Automatica 48 (2012) 2502–2511

Contents lists available at SciVerse ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

LS-SVM approximate solution to linear time varying descriptor systems☆

Siamak Mehrkanoon [1], Johan A.K. Suykens

KU Leuven, ESAT-SCD, Kasteelpark Arenberg 10, B-3001 Leuven (Heverlee), Belgium

ARTICLE INFO

Article history:
Received 7 November 2011
Received in revised form
20 April 2012
Accepted 17 June 2012
Available online 31 July 2012

Keywords:
Least squares support vector machines
Descriptor systems
Closed form approximate solution

ABSTRACT

This paper discusses a numerical method based on Least Squares Support Vector Machines (LS-SVMs) for solving linear time varying initial and boundary value problems in Differential Algebraic Equations (DAEs). The method generates a closed form (model-based) approximate solution. The results of numerical experiments on different systems with index from 0 to 3, are presented and compared with analytic solutions to confirm the validity and applicability of the proposed method.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Differential algebraic equations (DAEs) arise frequently in numerous applications including mathematical modeling, circuit and control theory (Bernan, 1986), chemistry (Froment & Bischoff, 1990; Ranade, 2002), fluid dynamic (Lötstedt & Petzold, 1986) and computer-aided design. DAEs have been known under a variety of names, depending on the area of application, for instance, they are also called descriptor, implicit or singular systems. The most general form of DAE is given by

$$F(\dot{x}, x, t) = 0 \tag{1}$$

where $\frac{\partial F}{\partial \dot{x}}$ may be singular. The rank and structure of this Jacobian matrix depends, in general, on the solution $x(t)$. DAEs are characterized by their index. In Brenan, Campbell, and Petzold (1989) the index of (1) is defined as the minimum number of differentiations of the system which would be required to solve for $\dot{x}$ uniquely in terms of x and t. The index of DAEs is a measure of the degree of singularity of the system and it is widely considered as an indication of certain difficulties for numerical methods. We note that DAEs with an index greater than 1 are often referred to as higher-index DAEs and that the index of an ODE is zero. See Ascher and Petzold (1998) and Campbell (1982) for a detailed discussion of the index of a DAE. The important special case of (1) is semi-explicit DAE or an ODE with constraints i.e.

$$\begin{aligned} \dot{x} &= f(x, y, t) \\ 0 &= g(x, y, t). \end{aligned} \tag{2}$$

The index is 1 if $\frac{\partial g}{\partial y}$ is nonsingular. x and y are considered as differential and algebraic variables respectively. Analytic solutions for these problems are not generally available and hence numerical methods must be applied. Some numerical methods have been developed for solving DAEs using backward differentiation formulas (BDFs) (Ascher, 1989; Ascher & Petzold, 1998; Ascher & Spiter, 1994; Brenan et al., 1989; Gear & Petzold, 1984) or implicit Runge–Kutta (IRK) methods (Ascher & Petzold, 1991; Ascher & Spiter, 1994; Brenan et al., 1989; Murugesh & Murugesan, 2009). These methods are only applicable to low-index problems and often require that the problem should have a special structure. Furthermore, these approaches approximate the solutions at discrete points only (discrete solution) and some interpolation techniques are needed in order to get a continuous solution. Thereby, recently attempts have been made to develop methods that produce a closed form approximate solution. Awawdeh, Jaradat, and Alsayyed (2009) applied the Homotopy analysis method to systems of DAEs. The authors in Guzel and Bayram (2006) used the Padé approximation method to estimate the solution of singular systems with index-2.

☆ This work was supported by Research Council KUL: GOA/10/09MaNet, PFV/10/002 (OPTEC), several PhD/postdoc & fellow grants. Flemish Government: IOF: IOF/KP/SCORES4CHEM; FWO: PhD/postdoc grants, projects: G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine), G.0377.09 (Mechatronics MPC); G.0377.12 (Structured systems) research community (WOG: MLDM); IWT: PhD Grants, projects: Eureka-Flite+, SBO LeCo-Pro, SBO Climaqs, SBO POM, O&O-Dsquare. Belgian Federal Science Policy Office: IUAP P7/(DYSCO, Dynamical systems, control and optimization, 2012–2017). IBBT. EU: ERNSI, FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B. COST: Action ICO806: IntelliCIS. Contract Research: AMINAL. Other: ACCM. Johan Suykens is a professor at the KU Leuven, Belgium. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Kok Lay Teo under the direction of Editor Ian R. Petersen.

E-mail addresses: Siamak.Mehrkanoon@esat.kuleuven.be (S. Mehrkanoon), Johan.Suykens@esat.kuleuven.be (J.A.K. Suykens).

[1] Tel.: +32 16328658; fax: +32 16321970.

0005-1098/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2012.06.095

In general, the higher the index, the greater the numerical difficulty one is going to encounter when solving differential algebraic equations numerically and an alternative treatment is the use of index reduction techniques which are based on repeated differentiation of the constraints until a low-index problem (an index 1 DAE or ODE) is obtained. There are several reasons to consider differential algebraic equations (2) directly, rather than convert it to a system of ODEs (Brenan et al., 1989; Wang, Jhu, Yung, & Wang, 2011). Therefore designing direct methods that do not require a reformulation (e.g. index reduction) of DAEs will not only speed up the solution, but also the system structure (e.g. the modeling changes and parameter variations) can be more readily explored.

It is known that the singular system can have instantaneous jumps due to inconsistent initial conditions. Several approaches for consistent initialization of DAEs have been studied in the literature and in general they fall into two categories: the rigorous initialization and the direct initialization method (we refer the reader to Pengfei, Yaoyu, and Seem (2012) and references therein for more details). Within the scope of this paper we assume that consistent initial conditions are available.

In Mehrkanoon, Falck, and Suykens (2012), solving ordinary differential equations using Least Squares Support Vector Machines has been discussed. In this paper the method developed in Mehrkanoon et al. (2012) is extended to approximate the solution of linear time varying initial and boundary value problems in differential algebraic equations.

This paper is organized as follows. In Section 2, a brief review about least squares support vector regression problems together with preliminary definitions are given. In Section 3, we formulate our least squares support vector machines method to the solution of the linear time-varying initial value problem (IVP) in DAEs. Section 4, is devoted to the formulation of the method for boundary value problems (BVPs) in DAEs. The model selection of the proposed method is discussed in Section 5. In Section 6, numerical experiments are tested to demonstrate the viability of the proposed method.

2. LS-SVM regression and preliminary definitions

Let us consider a given training set $\{x_i, y_i\}_{i=1}^N$ with input data $x_i \in \mathbb{R}$ and output data $y_i \in \mathbb{R}$. The goal in regression is to estimate a model of the form $y(x) = w^T\varphi(x) + b$.

The primal LS-SVM model for regression can be written as follows (Suykens, Van Gestel, De Brabanter, De Moor, & Vandewalle, 2002)

$$\begin{aligned} &\underset{w,b,e}{\text{minimize}}\ \frac{1}{2}w^Tw + \frac{\gamma}{2}e^Te \\ &\text{subject to } y_i = w^T\varphi(x_i) + b + e_i, \quad i = 1, \ldots, N \end{aligned} \tag{3}$$

where $\gamma \in \mathbb{R}^+$, $b \in \mathbb{R}$, $w \in \mathbb{R}^h$. $\varphi(\cdot) : \mathbb{R} \to \mathbb{R}^h$ is the feature map and h is the dimension of the feature space. The dual solution is then given by

$$\left[\begin{array}{c|c} \Omega + \boldsymbol{I}_N/\gamma & 1_N \\ \hline 1_N^T & 0 \end{array}\right]\left[\begin{array}{c} \alpha \\ \hline b \end{array}\right] = \left[\begin{array}{c} y \\ \hline 0 \end{array}\right],$$

where $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T\varphi(x_j)$ is the (i, j)-th entry of the positive definite kernel matrix. $1_N = [1; \ldots; 1] \in \mathbb{R}^N$, $\alpha = [\alpha_1; \ldots; \alpha_N]$, $y = [y_1; \ldots; y_N]$ and I_N is the identity matrix. When we deal with differential equations, the target values y_i are no longer directly available, so the regression approach will fail. Nevertheless we can incorporate the underlying differential equation in the learning process to find an approximation for the solution.

First we need to define the derivative of the kernel function. Making use of Mercer's Theorem (Vapnik, 1998), derivatives of the feature map can be written in terms of derivatives of the kernel function. Let us define the following differential operator which will be used in subsequent sections

$$\nabla_n^m \equiv \frac{\partial^{n+m}}{\partial x^n \partial y^m}. \tag{4}$$

If $\varphi(x)^T\varphi(y) = K(x, y)$, then one can show that

$$\begin{aligned} [\varphi^{(n)}(x)]^T\varphi^{(m)}(y) &= \nabla_n^m[\varphi(x)^T\varphi(y)] \\ &= \nabla_n^m[K(x, y)] = \frac{\partial^{n+m}K(x, y)}{\partial x^n \partial y^m}. \end{aligned} \tag{5}$$

Using formula (5), it is possible to express all derivatives of the feature map in terms of the kernel function itself (provided that the kernel function is sufficiently differentiable). For instance if we choose the RBF kernel, $K(x, y) = \exp\left(-\frac{(x-y)^2}{\sigma^2}\right)$, then the following relations hold,

$$\begin{aligned} &\nabla_0^0[K(x, y)] = K(x, y), \\ &\nabla_1^0[K(x, y)] = \frac{\partial(\varphi(x)^T\varphi(y))}{\partial x} \\ &\qquad = \varphi^{(1)}(x)^T\varphi(y) = -\frac{2(x-y)}{\sigma^2}K(x, y), \\ &\nabla_0^1[K(x, y)] = \frac{\partial(\varphi(x)^T\varphi(y))}{\partial y} \\ &\qquad = \varphi(x)^T\varphi^{(1)}(y) = \frac{2(x-y)}{\sigma^2}K(x, y). \end{aligned}$$

The superscript$^{(1)}$ stands for the derivative operator.

3. Formulation of the method for IVPs in DAEs

Consider the linear time varying initial value problem in DAEs of the following form

$$\begin{aligned} &Z(t)\dot{X}(t) = A(t)X(t) + B(t)u(t), \quad t \in [t_{in}, t_f], \\ &X(t_{in}) = X_0, \end{aligned} \tag{6}$$

where $Z(t) = [z_{ij}(t)], A(t) = [a_{ij}(t)] \in \mathbb{R}^{m\times m}$ and $B(t) \in \mathbb{R}^{m\times r}$. The state vector $X = [x_1, \ldots, x_m]^T \in \mathbb{R}^m$, the input vector $u \in \mathbb{R}^r$ and $\dot{X}(t) = \frac{dX}{dt}$. $Z(t)$ may be singular on $[t_{in}, t_f]$ with variable rank and the DAE may have an index that is larger than one. When Z is nonsingular, Eq. (6) can be converted to an equivalent explicit ODE system. In addition we assume that $Z(t)$, $A(t)$ and $B(t)u(t)$ are sufficiently smooth and the DAE (6) is solvable. (See Definition 2.1 in Clark and Petzold (1989) and references therein for a more detailed discussion about solvability.)

Let us assume that a general approximate solution to the i-th equation of (6) is of the form $\hat{x}_i(t) = w_i^T\varphi(t) + b_i$, where w_i and b_i are parameters of the model that have to be determined. To obtain the optimal value of these parameters, collocation methods can be used (Kincaid & Cheney, 2002) which assume a discretization of the interval $[t_{in}, t_f]$ into a set of collocation points $\Upsilon = \{t_{in} = t_1 < t_2 < \cdots < t_N = t_f\}$. Therefore the adjustable parameters w_i and b_i, for $i = 1, \ldots, m$, are to be found by solving the following optimization problem:

$$\begin{aligned} &\underset{\hat{X}}{\text{minimize}}\ \frac{1}{2}\sum_{i=1}^{N}\left[(Z\hat{X}' - A\hat{X} - Bu)(t_i)\right]^2 \\ &\text{subject to } \hat{X}(t_{in}) = X_0, \end{aligned} \tag{7}$$

where N is the number of collocation points (which is equal to the number of training points) used to undertake the learning

process. For simplicity let us assume that $X_0 = [p_1; \ldots; p_m]$ and $g(t) = B(t)u(t)$. In what follows we formulate the optimization problem in the LS-SVM framework for solving linear time varying differential algebraic equations. For notational convenience let us list the following notations which are used in the following sections,

$$[\nabla_n^m K](t,s) = [\nabla_n^m K(x,y)]|_{x=t,y=s},$$

$$\Omega_n^m(i,j) = \nabla_n^m[K(x,y)]|_{x=t_i,y=t_j} = \left.\frac{\partial^{n+m}K(x,y)}{\partial x^n \partial y^m}\right|_{x=t_i,y=t_j},$$

$$\Omega_0^0(i,j) = \nabla_0^0[K(x,y)]|_{x=t_i,y=t_j} = K(t_i,t_j),$$

where $\Omega_n^m(i,j)$ denotes the (i,j)-th entry of matrix Ω_n^m. The Matlab notation $M(k:l, m:n)$ is used for selecting a submatrix of matrix M consisting of rows k to l and columns m to n. $M(i,:)$ denotes the i-th row of matrix M. $M(:,j)$ denotes the j-th column of matrix M.

Remark 1. In order to keep the notations as simple as possible, we utilized the same feature map $\varphi(t)$ for all the states, i.e. $\hat{x}_i(t) = w_i^T\varphi(t) + b_i$. Nevertheless it is possible to use different mapping functions, as long as the corresponding kernel functions satisfy Mercer's Theorem (Vapnik, 1998), and then one has more hyper parameters to tune.

3.1. Singular ODE system

Let us consider DAEs (6). The approximate solution, $\hat{x}_i(t) = w_i^T\varphi(t)+b_i$, for $i = 1, \ldots, m$ is then obtained by solving the following optimization problem

$$\begin{aligned}
&\underset{w_i,b_i,e_\ell^i}{\text{minimize}}\ \frac{1}{2}\sum_{\ell=1}^m w_\ell^T w_\ell + \frac{\gamma}{2}\sum_{\ell=1}^m e_\ell^T e_\ell \\
&\text{subject to} \begin{bmatrix} z_{11}(t) & \cdots & z_{1m}(t) \\ \vdots & \ddots & \vdots \\ z_{m1}(t) & \cdots & z_{mm}(t) \end{bmatrix} \begin{bmatrix} w_1^T\varphi'(t_i) \\ \vdots \\ w_m^T\varphi'(t_i) \end{bmatrix} \\
&= \begin{bmatrix} a_{11}(t) & \cdots & a_{1m}(t) \\ \vdots & \ddots & \vdots \\ a_{m1}(t) & \cdots & a_{mm}(t) \end{bmatrix} \begin{bmatrix} w_1^T\varphi(t_i) + b_1 \\ \vdots \\ w_m^T\varphi(t_i) + b_m \end{bmatrix} \\
&\quad + \begin{bmatrix} g_1(t_i) \\ \vdots \\ g_m(t_i) \end{bmatrix} + \begin{bmatrix} e_1(t_i) \\ \vdots \\ e_m(t_i) \end{bmatrix}, \quad \text{for } i = 2, \ldots, N, \\
&\begin{bmatrix} w_1^T\varphi(t_1) + b_1 \\ \vdots \\ w_m^T\varphi(t_1) + b_m \end{bmatrix} = \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix}.
\end{aligned} \tag{8}$$

Lemma 3.1. *Given a positive definite kernel function $K : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ with $K(t,s) = \varphi(t)^T\varphi(s)$ and a regularization constant $\gamma \in \mathbb{R}^+$, the solution to (8) is given by the following dual problem:*

$$\left[\begin{array}{ccc} \mathcal{K} & \mathcal{S} & -F_A \\ \hline \mathcal{S}^T & \Omega_0^0(1,1)\times I_{m\times m} & I_{m\times m} \\ \hline -F_A^T & I_{m\times m} & 0_{m\times m} \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline \beta \\ \hline b \end{array}\right] = \left[\begin{array}{c} G \\ \hline P \\ \hline 0 \end{array}\right] \tag{9}$$

with

$$\alpha = \left[\begin{array}{c} \alpha^1 \\ \hline \vdots \\ \hline \alpha^m \end{array}\right] = [\alpha_2^1; \ldots; \alpha_N^1; \cdots; \alpha_2^m; \ldots; \alpha_N^m] \in \mathbb{R}^{m(N-1)},$$

$$\beta = [\beta_1; \cdots; \beta_m], \qquad b = [b_1; \ldots; b_m],$$

$$P = [p_1; \ldots; p_m],$$

$$G = [g_1(t_2); \cdots; g_1(t_N); \cdots; g_m(t_2); \cdots; g_m(t_N)] \in \mathbb{R}^{m(N-1)},$$

$$F_A = \begin{bmatrix} F_{A_{11}} & \cdots & F_{A_{1m}} \\ \vdots & & \vdots \\ F_{A_{m1}} & \cdots & F_{A_{mm}} \end{bmatrix} \in \mathbb{R}^{m(N-1)\times m},$$

$$F_{A_{kl}} = [a_{kl}(t_2); \cdots; a_{kl}(t_N)] \in \mathbb{R}^{N-1}, \quad \text{for } k,l = 1, \ldots, m$$

$$F_{Z_{kl}} = [z_{kl}(t_2); \cdots; z_{kl}(t_N)] \in \mathbb{R}^{N-1}, \quad \text{for } k,l = 1, \ldots, m$$

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_{11} & \cdots & \mathcal{K}_{1m} \\ \vdots & & \vdots \\ \mathcal{K}_{m1} & \cdots & \mathcal{K}_{mm} \end{bmatrix} \in \mathbb{R}^{m(N-1)\times m(N-1)},$$

$$\begin{aligned}
\mathcal{K}_{ii} &= \overline{D_{Z_i}}\bar{\Omega}_1^1\overline{D_{Z_i}}^T - \overline{D_{A_i}}\bar{\Omega}_1^0\overline{D_{Z_i}}^T - \overline{D_{Z_i}}\bar{\Omega}_0^1\overline{D_{A_i}}^T \\
&\quad + \overline{D_{A_i}}\bar{\Omega}_0^0\overline{D_{A_i}}^T + \boldsymbol{I}_{N-1}/\gamma, \quad i = 1, \ldots, m, \\
\mathcal{K}_{ij} &= \overline{D_{Z_i}}\bar{\Omega}_1^1\overline{D_{Z_j}}^T - \overline{D_{A_i}}\bar{\Omega}_1^0\overline{D_{Z_j}}^T - \overline{D_{Z_i}}\bar{\Omega}_0^1\overline{D_{A_j}}^T \\
&\quad + \overline{D_{A_i}}\bar{\Omega}_0^0\overline{D_{A_j}}^T, \quad i,j = 1, \ldots, m \text{ and } i \neq j,
\end{aligned}$$

$$\mathcal{S} = \begin{bmatrix} \mathcal{S}_{11} & \cdots & \mathcal{S}_{1m} \\ \vdots & & \vdots \\ \mathcal{S}_{m1} & \cdots & \mathcal{S}_{mm} \end{bmatrix} \in \mathbb{R}^{m(N-1)\times m},$$

$$\mathcal{S}_{ij} = D_{Z_{ij}}\Omega_0^1(1,:)^T - D_{A_{ij}}\Omega_0^0(1,:)^T, \quad i,j = 1, \ldots, m,$$

$$\overline{D_{Z_i}} = [D_{Z_{i1}}, \ldots, D_{Z_{im}}], \qquad \overline{D_{A_i}} = [D_{A_{i1}}, \ldots, D_{A_{im}}],$$

$$\bar{\Omega}_l^k = \begin{bmatrix} \tilde{\Omega}_k^l & & \\ & \ddots & \\ & & \tilde{\Omega}_k^l \end{bmatrix} \in \mathbb{R}^{m(N-1)\times m(N-1)}, \quad k,l = 0, 1.$$

$D_{A_{ij}}$ and $D_{Z_{ij}}$ are diagonal matrices with the elements of $F_{A_{ij}}$ and $F_{Z_{ij}}$ on the main diagonal respectively. $\Omega_k^l(1,:) = [\Omega_k^l(1,2); \ldots; \Omega_k^l(1,N)]^T$ and $\tilde{\Omega}_k^l = \Omega_k^l(2:N, 2:N)$ for $k,l = 0, 1$. Also note that $\mathcal{K} = \mathcal{K}^T$.

Proof. Consider the Lagrangian of problem (8):

$$\begin{aligned}
&\mathcal{L}(w_\ell, b_\ell, e_\ell^i, \alpha_i^\ell, \beta_\ell) \\
&= \frac{1}{2}\sum_{\ell=1}^m w_\ell^T w_\ell + \frac{\gamma}{2}\sum_{\ell=1}^m e_\ell^T e_\ell \\
&\quad - \sum_{\ell=1}^m \Bigg[\sum_{i=2}^N \alpha_i^\ell \Bigg[w_\ell^T\left(z_{\ell\ell}(t_i)\varphi'(t_i) - a_{\ell\ell}(t_i)\varphi(t_i)\right) \\
&\quad - \sum_{\substack{k=1 \\ k\neq\ell}}^m w_k^T\left(z_{\ell k}(t_i)\varphi'(t_i) - a_{\ell k}(t_i)\varphi(t_i)\right) \\
&\quad - \sum_{k=1}^m a_{\ell k}(t_i)b_k - e_\ell^i - g_\ell^i \Bigg]\Bigg] \\
&\quad - \sum_{\ell=1}^m \beta_\ell\left(w_\ell^T\varphi(t_1) + b_\ell - p_\ell\right),
\end{aligned}$$

where $\{\alpha_i^\ell\}_{i=2}^N$, for $\ell = 1, \ldots, m$ and $\{\beta_\ell\}_{\ell=1}^m$ are Lagrange multipliers. $g_\ell^i = g_\ell(t_i)$, $e_\ell^i = e_\ell(t_i)$ for $i = 2, \ldots, N$ and $\ell = 1, \ldots, m$. Then the Karush–Kuhn–Tucker (KKT) optimality conditions are as follows,

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial w_\ell} &= 0 \to w_\ell = \sum_{v=1}^m\sum_{i=2}^N \alpha_i^v\left(z_{v\ell}(t_i)\varphi'(t_i) - a_{v\ell}(t_i)\varphi(t_i)\right) \\
&\quad + \beta_\ell\varphi(t_1), \quad \ell = 1, \ldots, m,
\end{aligned}$$

$$\frac{\partial \mathscr{L}}{\partial b_\ell} = 0 \to \sum_{k=1}^{m}\sum_{i=2}^{N} \alpha_i^k a_{k\ell}(t_i) - \beta_\ell = 0, \quad \ell = 1, \ldots, m,$$

$$\frac{\partial \mathscr{L}}{\partial e_\ell^i} = 0 \to e_\ell^i = -\frac{\alpha_i^\ell}{\gamma}, \quad i = 2, \ldots, N,\ \ell = 1, \ldots, m,$$

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \alpha_i^\ell} \ &= \ 0 \to \sum_{k=1}^{m} w_k^T \left(z_{\ell k}(t_i)\varphi'(t_i) - a_{\ell k}(t_i)\varphi(t_i)\right) \\ &\quad - \sum_{k=1}^{m} a_{\ell k}(t_i) b_k - e_\ell^i = g_\ell^i, \\ &\quad i = 2, \ldots, N,\ \ell = 1, \ldots, m\end{aligned}$$

$$\frac{\partial \mathscr{L}}{\partial \beta_\ell} = 0 \to w_\ell^T \varphi(t_1) + b_\ell = p_\ell, \quad \ell = 1, \ldots, m.$$

Eliminating $\{w_\ell\}_{\ell=1}^m$ and $\{e_\ell^i\}_{i=2}^N$ for $\ell = 1, \ldots, m$ from the corresponding KKT optimality conditions yields the following set of equations

$$\begin{cases} g_\ell^i = \sum_{v=1}^{m}\Big[\sum_{j=2}^{N}\alpha_j^v\Big[\sum_{k=1}^{m} z_{\ell k}(t_i)\Omega_1^1(j,i)z_{vk}(t_j) \\ \quad - \sum_{k=1}^{m} a_{\ell k}(t_i)\Omega_1^0(j,i)z_{vk}(t_j) - \sum_{k=1}^{m} z_{\ell k}(t_i)\Omega_0^1(j,i)a_{vk}(t_j) \\ \quad + \sum_{k=1}^{m} a_{\ell k}(t_i)\Omega_0^0(j,i)a_{vk}(t_j)\Big]\Big] \\ \quad + \sum_{v=1}^{m}\beta_v\left(z_{\ell v}(t_i)\Omega_0^1(1,i) - a_{\ell v}(t_i)\Omega_0^0(1,i)\right) \\ \quad + \frac{\alpha_i^\ell}{\gamma} - \sum_{k=1}^{m} a_{\ell k}(t_i) b_k, \quad i = 2, \ldots, N, \text{ and } \ell = 1, \ldots, m, \\ p_\ell = \sum_{v=1}^{m}\Big[\sum_{j=2}^{N}\alpha_j^v\left(z_{v\ell}(t_j)\Omega_1^0(j,1) - a_{v\ell}(t_j)\Omega_0^0(j,1)\right)\Big] \\ \quad + \beta_\ell\,\Omega_0^0(1,1) + b_\ell, \quad \ell = 1, \ldots, m, \\ 0 = \sum_{v=1}^{m}\sum_{j=2}^{N}\alpha_j^v\, a_{v\ell}(t_j) - \beta_\ell, \quad \ell = 1, \ldots, m. \end{cases}$$

Finally writing these equations in matrix form will result in the linear system (9). □

The model in the dual form becomes

$$\begin{aligned}\hat{x}_\ell(t) \ &= \ \sum_{v=1}^{m}\sum_{i=2}^{N}\alpha_i^v\left(z_{v\ell}(t_i)[\nabla_1^0 K](t_i,t) - a_{v\ell}(t_i)[\nabla_0^0 K](t_i,t)\right) \\ &\quad + \beta_\ell\,[\nabla_0^0 K](t_1,t) + b_\ell, \quad \ell = 1, \ldots, m,\end{aligned}$$

where K is the kernel function.

3.2. Explicit ODE system

Let us consider the system of IVPs (6) with Z as the identity matrix as a special case. This results in an ODE system.

Corollary 3.1. *Given a positive definite kernel function $K : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ with $K(t,s) = \varphi(t)^T\varphi(s)$ and a regularization constant $\gamma \in \mathbb{R}^+$, the solution to (8) with the matrix $Z = I$, is then given by the following dual problem:*

$$\left[\begin{array}{c|c|c} \mathcal{K} & \mathcal{S} & -F_A \\ \hline \mathcal{S}^T & \Omega_0^0(1,1)\times I_{m\times m} & I_{m\times m} \\ \hline -F_A^T & I_{m\times m} & 0_{m\times m}\end{array}\right]\left[\begin{array}{c}\alpha\\ \hline \beta \\ \hline b\end{array}\right] = \left[\begin{array}{c} G \\ \hline P \\ \hline 0\end{array}\right], \quad (10)$$

where α, β, b, R, P and F_A are defined as in Eq. (9). Also with

$$F_{Akl} = [a_{kl}(t_2); \cdots; a_{kl}(t_N)] \in \mathbb{R}^{N-1}, \quad \text{for } k, l = 1, \ldots, m$$

$$\mathcal{K} = \begin{bmatrix} \mathcal{K}_{11} & \ldots & \mathcal{K}_{1m} \\ \vdots & & \vdots \\ \mathcal{K}_{m1} & \ldots & \mathcal{K}_{mm}\end{bmatrix} \in \mathbb{R}^{m(N-1)\times m(N-1)},$$

$$\mathcal{K}_{ii} = \tilde{\Omega}_1^1 - D_{ii}\tilde{\Omega}_1^0 - \tilde{\Omega}_0^1 D_{ii} + T_{ii} + \boldsymbol{I}_{N-1}/\gamma, \quad i = 1, \ldots, m,$$

$$\mathcal{K}_{ij} = -\tilde{\Omega}_0^1 D_{ji} - D_{ij}\tilde{\Omega}_1^0 + T_{ij}, \quad i, j = 1, \ldots, m \text{ and } i \neq j,$$

$$\mathcal{S} = \begin{bmatrix} \mathcal{S}_{11} & \ldots & \mathcal{S}_{1m} \\ \vdots & & \vdots \\ \mathcal{S}_{m1} & \ldots & \mathcal{S}_{mm}\end{bmatrix} \in \mathbb{R}^{m(N-1)\times m},$$

$$\mathcal{S}_{ii} = \Omega_0^1(1,:)^T - D_{ii}\Omega_0^0(1,:)^T, \quad i = 1, \ldots, m,$$

$$\mathcal{S}_{ij} = -D_{ij}\Omega_0^0(1,:)^T, \quad i, j = 1, \ldots, m \text{ and } i \neq j,$$

$$T_{ij} = \bar{D}_i\bar{\Omega}\bar{D}_j^T \in \mathbb{R}^{(N-1)\times(N-1)} \quad i, j = 1, \ldots, m,$$

$$\bar{D}_i = [D_{i1}, \ldots, D_{im}], \qquad \bar{D}_j = [D_{j1}, \ldots, D_{jm}],$$

$$\bar{\Omega} = \begin{bmatrix} \tilde{\Omega}_0^0 & & \\ & \ddots & \\ & & \tilde{\Omega}_0^0\end{bmatrix} \in \mathbb{R}^{m(N-1)\times m(N-1)}.$$

D_{ij} is a diagonal matrix with the elements of F_{ij} on the main diagonal. $\Omega_n^m(1,:) = [\Omega_n^m(1,2); \ldots; \Omega_n^m(1,N)]^T$ and $\tilde{\Omega}_n^m = \Omega_n^m(2:N, 2:N)$ for $n, m = 0, 1$. Also note that $\mathcal{K} = \mathcal{K}^T$.

The model in the dual form becomes

$$\begin{aligned}\hat{x}_\ell(t) \ &= \ \sum_{i=2}^{N}\alpha_i^\ell\left([\nabla_1^0 K](t_i,t) - a_{\ell\ell}(t_i)[\nabla_0^0 K](t_i,t)\right) \\ &\quad - \sum_{\substack{v=1\\ v\neq\ell}}^{m}\sum_{i=2}^{N}\alpha_i^v\left(a_{v\ell}(t_i)[\nabla_0^0 K](t_i,t)\right) \\ &\quad + \beta_\ell\,[\nabla_0^0 K](t_1,t) + b_\ell, \quad \ell = 1, \ldots, m,\end{aligned}$$

where K is the kernel function.

4. Formulation of the method for BVPs in DAEs

Consider the linear time varying boundary value problem in DAEs of the following form

$$\begin{aligned} &Z(t)\dot{X}(t) = A(t)X(t) + g(t), \quad t \in [t_{in}, t_f], \\ &FX(t_{in}) + HX(t_f) = X_0, \end{aligned} \quad (11)$$

where matrices $Z(t)$, $A(t)$ and the state vector $X(t)$ are defined as in Eq. (6). $F = [f_{ij}]$ and $H = [h_{ij}] \in \mathbb{R}^{m\times m}$. The input is $g(t)$ and $\dot{X}(t) = \frac{dX}{dt}$. $Z(t)$ may be singular on $[t_{in}, t_f]$ with variable rank and the DAE may have an index that is larger than one. As before we assume that $Z(t)$, $A(t)$ and $g(t)$ are sufficiently smooth and the DAE (11) is solvable. When Z is nonsingular, Eq. (11) can be converted to the equivalent explicit ODE system.

The approximate solution, $\hat{x}_i(t) = w_i^T\varphi(t) + b_i$, for $i = 1, \ldots, m$ is then obtained by solving the following optimization problem,

$$\underset{w_i, b_i, e_\ell^i, \xi_\ell}{\text{minimize}} \ \frac{1}{2}\sum_{\ell=1}^{m} w_\ell^T w_\ell + \frac{\gamma}{2}\sum_{\ell=1}^{m} e_\ell^T e_\ell + \frac{\zeta}{2}\sum_{\ell=1}^{m}\xi_\ell^2$$

$$\text{subject to} \begin{bmatrix} z_{11}(t) & \cdots & z_{1m}(t) \\ \vdots & \ddots & \vdots \\ z_{m1}(t) & \cdots & z_{mm}(t)\end{bmatrix}\begin{bmatrix} w_1^T\varphi'(t_i) \\ \vdots \\ w_m^T\varphi'(t_i)\end{bmatrix}$$

$$
\begin{aligned}
&= \begin{bmatrix} a_{11}(t) & \cdots & a_{1m}(t) \\ \vdots & \ddots & \vdots \\ a_{m1}(t) & \cdots & a_{mm}(t) \end{bmatrix} \begin{bmatrix} w_1^T\varphi(t_i)+b_1 \\ \vdots \\ w_m^T\varphi(t_i)+b_m \end{bmatrix} \\
&\quad + \begin{bmatrix} g_1(t_i) \\ \vdots \\ g_m(t_i) \end{bmatrix} + \begin{bmatrix} e_1(t_i) \\ \vdots \\ e_m(t_i) \end{bmatrix}, \quad \text{for } i = 2, \ldots, N-1, \\
&\begin{bmatrix} f_{11} & \cdots & f_{1m} \\ \vdots & \ddots & \vdots \\ f_{m1} & \cdots & f_{mm} \end{bmatrix} \begin{bmatrix} w_1^T\varphi(t_1)+b_1 \\ \vdots \\ w_m^T\varphi(t_1)+b_m \end{bmatrix} \\
&\quad + \begin{bmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{m1} & \cdots & h_{mm} \end{bmatrix} \begin{bmatrix} w_1^T\varphi(t_N)+b_1 \\ \vdots \\ w_m^T\varphi(t_N)+b_m \end{bmatrix} \\
&= \begin{bmatrix} p_1 \\ \vdots \\ p_m \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_m \end{bmatrix}. \qquad (12)
\end{aligned}
$$

Lemma 4.1. *Given a positive definite kernel function $K : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ with $K(t,s) = \varphi(t)^T\varphi(s)$ and a regularization constant $\gamma, \xi \in \mathbb{R}^+$, the solution to (12) is given by the following dual problem:*

$$
\left[\begin{array}{c|c|c} \mathcal{K} & \mathcal{U} & -F_A \\ \hline \mathcal{U}^T & \Delta & \Pi \\ \hline -F_A^T & \Pi^T & 0_{m\times m} \end{array}\right] \left[\begin{array}{c} \alpha \\ \hline \beta \\ \hline b \end{array}\right] = \left[\begin{array}{c} G \\ \hline P \\ \hline 0 \end{array}\right] \qquad (13)
$$

with

$$
\begin{aligned}
&\alpha = \left[\begin{array}{c} \alpha^1 \\ \hline \vdots \\ \hline \alpha^m \end{array}\right] = [\alpha_2^1; \ldots; \alpha_{N-1}^1; \cdots; \alpha_2^m; \ldots; \alpha_{N-1}^m] \in \mathbb{R}^{m(N-2)}, \\
&\beta = [\beta_1; \cdots; \beta_m], \qquad b = [b_1; \ldots; b_m], \\
&P = [p_1; \ldots; p_m], \\
&G = [g_1(t_2); \cdots; g_1(t_{N-1}); \cdots; g_m(t_2); \cdots; g_m(t_{N-1})] \in \mathbb{R}^{m(N-2)}, \\
&F_A = \begin{bmatrix} F_{A_{11}} & \ldots & F_{A_{1m}} \\ \vdots & & \vdots \\ F_{A_{m1}} & \ldots & F_{A_{mm}} \end{bmatrix} \in \mathbb{R}^{m(N-2)\times m}, \\
&F_{A_{kl}} = [a_{kl}(t_2); \cdots; a_{kl}(t_{N-1})] \in \mathbb{R}^{N-2}, \quad \textit{for } k,l = 1, \ldots, m, \\
&F_{Z_{kl}} = [z_{kl}(t_2); \cdots; z_{kl}(t_{N-1})] \in \mathbb{R}^{N-2}, \quad \textit{for } k,l = 1, \ldots, m, \\
&\Pi = F + H, \\
&\Delta = \left(FF^T\right)\Omega_0^0(1,1) + \left(FH^T + HF^T\right)\Omega_0^0(1,N) \\
&\qquad + \left(HH^T\right)\Omega_0^0(N,N) + \boldsymbol{I}_m/\zeta, \\
&\mathcal{K} = \begin{bmatrix} \mathcal{K}_{11} & \ldots & \mathcal{K}_{1m} \\ \vdots & & \vdots \\ \mathcal{K}_{m1} & \ldots & \mathcal{K}_{mm} \end{bmatrix} \in \mathbb{R}^{m(N-2)\times m(N-2)}, \\
&\mathcal{K}_{ii} = \overline{D_{Z_i}}\bar{\Omega}_1^1\overline{D_{Z_i}}^T - \overline{D_{A_i}}\bar{\Omega}_1^0\overline{D_{Z_i}}^T - \overline{D_{Z_i}}\bar{\Omega}_0^1\overline{D_{A_i}}^T \\
&\qquad + \overline{D_{A_i}}\bar{\Omega}_0^0\overline{D_{A_i}}^T + \boldsymbol{I}_{N-1}/\gamma, \quad i = 1, \ldots, m, \\
&\mathcal{K}_{ij} = \overline{D_{Z_i}}\bar{\Omega}_1^1\overline{D_{Z_j}}^T - \overline{D_{A_i}}\bar{\Omega}_1^0\overline{D_{Z_j}}^T - \overline{D_{Z_i}}\bar{\Omega}_0^1\overline{D_{A_j}}^T \\
&\qquad + \overline{D_{A_i}}\bar{\Omega}_0^0\overline{D_{A_j}}^T, \quad i,j = 1, \ldots, m \text{ and } i \neq j,
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{U} = \begin{bmatrix} \mathcal{U}_{11} & \ldots & \mathcal{U}_{1m} \\ \vdots & & \vdots \\ \mathcal{U}_{m1} & \ldots & \mathcal{U}_{mm} \end{bmatrix} \in \mathbb{R}^{m(N-2)\times m}, \\
&\mathcal{U}_{ij} = \left(\sum_{k=1}^m D_{Z_{ik}} f_{jk}\right)\Omega_0^1(1,:)^T + \left(\sum_{k=1}^m D_{Z_{ik}} h_{jk}\right)\Omega_0^1(N,:)^T \\
&\qquad - \left(\sum_{k=1}^m D_{A_{ik}} f_{jk}\right)\Omega_0^0(1,:)^T - \left(\sum_{k=1}^m D_{A_{ik}} h_{jk}\right)\Omega_0^0(N,:)^T, \\
&\qquad i,j = 1, \ldots, m, \\
&\overline{D_{Z_i}} = [D_{Z_{i1}}, \ldots, D_{Z_{im}}], \qquad \overline{D_{A_i}} = [D_{A_{i1}}, \ldots, D_{A_{im}}], \\
&\bar{\Omega}_l^k = \begin{bmatrix} \tilde{\Omega}_k^l & & \\ & \ddots & \\ & & \tilde{\Omega}_k^l \end{bmatrix} \in \mathbb{R}^{m(N-2)\times m(N-2)}, \quad k,l = 0, 1.
\end{aligned}
$$

$D_{A_{ij}}$ and $D_{Z_{ij}}$ are diagonal matrices with the elements of $F_{A_{ij}}$ and $F_{Z_{ij}}$ on the main diagonal respectively. $\Omega_k^l(1,:) = [\Omega_k^l(1,2); \ldots; \Omega_k^l(1,N-1)]^T$, $\Omega_k^l(N,:) = [\Omega_k^l(N,2); \ldots; \Omega_k^l(N,N-1)]^T$ and $\tilde{\Omega}_k^l = \Omega_k^l(2:N-1, 2:N-1)$ for $k,l = 0, 1$. Also note that $\mathcal{K} = \mathcal{K}^T$.

Proof. Consider the Lagrangian of problem (12):

$$
\begin{aligned}
\mathscr{L}(w_\ell, b_\ell, e_\ell^i, \xi_\ell, \alpha_i^\ell, \beta_\ell) &= \frac{1}{2}\sum_{\ell=1}^m w_\ell^T w_\ell + \frac{\gamma}{2}\sum_{\ell=1}^m e_\ell^T e_\ell \\
&\quad + \frac{\zeta}{2}\sum_{\ell=1}^m \xi_\ell^2 - \sum_{\ell=1}^m \Bigg[\sum_{i=2}^{N-1}\alpha_i^\ell \Bigg[w_\ell^T\left(z_{\ell\ell}(t_i)\varphi'(t_i) - a_{\ell\ell}(t_i)\varphi(t_i)\right) \\
&\quad - \sum_{\substack{k=1 \\ k\neq\ell}}^m w_k^T\left(z_{\ell k}(t_i)\varphi'(t_i) - a_{\ell k}(t_i)\varphi(t_i)\right) \\
&\quad - \sum_{k=1}^m a_{\ell k}(t_i)b_k - e_\ell^i - g_\ell^i \Bigg]\Bigg] \\
&\quad - \sum_{\ell=1}^m \beta_\ell \Bigg[\sum_{i=1}^m w_i^T\left(f_{\ell i}\varphi(t_1) + h_{\ell i}\varphi(t_N)\right) \\
&\quad + \sum_{i=1}^m b_i\left(f_{\ell i} + h_{\ell i}\right) - p_\ell - \xi_\ell \Bigg],
\end{aligned}
$$

where $\{\alpha_i^\ell\}_{i=2}^{N-1}$, for $\ell = 1, \ldots, m$ and $\{\beta_\ell\}_{\ell=1}^m$ are Lagrange multipliers. $g_\ell^i = g_\ell(t_i)$, $e_\ell^i = e_\ell(t_i)$ for $i = 2, \ldots, N-1$ and $\ell = 1, \ldots, m$. Then the Karush–Kuhn–Tucker (KKT) optimality conditions are as follows,

$$
\begin{aligned}
\frac{\partial\mathscr{L}}{\partial w_\ell} &= 0 \to w_\ell = \sum_{v=1}^m\sum_{i=2}^{N-1}\alpha_i^v\left(z_{v\ell}(t_i)\varphi'(t_i) - a_{v\ell}(t_i)\varphi(t_i)\right) \\
&\quad + \sum_{j=1}^m \beta_j\left(f_{j\ell}\varphi(t_1) + h_{j\ell}\varphi(t_N)\right), \quad \ell = 1, \ldots, m, \\
\frac{\partial\mathscr{L}}{\partial b_\ell} &= 0 \to \sum_{k=1}^m\sum_{i=2}^{N-1}\alpha_i^k a_{k\ell}(t_i) - \sum_{j=1}^m \beta_j\left(f_{j\ell} + h_{j\ell}\right) = 0, \\
&\quad \ell = 1, \ldots, m, \\
\frac{\partial\mathscr{L}}{\partial e_\ell^i} &= 0 \to e_\ell^i = -\frac{\alpha_i^\ell}{\gamma}, \quad i = 2, \ldots, N-1, \quad \ell = 1, \ldots, m, \\
\frac{\partial\mathscr{L}}{\partial \xi_\ell} &= 0 \to \xi_\ell = -\frac{\beta_\ell}{\zeta}, \quad \ell = 1, \ldots, m,
\end{aligned}
$$

$$\frac{\partial \mathscr{L}}{\partial \alpha_i^\ell} = 0 \to \sum_{k=1}^{m} w_k^T \left(z_{\ell k}(t_i)\varphi'(t_i) - a_{\ell k}(t_i)\varphi(t_i)\right) - \sum_{k=1}^{m} a_{\ell k}(t_i) b_k - e_\ell^i = g_\ell^i, \quad i = 2, \ldots, N-1,\ \ell = 1, \ldots, m$$

$$\frac{\partial \mathscr{L}}{\partial \beta_\ell} = 0 \to \sum_{i=1}^{m} w_i^T \left(f_{\ell i}\varphi(t_1) + h_{\ell i}\varphi(t_N)\right) + \sum_{i=1}^{m} b_i \left(f_{\ell i} + h_{\ell i}\right) - p_\ell - \xi_\ell = 0, \quad \ell = 1, \ldots, m.$$

Eliminating $\{w_\ell\}_{\ell=1}^m$, $\{e_\ell^i\}_{i=2}^{N-1}$ and ξ_ℓ for $\ell = 1, \ldots, m$ from the corresponding KKT optimality conditions yields the following set of equations:

$$\begin{aligned}
p_\ell &= \sum_{v=1}^{m}\Bigg[\sum_{j=2}^{N-1}\alpha_j^v\Bigg[\left(\sum_{k=1}^{m} z_{\ell k}(t_j) f_{vk}\right)\Omega_1^0(j,1) + \left(\sum_{k=1}^{m} z_{\ell k}(t_j) h_{vk}\right)\Omega_1^0(j,N) \\
&\quad - \left(\sum_{k=1}^{m} a_{\ell k}(t_j) f_{vk}\right)\Omega_0^0(j,1) - \left(\sum_{k=1}^{m} a_{\ell k}(t_j) h_{vk}\right)\Omega_0^0(j,N)\Bigg]\Bigg] \\
&\quad + \sum_{j=1}^{m}\beta_j\Bigg[\left(\sum_{k=1}^{m} f_{jk} f_{\ell k}\right)\Omega_0^0(1,1) + \left(\sum_{k=1}^{m} f_{jk} h_{\ell k} + \sum_{k=1}^{m} h_{jk} f_{\ell k}\right)\Omega_0^0(1,N) \\
&\quad + \left(\sum_{k=1}^{m} h_{jk} h_{\ell k}\right)\Omega_0^0(N,N)\Bigg] + \sum_{j=1}^{m} b_j \left(f_{\ell j} + h_{\ell j}\right) + \frac{\beta_\ell}{\xi}, \quad \ell = 1, \ldots, m,
\end{aligned}$$

$$\begin{aligned}
g_\ell^i &= \sum_{v=1}^{m}\Bigg[\sum_{j=2}^{N-1}\alpha_j^v\Bigg[\sum_{k=1}^{m} z_{\ell k}(t_i)\Omega_1^1(j,i) z_{vk}(t_j) - \sum_{k=1}^{m} a_{\ell k}(t_i)\Omega_1^0(j,i) z_{vk}(t_j) - \sum_{k=1}^{m} z_{\ell k}(t_i)\Omega_0^1(j,i) a_{vk}(t_j) \\
&\quad + \sum_{k=1}^{m} a_{\ell k}(t_i)\Omega_0^0(j,i) a_{vk}(t_j)\Bigg]\Bigg] + \sum_{v=1}^{m}\beta_v\Bigg[\left(\sum_{k=1}^{m} z_{\ell k}(t_i) f_{vk}\right)\Omega_0^1(1,i) + \left(\sum_{k=1}^{m} z_{\ell k}(t_i) h_{vk}\right)\Omega_0^1(N,i) \\
&\quad - \left(\sum_{k=1}^{m} a_{\ell k}(t_i) f_{vk}\right)\Omega_0^0(1,i) - \left(\sum_{k=1}^{m} a_{\ell k}(t_i) h_{vk}\right)\Omega_0^0(N,i)\Bigg] + \frac{\alpha_i^\ell}{\gamma} - \sum_{k=1}^{m} a_{\ell k}(t_i) b_k, \\
&\quad i = 2, \ldots, N-1, \text{ and } \ell = 1, \ldots, m,
\end{aligned}$$

$$0 = \sum_{v=1}^{m}\sum_{j=2}^{N-1}\alpha_j^v\, a_{v\ell}(t_j) - \sum_{j=1}^{m} b_j \left(f_{j\ell} + h_{j\ell}\right), \quad \ell = 1, \ldots, m.$$

Finally writing these equations in the matrix form will result in the linear system (9). □

The model in the dual form becomes

$$\hat{x}_\ell(t) = \sum_{v=1}^{m}\sum_{i=2}^{N-1}\alpha_i^v\left(z_{v\ell}(t_i)[\nabla_1^0 K](t_i, t) - a_{v\ell}(t_i)[\nabla_0^0 K](t_i, t)\right) + \sum_{v=1}^{m}\beta_v\left([\nabla_0^0 K](t_1, t) f_{v\ell} + [\nabla_0^0 K](t_N, t) h_{v\ell}\right) + b_\ell, \quad \ell = 1, \ldots, m,$$

where K is the kernel function.

5. Parameter tuning

The performance of the LS-SVM model depends on the choice of the tuning parameters. In this paper for all experiments the Gaussian RBF kernel is used. Therefore a model is determined by the regularization parameter γ and the kernel bandwidth σ.

It should be noted that as opposed to the regression case, we do not have the target values and we do not have noise. Therefore a quite large value can be taken for the regularization constant γ, so that the optimization problem (7) is sharply minimized, compared to the noisy case. In this work, the optimal values for γ and σ are obtained by evaluating the performance of the model on a validation set using a meaningful grid of possible (γ, σ) combinations. The validation set is defined to be the set of midpoints $V \equiv \{v_i = \frac{(t_i + t_{i+1})}{2}, i = 1, \ldots, N-1\}$. The values that minimize the mean squared error (MSE) on this validation set are then selected.

6. Numerical results

In this section, four experiments are performed to demonstrate the capability of the proposed method for solving initial and boundary value problems in DAEs. The accuracy of an approximate solution is measured by means of the mean square error (MSE) which is defined as follows:

$$\mathrm{MSE}_{\mathrm{test}} = \frac{\sum_{i=1}^{M}(x(t_i) - \hat{x}(t_i))^2}{M},$$

where M is the number of test points. In all the experiments M is set to 200 points on the given domain.

Problem 6.1. Consider the following nonsingular system of time varying ordinary differential equations

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ \frac{-2}{t^2} & \frac{2}{t^2}\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ t\log(t)\end{bmatrix} \tag{14}$$

subject to

$$x_1(1) = 1, \qquad x_2(1) = 0.$$

This problem is solved for $t \in [1, 10]$ and the approximate solution obtained by the proposed method is compared with the solution obtained by MATLAB built-in solver ode45 in Fig. 1. The obtained results with different number of training points are tabulated in Table 1. Note that the subroutine DSolve of Mathematica 6.0 failed to find the analytical solution for the above equation.

Problem 6.2. Consider the singular system of index-3 discussed as follows (Murugesan & Balakumar, 2011):

$$Z(t)\dot{X}(t) = A(t)X(t) + B(t)u(t), \quad t \in [0, 20],\ X(0) = X_0, \tag{15}$$

where

$$Z = \begin{bmatrix}0 & -t & 0\\ 1 & 0 & t\\ 0 & 1 & 0\end{bmatrix}, \qquad A = \begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\end{bmatrix}$$

Table 1
Numerical results of the proposed method for solving Problem 6.1 on time interval [1, 10], with N number of collocation points.

N	$\mathbf{MSE}_{\text{test}}$	
	x_1	x_2
20	3.1×10^{-2}	1.2×10^{-3}
40	3.9×10^{-5}	1.4×10^{-6}
60	2.6×10^{-7}	1.1×10^{-8}
80	4.8×10^{-9}	3.5×10^{-10}

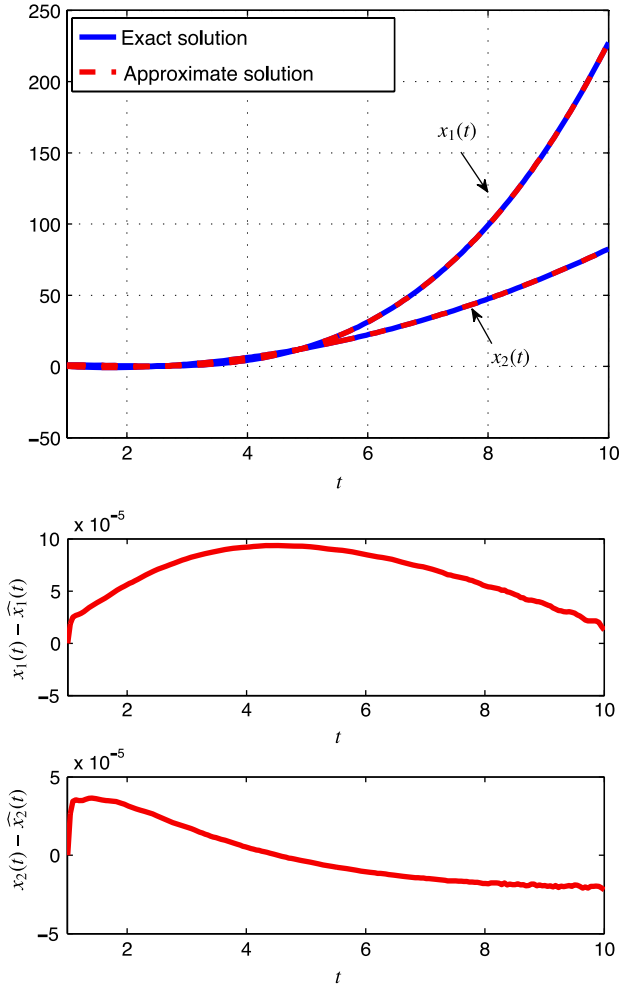

Fig. 1. The obtained approximate solution and model errors for Problem 6.1, when 80 equidistant mesh points on the interval [1, 10] are used for training.

and $B(t) = 0$ with $x(0) = [0, e^{-1}, e^{-1}]^T$. The exact solution is given by

$$x_1(t) = -t\exp(-(t+1)),$$
$$x_2(t) = x_3(t) = \exp(-(t+1)).$$

The problem is solved on domain $t \in [0, 20]$ for different N (number of collocation points) values. The approximate solution obtained by the proposed method is compared with the exact solution (see Fig. 2) and the results are recorded in Table 2. From Table 2, it is apparent that as N increases, the solution converges to the true solution. Note that the MATLAB built-in solver ode15i can solve DAEs up to index-1.

Problem 6.3. Consider the linear time varying singular system (Murugesan, Dhayabaran, & Amirtharaj, 2002)

$$\begin{aligned} &Z(t)\dot{X}(t) = A(t)X(t) + B(t)u(t), \quad t \in [0, 10],\ X(0) = X_0 \\ &y(t) = K(t)x(t), \end{aligned} \tag{16}$$

Table 2
Numerical results of the proposed method for solving Problem 6.2 on time interval [0, 20], with N number of collocation points.

N	$\mathbf{MSE}_{\text{test}}$		
	x_1	x_2	x_3
20	1.33×10^{-5}	4.82×10^{-8}	4.73×10^{-7}
40	1.38×10^{-8}	1.39×10^{-10}	3.14×10^{-9}
60	4.82×10^{-10}	3.54×10^{-12}	2.38×10^{-10}

Table 3
Numerical results of the proposed method for solving Problem 6.3 on time interval [0, 10], with N number of collocation points.

N	$\mathbf{MSE}_{\text{test}}$	
	y_1	y_2
10	1.23×10^{-3}	8.76×10^{-4}
20	1.25×10^{-5}	1.31×10^{-5}
30	1.42×10^{-6}	7.80×10^{-7}
40	5.35×10^{-9}	7.62×10^{-9}

where $y(t)$ is the output vector and

$$Z = \begin{bmatrix} 1+t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1+t & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -(1+t) & 1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & 1+t & 0 & -1 \\ 0 & 0 & 0 & (1+e^{-t})\sin(t) \end{bmatrix}$$

with $u = [1 + t + t^2/2, 0]^T$ and $x(0) = [0, -1, 0, -1]^T$ the exact solution is given by

$$y_1(t) = -\left(1 + t + t^2 + \frac{t^3}{3}\right) + \frac{1 + t + 3t^2/2 + t^3 + t^4/4}{1+t},$$
$$y_2(t) = -(1 + e^{-t})\sin(t)\left(\frac{1 + t + 3t^2/2 + t^3 + t^4/4}{1+t}\right).$$

The interval [0, 10] is discretized into N number of mesh points. The obtained mean square errors for test set are tabulated in Table 3. The results reveal that higher order accuracy can be achieved by an increasing number of collocation points. The analytical solution and the obtained approximate solution, with 50 equidistant mesh points on the interval [0, 10] as training points, are compared in Fig. 3. Note that the MATLAB built-in solver ode15i failed to solve Problem 6.3.

Problem 6.4. Consider the linear time varying index one boundary value problem of DAE given by Clark and Petzold (1989):

$$Z(t)\dot{X}(t) = A(t)X(t) + g(t), \quad t \in [0, 1], \tag{17}$$

where

$$Z = \begin{bmatrix} 1 & -t & t^2 \\ 0 & 1 & -t \\ 0 & 0 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & (t+1) & -(t^2+2t) \\ 0 & 1 & 1-t \\ 0 & 0 & -1 \end{bmatrix}$$

with $g(t) = [0, 0, \sin(t)]^T$ and boundary conditions

$$x_1(0) = 1, \qquad x_2(1) - x_3(1) = e.$$

The exact solution is given by

$$x_1(t) = e^{-t} + te^t,$$
$$x_2(t) = e^t + t\sin(t),$$
$$x_3(t) = \sin(t).$$

The computed residuals are displayed in Fig. 4. The mean square errors for the test set are also recorded in Table 4.

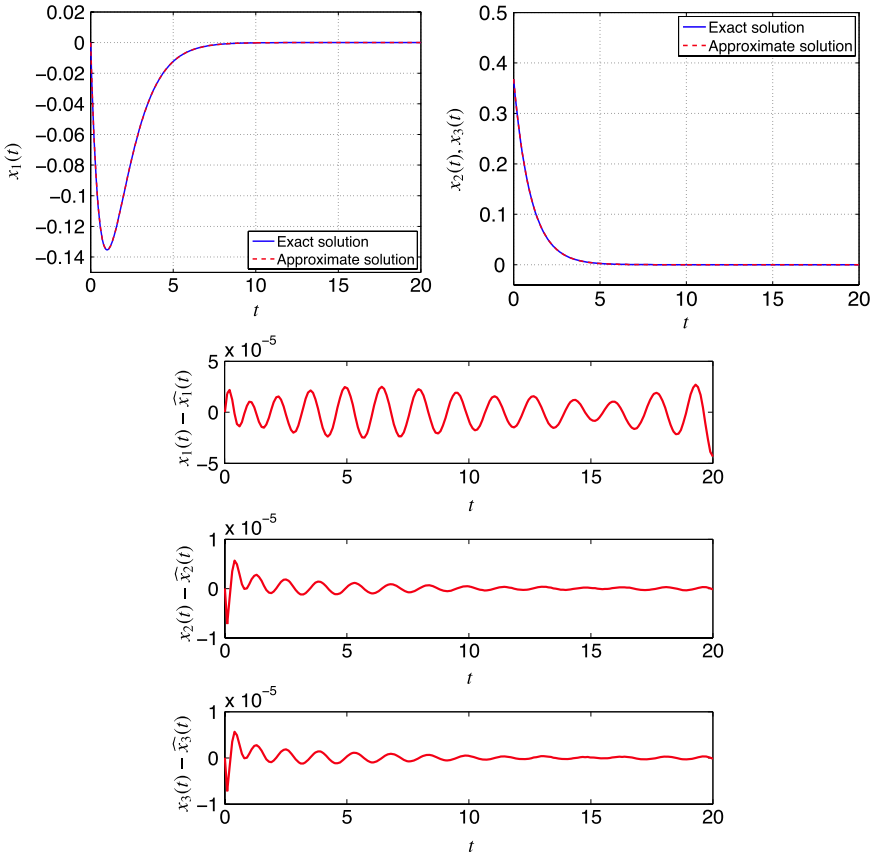

Fig. 2. The obtained approximate solution and model errors for Problem 6.2, when 70 equidistant mesh points on the interval [0, 20] are used for training.

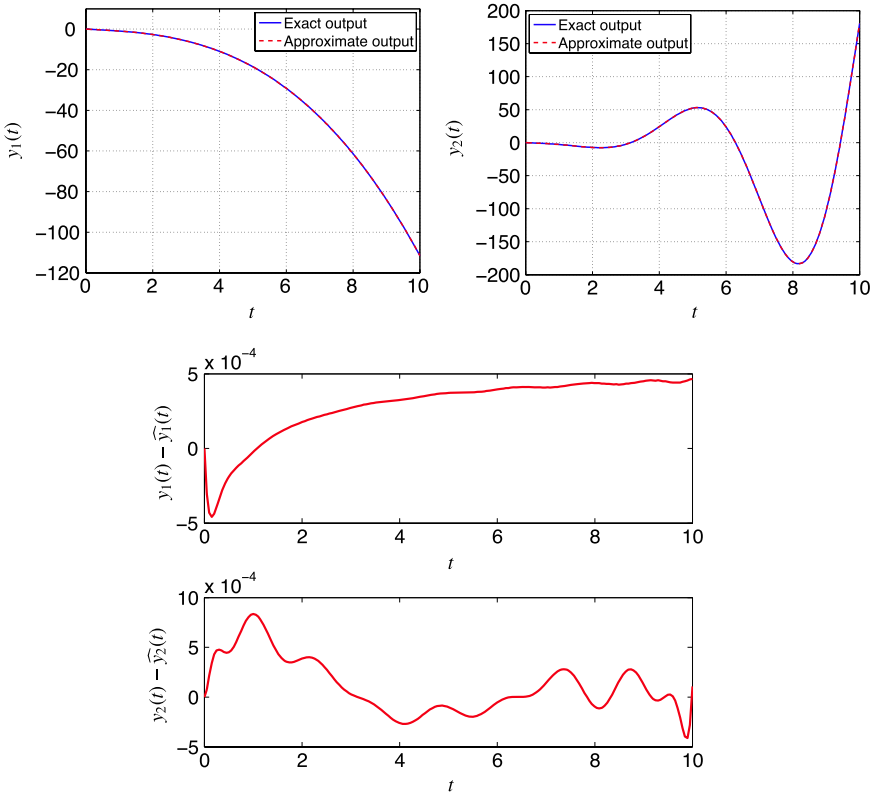

Fig. 3. The obtained approximate solution and model errors for Problem 6.3, when 50 equidistant mesh points on the interval [0, 10] are used for training.

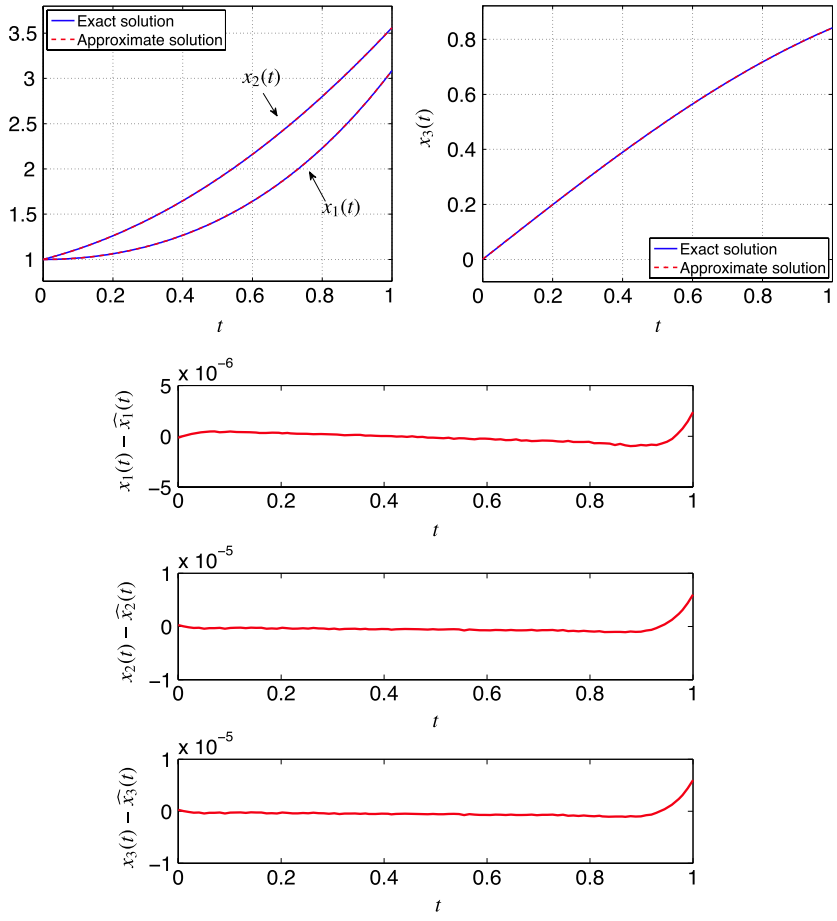

Fig. 4. The obtained approximate solution and model errors for Problem 6.4, when 10 equidistant mesh points on the interval [0, 1] are used for training.

Table 4
Numerical results of the proposed method for solving Problem 6.4 on time interval [0,1], with N number of collocation points.

N	$\mathbf{MSE}_{\text{test}}$		
	x_1	x_2	x_3
10	3.81×10^{-5}	2.65×10^{-5}	6.02×10^{-6}
20	2.60×10^{-13}	1.05×10^{-12}	1.65×10^{-12}
30	9.77×10^{-15}	2.78×10^{-13}	3.74×10^{-13}
40	1.04×10^{-15}	7.43×10^{-15}	6.12×10^{-15}

Remark 2. A singular system with a discontinuous input will exhibit a jump. The LS-SVM approximation with Gaussian kernel (which provides a smooth approximation) shows a spurious oscillation near the discontinuity. This oscillation behavior is a common phenomenon known as the Gibbs phenomenon that appears when the underlying function being approximated has jump discontinuities. Some methods have been suggested in the literature to reduce the effect of the Gibbs phenomenon (see Jerri, 2011). Another approach is to use a continuous approximation of the non smooth input signal as the new input for the system (Rabier & Rheinboldt, 1996).

Remark 3. Concerning practical application of the proposed method for finding the approximate solution to the given DAEs on a very long time interval, the approach described in Mehrkanoon et al. (2012) which is based on the domain decomposition technique can be utilized here as well.

Remark 4. In general, but not necessarily, one starts with equidistant mesh points. The residual error on the validation set is then monitored and for the subintervals in which the error is not sufficiently small one may refine the mesh in order to achieve adequate accuracy. The residual error at a given point t_i can be computed as

$$e(t_i) = Z(t_i)\dot{X}(t_i) - A(t_i)X(t_i) - g(t_i). \tag{18}$$

7. Conclusion and future work

In this paper, a new method based on least squares support vector machines is developed for solving linear time varying differential algebraic equations. The method is successfully applied for solving higher index DAEs (index larger than one). The results reveal that the method can obtain the desired accuracy with only a few number of collocation points. Also the method is stable when the non uniform sampling is used to create the collocation points.

References

Ascher, U. M. (1989). On symmetric schemes and differential–algebraic equations. *SIAM Journal on Scientific and Statistical Computing*, *10*, 937–949.
Ascher, U. M., & Petzold, L. R. (1991). Projected implicit Runge–Kutta methods for differential algebraic equations. *SIAM Journal on Numerical Analysis*, *28*, 1097–1120.
Ascher, U. M., & Petzold, L. R. (1998). *Computer methods for ordinary differential equations and differential algebraic equations*. Philadelphia: Society for Industrial and Applied Mathematics.
Ascher, U. M., & Spiter, R. J. (1994). Collocation software for boundary-value differential algebraic equations. *SIAM Journal on Scientific Computing*, *15*, 938–952.
Awawdeh, F., Jaradat, H. M., & Alsayyed, O. (2009). Solving system of DAEs by homotopy analysis method. *Chaos, Solitons and Fractals*, *42*(3), 1422–1427.
Bernan, K. (1986). Numerical simulation of trajectory prescribed path control problems. *IEEE Transactions on Automatic Control*, *AC-31*, 266–269.

Brenan, K. E., Campbell, S. L., & Petzold, L. R. (1989). *Numerical solution of initial-value problems in differential–algebraic equations*. New York: Elsevier.
Campbell, S. L. (1982). *Singular systems of differential equations II*. Marshfield, MA: Pitman.
Clark, K. D., & Petzold, L. R. (1989). Numerical solution of boundary value problems in differential–algebraic systems. *SIAM Journal on Scientific and Statistical Computing*, *10*(5), 915–936.
Froment, G. B., & Bischoff, K. B. (1990). *Chemical reactor analysis and design*. New York: John Wiley & Sons Inc.
Gear, C. W., & Petzold, L. R. (1984). ODE systems for the solution of differential–algebraic systems. *SIAM Journal on Numerical Analysis*, *21*, 716–728.
Guzel, N., & Bayram, M. (2006). Numerical solution of differential–algebraic equations with index-2. *Applied Mathematics and Computation*, *174*, 1279–1289.
Jerri, A. J. (2011). *Advances in the Gibbs phenomenon* (1st ed.). New York: Sampling Publishing.
Kincaid, D. R., & Cheney, E. W. (2002). *Numerical analysis: mathematics of scientific computing* (3rd ed.). Pacific Grove, CA: Brooks/Cole.
Lötstedt, P., & Petzold, L. (1986). Numerical solution of nonlinear differential equations with algebraic constraints: convergence results for the backward differentiation formulas. *Mathematics of Computation*, *46*, 491–516.
Mehrkanoon, S., Falck, T., & Suykens, J. A. K. (2012). Approximate solutions to ordinary differential equations using least squares support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(9), 1356–1367.
Murugesan, K., & Balakumar, V. (2011). Study on singular systems of index three via Runge–Kutta method of order-10. *International Journal of Pure and Applied Mathematics*, *70*(5), 723–733.
Murugesan, K., Dhayabaran, D. P., & Amirtharaj, E. C. H (2002). A study of second-order state-space systems of time-invariant and time-varying transistor circuits using the STWS technique. *International Journal of Electronics*, *89*(4), 305–315.
Murugesh, V., & Murugesan, K. (2009). RK-Butcher algorithms for singular system-based electronic circuit. *International Journal of Computer Mathematics*, *86*(3), 523–536.
Pengfei, L., Yaoyu, L., & Seem, J. E. (2012). Consistent initialization of system of differentialalgebraic equations for dynamic simulation of centrifugal chillers. *Journal of Building Performance Simulation*, *5*(2), 115–139.
Rabier, P. J., & Rheinboldt, W. C. (1996). Time-dependent linear DAEs with discontinuous inputs. *Linear Algebra and its Applications*, *247*(1), 1–29.
Ranade, V. V. (2002). *Computational fow modeling for chemical engineering*. San Diego: Academic Press.
Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., & Vandewalle, J. (2002). *Least squares support vector machines*. Singapore: World Scientific Pub. Co.
Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.
Wang, H. S., Jhu, W. L., Yung, C. F., & Wang, P. F. (2011). Numerical solutions of differential–algebraic equations and its applications in solving TPPC problems. *Journal of Marine Science and Technology*, *19*(1), 76–88.



Siamak Mehrkanoon received the B.S. degree in pure mathematics in 2005 and the M.S. degree in applied mathematics from the Iran University of Science and Technology, Tehran, Iran, in 2007. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium. His current research interests include machine learning, system identification, pattern recognition, and numerical algorithms.



Johan A.K. Suykens (SM'05) was born in Willebroek Belgium, on May 18 1966. He received the M.S. degree in Electro-Mechanical Engineering and the Ph.D. Degree in Applied Sciences from the Katholieke Universiteit Leuven, in 1989 and 1995, respectively. In 1996 he has been a Visiting Postdoctoral Researcher at the University of California, Berkeley. He has been a Postdoctoral Researcher with the Fund for Scientific Research FWO Flanders and is currently a Professor (Hoogleraar) with KU. Leuven. He is author of the books "Artificial Neural Networks for Modelling and Control of Non-linear Systems" (Kluwer Academic Publishers) and "Least Squares Support Vector Machines" (World Scientific), co-author of the book "Cellular Neural Networks, Multi-Scroll Chaos and Synchronization" (World Scientific) and editor of the books "Nonlinear Modeling: Advanced Black-Box Techniques" (Kluwer Academic Publishers) and "Advances in Learning Theory: Methods, Models and Applications" (IOS Press).

In 1998 he organized an International Workshop on Nonlinear Modeling with Time-series Prediction Competition. He is a Senior IEEE member and has served as associate editor for the IEEE Transactions on Circuits and Systems (1997–1999 and 2004–2007) and for the IEEE Transactions on Neural Networks (1998–2009). He received an IEEE Signal Processing Society 1999 Best Paper (Senior) Award and several Best Paper Awards at International Conferences. He is a recipient of the International Neural Networks Society INNS 2000 Young Investigator Award for significant contributions in the field of neural networks. He has served as a Director and Organizer of the NATO Advanced Study Institute on Learning Theory and Practice (Leuven 2002), as a program co-chair for the International Joint Conference on Neural Networks 2004 and the International Symposium on Nonlinear Theory and its Applications 2005, as an organizer of the International Symposium on Synchronization in Complex Networks 2007 and a co-organizer of the NIPS 2010 workshop on Tensors, Kernels and Machine Learning. He has been recently awarded an ERC Advanced Grant 2011.