LS-SVM based solution for delay differential equations

Siamak Mehrkanoon and Johan A. K. Suykens

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

E-mail: {Siamak.Mehrkanoon,Johan.Suykens}@esat.kuleuven.be

Abstract. In this paper a new approach based on Least Squares Support Vector Machines (LS-SVMs) is proposed for solving delay differential equations (DDEs) with single-delay. The proposed method provides a closed form approximate model based solution without using any interpolation techniques. The result of this paper can be extended for DDE with multi-lags. The results of some numerical experiments are presented and compared with analytic solutions to confirm the validity and applicability of the proposed method.

1. Introduction

Delay differential equations (DDEs) can be found in the mathematical formulation of real life phenomena in a wide variety of applications especially in science and engineering such as population dynamics, infectious disease and control problems [1]. In contrast with ordinary differential equations (ODEs) where the unknown function and its derivatives are evaluated at the same time instant, in a DDE the evolution of the system at a certain time, depends on the state of the system at an earlier time. A general form of first order multi-delay DDE is given by:

$$\begin{aligned} &\dot{x}(t) = f(t, x(t), x(t-\tau_1), \ldots, x(t-\tau_n)), \quad t \geq t_{in}, \\ &x(t) = g(t), \quad t \leq t_{in} \end{aligned} \tag{1}$$

where $g(t)$ is the initial function and $\{\tau_i\}_{i=1}^n$ are the delays or lags which are non-negative and can be constant, time dependent or state dependent. The term $x(t-\tau_i)$ is called the delay term. Due to the presence of the delay term, numerical methods that provide only discrete solutions at the grid points are not suitable for DDE and one needs to apply interpolation techniques, for approximating the delay term, in order to advance the solution of the given DDE.

Least Squares Support Vector Machines (LS-SVMs) is a methodology which has been applied in a wide variety of fields such as regression and classification [3]. Like in support vector machines [6], in this method one maps the data into a high dimensional feature space using a feature map. Solving ordinary differential equations (ODEs) and differential algebraic equations (DAEs) using LS-SVMs have been discussed in [4, 5]. In this paper the method developed in [4] is extended to approximate the solution of a scalar DDE. The method provides a closed form approximate model based solution without using any interpolation techniques.

Making use of Mercer's Theorem [6], 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 the subsequent section

$$\nabla_n^m \equiv \frac{\partial^{n+m}}{\partial p^n \partial q^m}. \tag{2}$$

When $\boldsymbol{\varphi}(p)^T\boldsymbol{\varphi}(q) = K(p,q)$, one can show then that

$$[\boldsymbol{\varphi}^{(n)}(p)]^T\boldsymbol{\varphi}^{(m)}(q) = \nabla_n^m[\boldsymbol{\varphi}(p)^T\boldsymbol{\varphi}(q)] = \nabla_n^m[K(p,q)] = \frac{\partial^{n+m}K(p,q)}{\partial p^n \partial q^m}. \tag{3}$$

Using formula (3), 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).

Suppose that $t_i \in \varrho$ and $v_j \in \tau$ are two arbitrary points and ϱ and $\tau \subseteq \mathbb{R}$. Now for notational convenience let us list the following notations which are used in the following sections.

$$\left[\nabla_n^m K\right](t,y) = \left[\nabla_n^m K(p,q)\right]\Big|_{p=t,q=y}, \quad \left[{}^{\varrho}\Omega^{\tau}\right]_{i,j} = \nabla_0^0[K(p,q)]\Big|_{p=t_i,q=v_j} = K(t_i,v_j),$$
$$\left[{}_n^{\varrho}\Omega_m^{\tau}\right]_{i,j} = \nabla_n^m\left[K(p,q)\right]\Big|_{p=t_i,q=v_j} = \frac{\partial^{n+m}K(p,q)}{\partial p^n\partial q^m}\Big|_{p=t_i,q=v_j},$$

where $\left[{}_n^{\varrho}\Omega_m^{\tau}\right]_{i,j}$ denotes the (i,j)-th entry of matrix ${}_n^{\varrho}\Omega_m^{\tau}$. In the case that $\varrho = \tau$, we denote the matrix by ${}_n^{\varrho}\Omega_m$.

2. Formulation of the method for DDE

Consider a scalar linear DDE with single-delay of the form

$$\begin{aligned} &\dot{x}(t) = a(t) + b(t)x(t) + c(t)x(t-\tau(t)), \quad t_{in} \le t \le t_f, \\ &x(t) = g(t), \quad t \le t_{in}, \end{aligned} \tag{4}$$

where $\dot{g}(t_{in}^-) = \dot{x}(t_{in}^+)$. Assume that a general approximate solution to equation (4) is of the form of $\hat{x}(t) = \boldsymbol{w}^T\boldsymbol{\varphi}(t) + d$, where $\boldsymbol{\varphi}(\cdot) : \mathbb{R} \to \mathbb{R}^h$ is the feature map and h is the dimension of the feature space. To obtain the optimal value of $\boldsymbol{w}$ and d, a collocation method is used [2] with a discretization of the interval $[t_{in}, t_f]$ into a set of points $\{t_i\}_{i=1}^N$. The estimated solution follows from

$$\begin{aligned} \min_{w,d,e_i} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^m e_i^2 \\ \text{s.t.} \quad & \boldsymbol{w}^T\left[\boldsymbol{\varphi}'(t_i) - b_i\boldsymbol{\varphi}(t_i) - c_i\boldsymbol{\varphi}(s_i)\right] - (b_i + c_i)d = \\ & a_i + e_i, \ i = 2,\ldots,N \\ & \boldsymbol{w}^T\boldsymbol{\varphi}(v_k) + d = g(v_k), \ k = 1,\ldots,M \end{aligned} \tag{5}$$

where $s_i = t_i - \tau(t_i)$ and $v_k = \{s_k \,|\, s_k \le t_{in}\}$. N is the number of collocation points (which is equal to the number of training points). Furthermore $a_i = a(t_i), b_i = b(t_i)$ and $c_i = c(t_i)$. Problem 5 is obtained by combining the LS-SVM cost function with constraints constructed by imposing the approximate solution $\hat{x}(t) = \boldsymbol{w}^T\boldsymbol{\varphi}(t) + d$, given by the LS-SVM model, to satisfy the given differential equation with corresponding initial function at collocation points $\{t_i\}_{i=1}^N$. Problem (5) is a quadratic minimization under linear equality constraints, which enables a straightforward solution.

Lemma 2.1. *Given a positive definite kernel function $K : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ with $K(t,s) = \boldsymbol{\varphi}(t)^T \boldsymbol{\varphi}(s)$ and a regularization constant $\gamma \in \mathbb{R}^+$, the solution to (5) is obtained by solving the following dual problem:*

$$\left[\begin{array}{c|c|c} \mathcal{K} + I_{N-1}/\gamma & \mathcal{U} & -\mathcal{H} \\ \hline \mathcal{U}^T & {}^{\mathcal{V}}_{0}\Omega_0 & 1_M \\ \hline -\mathcal{H}^T & 1_M & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline \boldsymbol{\beta} \\ \hline d \end{array}\right] = \left[\begin{array}{c} \boldsymbol{F_a} \\ \hline \boldsymbol{G} \\ \hline 0 \end{array}\right] \tag{6}$$

with

$\boldsymbol{\alpha} = [\alpha_2, \ldots, \alpha_N]^T, \boldsymbol{\beta} = [\beta_2, \ldots, \beta_N]^T, \boldsymbol{F_a} = [a(t_2), \ldots, a(t_N)]^T \in \mathbb{R}^{N-1}$,
$\boldsymbol{F_b} = [b(t_2), \ldots, b(t_N)]^T \in \mathbb{R}^{N-1}, \boldsymbol{F_c} = [c(t_2), \ldots, c(t_N)]^T \in \mathbb{R}^{N-1}, \boldsymbol{G} = [g(t_2), \ldots, g(t_N)]^T \in \mathbb{R}^{N-1}$,
$\mathcal{K} = \bar{D}\,\bar{\Omega}\,\bar{D}^T,\ \mathcal{U} = \bar{D}\,\Delta^T,\ \mathcal{H} = \boldsymbol{F_b} + \boldsymbol{F_c}$

$$\bar{\Omega} = \left[\begin{array}{c|c|c} {}^{\mathcal{T}}_{1}\Omega_{1} & {}^{\mathcal{T}}_{0}\Omega_{1} & {}^{\mathcal{S}}_{0}\Omega^{\mathcal{T}}_{1} \\ \hline {}^{\mathcal{T}}_{1}\Omega_{0} & {}^{\mathcal{T}}_{0}\Omega_{0} & {}^{\mathcal{S}}_{0}\Omega^{\mathcal{T}}_{0} \\ \hline {}^{\mathcal{T}}_{1}\Omega^{\mathcal{S}}_{0} & {}^{\mathcal{T}}_{0}\Omega^{\mathcal{S}}_{0} & {}^{\mathcal{S}}_{0}\Omega_{0} \end{array}\right] \in \mathbb{R}^{3(N-1)\times 3(N-1)},\ \bar{D} = [\mathbb{1}, -D_b, -D_c],\ \Delta = [{}^{\mathcal{V}}_{0}\Omega^{\mathcal{T}}_{1},\ {}^{\mathcal{V}}_{0}\Omega^{\mathcal{T}}_{0},\ {}^{\mathcal{V}}_{0}\Omega^{\mathcal{S}}_{0}]$$

D_b and D_c are diagonal matrices with the elements of $\boldsymbol{F_b}$ and $\boldsymbol{F_c}$ on the main diagonal respectively. $\mathbb{1}$ is the identity matrix. In addition $\mathcal{S} = \{s_i\}_{i=2}^N$, $\mathcal{V} = \{v_i\}_{i=1}^M$ and $\mathcal{T} = \{t_i\}_{i=2}^N$.

The solution in the dual form becomes:

$$\hat{x}(t) = \sum_{i=2}^{N} \alpha_i \Big([\nabla_1^0 K](t_i, t) - b(t_i)[\nabla_0^0 K](t_i, t) - c(t_i)[\nabla_0^0 K](s_i, t) \Big) + \sum_{k=1}^{M} \beta_k \, [\nabla_0^0 K](v_k, t) + d,$$

where $[\nabla_0^0 K](t,s) = K(t,s) = \boldsymbol{\varphi}(t)^T \boldsymbol{\varphi}(s)$ is the kernel function and $[\nabla_1^0 K](t,s) = \boldsymbol{\varphi}'(t)^T \boldsymbol{\varphi}(s)$ is its derivative. Here the Gaussian RBF kernel is used. α_i and β_k are Lagrange multipliers associated with (3). The result of this paper can be extended for linear DDE with multi-lags.

3. Numerical Result

Problem 3.1: Consider the following scalar DDE,

$$\begin{aligned} &\dot{x}(t) = -x(t - 1 + \exp(-t)) + \sin(t - 1 + \exp(-t)) + \cos(t),\ t \geq 0, \\ &x(t) = \sin(t),\ t \leq 0 \end{aligned}$$

The problem is solved on domain $t \in [0, 40]$ for $N = 200$. Figure 1 shows the residuals $e(t) = x(t) - \widehat{x}(t)$.

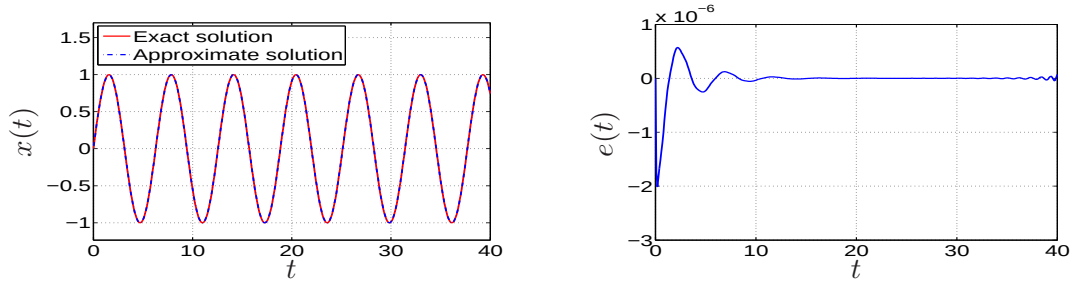
Problem 3.2: Consider the delay equation with asymptotically vanishing-lag:

$$\begin{aligned} &\dot{x}(t) = \frac{t^4 - 3}{(t^5 + t)\ln(t - t^{-3} + [t - t^{-3}]^{-3})}\, x(t - t^{-3}), \quad t \in [2, 30], \\ &x(t) = \ln(t + t^{-3}), \quad t \in [1.5, 2] \end{aligned}$$

whose exact solution is $x(t) = \ln(t + t^{-3})$. The result obtained by the proposed method method, with $N = 400$, is depicted in Figure 2.

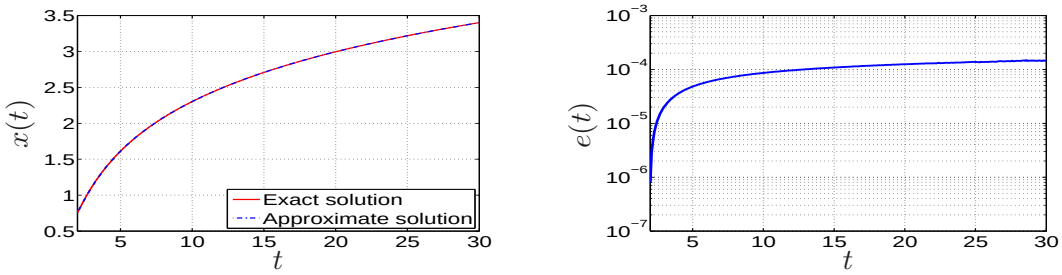
4. Conclusion

In this paper, we have presented a method based on LS-SVMs for the numerical solution of delay differential equations. The proposed method provides a close form approximate solution for the problem without using any interpolation techniques.



(a) (b)

Figure 1. (a) Numerical results for Problem 3.1, when 200 equidistant points in [0,40] are used for training. (b) Obtained model errors for problem 3.1.



(a) (b)

Figure 2. (a) Numerical results for Problem 3.2, when 400 equidistant points in [2,30] are used for training. (b) Obtained model errors for problem 3.2.

Acknowledgments

This work was supported by: • Research Council KUL: GOA/10/09 MaNet, 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 LeCoPro, 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.

References

[1] Driver R D 1977 *Ordinary and Delay Differential Equations* (New York: Springer-Verlag New York Inc).

[2] Kincaid D R and Cheney E W 2002 *Numerical Analysis*, (Mathematics of Scientific Computing, third ed., Brooks/Cole, Pacific Grove, CA).

[3] Suykens J A K, Van Gestel T, De Brabanter J, De Moor B and Vandewalle J 2002 *Least Squares Support Vector Machines* (World Scientific, Singapore.)

[4] Mehrkanoon S, Falck T and 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** 1356-67.

[5] Mehrkanoon S and Suykens J A K 2012 LS-SVM approximate solution to linear time varying descriptor systems *Automatica*, **48**, 2502-2511.

[6] Vapnik V 1998 *Statistical learning theory* (New York: Wiley)