Kybernetika

Mitra Bolhassani; Hassan Dana Mazraeh; Kourosh Parand

A new method based on least-squares support vector regression for solving optimal control problems

Kybernetika, Vol. 60 (2024), No. 4, 513–534

Persistent URL: http://dml.cz/dmlcz/152617

KYBERNETIKA — VOLUME *60* (2024), NUMBER 4, PAGES 513–534

A NEW METHOD BASED ON LEAST-SQUARES SUPPORT VECTOR REGRESSION FOR SOLVING OPTIMAL CONTROL PROBLEMS

Mitra Bolhassani, Hassan Dana Mazraeh, Kourosh Parand

In this paper, a new application of the Least Squares Support Vector Regression (LS-SVR) with Legendre basis functions as mapping functions to a higher dimensional future space is considered for solving optimal control problems. At the final stage of LS-SVR, an optimization problem is formulated and solved using Maple optimization packages. The accuracy of the method are illustrated through numerical examples, including nonlinear optimal control problems. The results demonstrate that the proposed method is capable of solving optimal control problems with high accuracy.

Keywords: Least squares support vector machines, Optimal control problems, Legendre orthogonal polynomials, Regression, Artificial intelligence

Classification: 68T20, 49Mxx

1. INTRODUCTION

The field of optimal control focuses on efficiently managing dynamic systems over a specified duration, aiming to minimize a designated performance measure while satisfying additional constraints. These problems involve two key variables: the state variable, which describes the system's behavior at each stage, and the control variable, which indicates the system's evolution from one stage to the next. Optimal control problems manifest as mathematical models in various domains, including industrial applications [1, 10], health sciences [14], and aerospace sciences [7]. Due to the often unattainable nature of analytical solutions for these problems, seeking approximate solutions is a pragmatic approach.

The field of numerical methods has garnered significant interest among researchers in mathematical sciences, leading to the development of various computational techniques and efficient algorithms for solving optimal control problems. Providing effective numerical methods for these problems is of paramount importance. Numerical methods used to solve optimal control problems can be classified into two categories: direct and indirect methods [2].

Indirect methods involve deriving solutions using necessary optimality conditions from the calculus of variations and Pontryagin's maximum principle [3, 18]. These con-

DOI: 10.14736/kyb-2024-4-0513

ditions form a Hamiltonian boundary value problem (HBVP), which can be solved using numerical methods. A primary advantage of these methods is the high confidence in the numerical solutions obtained, as they satisfy the first-order necessary conditions. However, there are several disadvantages: obtaining the first-order conditions analytically is often challenging, and they require a highly accurate initial guess for the unknown boundary conditions due to their small convergence radius.

In contrast, direct methods discretize the optimal control problem, converting it into a nonlinear programming problem. Their advantages include the absence of the need to find first-order necessary conditions and a much larger convergence radius. These methods rely on robust theories such as approximation, control, and optimization theories. They often use orthogonal functions to approximate solutions and can be categorized into methods that parameterize control variables and those that parameterize both state and control variables. Only control variables are discretized in the former, and the resulting equations are solved using numerical integration. In the latter, both state and control variables are discretized simultaneously, resulting in a set of algebraic constraints [3, 22, 26].

In recent decades, spectral methods—techniques that parameterize both state and control variables—have attracted significant attention. For example, Parand et al. used the collocation method with various approaches to solving nonlinear optimal control problems involving the classical diffusion equation, inequality-constrained optimal control problems of arbitrary order, and nonlinear 2D optimal control problems [9, 15, 19]. Similarly, Razzaghi et al. employed the collocation method to address linear quadratic optimal control problems and control of linear systems [5, 20]. They have also used various methods to solve diverse optimal control problems, including fractional optimal control [29, 30, 31, 32, 33, 34, 35].

Recently, support vector machine (SVM) techniques have achieved notable success in tackling various problems. Initially introduced by Cortes and Vapnik for binary classification [4], SVMs transform classification problems into quadratic programming problems with inequality constraints, ensuring a unique solution under specific conditions. The LS-SVM method, proposed by Suykens et al. [24], attains a globally optimal value by solving a system of linear equations derived from equality constraints, offering significant computational efficiency compared to traditional SVMs.

Two years later, Suykens et al. [38] introduced the use of least squares support vector machines (LS-SVM) for the optimal control of nonlinear systems. The present study introduces a novel method using LS-SVR with Legendre basis functions for solving optimal control problems.

While the foundational work by Suykens et al. (2001) demonstrated the effectiveness of LS-SVM for N-stage optimal control, our approach significantly extends this framework in several key aspects. Firstly, this approach employs a training technique called collocation least squares support vector regression (CLS-SVR). We use Legendre polynomials as the network's mapping functions for our computations, resulting in an optimization problem that can be reduced to solving a set of algebraic equations. By employing Legendre polynomials as basis functions, our method leverages their orthogonality and boundedness properties, which enhance computational accuracy and manage error propagation effectively. This choice contrasts with the standard polynomial basis used in the LS-SVM framework by Suykens et al., offering a substantial improvement in precision and computational cost.

Secondly, we employed Maple's Optimization Package to solve the optimization problem related to the LS-SVR model. This integration signifies a significant advancement over the nonlinear programming techniques used in the prior methods.

Thirdly, the proposed LS-SVR method is being used for the first time to solve nonlinear optimal control problems and nonlinear two-dimensional fractional optimal control problems. The presented examples demonstrate the high accuracy of this method.

Finally, we selected two distinct values for the γ_1 and γ_2 parameters in the LS-SVR model, with one for the coefficient of $\sum i = 1^N e_i^2$ and another for the coefficient of J_N. Therefore, the optimization cost function is as follows:

$$\min_{\alpha,\beta,e} \quad \frac{1}{2}\left(\sum_{i=0}^{M_1} {\alpha_i}^2 + \sum_{i=0}^{M_2} {\beta_i}^2\right) + \frac{\gamma_1}{2}\sum_{i=1}^{k} e_i^2 + \gamma_2 J_N.$$

It is important to note that, according to our experiments, the best results are obtained in some problems when $\gamma_1 = \gamma_2$. According to our findings, this assumption improves the accuracy of solutions. In contrast, previous works by other researchers only used a single parameter γ_1.

This method has been used in many applications, including the approximation of ordinary differential equations, partial differential equations, and integral equations. For example, refer to the articles by Parand et al. in this field [12, 16, 17].

The structure of the remaining sections in this paper is as follows. In Section 2, we introduce the necessary preliminaries and notations related to polynomial spaces, along with an overview of least squares support vector regression. Section 3 presents the formulation of least squares support vector regression method specifically designed for solving optimal control problems, detailing the approach and methodology employed. In the final section, we conduct several numerical experiments to assess the accuracy of the proposed method.

2. PRELIMINARIES

Considering that concepts such as Legendre polynomials and least squares support vector regression are needed in the following sections, we will examine these concepts in this section.

2.1. Legendre polynomials

The Legendre polynomials, represented as $P_n(x)$, are a set of orthogonal polynomials, with n denoting the polynomial order and x representing the variable. The order n corresponds to the degree of the polynomial [23]. They exhibit orthogonality over the range [-1, 1], where the weight function is defined as $\omega(x) = 1$. The following recursive relation generates the basis functions:

$$P_{n+1}(x) = \frac{(2n+1)}{(n+1)} x P_n(x) - \frac{n}{(n+1)} P_{n-1}(x), n \geqslant 1, \tag{1}$$

$$P_0(x) = 1, P_1(x) = x. \tag{2}$$

These polynomials are derived as the solution of the Sturm-Liouville differential equation:

$$\frac{\mathrm{d}}{\mathrm{d}x}\left((1-x^2)\frac{\mathrm{d}P_n(x)}{\mathrm{d}x}\right) + n(n+1)P_n(x) = 0. \tag{3}$$

These properties exhibit both symmetry and boundedness:

$$P_n(-x) = (-1)^n P_n(x), \quad n \geqslant 0, \tag{4}$$

$$|P_n(x)| \leqslant 1, \quad \forall x \in [-1,1]. \tag{5}$$

The utilization of equality (4) helps in minimizing computational expenses, while inequality (5) serves to manage error propagation. The roots of Legendre polynomials are used in the Gauss-quadrature rules. Imagine we want to approximate the integral of the function $f(x)$ as follow :

$$\int_{-1}^{1} f(x)\,\mathrm{d}x \approx \sum_{i=1}^{N} \omega_i f(x_i). \tag{6}$$

With x_i representing the zeros of the Legendre polynomials and ω_i as the coefficients, we have:

$$\omega_i = \frac{2}{(1-x_i^2)[{P'}_{N+1}(x_i)]^2}, \quad 0 \leqslant i \leqslant N. \tag{7}$$

2.2. LS-SVR formulations

Considering a training dataset comprised of pairs $\{x_i, y_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}$. Suykens presented the LS-SVR model in the following manner [25]:

$$y(x) = w^T \varphi(x) + b, \tag{8}$$

which is the weights $w = [w_1, \ldots, w_m]^T$, the basis functions $\varphi(x) = [\varphi_1(x), \ldots, \varphi_m(x)]^T$ and the bias b. The appropriate values for w and b in this equation are found by solving the following optimization problem:

$$\begin{aligned} \min_{w,b,e} \mathcal{J}(w,e) &= \frac{1}{2}w^T w + \gamma\frac{1}{2}\sum_{k=1}^{N} e_k^2 \\ \text{subject to} \qquad y_k &= w^T\varphi(x_k) + b + e_k, \quad k = 1,\ldots,n. \end{aligned} \tag{9}$$

In this equation, γ represents the Tikhonov regularization parameter, $e_i \in \mathbb{R}$ denotes the error at the ith training point, $\omega \in \mathbb{R}^m$ denotes the weight vector, and $b \in \mathbb{R}$ is the bias term [24]. Using Lagrange multiplier method, we will have the following form:

$$\mathcal{L}(w,b,e;\alpha)=\mathcal{J}(w,e)-\sum_{k=1}^{n}\alpha_k\left\{w^T\varphi\left(x_k\right)+b+e_k-y_k\right\}, \tag{10}$$

where α_k represents the Lagrange multipliers. The solution involves maximizing α while simultaneously minimizing w, b, and e, leading to the identification of the saddle point in the Lagrangian function. Subsequently, by applying the Karush-Kuhn-Tucker (KKT) conditions, the unknown variables can be determined by solving the following linear system [21]:

$$\left[\begin{array}{c|c} 0 & {1_n}^T \\ \hline 1_n & \Omega+\gamma^{-1}I \end{array}\right]\left[\begin{array}{c} b \\ \hline \alpha \end{array}\right]=\left[\begin{array}{c} 0 \\ \hline y \end{array}\right], \tag{11}$$

where I denotes the identity matrix, $\Omega_{ij}=\varphi(x_i)^T\varphi(x_j)$ for $i,j=1,\ldots,n$, $y=[y_1,\ldots,y_n]^T$, $1_n=[1,\ldots,1]^T$ and $\alpha=[\alpha_1,\ldots,\alpha_n]^T$.

3. LS-SVR FOR OPTIMAL CONTROL PROBLEMS

Let's take into account the following system of nonlinear differential equations that characterizes a process taking place over a specified time interval $[t_a,t_b]$:

$$\frac{\mathrm{d}X(\tau)}{\mathrm{d}t}=f(\tau,X(\tau),U(\tau)), \tag{12}$$

The initial conditions for the system are given as follows:

$$X\left(t_a\right)=x_0,\quad X\left(t_b\right)=x_1. \tag{13}$$

The system is characterized by the state variable $X(\cdot):[t_a,t_b]\to\mathbb{R}$, the control variable $U(\cdot):[t_a,t_b]\to\mathbb{R}$, and a real-valued continuously differentiable function $f(.)$. The objective of the optimal control problem is to determine the control variable $U(\cdot)$ that guides the system described by equation (12) from an initial position $X\left(t_a\right)=x_0$ to a desired position $X\left(t_b\right)=x_1$ within the time interval (t_b-t_a), while optimizing a performance index J, given by:

$$\min J[X,U]=\int_{t_a}^{t_b}L(\tau,X(\tau),U(\tau))\,\mathrm{d}\tau. \tag{14}$$

Assuming the existence of permissible controls passing through (t_a,x_0) and (t_b,x_1), our objective within this control set is to find the control variable that minimizes J and designate it as the optimal control. If $t_a\neq-1$ or $t_b\neq1$, we introduce the transformation using Legendre polynomials:

$$\tau=\frac{t_b-t_a}{2}t+\frac{t_b+t_a}{2}. \tag{15}$$

The variable transformation results in t being in the interval $[-1,1]$, which corresponds to τ being in the range $[t_a,t_b]$.

By utilizing Equation (15), we can express the optimal control problem stated in Equations (12) – (14) as follows:

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} = f\left(\frac{t_b - t_a}{2}t + \frac{t_b + t_a}{2}, x(t), u(t)\right) \tag{16}$$

and the trajectory $x(t)$ that corresponds to the specified initial conditions:

$$x(-1) = x_0, \quad x(1) = x_1, \tag{17}$$

minimizes:

$$\min J[x,u] = \frac{t_b - t_a}{2} \int_{-1}^{1} L\left(\frac{t_b - t_a}{2}t + \frac{t_b + t_a}{2}, x(t), u(t)\right) \mathrm{d}t. \tag{18}$$

We express the approximate solution $x(t)$ and $u(t)$ of Equation (14) by expanding it in terms of Legendre polynomials, using unknown coefficients. The coefficients are then determined through LS-SVR procedure. We now seek Nth-degree Legendre interpolating polynomials to $x(t)$ and $u(t)$

$$x_N(t) := \sum_{i=0}^{M_1} \alpha_i P_i(t), \quad u_N(t) := \sum_{i=0}^{M_2} \beta_i P_i(t). \tag{19}$$

Here, P_i represents the Legendre polynomial of degree i as defined in equation (2), and M_1, M_2 denotes the number of basis functions. It is important to mention that we exclude the bias term b because the first Legendre polynomial, as defined in formula (2), serves the same purpose. We next approximate problem (18) by the following one:

$$J_N = \frac{t_b - t_a}{2} \int_{-1}^{1} L\left(\frac{t_b - t_a}{2}t + \frac{t_b + t_a}{2},\ x_N(t),\ u_N(t)\right) \mathrm{d}t, \tag{20}$$

Valuation of dynamical system in zeros of Legendre polynomial:

$$\begin{aligned} &\left\{\frac{\mathrm{d}x(ih)}{\mathrm{d}t} = e_l | h = 0.1, i = 0, \ldots, N, l = 1, \ldots, k\right\} \\ &\cup \{x_N(t_a) - x_0 = e_z | z = k+1, \ldots, 2k\} \\ &\cup \{x_N(t_b) - x_1 = e_s | s = 2k, \ldots, 3k\}. \end{aligned} \tag{21}$$

Here k is the cardinality of the training. To obtain the unknown coefficients of equation (19) within the network and apply the LS-SVR model to solve nonlinear optimal control problems, it is necessary to modify the general form of LS-SVR to incorporate the specific form of the problem and its constraints. This is done in an equivalent formulation of the standard LS-SVR problem (9). we consider the following form of optimization problem associated with (12) – (14):

$$\begin{aligned} \min_{\alpha,\beta,e} \quad & \frac{1}{2}\left(\sum_{i=0}^{M_1} {\alpha_i}^2 + \sum_{i=0}^{M_2} {\beta_i}^2\right) + \frac{\gamma_1}{2}\sum_{i=1}^{k} e_i^2 + \gamma_2 J_N \\ \text{s.t.} \quad & \frac{\mathrm{d}x(ih)}{\mathrm{d}t} = e_l, \\ & x_N\left(t_a\right) - x_0 = e_z, \\ & x_N\left(t_b\right) - x_1 = e_s, \end{aligned} \tag{22}$$

where the regularization parameters γ_1 and γ_2 are the regularization parameters for the error.

4. NUMERICAL ANALYSIS

In this section, we present the numerical outcomes by applying our method to the given examples using Maple software on a Windows operating system. We utilized a Core i5 processor, setting the precision to 60 digits. The mean absolute error (MAE) is reported for comparing the difference between the exact solutions and LS-SVR approximate solutions in the following tables, corresponding to each example. To report the MAE for the first three examples, we provide two tables. The first table for the first three examples presents the MAE for different values of M_1,M_2, with a fixed value of $\gamma_1, \gamma_2 = 10^{14}$. The second table for the first three examples reports the MAE different values of γ_1,γ_2 while M_1,M_2 are set to 20. In each figure, we compare the exact solution and LS-SVR approximate solution for M_1,$M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$. In each of the last three examples, we included a table that displays the MAE for different values of γ_1,γ_2. Additionally, the values of M_1 and M_2 are considered constant for these examples.

Before presenting the experimental results for the problem under consideration, we provide an algorithmic view of the entire procedure and explain how the constraints are constructed in this study. Additionally, the complete code in Maple for the final example, which is a two-dimensional fractional order problem, is included in Appendix A.

Example 1: We examine the first optimal control problem, which includes state $X(t)$ and control $U(t)$ variables as follows:

$$\min J[X,U] = \int_0^1 \left(U^2(t) + X^2(t)\right)\mathrm{d}t, \quad t \in [0,1],$$

the dynamic system of this problem is as follows:

$$U(t) = \frac{\mathrm{d}X(t)}{\mathrm{d}t},$$

under the following boundary conditions:

$$X(0) = 0, \quad X(1) = \frac{1}{2}.$$

Algorithm 1 Pseudocode of the whole procedures and constraints construction

Require: Some packages such as the Optimization package.

Set the parameters such as γ_1, γ_2, and an interval $[a, b]$.
Set the degree N used to form the unknown function based on the Legendre polynomials.
Shifted_interval $\leftarrow \frac{2x-a-b}{b-a}$.
Calculate the *training points*. The *training points* are a set of the roots of the first $N+1$ Legendre polynomials in the Shifted_interval.
If it exists, define the exact solution for comparison purposes.
Define the unknown function based on the Legendre polynomials of degree N.
Define the function J according to the optimal control problem at hand.
Define condition functions based on the conditions of the optimal control under consideration.
Create an empty set of constraints: $constraints \leftarrow \{\}$.
Set the counter variable k: $k \leftarrow 1$. This counter holds the number of constraints created.
for each $condition_j(\cdot)$ **do**
    for each i in *training points* **do**
        $constraints \leftarrow constraints \cup \{condition_j(i) = e[k]\}$.
        $k \leftarrow k + 1$.
    end for
end for
Set the cost function of the LS-SVR model:

$$cost \leftarrow \frac{1}{2} w^T w + \frac{\gamma_1}{2} \sum_{i=0}^{k} e[i]^2 + \gamma_2 J.$$

Solve the optimization problem ($cost$, $constraints$) obtained from the primal form of the LS-SVR model.

The analytical solution is given by [6]:

$$X(t) = \frac{e(e^t - e^{-t})}{2(e^2 - 1)},$$
$$U(t) = \frac{e(e^t + e^{-t})}{2(e^2 - 1)}.$$

The MAE is reported in Table 1 for different values of M_1,M_2 with a fixed value of $\gamma_1, \gamma_2 = 10^{14}$. By increasing the values of M_1,M_2 the error decreased so that the lowest error was obtained in M_1,$M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$. Furthermore, the MAE is reported in Table 2 for different values of γ_1, γ_2 with a fixed value of M_1,$M_2 = 20$. By increasing the value of γ_1, γ_2 the error decreased so that the lowest error was obtained in M_1,$M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$.

Figure 1 illustrate the exact and approximated state variable, $X(t)$ and also the exact and approximated control variable, $U(t)$.

M_1=M_2	MAE for $X(t)$	MAE for $U(t)$
5	5.27×10^{-8}	1.58×10^{-6}
10	1.38×10^{-15}	1.03×10^{-13}
15	5.17×10^{-17}	2.01×10^{-16}
20	2.25×10^{-19}	3.78×10^{-18}

Tab. 1: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of M_1,M_2 and a fixed $\gamma_1, \gamma_2 = 10^{14}$.

γ_1=γ_2	MAE for $X(t)$	MAE for $U(t)$
10^5	5.31×10^{-8}	2.37×10^{-7}
10^8	2.51×10^{-13}	5.71×10^{-12}
10^{11}	5.11×10^{-18}	2.13×10^{-17}
10^{14}	2.25×10^{-19}	3.78×10^{-18}

Tab. 2: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of γ_1, γ_2 and a fixed $M_1, M_2 = 20$.

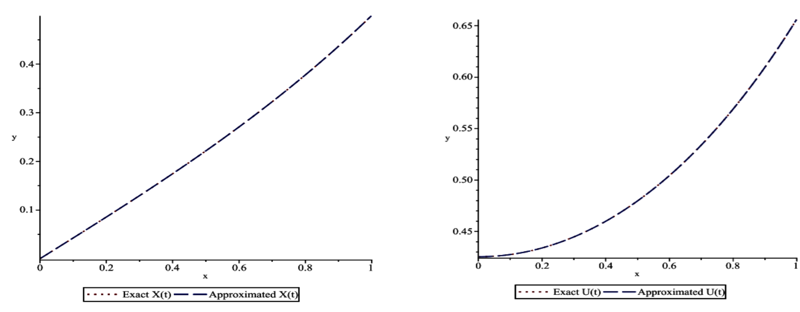

Fig. 1: The left figure presents the exact and approximated state variables $X(t)$, while the right figure presents the exact and approximated state variables $U(t)$.

Example 2: We will investigate another problem which is as follows:

$$\min J[X,U] = \frac{1}{2}\int_0^1 (2X^2(t) + U^2(t))\,\mathrm{d}t, \quad t \in [0,1],$$

under the dynamic system:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = U(t) - \frac{1}{2}X(t),$$

and boundary conditions:

$$X(0) = 1.$$

The analytical solution is as follows [11]:

$$\begin{aligned} X(t) &= \frac{1}{2+e^{-3}}[2e^{-\frac{3}{2}t} + e^{-3+\frac{3}{2}t}], \quad 0 \leqslant t \leqslant 1, \\ U(t) &= \frac{-2}{2+e^{-3}}[e^{-\frac{3}{2}t} - e^{-3+\frac{3}{2}t}], \quad 0 \leqslant t \leqslant 1. \end{aligned}$$

The MAE is reported in Table 3 for different values of M_1,M_2 with a fixed value of $\gamma_1, \gamma_2 = 10^{14}$. By increasing the values ofM_1,M_2 the error decreased so that the lowest error was obtained in $M_1, M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$. Furthermore, the MAE is reported in Table 4 for different values of γ_1, γ_2, with a fixed value of $M_1, M_2 = 20$. By increasing the value of γ_1, γ_2, the error decreased so that the lowest error was obtained in $M_1, M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$.

Figure 2 show the exact and approximated state variable, $X(t)$ and also the exact and approximated control variable, $U(t)$.

M_1=M_2	MAE for $X(t)$	MAE for $U(t)$
5	1.16×10^{-6}	2.04×10^{-5}
10	1.17×10^{-13}	1.01×10^{-11}
15	5.23×10^{-17}	2.16×10^{-16}
20	5.21×10^{-17}	2.11×10^{-16}

Tab. 3: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of M_1,M_2 and a fixed $\gamma_1, \gamma_2 = 10^{14}$.

γ_1=γ_2	MAE for $X(t)$	MAE for $U(t)$
10^5	5.07×10^{-7}	2.07×10^{-6}
10^8	5.03×10^{-10}	2.01×10^{-9}
10^{11}	5.12×10^{-13}	2.11×10^{-12}
10^{14}	5.21×10^{-17}	2.11×10^{-16}

Tab. 4: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of γ_1, γ_2 and a fixed $M_1, M_2 = 20$.

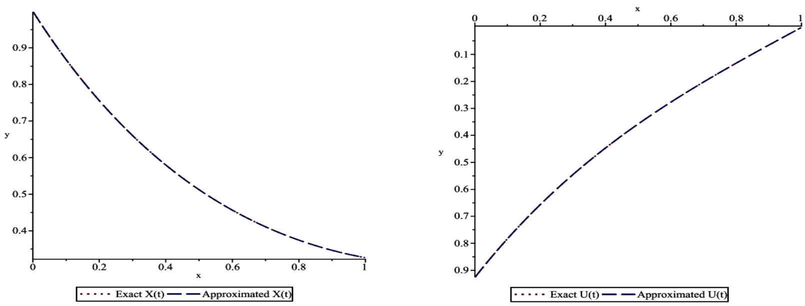

Fig. 2: The left figure presents the exact and approximated state variables $X(t)$, while the right figure presents the exact and approximated state variables $U(t)$.

Example 3: In the following example [13], the objective is to minimize:

$$\min J[X, U] = \int_0^1 (X(t) - \frac{1}{2}U^2(t))\, \mathrm{d}t, \quad t \in [0, 1],$$

with the dynamic system:

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = U(t) - X(t),$$

with boundary conditions:

$$X(0) = 0, \qquad X(1) = \frac{1}{2}\left(1 - \frac{1}{e}\right)^2,$$

and the analytical solution is:

$$X(t) = 1 - \frac{1}{2}e^{t-1} + \left(\frac{1}{2e} - 1\right)e^{-t},$$
$$U(t) = 1 - e^{t-1}.$$

The MAE is reported in Table 5 for different values of M_1,M_2 with a fixed value of $\gamma_1, \gamma_2 = 10^{14}$. By increasing the value of M_1,M_2 the error decreased so that the lowest error was obtained in $M_1, M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$. Furthermore, the MAE is reported in Table 6 for different values of γ_1, γ_2 with a fixed value of $M_1, M_2 = 20$. By increasing the value of γ_1, γ_2, the error decreased so that the lowest error was obtained in $M_1, M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$.

Figure 3 illustrate the exact and approximated state variable, $X(t)$ and also the exact and approximated control variable, $U(t)$.

M_1=M_2	MAE for $X(t)$	MAE for $U(t)$
5	7.41×10^{-7}	1.41×10^{-6}
10	2.61×10^{-15}	3.03×10^{-13}
15	1.71×10^{-16}	3.01×10^{-15}
20	2.55×10^{-18}	1.43×10^{-17}

Tab. 5: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of M_1,M_2 and a fixed $\gamma_1, \gamma_2 = 10^{14}$.

γ_1=γ_2	MAE for $X(t)$	MAE for $U(t)$
10^5	3.23×10^{-7}	3.81×10^{-8}
10^8	1.81×10^{-14}	4.01×10^{-11}
10^{11}	1.01×10^{-17}	4.32×10^{-16}
10^{14}	2.55×10^{-18}	1.43×10^{-17}

Tab. 6: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of γ_1, γ_2 and a fixed $M_1, M_2 = 20$.

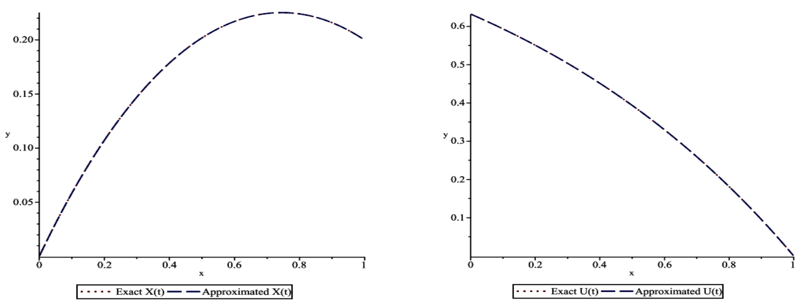

Fig. 3: The left figure presents the exact and approximated state variables $X(t)$, while the right figure presents the exact and approximated state variables $U(t)$.

In the first three examples increasing the values of M_1,M_2 and γ_1, γ_2 caused a decrease in mean absolute error. In act, the best error was obtained in $M_1, M_2 = 20$ and $\gamma_1\gamma_2 = 10^{14}$. It is worth mentioning that further increase in M_1,M_2 and γ_1, γ_2 more than $M_1, M_2 = 20$ and $\gamma_1, \gamma_2 = 10^{14}$ do not improve the accuracy considerably.

Example 4: Take into account the subsequent nonlinear optimal control problem

$$\min J[X,U] = \int_0^1 (U^2(t) + X^6(t))\,\mathrm{d}t, \quad t \in [0,1],$$

subject to

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = X(t) + 2U^2(t), \quad X(0) = 1.$$

The analytical solution is given by [36]:

$$X(t) = e^t.$$

The MAE is reported in Table 7 for different values of γ_1 and γ_2, with a fixed values of $M_1 = 2, M_2 = 4$. The rationale behind reporting γ_2 for 10^0 and 10^1 is that the other values for this parameter did not significantly improve the accuracy. Furthermore, γ_1 has been reported for values from 10^4 to 10^8. As evident from Table 7, the best results are obtained when γ_1 is set to 10^8 and γ_2 is 10^1. Figure 4 show the exact and approximated state variable, $X(t)$, and also presents the absolute error between the exact and the approximated $X(t)$.

γ_1	γ_2	MAE for $X(t)$
10^4	10^0	3.7×10^{-2}
10^5	10^1	3.1×10^{-2}
10^5	10^0	3.6×10^{-3}
10^6	10^1	3.1×10^{-3}
10^6	10^0	3.4×10^{-4}
10^7	10^1	3.1×10^{-4}
10^7	10^0	3.3×10^{-5}
10^8	10^1	3.1×10^{-5}

Tab. 7: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of γ_1, γ_2 and a fixed $M_1 = 2, M_2 = 4$.

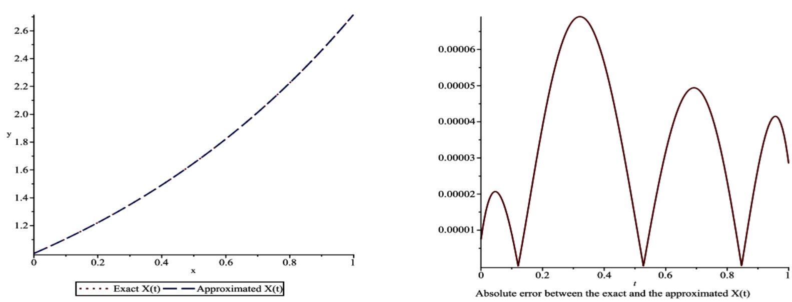

Fig. 4: The left figure presents the exact and approximated state variables $X(t)$, while the right figure presents the absolute error between the exact and the approximated $X(t)$.

Example 5: Consider the following nonlinear optimal control problem

$$\min J[X,U] = \int_0^1 (U^2(t) + X^2(t))\,\mathrm{d}t, \quad t \in [0,1],$$

subject to

$$\frac{\mathrm{d}X(t)}{\mathrm{d}t} = -X^2(t) - tU^2(t), \quad X(0) = 2.$$

The analytical solution is given by [36]:

$$X(t) = \frac{1}{(t+1/2)}.$$

The MAE in the example is reported in Table 8 for different values of γ_1 and γ_2, with a fixed values of $M_1 = 12$,$M_2 = 12$. The rationale behind reporting γ_2 for 10^0 and 10^1 is that the other values for this parameter did not significantly improve the accuracy. Furthermore, γ_1 has been reported for values from 10^1 to 10^6. As evident from Table 8, the best results are obtained when γ_1 is set to 10^6 and γ_2 is 10^0. Figure 5 show the exact and approximated state variable, $X(t)$, and also presents the absolute error between the exact and the approximated $X(t)$.

γ_1	γ_2	MAE for $X(t)$
10^1	10^0	3.7×10^{-2}
10^2	10^1	2.1×10^{-2}
10^2	10^0	3.8×10^{-3}
10^3	10^1	2.2×10^{-3}
10^3	10^0	4.7×10^{-4}
10^4	10^1	2.2×10^{-4}
10^4	10^0	2.8×10^{-5}
10^5	10^1	2.1×10^{-5}
10^6	10^1	1.2×10^{-5}
10^5	10^0	2.1×10^{-6}
10^6	10^0	1.3×10^{-6}

Tab. 8: Mean absolute error between the exact and approximated $X(t)$ and $U(t)$ for different values of γ_1, γ_2 and a fixed $M_1 = 12$,$M_2 = 12$.

Example 6: Consider the following nonlinear 2d fractional optimal control problem

$$\min J[U,Y] = \int_0^1 \int_0^1 \left[\big(U(x,t) - t^4 \sin(x)\big)^2 + \big(Y(x,t) - t^3 \cos(x)\big)^2 \right] \mathrm{d}x\mathrm{d}t,$$

subject to the nonlinear fractional dynamical system

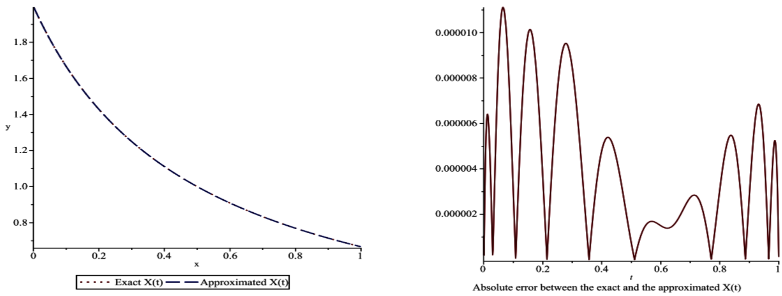

Fig. 5: The left figure presents the exact and approximated state variables $X(t)$, while the right figure presents the absolute error between the exact and the approximated $X(t)$.

$$
\begin{aligned}
D^{0.5}U(x,t) &= \cos(U(x,t)) + 2\sin(x)U_x(x,t) + U_{xx}(x,t) + 6\sin(x)Y(x,t) \\
&\quad - \left(\cos(t^4\sin(x)) + t^3\left(t\sin(2x) - t\sin(x) + 3\sin(2x)\right)\right) + \frac{\Gamma(5)\sin(x)\,t^{3.5}}{\Gamma(4.5)},
\end{aligned}
$$

and the Goursat-Darboux conditions

$$U(x,0) = 0, \quad U(0,t) = 0.$$

The analytical optimal solution of this example is $\langle U(x,t), Y(x,t)\rangle = \langle t^4\sin(x), t^3\cos(x)\rangle$ [37]. In this example, $U(x,t)$ is the state variable, and $Y(x,t)$ is the control variable. The MAE is reported in Table 9 for different values of γ_1 and γ_2, with a fixed values of $M_1, M_2, M_2, M_2 = 4$. The rationale behind reporting γ_1 for 10^0 and 10^1 is that the other values for this parameter did not significantly improve the accuracy. Furthermore, γ_2 has been reported for values from 10^2 to 10^7. As evident from Table 9, the best results are obtained when γ_1 is set to 10^0 and γ_2 is 10^7. Figure 6 shows the approximated $U(x,t)$, exact $U(x,t)$, both approximated and exact $U(x,t)$, and the absolute error between them are presented. Similarly, figures related to the approximated $Y(x,t)$, exact $Y(x,t)$, both approximated and exact $Y(x,t)$, and the absolute error between them are also presented.

5. CONCLUSIONS

This paper introduces a numerical algorithm utilizing least squares support vector regression for solving optimal control problems using machine learning. The method proposed in this study, called CLS-SVR, utilizes the Legendre orthogonal polynomials as the mapping functions to a higher dimensional future space for optimal control problems. This method was implemented for several nonlinear optimal control examples and a nonlinear 2d fractional optimal control problem. Some of the primary advantages of the proposed method include sparsity, well-conditioned generated matrices, rapid convergence, and low computational cost. The numerical findings showed that the suggested method is highly effective in resolving these problems. As future studies, researchers can employ this method to solve various optimal control problems, including fractional optimal con-

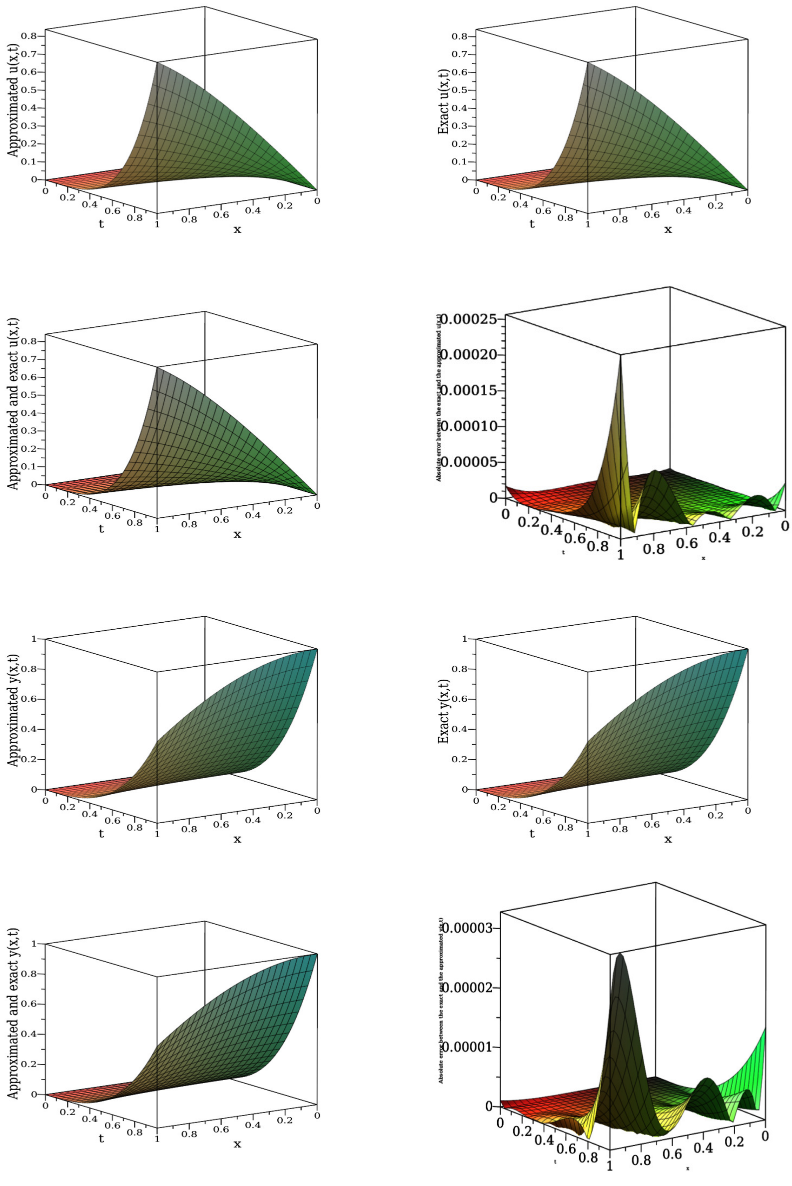

Fig. 6: Figures related to the approximated $U(x,t)$, exact $U(x,t)$, both approximated and exact $U(x,t)$, and the absolute error between them are presented. Similarly, figures related to the approximated $Y(x,t)$, exact $Y(x,t)$, both approximated and exact $Y(x,t)$, and the absolute error between them are also presented.

γ_1	γ_2	MAE for $U(x,t)$	MAE for $Y(x,t)$
10^0	10^3	7.75×10^{-2}	1.2×10^{-2}
10^0	10^2	1.33×10^{-2}	1.14×10^{-2}
10^1	10^2	1.4×10^{-2}	6.4×10^{-3}
10^1	10^3	1.2×10^{-2}	3.77×10^{-3}
10^1	10^4	7.3×10^{-3}	3.2×10^{-3}
10^0	10^4	2.62×10^{-3}	8.86×10^{-4}
10^1	10^5	2.1×10^{-3}	8.2×10^{-4}
10^0	10^5	6.4×10^{-4}	1.54×10^{-4}
10^1	10^6	6.2×10^{-4}	1.1×10^{-4}
10^0	10^6	8.54×10^{-5}	1.82×10^{-5}
10^1	10^7	8.2×10^{-5}	1.2×10^{-5}
10^0	10^7	8.88×10^{-6}	1.86×10^{-6}

Tab. 9: Mean absolute error between the exact and approximated $U(x,t)$ and $Y(x,t)$ for different values of γ_1, γ_2 and a fixed $M_1, M_2, M_2, M_2 = 4$.

trol. Furthermore, the hyper-parameters tune, which was done manually in this paper, could be done by some search algorithms such as meta-heuristic algorithms.

(Received February 5, 2024)

REFERENCES

[1] H. M. Amman and D. A. Kendrick: Computing the *58* (1998), 2, 185–191. DOI:10.1016/S0165-1765(97)00263-2

[2] J. T. Betts: Survey of numerical methods for trajectory optimization. J. Guidance Control Dynamics (1998), 98124–2207.

[3] H. G. Bock and K. J. Plitt: A multiple shooting algorithm for direct solution of optimal control problems. In: IFAC 9th congress 1984, pp. 1603–1608. DOI:10.1016/S1474-6670(17)61205-9

[4] C. Cortes and V. Vapnik: Support-vector networks. Mach. Learn *20* (1995), 3, 273–297. DOI:10.1007/BF00994018

[5] G. N. Elnagar and M. Razzaghi: A collocation-type method for linear quadratic optimal control problems. Opt. Control Appl. Methods *8* (1997), 3, 227–235.

[6] B. Kafash, A. Delavarkhalafi, and S. M. Karbassi: Application of Chebyshev polynomials to derive efficient algorithms for the solution of optimal control problems. Scientia Iranica *19* (2012), 3, 795–805. DOI:10.1016/j.scient.2011.06.012

[7] S. Kang and J. Wang, C. Li, and J. Shan: Nonlinear optimal control with disturbance rejection for asteroid landing. J. Franklin Inst. *355* (2018), 16, 8027–8048. DOI:10.1016/j.jfranklin.2018.06.028

[8] D. E. Kirk: Optimal Control Theory. Prentice-Hall, Englewood Cliffs 1970.

[9] S. Latifi, K. Parand, and M. Delkhosh: Generalized Lagrange–Jacobi–Gauss–Radau collocation method for solving a nonlinear optimal control problem with the classical diffusion equation. European Phys. J. Plus, 2020.

[10] R. C. Loxton, K. L. Teo, V. Rehbock, and W. K. Ling: Optimal switching instants for a switched-capacitor DC/DC power converter. Automatica J. IFAC *45* (2009), 4, 973–980. DOI:10.1016/j.automatica.2008.10.031

[11] H. R. Marzban and M. Razzaghi: Optimal control of linear systems via hybrid of block-pulse functions and Legendre polynomials. J. Franklin Inst. *341* (2004), 279–293. DOI:10.1016/j.jfranklin.2003.12.011

[12] K. M. Mohammadi, M. Jani, and K. Parand: A least squares support vector regression for anisotropic diffusion filtering. arXiv preprint arXiv, 2022.

[13] E. Naevdal: Solving continuous-time optimal-control problems with a spreadsheet. J. Econom. Educ. *34* (2003), 2, 99–122. DOI:10.3928/0022-0124-20030501-03

[14] L. O. Naraigh and A. Byrne: Piecewise-constant optimal control strategies for controlling the outbreak of COVID-19 in the Irish population. Math. Biosci *330* (2020), 108496. DOI:10.1016/j.mbs.2020.108496

[15] K. Parand, S. Latifi, M. Delkhosh, and M. M. Moayeri: Generalized Lagrangian Jacobi-Gauss-Radau collocation method for solving a nonlinear 2-D optimal control problem with the classical diffusion equation. arXiv preprint arXiv, 2018.

[16] K. Parand, M. Razzaghi, R. Sahleh, and M. Jani: Least squares support vector regression for solving Volterra integral equations. Engineering with Computers, 2020.

[17] A. Pakniyat, K. Parand, and M. Jani: Least squares support vector regression for differential equations on unbounded domains. Chaos Solitons Fractals, 2021.

[18] L. S. Pontryagin, V. Boltyanskii, R. Gamkrelidze, and E. Mischenko: The Mathematical Theory of Optimal Processes. Wiley Interscience, 1962.

[19] K. Rabiei and K. Parand: Collocation method to solve inequality-constrained optimal control problems of arbitrary order. Engrg. Comput. *36* (2020), 1, 115–125.

[20] M. Razzaghi, J. Nazarzadeh, and K. Y. Nikravesh: A collocation method for optimal control of linear systems with inequality constraints. Math.Problems Engrg. *3* (1998), 6, 503–515.

[21] R. Rifkin, G. Yeo, and T. Poggio: Regularized least-squares classification. Nato Sci. Ser. Sub Ser. III Comput. Syst. Sci. *190* (2003), 131–154.

[22] A. Schwartz: Theory and Implementation of Numerical Mathod Based on Runge–Kutta Integration for Solving Optimal Control Problems. Ph.D. Thesis, University of California, 1996.

[23] J. Shen, T. Tang, and L-L. Wang: Spectral Methods: Algorithms, Analysis and Applications. Springer Science and Business Media, 2011.

[24] J. A. Suykens and J. Vandewalle: Least squares support vector machine classifiers. Neural Process. Lett. *9* (1999), 3, 293–300. DOI:10.1023/A:1018628609742

[25] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle: Least Squares Support Vector Machines. World Scientific, 2002.

[26] K. L. Teo, C. J. Goh, and K. H. Wong: Unified Computational Approach to Optimal Control Problems. Longmann Scientific and Technical, 1991.

[27] S. Sabermahani, Y. Ordokhani, K. Rabiei, and M. Razzaghi: Solution of optimal control problems governed by Volterra integral and fractional integro-differential equations. J. Vibration Control *29* (2023), 15.-16, 3796–3808. DOI:10.1177/10775463221105923

[28] K. Rabiei and M. Razzaghi: An approach to solve fractional optimal control problems via fractional-order Boubaker wavelets. J. Vibration Control *29* (2023), 7–8, 1806–1819. DOI:10.1177/10775463211070902

[29] M. H. Heydari, M. Razzaghi, and Z. Avazzadeh: Orthonormal piecewise Bernoulli functions: Application for optimal control problems generated. Vibration Control *29* (2023), 5–6, 1164–1175. DOI:10.1177/10775463211059364

[30] M. H. Heydari, R. Tavakoli, and M. Razzagh: Application of the extended Chebyshev cardinal wavelets in solving fractional optimal control problems with ABC fractional derivative. Int. J. Systems Sci. *53* (2022), 12, 2694–2708. DOI:10.1080/00207721.2022.2058640

[31] G. Ghanbari and M. Razzaghi: Numerical solutions for fractional optimal control problems by using generalised fractional-order Chebyshev wavelets. Int. J. Systems Sci. *53* (2022), 4, 778–792. DOI:10.1080/00207721.2021.1972357

[32] M. H. Heydari and M. Razzaghi: A new class of orthonormal basis functions: application for fractional optimal control problems. Int. J. Systems Sci. *53* (2022), 2, 240–252.

[33] M. Lakestani, Y. Edrisi-Tabrizi, and M. Razzaghi: Study of B-spline collocation method for solving fractional optimal control problems. Trans. Inst. Measurement Control *43* (2021), 11, 2425–2437. DOI:10.1177/0142331220987537

[34] H. Dehestani, Y. Ordokhani, and M. Razzaghi: Fractional-order Bessel wavelet functions for solving variable order fractional optimal control problems with estimation error. Int. J. Systems Sci. *51* (2020), 6, 1032–1052. DOI:10.1080/00207721.2020.1746980

[35] S. Mashayekhi and M. Razzaghi: An approximate method for solving fractional optimal control problems by hybrid functions. J. Vibration Control *24* (2018), 9, 1621–1631.

[36] J. K. Mohammed and A. R. Khudair: A novel numerical method for solving optimal control problems using fourth-degree hat functions. J. Partial Differential Equations Appl. Math. *7* (2023), 9, 100507. DOI:10.1016/j.padiff.2023.100507

[37] H. Hassani, J. A. Tenreiro Machado, Z. Avazzadeh, E. Naraghirad, and M. Sh. Dahaghin: Generalized Bernoulli polynomials: Solving nonlinear 2D fractional optimal control problems. J. Scient. Comput. *83* (2020), 2, 1–21.

[38] J. A. K. Suykens, J. Vandewalle, and B. D. Moor: Optimal control by least squares support vector machines. J. Partial Differential Equations Appl. Math. *14* (2001), 1, 23–35.

Mitra Bolhassani, Department of Applied Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran. Iran.
e-mail: m_bolhassani@sbu.ac.ir

Hassan Dana Mazraeh, Department of Computer and Data sciences, Faculty of Mathematical Sciences, Shahid Beheshti University. Tehran, Iran.
e-mail: h_danamazraeh@sbu.ac.ir

Kourosh Parand, (Corresponding author.) Department of Computer and Data sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran.
Department of Cognitive Modeling, Institute for Cognitive and Brain Sciences, Shahid Beheshti University, Tehran. Iran.
e-mail: k_parand@sbu.ac.ir

APPENDIX A

```
restart:
with(orthopoly):
with(Optimization):
with(Student[NumericalAnalysis]):
Digits := 20:
a, b := 0, 1:
mygamma1 := 10^0:
mygamma2 := 10^7:
Numbers := 5:
N1 := Numbers - 1:
N2 := Numbers - 1:
N3 := Numbers - 1:
N4 := Numbers - 1:
shift := (2*x - a - b)/(b - a):
train := fsolve(P(Numbers, shift)):
u_exact := (x, t) -> t^4*sin(x):
y_exact := (x, t) -> t^3*cos(x):
y := unapply(sum(sum(w[z, zz]*P(z, 2*x - 1)*P(zz, 2*
    t - 1), zz = 0,\ldots,N2), z = 0,\ldots, N1), x,
    t):
u := unapply(sum(sum(v[z, zz]*P(z, 2*x - 1)*P(zz, 2*
    t - 1), zz = 0,\ldots,N4), z = 0,\ldots,N3), x, t
    ):
J := 1/2*int((u(x, t) - t^4*sin(x))^2 + (y(x, t) - t
    ^3*cos(x))^2, x = a,\ldots,b, t = a,\ldots,b):
condition1 := unapply(fracdiff(u(x, t), t, 0.5) -
    cos(u(x, t)) - 2*sin(x)*diff(u(x, t), x) - diff(u
    (x, t), xx) - 6*sin(x)*y(x, t) + cos(t^4*sin(x))
    + t^3*(t*sin(2*x) - t*sin(x) + 3*sin(2*x)) -
    GAMMA(5)*sin(x)*t^3.5/GAMMA(4.5), x, t):
condition2 := unapply(u(x, 0), x):
condition3 := unapply(u(0, t), t):
constraints := {}:
k := 1:
for i in train do
for j in train do
constraints := constraints union {condition1(i, j) =
     e[k]}:
k := k + 1:
end do:
end do:
for i in train do
```

```
        constraints := constraints union {condition2(i) = e[
            k]}:
        k := k + 1:
        constraints := constraints union {condition3(i) = e[
            k]}:
        k := k + 1:
        end do:
        cost := 1/2*(sum(sum(w[c1, c2]^2, c2 = 0,\ldots,N2),
             c1 = 0,\ldots,N1) + sum(sum(v[c3, c4]^2, c4 =
            0,\ldots,N4), c3 = 0,\ldots,N3)) + mygamma1/2*sum
            (e[c5]^2, c5 = 1,\ldots,k - 1) + mygamma2*J:
        result := NLPSolve(cost, constraints):
        assign(result[2]):
        # Visualization and calculating the mean absolute
            error`
        plot3d(u(x, t), x = 0,\ldots,1, t = 0,\ldots,1,
            labels = ["x", "t", "Approximated u(x,t)"],
            labeldirections = [horizontal, horizontal,
            vertical], axesfont = [Times, Normal, 15],
            labelfont = [Times, Normal, 20]);
        plot3d(u_exact(x, t), x = 0,\ldots,1, t = 0,\ldots
            ,1, labels = ["x", "t", "Exact u(x,t)"],
            labeldirections = [horizontal, horizontal,
            vertical], axesfont = [Times, Normal, 15],
            labelfont = [Times, Normal, 20]);
        plot3d([u(x, t), u_exact(x, t)], x = 0,\ldots,1, t =
             0,\ldots,1, labels = ["x", "t", "Approximated
            and exact u(x,t)"], labeldirections = [horizontal
            , horizontal, vertical], axesfont = [Times,
            Normal, 15], labelfont = [Times, Normal, 20]);
        plot3d(abs(u(x, t) - u_exact(x, t)), x = 0,\ldots,1,
             t = 0,\ldots,1, labels = ["x", "t", "Absolute
            error between the exact and the approximated u(x,
            t)"], labeldirections = [horizontal, horizontal,
            vertical], axesfont = [Times, Normal, 15],
            labelfont = [Times, Normal, 12]);
        plot3d(y(x, t), x = 0,\ldots,1, t = 0,\ldots,1,
            labels = ["x", "t", "Approximated y(x,t)"],
            labeldirections = [horizontal, horizontal,
            vertical], axesfont = [Times, Normal, 15],
            labelfont = [Times, Normal, 20]);
        plot3d(y_exact(x, t), x = 0,\ldots,1, t = 0,\ldots
            ,1, labels = ["x", "t", "Exact y(x,t)"],
            labeldirections = [horizontal, horizontal,
            vertical], axesfont = [Times, Normal, 15],
```

```
        labelfont = [Times, Normal, 20]);
    plot3d([y(x, t), y_exact(x, t)], x = 0,\ldots,1, t =
         0 ,\ldots,1, labels = ["x", "t", "Approximated
        and exact y(x,t)"], labeldirections = [horizontal
        , horizontal, vertical], axesfont = [Times,
        Normal, 15], labelfont = [Times, Normal, 20]);
    plot3d(abs(y(x, t) - y_exact(x, t)), x = 0,\ldots,1,
         t = 0,\ldots,1, labels = ["x", "t", "Absolute
        error between the exact and the approximated y(x,
        t)"], labeldirections = [horizontal, horizontal,
        vertical], axesfont = [Times, Normal, 15],
        labelfont = [Times, Normal, 12]);
    E1 := 0:
    E2 := 0:
    for i in train do
    for j in train do
    E1 := E1 + abs(u(i, j) - u_exact(i, j)); E2 := E2 +
        abs(y(i, j) - y_exact(i, j)):
    end do:
    end do:
    MAE_u := evalf(E1/(Numbers*Numbers));
    MAE_y := evalf(E2/(Numbers*Numbers));
```

Listing 1: Maple Code of Example 6