z}(t)=\Phi(t)\cdot\boldsymbol{z}(0)+\sum_{k=1}^{K}R_k(t)\cdot\boldsymbol{a}_k, \tag{5}$$

where matrices $\Phi(t)$ and $R_k(t)$ are solutions of the following Cauchy problems:

$$\frac{d}{dt}\Phi(t)=C\Phi(t),\quad \Phi(0)=E, \tag{6}$$

$$\frac{d}{dt}R_k(t)=CR_k(t)+f_k(t)E,\quad R_k(0)=0, \tag{7}$$

$$(k=1,\ldots,K);$$

where E is a unit matrix. The size of all matrices is $S{\times}S$.

Let's remark that the equations of these problems depend only on functions $f_k(t)$ and do not depend on their derivatives.

Let's remember that $z_1(t)=e(t)$, then the first row of solution $\boldsymbol{z}(t)$ (5) can be noted in the form of $the\ auxiliary\ LSModel\ with\ residual$:

$$\sum_{s=1}^{S}\phi_{1s}(t)z_{s0}+\sum_{k=1}^{K}\sum_{s=1}^{S}r_{k,1s}(t)a_{ks}=e(t), \tag{8}$$

where

$\phi_{1s}(t)$ are elements of the first row of the matrix $\Phi(t)$,

$r_{k,1s}(t)$ are elements of the first rows of the matrices $R_k(t)$.

The initial and the auxiliary LSModels with residual are equivalent from the parameter identification point of view, in other words, all solutions of the parameter identification problem obtained using the auxiliary LSModel are solutions to the initial problem and vice versa.

Let's remark that all Cauchy Problems (6,7) and the residual relation (4) have free parameters $\{c_s\colon s{=}1,...,S\}$ and, therefore, an additional problem of their selection may be set.

For example, the equation for the first element of the sth column of the matrix $R_k(\boldsymbol{t})$:

$$r_{k,1s}^{(S)}(\boldsymbol{t})-c_1\,r_{k,1s}^{(S-1)}(\boldsymbol{t})-\cdots-c_{S-1}\,r_{k,1s}^{(1)}(\boldsymbol{t})\ ...-\ c_S\,r_{k,1s}(\boldsymbol{t})=f_k^{(S-s)}(\boldsymbol{t})$$

is an equation of a linear filter of LSModel element $f_k^{(S-s)}(\boldsymbol{t})$, that's why it's possible to choose the order S and arbitrary coefficients c_s of this and other equations based on conditions of filtration of input functions $f_k^{(S-s)}(\boldsymbol{t})$ into output functions $r_{k,1s}(\boldsymbol{t})$.

It's not by chance that the transformations of elements $f_k^{(S-s)}(\boldsymbol{t})$ of the initial LSModel to elements of the auxiliary LSModel $r_{k,1s}(\boldsymbol{t})$ were called FR-$transformations$.

In a new notation:

$$r_{k,1s}(\boldsymbol{t})=FR\big(\,f_k^{(S-s)}(\boldsymbol{t})\,\big),\quad e(\boldsymbol{t})=FR\big(\,\varepsilon(\boldsymbol{t})\,\big).$$

Based on the above, the following algorithm may be proposed.

2.1. Iterative Algorithm of Identification of Order, Parameters and State Estimation

$Step\,0:$ Select initial LSModels (1) and auxiliary LSModels with residual (8) and obtain the initial observation data:

$$f_k^{[0]}(t,y,g),\ t\in\boldsymbol{t} \qquad k=1,...,K.$$

$Step\,1$: Solve Cauchy problems for the calculation of matrices $\Phi(\boldsymbol{t})$ (6) and $R_k(\boldsymbol{t})$ (7) on the observation interval $\boldsymbol{t}$.

$Step\,2$: Solve the problem of identification via the auxiliary LSModel with residual (8), i.e. calculate

$$z_{s0},\ a_{ks} \qquad k=1,...,K;\ s=1,...,S.$$

$Step\,3$: Calculate $\boldsymbol{z(t)}$ (5), $d/dt\,\boldsymbol{z(t)}$ (3).

If the residual $e(\boldsymbol{t})=z_1(\boldsymbol{t})$ or other circumstances meet the identification criterion,

$finish\ the\ calculations.$

If it is possible, again calculate components $f_k^{[i]}(\boldsymbol{t})$ from (2),

$Go\ to\ Step\,1$ and continue calculations on the basis of the newly calculated components $f_k^{[i]}(\boldsymbol{t})$ of the state vector of the initial LSModel. Here $[i]$ is an iteration number.

$End\ of\ iterative\ algorithm.$

I'd like to note that all iterations are calculated without additional measurements on the basis of the smoothing effect (filtering) of the functions $f_k^{[i]}(\boldsymbol{t})$.

2.2. Properties of FR-transformations

Let's prove the properties of solutions of Cauchy Problems for the matrices Φ and R_k that make their calculation process much easier.

Let's consider the sth columns $\boldsymbol{\phi}_s(t)$, $\boldsymbol{r}_{k,s}(t)$ and $\boldsymbol{e}_s$ of the matrices $\Phi(t)$, $R_k(t)$ and the identity matrix E. The following relations are true for these columns:

$$\boldsymbol{e}_{s-1} = C\boldsymbol{e}_s,$$

$$\boldsymbol{\phi}_{s-1}(t) = C\boldsymbol{\phi}_s(t),$$

$$\boldsymbol{r}_{k,s-1}(t) = C\boldsymbol{r}_{k,s}(t).$$

The validity of the first relation can be verified by direct substitution.

To prove the second one, let's write the Cauchy Problem for the $(s-1)$th column

$$\frac{d}{dt}\boldsymbol{\phi}_{s-1}(t) = C\boldsymbol{\phi}_{s-1}(t),\ \boldsymbol{\phi}_{s-1}(0)=\boldsymbol{e}_{s-1};$$

and, having pre-multiplied all equations by the matrix C, we get

$$\frac{d}{dt}C\boldsymbol{\phi}_{s-1}(t) = C\,C\boldsymbol{\phi}_{s-1}(t),\ C\boldsymbol{\phi}_{s-1}(0)=C\boldsymbol{e}_{s-1}.$$

And, taking the proved relations into account, we get

$$\frac{d}{dt}\boldsymbol{\phi}_s(t) = C\boldsymbol{\phi}_s(t),\ \boldsymbol{\phi}_s(0)=\boldsymbol{e}_s,$$

i.e. the Cauchy Problem for the sth column.

Let's write the Cauchy Problem for the $(s-1)$th column of the matrix $R_k(t)$ in a similar way:

$$\frac{d}{dt}\boldsymbol{r}_{k,s-1}(t) = C\boldsymbol{r}_{k,s-1}(t) + f_k(t)\,\boldsymbol{e}_{s-1},\ \boldsymbol{r}_{k,s-1}(0)=0;$$

having pre-multiplied all equations by the matrix C, we get

$$\frac{d}{dt}C\boldsymbol{r}_{k,s-1}(t) = C\,C\boldsymbol{r}_{k,s-1}(t) + C f_k(t)\,\boldsymbol{e}_{s-1},$$

$$C\boldsymbol{r}_{k,s-1}(0)=C\,0.$$

Taking the proved relations into account, we get

$$\frac{d}{dt}\boldsymbol{r}_{k,s}(t) = C\boldsymbol{r}_{k,s}(t) + f_k(t)\,\boldsymbol{e}_s,\ \boldsymbol{r}_{k,s}(0)=0,$$

i.e. the Cauchy Problem for the sth column.

So, to solve the Cauchy Problems for the matrices Φ and R_k, it's necessary to solve the Cauchy Problems for the last columns of these matrices and to calculate the other columns according to relations. Such calculation process organization results in much faster obtaining of solutions.

For example, from the properties it follows that the solution of the Cauchy Problem

$$\frac{d}{dt}R(t)=\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}R(t)+f(t)\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix},\ R(0)=\begin{bmatrix}0&0&0\\0&0&0\\0&0&0\end{bmatrix},$$

looks like

$$R = \begin{bmatrix}r_3&r_2&r_1\\0&r_3&r_2\\0&0&r_3\end{bmatrix}$$

where

$$r_3(t)=\int_0^t f(\tau)d\tau,\ r_2(t)=\int_0^t r_3(\tau)d\tau,\ r_1(t)=\int_0^t r_2(\tau)d\tau.$$

2.3. Analytical example

In this example, I'd like to demonstrate the basic principle of object parameters identification using an auxiliary LSModel, which consists in the fact that the general solution of the problem for the main and auxiliary models coincide.

For this purpose let's consider the object going throughout the entire article and the measurements obtained in experiment #1 $g(x,t)=0.2$ and $w(x,t)=t+x(x-1)/2$, only this time we'll take another model:

$$a_1\,w_t(x,t) + a_2\,w_{xx}(x,t) + a_3\,w(x,t) = g(x,t).$$

Let's calculate $w_t(x,t)=1$ and $w_{xx}(x,t)=1$ and having substituted them into the equation we get

$$a_1{\cdot}1 + a_2{\cdot}1 + a_3(t{+}x(x{-}1)/2) = 0.2\,.$$

Obviously, the parameter identification problem has an ambiguous general solution: $a_1{+}a_2{=}0.2$, $a_3{=}0$.

I'd like to draw your attention to the fact that in an ambiguous solution, some parameters may be determined unambiguously [12].

It is also interesting that having taken another model of our object we got $a_3{=}0$. This fact confirms that the previous model with fewer elements is adequate, what has a certain physical meaning.

Now we will analytically solve this problem with the help of an auxiliary LSModel.

Let's fix $x{=}0$ then $w(0,t){=}t$, $w_{xx}(0,t){=}const$ and LSModel (1) is as follows:

$$a_1\, w_t(0,t) + a_3\, w(0,t) + b\, 1(t) = 0,$$

where $b{=}a_2 w_{xx}(0,t){-}g(0,t)$.

In notation of equation (1) $f_1(t){=}w(0,t)$, $f_2(t){=}1$, $K{=}2$, $S{=}2$, $a_{11}{=}a_1$, $a_{12}{=}a_3$, $a_{21}{=}0$, $a_{22}{=}b$.

For the sake of the simplicity of calculations here, let

$$c_1{=}c_2{=}0,$$

then the auxiliary vector $\boldsymbol{z}(t)$ (2) will have the following components

$$z_1(t) = 0,$$
$$z_2(t) = -a_1 w(0,t),$$

and the auxiliary LSModel in vector form (3) will appear as follows:

$$\frac{d}{dt}\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} + w(0,t)\begin{bmatrix} a_1 \\ a_3 \end{bmatrix} + 1\begin{bmatrix} 0 \\ b \end{bmatrix},$$

and the auxiliary LSModel in matrix form (5) will appear as follows:

$$\begin{bmatrix} z_1(t) \\ z_2(t) \end{bmatrix} {=}\Phi(t)\begin{bmatrix} z_1(0) \\ z_2(0) \end{bmatrix} {+}R_1(t)\begin{bmatrix} a_1 \\ a_3 \end{bmatrix} {+}R_2(t)\begin{bmatrix} 0 \\ b \end{bmatrix}.$$

Here $\Phi(t)$, $R_1(t)$, $R_2(t)$ are the solutions of three Cauchy problems:

$$\frac{d}{dt}\Phi(t){=}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\Phi(t), \quad \Phi(0){=}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix};$$

$$\frac{d}{dt}R_1(t){=}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}R_1 + \begin{bmatrix} w(0,t) & 0 \\ 0 & w(0,t) \end{bmatrix}, \; R_1(0){=}\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix};$$

$$\frac{d}{dt}R_2(t){=}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}R_2 + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R_2(0){=}\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Thus, we've got the auxiliary LSModel with residual (8):

$$\phi_{11}(t)z_1(0) + \phi_{12}(t)z_2(0) + ... + r^1_{11}(t)a_1 + r^1_{12}(t)a_3 + r^2_{12}(t)b = 0.$$

It is easy to show that

$$\Phi(t){=}\begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix}, \; R_1(t){=}\begin{bmatrix} t^2/2 & t^3/6 \\ 0 & t^2/2 \end{bmatrix}, \; R_2(t){=}\begin{bmatrix} t & t^2/2 \\ 0 & t \end{bmatrix},$$

then the auxiliary LSModel will be as follows:

$$1\, z_1(0) + t\, z_2(0) + t^2/2\, a_1 + t^3/6\, a_3 + t^2/2\, b = 0.$$

The general solution of the identification problem is as follows:

$$z_1(0){=}0, \; z_2(0){=}0, \; a_1 + b = 0, \; a_3{=}0.$$

Let me remind that $z_2(t){=}{-}a_1\, w(0,t)$, consequently $d/dt\, z_2(t){=}{-}a_1\, w_t(0,t)$, and from the model in vector form it follows that $d/dt\, z_2(t){=}w(0,t)\, a_3 + 1\, b{=}b$.

Then $w_t(0,t){=}{-}b/a_1$, and from the obtained general solution $b{=}{-}a_1$, in the result $w_t(0,t){=}1$.

And so with the help of an auxiliary model, we calculated the general solution and derivative $w_t(0,t)$ without resorting to differentiation of measurements, but only by solving Cauchy problems.

Now let's fix $t{=}0$ then $w(x,0){=}x(x{-}1)/2$, $w_t(x,0){=}const$ and LSModel (1) is as follows:

$$a_2\, w_{xx}(x,0) + a_3\, w(x,0) + d\, 1(x) = 0,$$

where $d{=}a_1 w_t(x,0){-}g(x,0)$.

In notation of equation (1) $f_1(x){=}w(x,0)$, $f_2(x){=}1$, $K{=}2$, $S{=}3$, $a_{11}{=}a_2$, $a_{12}{=}0$, $a_{13}{=}a_3$, $a_{21}{=}0$, $a_{22}{=}0$, $a_{23}{=}d$.

To simplify calculations here, let

$$c_1{=}c_2{=}c_3{=}0,$$

then the auxiliary vector $\boldsymbol{z}(x)$ (2) will have the following components

$$\begin{aligned} z_1(x) &= 0,\\ z_2(x) &= -a_2 w(x,0),\\ z_3(x) &= -a_2 w_x(x,0), \end{aligned}$$

and the auxiliary LSModel in vector form (3) will appear as follows:

$$\frac{d}{dx}\begin{bmatrix} z_1(x)\\ z_2(x)\\ z_3(x)\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix} z_1(x)\\ z_2(x)\\ z_3(x)\end{bmatrix} + w(x,0)\begin{bmatrix} a_2\\ 0\\ a_3\end{bmatrix} + 1\begin{bmatrix} 0\\ 0\\ d\end{bmatrix},$$

and the auxiliary LSModel in matrix form (5) will appear as follows:

$$\begin{bmatrix} z_1(x)\\ z_2(x)\\ z_3(x)\end{bmatrix} {=}\Phi(x)\begin{bmatrix} z_1(0)\\ z_2(0)\\ z_3(0)\end{bmatrix} {+}R_1(x)\begin{bmatrix} a_2\\ 0\\ a_3\end{bmatrix} {+}R_2(x)\begin{bmatrix} 0\\ 0\\ d\end{bmatrix}.$$

Here $\Phi(x)$, $R_1(x)$, $R_2(x)$ are the solutions of three Cauchy problems:

$$\frac{d}{dx}\Phi(x) = \begin{bmatrix} 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\Phi(x), \quad \Phi(0){=}\begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix};$$

$$\frac{d}{dx}R_1(x){=}\begin{bmatrix} 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}R_1(x) + w(x,0)\begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix},\ R_1(0){=}0;$$

$$\frac{d}{dx}R_2(x){=}\begin{bmatrix} 0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}R_2(x) + \begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix},\ R_2(0){=}0.$$

Thus, we've got the auxiliary LSModel with residual (8):

$$\phi_{11}(x)z_1(0) + \phi_{12}(x)z_2(0) + \phi_{13}(x)z_3(0) + ... +\ r^1_{11}(x)a_2 + r^1_{13}(x)a_3 + r^2_{13}(x)d = 0.$$

It is easy to show that

$$\Phi(x){=}\begin{bmatrix} 1 & x & x^2/2\\ 0 & 1 & x\\ 0 & 0 & 1\end{bmatrix}, \quad R_2(x){=}\begin{bmatrix} x & x^2/2 & x^3/6\\ 0 & x & x^2/2\\ 0 & 0 & x\end{bmatrix},$$

$$R_1(x){=}\begin{bmatrix} x^3/6{-}x^2/4 & x^4/24{-}x^3/12 & x^5/120{-}x^4/48\\ 0 & x^3/6{-}x^2/4 & x^4/24{-}x^3/12\\ 0 & 0 & x^3/6{-}x^2/4\end{bmatrix},$$

then the auxiliary LSModel will be as follows:

$$1\, z_1(0) + x\, z_2(0) + (x^2/2)\, z_3(0) + ... + (x^3/6{-}x^2/4)a_2 + (x^5/120{-}x^4/48)a_3 + (x^3/6)d = 0.$$

The general solution of the identification problem is as follows:

$$z_1(0){=}0,\ z_2(0){=}0,\ z_3(0)/2{-}a_2/4{=}0, a_2{+}d{=}0,\ a_3{=}0.$$

Let me remind that $z_3(x){=}{-}a_2\, w_x(x,0)$, and from the model in matrix form it follows that

$$z_3(x) = \phi_{33}(x)\, z_3(0){+}r^1_{31}(x)\, a_2{+}r^1_{33}(x)\, a_3{+}r^2_{33}(x)\, d{=}\ a_2/2 + x\, d = a_2/2 - x\, a_2,$$

then

$$w_x(x,0){=}{-}z_3(x)/a_2 = x{-}1/2.$$

From $z_3(x){=}{-}a_2\, w_x(x,0)$ it also follows that $d/dx\, z_3(x){=}{-}a_2\, w_{xx}(x,0)$, and from the model in vector form it follows that

$$\frac{d}{dx} z_3(x){=}w(x,0)\, a_3 + 1\, d{=}d.$$

then $w_{xx}(x,0){=}{-}d/a_2$, and from the obtained general solution $d{=}{-}a_2$, in the result $w_{xx}(x,0){=}1$.

Let me remind that for $t{=}0$ $d{=}a_1 w_t(x,0){-}g(x,0)$, and $g(x,t){=}0.2$, and now we have calculated that $d{=}{-}a_2$. And earlier for $x{=}0$ we have calculated that $w_t(0,t){=}1$. Having substituted all calculations into the expression for d we'll get in the end $-a_2{=}a_1{-}0.2$ or $a_1{+}a_2{=}0.2$.

Similarly, when $x{=}0$ $b{=}a_2 w_{xx}(0,t){-}g(0,t)$ and $b{=}{-}a_1$ and when $t{=}0$ $w_{xx}(x,0){=}1$.

This time $-a_1{=}a_2{-}0.2$ or $a_1{+}a_2{=}0.2$.

The total result of our calculations $a_1{+}a_2{=}0.2$, $a_3{=}0$ completely coincides with the general solution of the identification problem according to the basic model.

I'd like to draw your attention to the fact that despite the fact that the general solution was ambiguous, we got exact expressions for derivatives $w_t(0,t)$, $w_x(x,0)$, $w_{xx}(x,0)$. This is not surprising, because the essence of the general solution is that all models included in this solution have indiscernible (identical) states.

In particular, this is why the normal solution obtained using the least squares method has gained such popularity in cybernetics, although it often does not give the correct parameters, but being one of the solutions included in the general solution, it allows you to assess the state of the object correctly.

2.4. Example of Program

In contrast to the analytical example, let's take the previous model of our object to build the program:

$$a_1\, w_t(x,t) + a_2\, w_{xx}(x,t) + a_3\, g(x,t) = 0.$$

Let me remind you that experiment #3 guaranteed identification (see part 1). However, now having fixed $t{=}t_j$ we get

$$g(\boldsymbol{x},t_j) = sin(\boldsymbol{x})\, sin(t_j),$$

$$g_t(\boldsymbol{x},t_j) = sin(\boldsymbol{x})\, cos(t_j),$$

$$g_{xx}(\boldsymbol{x},t_j) = -sin(\boldsymbol{x})\, sin(t_j).$$

It is evident that now $g_t(\boldsymbol{x},t_j)$ and $g_{xx}(\boldsymbol{x},t_j)$ are linearly dependent and experiment #3 does not guarantee identification. Therefore, we will make a program for experiment #4 in which

$$g(\boldsymbol{x},\boldsymbol{t}) = sin(\boldsymbol{x})\, sin(\boldsymbol{t}) + 0.8\,\boldsymbol{t},$$

and then

$$g_t(\boldsymbol{x},t_j) = sin(\boldsymbol{x})\, cos(t_j) + 0.8,$$

$$g_{xx}(\boldsymbol{x},t_j) = -sin(\boldsymbol{x})\, sin(t_j).$$

Now $g_t(\boldsymbol{x},t_j)$ and $g_{xx}(\boldsymbol{x},t_j)$ are linear independent and experiment #4 guarantees identification.

Simulation of the object measurements, as before, occupies the beginning of the program. The full text of the program is given in the appendix, hereinafter we will consider only the main aspects.

$Step\, 0$: Select initial LSModels (1) and auxiliary LSModels with residual (8) and obtain the initial observation data.

The algorithm proposed in the analytical example was based on the fact that $w_t(\boldsymbol{x},t_j)$ and $w_{xx}(x_i,\boldsymbol{t})$ are constants, what simplified all calculations, but is incorrect in the general case.

Therefore, as before, we will calculate $w_t(\boldsymbol{x},t_j)$ according to formulas of central difference derivatives at every step of the outer loop of iterations $^{[i]}$:

$$w_t^{[i]}(\boldsymbol{x},t_j){=}\big(w^{[i]}(\boldsymbol{x},t_{j+1}){-}w^{[i]}(\boldsymbol{x},t_{j-1})\big)/(t_{j+1}{-}t_{j-1})$$

and the initial LSModel will be as follows:

$$\sum_{k=1}^{3}\sum_{s=1}^{3} a_{ks} f_k^{(S-s)}(\boldsymbol{x},t_j) = \varepsilon(\boldsymbol{x},t_j),$$

here

$$f_1{=}w_t^{[i]}(\boldsymbol{x},t_j),\quad a_{11}{=}0,\quad a_{12}{=}0,\quad a_{13}{=}a_1;$$

$$f_2{=}w^{[i]}(\boldsymbol{x},t_j),\quad a_{21}{=}a_2,\quad a_{22}{=}0,\quad a_{23}{=}0;$$

$$f_3{=}g(\boldsymbol{x},t_j),\quad a_{31}{=}0,\quad a_{32}{=}0,\quad a_{33}{=}a_3.$$

In this example, we fixed t_j in order to calculate the first derivative $w_t(\boldsymbol{x},t_j)$, and not the second $w_{xx}(x_i,\boldsymbol{t})$ when fixing x_i.

The outer loop of iterations begins with the operators

```
j1=13;    j2=15;                                                  % 69
                                                                  % 70
wi=w;  wi_t=w_t;                                                  % 71
                                                                  % 72
while 1, Iteration=Iteration+1                           %%%%%%%%%%% 73
for j=j1:j2,                                                      % 74
```

Operators `% 69` specify the interval $[t_{j_1}, t_{j_2}]$ over which further calculations will be performed.

Operators `% 71` set initial values for

$w^{[i]}(\boldsymbol{x},\boldsymbol{t})$: `wi=w;` and $w_t^{[i]}(\boldsymbol{x},\boldsymbol{t})$: `wi_t=w_t;`

The outer loop of iterations ends with the operators

```
end; %for tj                                                    %161
                                                                %162
it = input('Continue? Y/N [Y]: ', 's');                         %163
if isempty(it), it='Y'; end;                                    %164
if it=='n' || it=='N',  break; end;                             %165
                                                                %166
if j2-j1>1 && j1>=1 && j2<=st ,                                 %167
for j = j1+1 : j2-1,                                            %168
for i = 1 : sx,                                                 %169
wi_t(i,j)=(wi(i,j+1)-wi(i,j-1))...                              %170
/(t(j+1)-t(j-1));                                               %171
end;                                                            %172
end;                                                            %173
j1=j1+1;  j2=j2-1;                                              %174
else                                                            %175
disp('Continuation is not possible.');                          %176
break;                                                          %177
end;                                                            %178
                                                                %179
end %for Iteration                          %%%%%%%%%%%%%%%%%%%%%%180
```

After the interval $[t_{j_1}, t_{j_2}]$ (operator `%161`) is exhausted, a question is asked whether to continue the outer loop of iterations and, if the answer is positive, new values $w_t(\boldsymbol{x}, t_j)$ (operators `%168-173`) are calculated and the interval narrows by one on both sides (operators `%174`), as the derivative is calculated on a reduced interval. If the interval $[t_{j_1}, t_{j_2}]$ is not sufficient (operator `%167`), a message is displayed (operator `%176`) and calculations are terminated (operator `%177`).

In the inner loop of iterations, which begins with the operator

```
for i=1:2;                              %%%%%%%%%%%%%%%%%%%%%%%%%% 78
```

and ends with the operator

```
end; %for i                                 %%%%%%%%%%%%%%%%%%%%%% 160
```

two internal iterations of the algorithm, which we discuss throughout the article, are performed.

Iterative algorithm of identification of order, parameters and state estimation:

Step 1: Solve Cauchy problems for the calculation of the matrices $\Phi(\boldsymbol{x}, t_j)$ (6) and $R_k(\boldsymbol{x}, t_j)$ (7) on the observation interval $[\boldsymbol{x}, t_j]$, begins with comments

```
%% Step 1                                                       % 80
%% CAUCHY PROBLEMS                                              % 81
```

and contains operators necessary to solve all Cauchy Problems, except the Cauchy Problem for the matrix $\Phi(\boldsymbol{x}, t_j)$, which does not depend on iterable variables and is solved in zero iteration once (operators `% 61-63`).

To solve the Cauchy Problems, M-function ode45 is taken as the most universal, although the choice must be made in accordance with the peculiarity of the problem being solved. To use it, you need a separate file named, for example, `OYuK_R2` $\big($to calculate the matrix $R_2(\boldsymbol{x})\big)$ containing M-function, which is also called `OYuK_R2`, and calculates the derivative $d/dx\, R_2(\boldsymbol{x})$ of the Cauchy Problem being solved.

To calculate derivatives of the matrices $R_k(\boldsymbol{x}, t_j)$ cubic spline approximation (M-function spline) is used, what makes it possible to work with a given opt accuracy and with uneven readings of measured signals, that spring up, in particular, when anomalous measurements are removed (see part 1).

```
function[dR]=OYuK_R2(xr,R)                                      %1
global   x  j  wi  C                                            %2
dR = C*R + [ 0; 0; spline(x,wi(:,j),xr) ];                      %3
end                           %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%4
```

The necessary information is passed to the functions via shared variables `global`. The matrix `C` also sets the order of the auxiliary model.

The Cauchy Problems are solved for the third columns, for example, `R2_3` of the matrix $R_2(\boldsymbol{x}, t_j)$. M-function ode45 returns an array of dimension $\mathtt{sx}\times 3$, and for further calculations $3\times\mathtt{sx}$ is needed. Therefore, the result is transposed `Phi_3=Phi_3'` and then the previous columns of the matrix $\Phi(\boldsymbol{x})$ are calculated by multiplying by the matrix `C`:

```
[xR2,R2_3]=ode45(@OYuK_R2,x,[0;0;0],opt);
R2_3=R2_3';
R2_2=C*R2_3; R2_1=C*R2_2;
```

Other necessary Cauchy Problems are solved in a similar way.

Step 2 : Solve the problem of identification via the auxiliary LSModel with residual (8), begins with comments

```
%% Step 2
```

and contains three ways to calculate initial conditions and parameters. Every method ends with the display of solutions and singular numbers. By this I want to draw your attention to the need for a singular analysis. The use of the auxiliary model is just the beginning, because the general solution according to the initial model and the general solution according to the auxiliary model coincide. The choice of an unambiguous correct solution occurs at this step and here you need to use everything that was written earlier.

We have already discussed two methods: `SVD` and `MIXED LS & TLS`, but, as you understand, it is both possible and necessary to apply many others.

Below we'll discuss `PROJECTION on the FR-GUARANTORS` method.

In part 1 of this article, the method of projection onto the hyperplane of guarantors was presented. In it, the elements of the model were projected onto the span of guarantors.

Now the auxiliary model includes elements of the first row ϕ_{1s} of the matrix Φ and FR-transformed elements of the initial model $r_{k,1s}{=}FR\big(f_k^{(S-s)}\big)$, and, therefore the projection should be carried out onto the span of the first row of the matrix Φ and FR-transformed guarantors ($span\{\phi_{1s},\, r_{k,1s}\}$).

This is how calculations using this method look in the program:

```
%% PROJECTION on the FR-guarantors
[Qpr,Tpr]=...
qr([Phi R4_3(1,:)' R3_1(1,:)']);
Ppr=Qpr'*[Phi R];
[Upr,Spr,Vpr]=svd(Ppr(1:5,:),0);
z0a_pr=-Vpr(:,6)'/Vpr(6,6)
singular_value_pr=diag(Spr)'
%%
```

Step2 ends with the choice of a method whose calculated initial conditions and parameters will participate in further calculations:

```
%% CHOICE OF METHOD
%z0a=z0a_svd; disp('SVD');
%z0a=z0a_mix; disp('MIXED LS & TLS');
z0a=z0a_pr; disp('PROJECTION');
%%
```

The choice is made in the program by removing the `%` note sign. The current selection is displayed on the screen. The PROJECTION on the FR-GUARANTORS method is chosen here.

Step 3: Calculate $\boldsymbol{z}(\boldsymbol{x})$ (5), $d/dx\,\boldsymbol{z}(\boldsymbol{x})$ (3), begins with comment

```
%% Step 3
```

and contains operators for calculation of $\boldsymbol{z}(\boldsymbol{x})$

```
z0=z0a(1:3); a=z0a(4:6);
z=Phi_1*z0(1)+Phi_2*z0(2)+Phi_3*z0(3)...
+R1_3*a(1) +R2_1*a(2) +R3_3*a(3);
```

and $d/dx\,\boldsymbol{z}(\boldsymbol{x})$

```
dz=C*z;                                    %139
dz(1,:)=dz(1,:)+a(2)*wi(:,j)';             %140
dz(3,:)=dz(3,:)...                         %141
+a(1)*wi_t(:,j)'+a(3)*g(:,j)';             %142
```

Let me remind that

$z_1(\boldsymbol{x},t_j) = e(\boldsymbol{x},t_j)$,

$z_2(\boldsymbol{x},t_j) = z_1^{(1)}(\boldsymbol{x},t_j) \; - \; c_1 z_1(\boldsymbol{x},t_j) \; - \; a_2 w^{[i]}(\boldsymbol{x},t_j)$,

$z_3(\boldsymbol{x},t_j) = z_2^{(1)}(\boldsymbol{x},t_j) \; - \; c_2 z_1(\boldsymbol{x},t_j)$.

and assuming that $e(\boldsymbol{x},t_j){\approx}0$, then its derivative $e^{(1)}(\boldsymbol{x},t_j){=}z_1^{(1)}(\boldsymbol{x},t_j){\approx}0$ too, and hence

$w^{[i]}(\boldsymbol{x},t_j){=}{-}z_2(\boldsymbol{x},t_j)/a_2, \qquad w_x^{[i]}(\boldsymbol{x},t_j){=}{-}z_3(\boldsymbol{x},t_j)/a_2,$

$w_{xx}^{[i]}(\boldsymbol{x},t_j){=}{-}\frac{d}{dx} z_3(\boldsymbol{x},t_j)/a_2$.

In the program it looks like this

```
wi(:,j)=-z(2,:)'/a(2);                     %136
wi_x(:,j)=-z(3,:)'/a(2);                   %137
wi_xx(:,j)=-dz(3,:)'/a(2);                 %143
```

To check the assumptions about the smallness of $e(\boldsymbol{x},t_j)$, the program calculates the T-norm of $z_1(\boldsymbol{x},t_j)$

```
z1max=max(abs(z(1,:)))                     %134
```

Next, plots of the calculated model states are displayed on the screen (operators `%145-158`) and the internal iteration is repeated one more time.

In practice, of course, the need for repetition should proceed from the convergence of the algorithm, which can be controlled by T-norm of the residual, or other considerations based on the specifics of the identification problem.

The proposed auxiliary model assumes that the order of the differential equation of the initial model is unknown, but is limited by the value S. The class of models (1) is very wide, so the convergence of the identification algorithm cannot be guaranteed.

Hereinafter I want to consider another auxiliary model, for fixed-order equations S, which leads to a simple iteration $w^{[i+1]}{=}\Psi(w^{[i]})$.

3. Auxiliary Linear Structure Model 2

Let's consider the following Linear Structure Model [15]

$$w^{(S)}(t){+}\sum_{s=1}^{S} a_{0s} w^{(S-s)}(t){+}\sum_{k=1}^{K}\sum_{s=1}^{S} a_{ks} f_k^{(S-s)}(t,w,g){=}0,$$
$$\tilde{w}(t) = w(t) + \Delta w(t),$$

where

- a_{ks} are LSModel parameters;
- $f_k(t,w,g)$ are LSModel elements, depending on input $g{=}g(t)$, output $w{=}w(t)$ and on independent variable t ($t{\in}\boldsymbol{t}$);
- $f_k^{(S-s)}(t,w,g)$ are $(S{-}s)$-derivatives of function $f_k(t,w,g)$ with respect to t;
- $\tilde{w}(t)$ output measurements $w(t)$;
- $\boldsymbol{t}$ is an observation interval.

Let's introduce *a state vector* $\boldsymbol{z}(t)$ with the following components:

$$\begin{aligned}
&z_1(t){=}w(t),\\
&z_2(t){=}z_1^{(1)}(t) \quad - \quad a_{0,1}w(t) \quad - \quad \sum_{k=1}^{K} a_{k1} f_k(t,w,g),\\
&\vdots\\
&z_S(t){=}z_{S-1}^{(1)}(t) \quad - \quad a_{0,S-1}w(t) \quad - \quad \sum_{k=1}^{K} a_{k,S-1} f_k(t,w,g).
\end{aligned}$$

Then it's possible to write the initial model as

$$\frac{d}{dt}\boldsymbol{z}(t) = A\boldsymbol{z}(t) + \sum_{k=1}^{K} \boldsymbol{a}_k f_k(t,w,g),$$

where

$$A = \begin{bmatrix} a_{01} & 1 & 0 & \cdots & 0 \\ a_{02} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ a_{0,S-1} & 0 & 0 & \cdots & 1 \\ a_{0S} & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{a}_k = \begin{bmatrix} a_{k1} \\ \vdots \\ a_{kS} \end{bmatrix}.$$

Having added $\boldsymbol{c}z_1(t)-\boldsymbol{c}w(t)=0$, $\boldsymbol{c}=[c_1...c_S]'$ to the left side of the equation and denoted $w(t)=f_0(t)$ we get the following auxiliary LSModel in vector form:

$$\frac{d}{dt}\boldsymbol{z}(t) = C\boldsymbol{z}(t) + \sum_{k=0}^{K} \hat{\boldsymbol{a}}_k \cdot f_k(t, w^{[i]}, g),$$

where

$$C = \begin{bmatrix} c_1 & 1 & 0 & \cdots & 0 \\ c_2 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ c_{S-1} & 0 & 0 & \cdots & 1 \\ c_S & 0 & 0 & \cdots & 0 \end{bmatrix},$$

$\boldsymbol{c}=[c_1...c_S]'$ are free auxiliary LSModel parameters;

$\hat{\boldsymbol{a}}_k=[\hat{a}_{k1}...\hat{a}_{kS}]'$ are auxiliary LSModel parameters;

$w^{[i]}(t)$ is the ith estimation of $w(t)$ and the primary measurement of $w(t)$ is $w^{[0]}(t)=w(t)+\Delta w(t)$.

Notice that when $\boldsymbol{c}+\hat{\boldsymbol{a}}_0=\boldsymbol{a}_0$, $\hat{\boldsymbol{a}}_k=\boldsymbol{a}_k$ $(k=1...K)$ and $w^{[i]}(\boldsymbol{t})=w(\boldsymbol{t})$ the auxiliary LSModel and the initial LSModel are equivalent in parameters and state.

As before, let's write the solution of the Cauchy Problem via the auxiliary LSModel in vector form (2) with initial conditions $\boldsymbol{z}(0)=[\,z_{10}^{[i]} \ldots z_{S0}^{[i]}\,]'$ as the auxiliary LSModel in matrix form:

$$\boldsymbol{z}(t) = \Phi^{[i]}(t)\,\boldsymbol{z}(0) + \sum_{k=0}^{K} R_k^{[i]}(t)\,\hat{\boldsymbol{a}}_k,$$

where the matrices $\Phi^{[i]}(t)$ and $R_k^{[i]}(t)$ are solutions of the following Cauchy Problems:

$$\frac{d}{dt}\Phi^{[i]}(t) = C\Phi^{[i]}(t),\ \Phi^{[i]}(0)=E;$$

$$\frac{d}{dt}R_k^{[i]}(t) = CR_k^{[i]}(t) + f_k(t, w^{[i]}, g)E,\ R_k^{[i]}(0)=0;$$

where E is an identity matrix.

All matrixes have $(S\times S)$-dimension and all vectors have S-dimension.

Let's introduce two auxiliary LSModels: with the residual $e_1^{[i]}(t)$ and with the basic model element $w^{[i]}(t)$:

$$\sum_{s=1}^{S} \phi_{1s}^{[i]}(t)\cdot z_{s0} + \sum_{k=0}^{K}\sum_{s=1}^{S} r_{k,1s}^{[i]}(t)\cdot \hat{a}_{ks} = w^{[i]}(t) + e_1^{[i]}(t)$$

and with the residual $e_2^{[i]}(t)$ and with the basic model element $\tilde{w}(t)$:

$$\sum_{s=1}^{S} \phi_{1s}^{[i]}(t) z_{s0} + \sum_{k=0}^{K}\sum_{s=1}^{S} r_{k,1s}^{[i]}(t)\hat{a}_{ks} = \tilde{w}(t) + e_2^{[i]}(t),$$

where $\phi_{1s}^{[i]}(t)$ are elements of the first row of the matrix $\Phi^{[i]}(t)$, $r_{k,1s}^{[i]}(t)$ are elements of the first rows of the matrices $R_k^{[i]}(t)$.

The first auxiliary LSModel with residual is equivalent to

$$z_1(t) = w^{[i]}(t) + e_1^{[i]}(t),$$

and now, minimizing $e_1^{[i]}(t)$ as a residual, we are going to move $z_1(\boldsymbol{t})$ to $w^{[i]}(\boldsymbol{t})$ in the first model.

The second auxiliary LSModel with residual is equivalent to

$$z_1(t) = \tilde{w}(t) + e_2^{[i]}(t),$$

and minimizing $e_2^{[i]}(t)$ as a residual and taking

$$w^{[i+1]}(\boldsymbol{t}) = z_1^{[i]}(\boldsymbol{t})$$

for the next iteration we will get

$$w^{[i+1]}(\boldsymbol{t}) = \tilde{w}(t) + e_2^{[i]}(t),$$

or

$$e_2^{[i]}(t) = -\Delta w(t)$$

in the second model.

Thus, the uncertainty in the LSModel became a residual in the auxiliary LSModel.

Let's take $w^{[0]}(t){:=}\tilde{w}(t)$, $\boldsymbol{c}^{[0]}{:=}\boldsymbol{c}$.

Let's solve the corresponding Cauchy Problems and obtain matrices $\Phi^{[i]}(\boldsymbol{t})$ and $R_k^{[i]}(\boldsymbol{t})$.

Let's solve the Identification Problem via one of auxiliary LSModels with residual and find $z_{s0}^{[i]}$ and $a_{ks}^{[i]}$.

Let's substitute the found matrices and parameters into the right part of solution of the Cauchy Problem of the auxiliary LSModel in matrix form and let's define the value $\boldsymbol{z}^{[i]}(\boldsymbol{t})$ as follows:

$$\boldsymbol{z}^{[i]}(\boldsymbol{t}){:=}\Phi^{[i]}(\boldsymbol{t})\,\boldsymbol{z}^{[i]}(0) + \sum_{k=0}^{K} R_k^{[i]}(\boldsymbol{t})\,\hat{\boldsymbol{a}}_k^{[i]},$$

and calculate

$$\frac{d}{dt}\boldsymbol{z}^{[i]}(\boldsymbol{t}){:=}C^{[i]}\boldsymbol{z}^{[i]}(\boldsymbol{t}) + \sum_{k=0}^{K} f_k(\boldsymbol{t}, z_1^{[i]}, g)\,\hat{\boldsymbol{a}}_k^{[i]}$$

from the equation of the auxiliary LSModel in vector form.

Then for the next iteration let's take

$$w^{[i+1]}(\boldsymbol{t}){:=}\ z_1^{[i]}(\boldsymbol{t}).$$

To assess the convergence of the identification process, the t-norm of the residual

$$||e_1^{[i]}(\boldsymbol{t})||_t{:=}\ \max_{t\in\boldsymbol{t}} |w^{[i+1]}(t) - w^{[i]}(t)|,$$

may be used and may be calculated

$$\Delta w^{[i]}(\boldsymbol{t}){:=}\ \tilde{w}(\boldsymbol{t}) - w^{[i]}(\boldsymbol{t}).$$

We may change the value of the matrix $C^{[i+1]}$ by taking, for example $\boldsymbol{c}^{[i+1]}{=}\boldsymbol{a}_0^{[i]}$, and we may not change $C^{[i]}$, what is more practical from the point of view of solution of Cauchy Problems for the matrices $\Phi^{[i]}(t)$ and $R_k^{[i]}(t)$.

3.1. Example of Nonlinear LSModel

Let's consider a nonlinear model from part 1:

$$\alpha_1 w_t(\boldsymbol{x},\boldsymbol{t}) + \alpha_2 w_{xx}(\boldsymbol{x},\boldsymbol{t}) + \alpha_3 w(\boldsymbol{x},\boldsymbol{t})... + \beta_1 \left(w_x(\boldsymbol{x},\boldsymbol{t})\right)^2 + \beta_2\, g(\boldsymbol{x},\boldsymbol{t}) = 0.$$

Let's fix $\boldsymbol{t}{=}t_j$, then the initial LSModel will be as follows:

$$w_{xx}(\boldsymbol{x},t_j) + a_{02}\, w(\boldsymbol{x},t_j)... + a_{12}\, w_t(\boldsymbol{x},t_j) + a_{22}\, w_x^2(\boldsymbol{x},t_j) + a_{32}\, g(\boldsymbol{x},t_j) = 0,$$

here

$a_{02}{=}\alpha_3/\alpha_2,\ a_{12}{=}\alpha_1/\alpha_2,\ a_{22}{=}\beta_1/\alpha_2,\ a_{32}{=}\beta_2/\alpha_2,$

the others $a_{ks}{=}0$.

The state vector $\boldsymbol{z}(\boldsymbol{x},t_j)$ has the following components:

$$z_1(\boldsymbol{x},t_j) = w(\boldsymbol{x},t_j),$$

$$z_2(\boldsymbol{x},t_j) = z^{(1)}(\boldsymbol{x},t_j) - a_{01}\, w(\boldsymbol{x},t_j) = w_x(\boldsymbol{x},t_j),$$

because $a_{01}{=}0$.

The auxiliary LSModel in vector form will be as follows:

$$\frac{d}{dx}\boldsymbol{z}(\boldsymbol{x},t_j) = \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \boldsymbol{z}(\boldsymbol{x},t_j) + \begin{bmatrix} \hat{a}_{01} \\ \hat{a}_{02} \end{bmatrix} w(\boldsymbol{x},t_j)\ ... + \begin{bmatrix} 0 \\ \hat{a}_{12} \end{bmatrix} w_t(\boldsymbol{x},t_j) + \begin{bmatrix} 0 \\ \hat{a}_{22} \end{bmatrix} w_x^2(\boldsymbol{x},t_j) + \begin{bmatrix} 0 \\ \hat{a}_{32} \end{bmatrix} g(\boldsymbol{x},t_j),$$

here $\hat{a}_{01}{=}{-}c_1$ (because $a_{01}{=}0$), $\hat{a}_{02}{=}a_{02}{-}c_2$, $\hat{a}_{12}{=}a_{12}$, $\hat{a}_{22}{=}a_{22}$, $\hat{a}_{32}{=}a_{32}$.

$Step$ 0:

Let's take $w^{[0]}(\boldsymbol{x},t_j){:=}\tilde{w}(\boldsymbol{x},t_j)$.

Let's calculate $w_t^{[0]}(\boldsymbol{x},t_j)$ and $w_x^{[0]}(\boldsymbol{x},t_j)$ according to formulas of central-difference derivatives:

$$w_t^{[0]}(x_i,t_j){:=}\big(\tilde{w}(x_i,t_{j+1}){-}\tilde{w}(x_i,t_{j-1})\big)/(t_{j+1}{-}t_{j-1}),\, w_x^{[0]}(x_i,t_j){:=}\big(\tilde{w}(x_{i+1},t_j){-}\tilde{w}(x_{i-1},t_j)\big)/(x_{i+1}{-}x_{i-1}).$$

Let's solve the Cauchy Problem:

$$\frac{d}{dx}\begin{bmatrix} \phi_{12}(\boldsymbol{x}) \\ \phi_{22}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} \phi_{12}(\boldsymbol{x}) \\ \phi_{22}(\boldsymbol{x}) \end{bmatrix}, \quad \begin{bmatrix} \phi_{12}(0) \\ \phi_{22}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix};$$

and calculate

$$\begin{bmatrix} \phi_{11}(\boldsymbol{x}) \\ \phi_{21}(\boldsymbol{x}) \end{bmatrix} := \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} \phi_{12}(\boldsymbol{x}) \\ \phi_{22}(\boldsymbol{x}) \end{bmatrix}.$$

$Step$ 1:

Let's solve the Cauchy Problem:

$$\frac{d}{dx}\begin{bmatrix} r_{0,12}(\boldsymbol{x}) \\ r_{0,22}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} r_{0,12}(\boldsymbol{x}) \\ r_{0,22}(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ w^{[i]}(\boldsymbol{x},t_j) \end{bmatrix}, \begin{bmatrix} r_{0,12}(0) \\ r_{0,22}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

and calculate

$$\begin{bmatrix} r_{0,11}(\boldsymbol{x}) \\ r_{0,21}(\boldsymbol{x}) \end{bmatrix} := \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} r_{0,12}(\boldsymbol{x}) \\ r_{0,22}(\boldsymbol{x}) \end{bmatrix};$$

and the Cauchy Problem:

$$\frac{d}{dx}\begin{bmatrix} r_{1,12}(\boldsymbol{x}) \\ r_{1,22}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} r_{1,12}(\boldsymbol{x}) \\ r_{1,22}(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ w_t^{[i]}(\boldsymbol{x},t_j) \end{bmatrix}, \begin{bmatrix} r_{1,12}(0) \\ r_{1,22}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

and the Cauchy Problem:

$$\frac{d}{dx}\begin{bmatrix} r_{2,12}(\boldsymbol{x}) \\ r_{2,22}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} r_{2,12}(\boldsymbol{x}) \\ r_{2,22}(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ \left(w_x^{[i]}(\boldsymbol{x}, t_j)\right)^2 \end{bmatrix}, \begin{bmatrix} r_{2,12}(0) \\ r_{2,22}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

and the Cauchy Problem:

$$\frac{d}{dx}\begin{bmatrix} r_{3,12}(\boldsymbol{x}) \\ r_{3,22}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} c_1 & 1 \\ c_2 & 0 \end{bmatrix} \begin{bmatrix} r_{3,12}(\boldsymbol{x}) \\ r_{3,22}(\boldsymbol{x}) \end{bmatrix} + \begin{bmatrix} 0 \\ g(\boldsymbol{x}, t_j) \end{bmatrix}, \begin{bmatrix} r_{3,12}(0) \\ r_{3,22}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

I'd like to note that all Cauchy Problems are linear with constant coefficients, despite the fact that the equation of the initial model is nonlinear.

In particular, this circumstance makes it possible to solve analytical examples very easily, even for nonlinear models.

$Step$ 2:

Let's solve the Identification Problem via one of auxiliary LSModels with residual and find $z_{s0}^{[i]}$ and $a_{ks}^{[i]}$:

$$\phi_{11}(\boldsymbol{x})\, z_{10}^{[i]} + \phi_{12}(\boldsymbol{x})\, z_{20}^{[i]} - r_{0,11}(\boldsymbol{x})\, c_1 - r_{0,12}(\boldsymbol{x})\, c_2 ... + r_{0,12}(\boldsymbol{x})\, a_{02}^{[i]} + r_{1,12}(\boldsymbol{x})\, a_{12}^{[i]} + r_{2,12}(\boldsymbol{x})\, a_{22}^{[i]} ... + r_{3,12}(\boldsymbol{x})\, a_{32}^{[i]} = w^{[i]}(\boldsymbol{x}, t_j)$$

or

$$\phi_{11}(\boldsymbol{x})\, z_{10}^{[i]} + \phi_{12}(\boldsymbol{x})\, z_{20}^{[i]} - r_{0,11}(\boldsymbol{x})\, c_1 - r_{0,12}(\boldsymbol{x})\, c_2 ... + r_{0,12}(\boldsymbol{x})\, a_{02}^{[i]} + r_{1,12}(\boldsymbol{x})\, a_{12}^{[i]} + r_{2,12}(\boldsymbol{x})\, a_{22}^{[i]} ... + r_{3,12}(\boldsymbol{x})\, a_{32}^{[i]} = \tilde{w}(\boldsymbol{x}, t_j).$$

$Step$ 3:

Now we can calculate

$$\begin{bmatrix} z_1^{[i]}(\boldsymbol{x}) \\ z_2^{[i]}(\boldsymbol{x}) \end{bmatrix} := \begin{bmatrix} \phi_{11}(\boldsymbol{x}) \\ \phi_{21}(\boldsymbol{x}) \end{bmatrix} z_{10}^{[i]} + \begin{bmatrix} \phi_{12}(\boldsymbol{x}) \\ \phi_{22}(\boldsymbol{x}) \end{bmatrix} z_{20}^{[i]} ... - \begin{bmatrix} r_{0,11}(\boldsymbol{x}) \\ r_{0,21}(\boldsymbol{x}) \end{bmatrix} c_1 + \begin{bmatrix} r_{0,12}(\boldsymbol{x}) \\ r_{0,22}(\boldsymbol{x}) \end{bmatrix} (a_{02}^{[i]} - c_2) ...$$

$$+ \begin{bmatrix} r_{1,12}(\boldsymbol{x}) \\ r_{1,22}(\boldsymbol{x}) \end{bmatrix} a_{12}^{[i]} + \begin{bmatrix} r_{2,12}(\boldsymbol{x}) \\ r_{2,22}(\boldsymbol{x}) \end{bmatrix} a_{22}^{[i]} + \begin{bmatrix} r_{3,12}(\boldsymbol{x}) \\ r_{3,22}(\boldsymbol{x}) \end{bmatrix} a_{32}^{[i]},$$

$$e^{[i+1]}(\boldsymbol{x}) := z_1^{[i]}(\boldsymbol{x}) - w^{[i]}(\boldsymbol{x}, t_j), ||e^{[i+1]}(\boldsymbol{x})||_t := \max_{x \in \boldsymbol{x}} |e^{[i+1]}(x)|, e_x^{[i+1]}(\boldsymbol{x}) := z_2^{[i]}(\boldsymbol{x}) - w_x^{[i]}(\boldsymbol{x}, t_j), ||e_x^{[i+1]}(\boldsymbol{x})||_t := \max_{x \in \boldsymbol{x}} |e_x^{[i+1]}(x)|,$$

and can take

$$w^{[i+1]}(\boldsymbol{x}, t_j) := z_1^{[i]}(\boldsymbol{x}), \qquad w_x^{[i+1]}(\boldsymbol{x}, t_j) := z_2^{[i]}(\boldsymbol{x}), \Delta w^{[i+1]}(\boldsymbol{x}, t_j) := \tilde{w}(\boldsymbol{x}, t_j) - w^{[i+1]}(\boldsymbol{x}, t_j).$$

$Go\ to\ Step$ 1.

The reader is invited to perform the calculation of the analytical example and create a program independently. And the main thing is to check the program and theory for a complete match. I warn you right away that most likely there will be errors both in the theoretical part and in the program. But a match will mean that the program is ready for use.

I hope that I have corrected my mistakes. If you find any, would you, please, take the trouble to write to the mail. I will be pleased that you have read my article carefully.

4. Conclusion

The auxiliary model method itself is analytical, but its software implementation has a lot of options that depend on:

1. measurement filtering method (matrix C and S model order);
2. method of solving the Cauchy Problem;
3. approximation method and measurement frequency;
4. method for solving the auxiliary problem of parameter and initial state identification.

All this gives good opportunities to adjust the program to the conditions of the problem you are facing.

In this part of the article examples are given in which only the problems of organizing calculations using the auxiliary model method are discussed. But do not forget about the problems described in two previous parts of the article, which have now become the problem of identification of the auxiliary LSModel parameters.

Abbreviations

LSModel	Linear Structure Model
LSModels	Linear Structure Model Class
M-function	MATLAB function

ORCID

0009-0005-0593-1685 (Oleg Yu. Kopysov)

Author Contributions

Oleg Yu. Kopysov is the sole author. The author read and approved the final manuscript.

Conflicts of Interest

The author declare no conflicts of interest.

Appendix: MATLAB Program

These program files were transferred to the article by copying from the MATLAB editor. You can copy each file back to the MATLAB editor without changing anything. The line numbers on the right are comments of the MATLAB language and do not affect the operation of the program. After copying to the MATLAB editor, you will see the line numbers on the left, they must match the numbers on the right. Next, you need to save the file with the appropriate name.

File OYuK (main program)

```
clear; clf; format short e                                                   %  1
opt=odeset('RelTol', 2^(-24));                                               %  2
                                                                             %  3
global  x t j  wi wi_t  g g_t  C                                             %  4
                                                                             %  5
%%              HEAT EQUATION                                               %%  6
%% 0.8 w_t(x,t) - 0.6 w_xx(x,t) = g(x,t)                                    %%  7
                                                                             %  8
Experiment=4                                            %%%%%%%%%%%%%%%%%%%%%%%%%%%%  9
hx=2^(-3); x1=0;      xf=6;                                                  % 10
ht=2^(-2); t1=-1*ht;  tf=6+ht;                                               % 11
                                                                             % 12
x=(x1:hx:xf);  sx=size(x,2);                                                 % 13
t=(t1:ht:tf);  st=size(t,2);                                                 % 14
                                                                             % 15
g0=zeros(sx,st);   g0_t=zeros(sx,st);                                        % 16
w0=zeros(sx,st);   w0_t=zeros(sx,st);                                        % 17
w0_x=zeros(sx,st); w0_xx=zeros(sx,st);                                       % 18
F0=zeros(sx*st,3);                                                           % 19
m=0; teta=-atan(4/3);                                                        % 20
for i=1:sx, for j=1:st,                                                      % 21
g0(i,j)=sin(x(i))*sin(t(j))-0.8*t(j);                                        % 22
g0_t(i,j)=sin(x(i))*cos(t(j))-0.8;                                           % 23
w0(i,j)=sin(x(i))*sin(t(j)+teta)-t(j)^2/2;                                   % 24
w0_t(i,j)=sin(x(i))*cos(t(j)+teta)-t(j);                                     % 25
w0_x(i,j)=cos(x(i))*sin(t(j)+teta);                                          % 26
w0_xx(i,j)=-sin(x(i))*sin(t(j)+teta);                                        % 27
m=m+1;                                                                       % 28
F0(m,:)=[w0_t(i,j) w0_xx(i,j) g0(i,j)];                                      % 29
end; end;                                                                    % 30
                                                                             % 31
[U0,S0,V0]=svd(F0,0); a0=-V0(:,3)/V0(3,3);                                   % 32
                                                                             % 33
deviation=0.01                                                               % 34
w=w0+deviation*randn([sx,st]);                                               % 35
```

```
g=g0;

g0=g0(:,2:st-1); g0_t=g0_t(:,2:st-1);
w0=w0(:,2:st-1); w0_t=w0_t(:,2:st-1);
w0_x =w0_x(:,2:st-1);
w0_xx=w0_xx(:,2:st-1);

Iteration=0;                            %%%%%%%%%%%%%%%%%%%%%%%%%%%%

w_t=zeros((sx),(st-2));
g_t=zeros((sx),(st-2));
for i=1:sx, for j=2:st-1,
w_t(i,j-1)=(w(i,j+1)-w(i,j-1))...
/(t(j+1)-t(j-1));
g_t(i,j-1)=(g(i,j+1)-g(i,j-1))...
/(t(j+1)-t(j-1));
end; end;

t=t(2:st-1);
g=g(:,2:st-1);    w=w(:,2:st-1);
st=st-2;

c1=-1; c2=-2^2; c3=0;
C=[ c1 1 0; c2 0 1; c3 0 0 ];

[xP,Phi_3]=ode45(@OYuK_Phi,x,[0;0;1],opt);
Phi_3=Phi_3';
Phi_2=C*Phi_3; Phi_1=C*Phi_2;
Phi=[Phi_1(1,:)' Phi_2(1,:)' Phi_3(1,:)'];

wi_x =zeros(sx,st);
wi_xx=zeros(sx,st);

j1=13;   j2=15;

wi=w;  wi_t=w_t;

while 1, Iteration=Iteration+1          %%%%%%%%%%%%
for j=j1:j2,

fprintf('%s%d%s%1.3f\n','t(',j,')=',t(j));

for i=1:2;                              %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Step 1
%% CAUCHY PROBLEMS
[xR1,R1_3]=ode45(@OYuK_R1,x,[0;0;0],opt);
R1_3=R1_3';

[xR2,R2_3]=ode45(@OYuK_R2,x,[0;0;0],opt);
R2_3=R2_3';
R2_2=C*R2_3; R2_1=C*R2_2;

[xR3,R3_3]=ode45(@OYuK_R3,x,[0;0;0],opt);
R3_3=R3_3';
R3_2=C*R3_3; R3_1=C*R3_2;
```

```
[xR4,R4_3]=ode45(@OYuK_R4,x,[0;0;0],opt);
R4_3=R4_3';

R=[R1_3(1,:)' R2_1(1,:)' R3_3(1,:)' ];

if 1, clf; plot([Phi R]); pause(2); end;

%% Step 2
%% SVD
[Ux,Sx,Vx]=svd([Phi R],0);
z0a_svd=-Vx(:,6)'/Vx(6,6)
singular_value=diag(Sx)'
%%
%% MIXED LS & TLS
[Q,T]=qr([Phi R3_3(1,:)']);
T1=T(1:4,1:4)^(-1);
P=Q'*[R1_3(1,:)' R2_1(1,:)'];
[Up,Sp,Vp]=svd(P(5:st,:),0);
b=-T1*P(1:4,:)*Vp(:,2);
z0a_mix=[b(1) b(2) b(3) Vp(:,2)' b(4)]';
z0a_mix=-z0a_mix'/z0a_mix(6)
singular_value_mix=diag(Sp)'
%%
%% PROJECTION on the FR-guarantors
[Qpr,Tpr]=...
qr([Phi R4_3(1,:)' R3_1(1,:)']);
Ppr=Qpr'*[Phi R];
[Upr,Spr,Vpr]=svd(Ppr(1:5,:),0);
z0a_pr=-Vpr(:,6)'/Vpr(6,6)
singular_value_pr=diag(Spr)'
%%
%% CHOICE OF METHOD
%z0a=z0a_svd; disp('SVD');
%z0a=z0a_mix; disp('MIXED LS & TLS');
z0a=z0a_pr; disp('PROJECTION');
%%
%% Step 3
z0=z0a(1:3); a=z0a(4:6);
z=Phi_1*z0(1)+Phi_2*z0(2)+Phi_3*z0(3)...
+R1_3*a(1) +R2_1*a(2) +R3_3*a(3);

z1max=max(abs(z(1,:)))

wi(:,j)=-z(2,:)'/a(2);
wi_x(:,j)=-z(3,:)'/a(2);

dz=C*z;
dz(1,:)=dz(1,:)+a(2)*wi(:,j)';
dz(3,:)=dz(3,:)...
+a(1)*wi_t(:,j)'+a(3)*g(:,j)';
wi_xx(:,j)=-dz(3,:)'/a(2);

clf;   xtj=['(x,',num2str(t(j),3),')'];
subplot(221);
plot(x,wi(:,j),'r','LineWidth',2);
title(['{\bfw^{[i]}}' xtj ]);
subplot(222);
```

```
plot(x,wi_t(:,j),'c','LineWidth',2);
title(['{\bfw_t^{[i]}}' xtj ]);
subplot(223);
plot(x,wi_x(:,j),'m','LineWidth',2);
title(['{\bfw^{[i]}_x}' xtj ]);
subplot(224);
plot(x,wi_xx(:,j),'b','LineWidth',2);
title(['{\bfw^{[i]}_{xx}}' xtj ]);
pause(3);

end; %for i
end; %for tj

it = input('Continue? Y/N [Y]: ', 's');
if isempty(it), it='Y'; end;
if it=='n' || it=='N',  break; end;

if j2-j1>1 && j1>=1 && j2<=st ,
for j = j1+1 : j2-1,
for i = 1 : sx,
wi_t(i,j)=(wi(i,j+1)-wi(i,j-1))...
/(t(j+1)-t(j-1));
end;
end;
j1=j1+1;  j2=j2-1;
else
disp('Continuation is not possible.');
break;
end;
end %for Iteration
```

File OYuK_Phi for calculation of the 3rd column of $\Phi(\boldsymbol{x}, t_j)$

```
function[dPhi]=OYuK_Phi(x_p,Phi)
global  C
dPhi = C*Phi;
end
```

File OYuK_R1 for calculation of the 3rd column of $R_1(\boldsymbol{x}, t_j)$

```
function[dR]=OYuK_R1(xr,R)
global   x  j  wi_t  C
dR = C*R + [ 0; 0; spline(x,wi_t(:,j),xr) ];
end
```

File OYuK_R2 for calculation of the 3rd column of $R_2(\boldsymbol{x}, t_j)$

```
function[dR]=OYuK_R2(xr,R)
global   x  j  wi  C
dR = C*R + [ 0; 0; spline(x,wi(:,j),xr) ];
end
```

File OYuK_R3 for calculation of the 3rd column of $R_3(\boldsymbol{x}, t_j)$

```
function[dR]=OYuK_R3(xr,R)
global   x  j  g  C
dR = C*R + [ 0; 0; spline(x,g(:,j),xr) ];
end
```

File OYuK_R4 for calculation of the 3rd column of $R_4(\boldsymbol{x}, t_j)$

```
function[dR]=OYuK_R4(xr,R)                                    %1
global   x  j  g_t  C                                         %2
dR = C*R + [ 0; 0; spline(x,g_t(:,j),xr) ];                   %3
end                                %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%4
```

References

[1] Bellman R., Kagiwada H., Kalaba R., Wengert's Numerical Method for Partial Derivatives, Orbit Determination, and Quasilinearization. *The RAND Corporation, RM-4354-PR, November*, 1964.

[2] Bellman R., Kalaba R. *Quasilinearization and nonlinear boundary-value problems.* American Elsevier Publishing Company, Inc. 1965, 218p.

[3] Bellman R., Wing G. M. *An introduction to invariant imbedding.* SIAM, 1992, 267p.

[4] Bellman R., Kalaba R., Wing G. M., Invariant Imbedding and Mathematical Physics–I: Particle Processes. *J. Math. Phys.* 1960, 1, pp. 280-308.

[5] Shimizu R., Aoki R., *Application of Invariant Imbedding to Reactor Physics*, Academic, New York, 1972.

[6] Wall D., Olsson P. Invariant imbedding and hyperbolic heat waves. *Technical report CODEN: LUTEDX (TEAT-7045/1-32/(1996).*

[7] Kalman R. E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Engineering,* 1960, 82D, pp. 35–45.

[8] Grewal M. S., Andrews A. P. *Kalman Filtering: Theory and Practice Using MATLAB*. Third Edition. John Wiley & Sons, Inc. 2008.

[9] Eykhoff P. *System Identification Parameter and State Estimation*. John Wiley & Sons, 1974.

[10] Sage A.P., Melsa J.L. *Estimation Theory with Application to Communication and Control*. Mc Graw Hill, 1971.

[11] Kopysov O. Yu., Prokopov B. I., Pupkov K. A. Identification of parameters and the state of one class of nonlinear system. *Problem of Control and Information Theory*, 1982, 11(3), pp. 205–216.

[12] Kopysov O. Yu. *Identification via linear structure models*. Varna: Publishing house DesCartes Science Center; 2012, 368 p. ISBN 978-954-92807-2-2.

[13] Kopysov O. Yu. Identification of Parameters and Restoring of State Vector of Dynamical Plant. *IFAC Proceedings Volumes (IFAC-PapersOnLine)*, 2013, 46(9), pp. 1494-1499.

[14] Kopysov O. Yu. Elements of the Model Identification. *IFAC-PapersOnLine*, 2017, 50(1), pp. 2260–2265. 20th IFAC World Congress.

[15] Kopysov O. Yu. Identification of Order, Parameters and State Estimation via Projection onto the Plane of Guarantors. *IFAC-PapersOnLine*, 2023, 56(2), pp. 7771–7777. 22th IFAC World Congress.