A Computational Framework for Optimal Control Using Polynomial-Based Approximation Methods with Applications in Intelligent Systems

Wydian Razaq Al-Dulaimi and Suha Shihab
College of Applied Science, University of Technology, 10066 Baghdad, Iraq
wadian973@gmail.com , *suha.n.shihab@uotechnology.edu.iq*

Keywords: Optimal Control Problem, Modified Vieta-Pell Functions, Direct Approximation Algorithms, Intelligent Robotic System, IT Applications.

Abstract: This work suggests an approximate direct technique to treat special kind of optimal control problem (OCP) in the finite domain [0, T] based on new modified family of Vieta-Pell functions. The proposed method employs a direct parameterization approach. In such technique, we approximate the state variables in terms modified family of Vieta-Pell functions and then transform the original optimal control problem to a constrained nonlinear programming problem. The aim of the presented algorithm is to reduce the numerical complexity with computational efficiency. The algorithm is well-suited for implementation in scientific computing environments such as MATLAB and Python, making it applicable to software-based control systems. To demonstrate its practical relevance to information technology applications, the proposed method is applied to optimal control problems arising in intelligent robotic systems, particularly robotic arm motion planning. Numerical simulations confirm fast convergence, high accuracy, and robustness, making the method suitable for integration into digital control architectures, intelligent automation systems, and IT-based robotic platforms.

1 INTRODUCTION

Optimal control problem (OCP) has been extensively considered in recent years. The major reason for the great importance of such problems are found in many applications in science, engineering, medicine, economics, finance, and a variety of other disciplines [1]-[3]. In recent years, the different algorithms to solve optimal control problem have been studied, such as the state parameterization technique with operation matrix of derivative for modified wavelets functions [4], the novel variational finite difference method [5], the difference method [6]-[7] and the finite element method to solve an OCP governed by e fractional wave equation [8], a wavelet collocation method for nonlinear partial OCP [9], Chebyshev wavelet and Müntz–Legendre Wavelet methods for solving various OCP [10-12], the Haar wavelet collocation method and Boubaker Wavelets [13]-[15]. The orthogonal functions are the most popular numerical approximates in solving an optimal control problem [16]-[18]. The ease and accuracy of applying the basic functions for treating an OCP and other many problems are our aim. Therefore, the authors in [19]-[21] extended solving different complicated problem based on numerical techniques together with the interesting basis functions called Vieta-Pell polynomials, such polynomials have utilized as a suitable technique for treating approximately a nonlinear fractional OCP [22]. A special Burgers an equation (BE) with fraction order was studied in [23] to solve BE approximately with the aid of certain family for Vieta Pell polynomials. In addition, the Vieta Pell polynomials are used in [24], [25] to solve stochastic fractional differential equations using direct technique. Some other work [26]-[29] used the generalized of Vieta Pell polynomials and their incomplete families.

The aim of the present work is the approximate investigation of the following optimal control problem:

$$\min J(x,u) = \int_0^T F(x(t),u(t),t)dt. \quad (1)$$

Subject to

$$\dot{x}(t) = g\big(x(t),u(t)\big), \quad (2)$$

and

$$x(0) = x_0, x(T) = x_T. \quad (3)$$

Here the two state and control vectors are $x(t)$ and $u(t)$ respectively.

The proposed modified version of the Vieta-Pell polynomials can use for many problems with finite intervals. In modern information technology applications, optimal control plays a vital role in intelligent systems such as robotics, autonomous platforms, embedded controllers, and cyber-physical systems. The effectiveness of these systems relies heavily on fast and accurate numerical algorithms that can be efficiently implemented in software environments. Consequently, developing computationally efficient basis functions and operational matrices is essential for real-time optimization and control. In this regard, the proposed MSIVP-based method provides a practical numerical framework that can be directly integrated into software-based control architectures used in intelligent robotic and IT-driven systems.

The paper is organized as follows. In Section 2, we introduce a new definition of MSIVP together with some interesting properties. In Section 3, we suggest a new approximated method to show how to transform an OCP (1-3) into an optimization one. We give two examples of our method in Section 4. In Section 5, the verification method on the IT applications is demonstrated and list conclusions in Section 6.

2 MODIFIED SHIFTED INCOMPLETE VIETA-PELL FUNCTIONS

In this section, we show how to modify the incomplete polynomials in [26] over limited domain to get an interesting basic functions on the domain $[0, T]$.

Definition 1. Let $M_n(t)$ denotes the modified incomplete Vieta Pell functions and s be any real positive numbers, then

$$M_n(t) = 2^s\left(\frac{2t}{T} - 1\right)M_{n-1}(t), n \geq 2, \quad (4)$$

where $M_1 = 1$.

In general, modified polynomials can draw as follows

$$M_n(t) = 2^{(n-1)s}\left(\frac{2t}{T} - 1\right)^{n-1}, n = 1, 2, \ldots. \quad (5)$$

Some important properties are given in the next subsections to reduce the computations in working with new basis functions.

2.1 The MSIVP Product Operation Matrix

Theorem 1. Let $M_n(t)$ and $M_m(t)$ be as chosen in (4). Then for all $n, m \geq 1$, one has

$$M_n(t)M_m(t) = M_{|n+m-1|}(t). \quad (6)$$

Proof. Assume$n = 1$, then:

$$M_1(t)M_m(t) = M_m(t), m \geq 1. \quad (7)$$

Consider that {4} is true for k, yields

$$M_k(t)\, M_m(t) = M_{|k+m-1|}(t). \quad (8)$$

We must prove that (4) is true for $k + 1$. By multiplying both sides of (8) by $2^s z$ to get,

$$2^s(z)M_k(t)\, M_m(t) = 2^s(z)M_{|k+m-1|}(t).$$

$z = \frac{2t}{T} - 1$ and making use of relation (4), which immediately gives

$$M_{k+1}(t)\, M_m(t) = M_{|k+m-1|}(t).$$

and this completes the proof of Theorem 1.

2.2 Operation Matrix of Derivative of MSIVP

Theorem 2. Assume that $M_n(t)$ as defined in (4). Then $\dot{M}_n(t), n \geq 2$, can be defined in below

$$\dot{M}_n(t) = (n-1)\frac{2}{T}2^S M_{n-1}(t), n \geq 2\,. \quad (10)$$

Proof. Let $n = 2$, then $\dot{M}_2(t) = 2^S\frac{2}{T}M_1(t)$.

Assume that (10) is true for $(n - 1)$. Using (4), yields

$$\dot{M}_n(t) = 2^S\left[(z)\dot{M}_{n-1}(t) + \frac{2}{T}M_{n-1}(t)\right]. \quad (11)$$

Then from (4) one can get

$$\dot{M}_n(t) = M_2(t)\dot{M}_{n-1}(t) + 2^S\frac{2}{T}M_{n-1}(t). \quad (12)$$

Now, application of the induction step on $\dot{M}_{n-1}(t)$ in (12), yields

$$\dot{M}_n(t) = (n-2)\frac{2}{T}2^S M_2(t)M_{n-2}(t) + 2^S\frac{2}{T}M_{n-1}(t).$$

It can simplify based on the results of theorem 1, to get

$$\dot{M}_n(t) = (n-2)\frac{2}{T}2^S M_{n-1}(t) + 2^S\frac{2}{T}M_{n-1}(t).$$

3 THE MSIVP ALGORITHM

The step by step MSIVP algorithm for the established method is outlined below.

Input: Initial position $(0) = x_0$, Desired final position $x(T) = x_f$, Final time $T = 1$, Approximation order n, Cost functional, Control definition.

Output: Approximate state trajectory $x(t)$, Approximate control signal $u(t)$, Optimal cost value J, Error norms L_2, L_∞.

Initialize MSIVP Basis Functions. Construct the MSIVP approximation of order n: $x_n(t) = \sum_{i=1}^{n} a_i 2^{(i-1)s}(z)^{i-1}$, Where $s = 1, T = 1$.

Compute the State Derivative. Evaluate the derivative:$\dot{x}_n(t)$

Define the Control Input.

Apply Boundary Conditions. Enforce: $(0) = x_0$, $x(T) = x_f$.

Formulate the Cost Functional. Substitute $x_n(t)$ and $u_n(t)$ into:

Solve the Optimization Problem. Minimize $J(a)$ subject to boundary constraints using a nonlinear programming solver to obtain the optimal coefficient vector: $a^* = (a, a_1, a_2, \ldots\ldots, a_n)$

Construct Optimal Trajectories. Substitute a^* into: $x^*(t) = u^*(t), x_n(t; a^*) = u_n(t; a^*)$

Compute Error Measures. Evaluate approximation accuracy using: $L_{2,z} = \sqrt{\sum_{j=1}^{N} \left|z_{ex.}(t_j) - z_n(t_j)\right|^2}$, $L_{\infty,z} = \max_j \left[z_{ex.}(t_j) - z_n(t_j)\right], j = 1, 2, \ldots, N$, where $z = (x, u)$.

4 NUMERICAL RESULTS

The accuracy of the proposed method is determined based on the following norms:

$$L_{2,x} = \sqrt{\sum_{j=1}^{N} \left|x_{ex.}(t_j) - x_n(t_j)\right|^2},$$
$$L_{\infty,x} = \max_j \left[x_{ex.}(t_j) - x_n(t_j)\right], j = 1, 2, \ldots, N.$$
$$L_{2,u} = \sqrt{\sum_{j=1}^{N} \left|u_{ex.}(t_j) - u_n(t_j)\right|^2},$$
$$L_{\infty,x} = \max_j \left[u_{ex.}(t_j) - u_n(t_j)\right], j = 1, 2, \ldots, N.$$

where $x_n(t)$, $u_n(t)$, $x_{ex.}(t)$ and $u_{ex.}(t)$ represent the approximate solution for the state variable, the approximate solution for the control variable, the approximate exact for the state variable and the exact solution for the control variable respectively. The absolute errors E for the performance index value is calculated using varies basis function orders as follow

$$E = |J_{ex.} - J_n|.$$

Here J_n and $J_{ex.}$ Represents respectively the approximate and exact cost values.

Example 1. In this problem, an object is to find the optimal control that minimizes

$$J = \int_0^1 (u^2 + x^2) dt.$$

Subject to $u(t) = \dot{x}(t)$,
Together with $x(0) = 0, x(1) = 0.5$.
while $x(t) = \frac{e(e^t - e^{-t})}{2(e^2 - 1)}$, $u(t) = \frac{e(e^t + e^{-t})}{2(e^2 - 1)}$ and $J_{exact} = 0.328258821379$ are the exact values for the state, control and the performance index value respectively.

The numerical results are obtained for both the state and the control variables at $n = 3,4$ and 5 and plotted in Figures 1 and 2. The analysis error norms are given in Tables 1 and 2.

Table 1: Errors for state of Example 1.

n	x(t)	
	L_2	Infinity norm
3	1.169575e-2	3.96544e-3
4	2.971882e-4	5.84650e-5
5	2.51801e-5	3.42773e-6

Table 2: Errors for control of Example 1.

n	u(t)		J
	L_2	Infinity norm	E
3	0.120678212	4.2881e-2	3.396e-4
4	5.110e-5	2.1836e-3	5.161e-7
5	9.4961 e-4	2.9811 e-4	9.330e-9

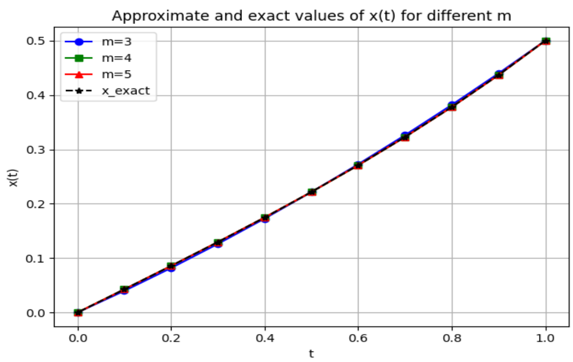

Figure 1: Approximate for state of Example 1.

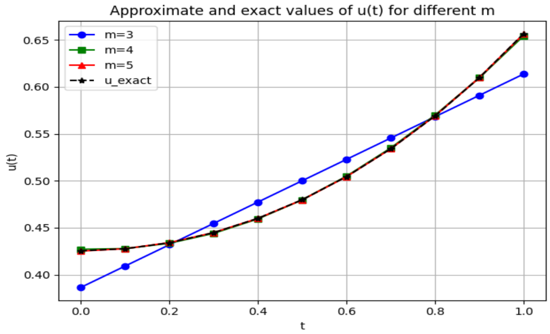

Figure 2: Approximate for control of Example 1.

A comparison is made between the present algorithm and the solution obtained by wavelets functions technique [4] for different values of n. It is based on the special wavelets basis functions to approximate the solution of the application 1 by using the state parameterization method. The absolute errors values E are respectively 0.000339663469485, 0.000000516182663 and 0.00000000933001 with $n = 3, 4{,}5$. The advantages of the suggested incomplete technique on solving OCP is the ability for solving the OCP directly with the interval $t \in [0, T]$ while other methods must introduce a suitable transformation according to the base functions.

Example 2. The aim is to minimize the following performance index

$$J = \frac{1}{2}\int_0^2 u^2(t)dt, \qquad 0 \le t \le 2,$$

with $u(t) = \dot{x}(t) + \ddot{x}(t)$ and $x(0) = 0, \dot{x}(0) = 0, x(2) = 5, \dot{x}(2) = 2$.

While $x(t) = K_1 + K_2 e^{-t} + C_1 t + \frac{C_2}{2} e^t$, $u(t) = C_1 + C_2 e^t$ is the exact solutions where $K_1 = -\frac{3e^4 - 2e^2 + 7}{4(e^2 - 1)}$,

$K_2 = \frac{e^2(1 + 3e^2)}{4(e^2 - 1)}$, $C_1 = \frac{7 + 3e^2}{4}$, $C_2 = \frac{7 - 3e^2}{2(e^2 - 1)}$.

$J_{exact} = 16.750723396861265$.

The obtained values of $L_{2,x}$. $L_{\infty,x}$, $L_{2,u}$. $L_{\infty,u}$ and E are illustrated at some selected n within the interval $t \in [0,2]$ and the results are given in Tables 3 and 4. In addition, Figures 3 and 4 plot the absolute errors for the state and control variables respectively.

Table 3: Errors for state of Example 2.

n	x(t)	
	L_2	Infinity norm
6	4.858775e-4	5.460372e-5
7	1.188900e-4	2.705501e-5
8	1.893695e-6	2.346852e-7

Table 4: Errors for control of Example 1.

n	u(t)		J
	L_2	Infinity norm	E
6	2.753e-2	4.129e-3	1.005e-5
7	1.420e-3	4.4723e-3	1.858e-8
8	2.706e-4	5.4293e-4	5.57e-10

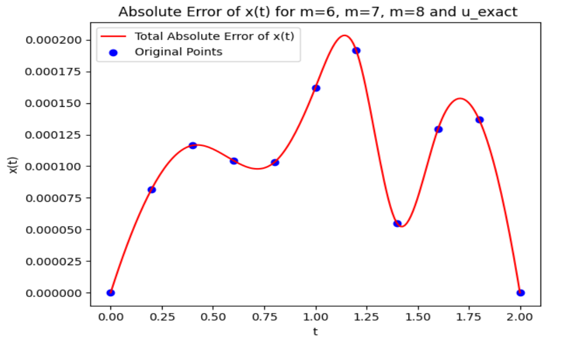

Figure 3: Absolute errors for state of Example 2.

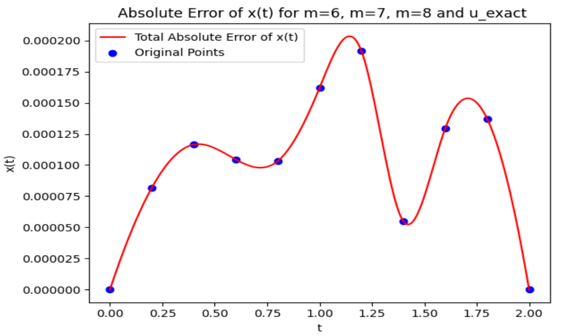

Figure 4: Absolute error for control of Example 2.

According to the Tables, as n increases, the values of E, L_2 and L_∞ decrease rapidly and the results reach to the exact values, the accuracy increases as the number of terms of the basis MSIVP expansion increases. Tables (1-4) illustrate that the suggested algorithm has a good convergence rate.

5 COMPUTATIONAL IMPLEMENTATION AND IT APPLICATION

The proposed MSIVP algorithm is developed with practical computational implementation as a primary consideration. Owing to the simplicity of the derived operational matrices, the algorithm can be efficiently implemented using standard scientific computing environments such as MATLAB, Python, and C-based numerical libraries which can allow the algorithm to be useful for modern IT infrastructures. The presented modified basis functions can be used to find optimal state trajectory and control input in real time with a few computational cost of the suggested algorithm. Furthermore, the proposed technique can apply in simulation-based control design and enable the engineers to calculate the system performance with the aid of numerical simulations before real-world implementation. This illustrates the relevance of the suggested algorithm to applications of information technology, especially in robotic control software and intelligent automation,

5.1 The Formulation for Robotic Arm Motion Problem

Here, a robotic arm motion in one dimension is presented. The arm moves from the initial position $x(0) = 1$ to the desired terminal position $x(1) = 0.2818$ within the time interval $0 \leq t \leq 1$ as illustrated in Figure 5. The aim is to find an optimal state trajectory and a control input defined as $x(t)$ and $u(t)$ respectively. The relation between them is made as

$$u(t) = \dot{x}(t) + x(t).$$

In which an accurate terminal position tracking is reached, the motion stays smooth to avoid vibrations and minimized the control effort. The cost of the control system is determine using the following quadratic cost functional

$$J = \frac{1}{2}\int_0^1 u^2(t) + x^2(t)dt.$$

The cost value J defines the balancing of energy consumption with deviation from the desired position. This formulation is widely used in optimal control problems of robotic manipulators to ensure efficient and precise motion while extending the operational lifespan of mechanical components.

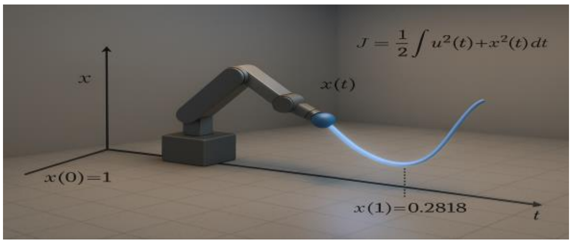

Figure 5: Illustration of robotic arm motion.

5.2 Robotic Arm Control Code in Python

```
import numpy as np
import matplotlib.pyplot as plt
# =====================================
# 1) Constants
# =====================================
sqrt2 = np.sqrt(2)
# Constant A from exact formula
A = (2*sqrt2 - 3) / (-np.exp(2*sqrt2) + 2*sqrt2 - 3)
# =====================================
# 2) Exact analytical solutions
# =====================================
def x_exact(t):
 return A*np.exp(sqrt2*t) + (1 - A)*np.exp(-sqrt2*t)
def u_exact(t):
 return A*(sqrt2 + 1)*np.exp(sqrt2*t) - (1 - A)*(sqrt2 -
1)*np.exp(-sqrt2*t)
# =====================================
# 3) Time grid
# =====================================
t = np.linspace(0.0, 1.0, 11)
# =====================================
# 4) Direct evaluation
# =====================================
x = x_exact(t)
u = u_exact(t)
# =====================================
# 5) Boundary condition check
# =====================================
print("x(0) =", format(x_exact(0.0), ".10f"))
print("x(1) =", format(x_exact(1.0), ".10f"))
# =====================================
# 6) Table of values
# =====================================
print("\n t x(t) u(t)")
print("-----------------------------------")
for ti, xi, ui in zip(t, x, u):
 print(f"{ti:3.1f} {xi:.10f} {ui:.10f}")
# =====================================
# 7) Cost functional J (numerical integration only)
# =====================================
J = 0.5 * np.trapz(u**2 + x**2, t)
print("\nJ_exact =", format(J, ".15f"))
# =====================================
# 8) Plot
# =====================================
plt.figure(figsize=(8,4))
plt.plot(t, x, 'o-', label='x(t)')
plt.plot(t, u, 's-', label='u(t)')
plt.xlabel('t')
plt.ylabel('Value')
plt.grid(True)
plt.legend()
plt.show().
```

5.3 Numerical Results for Robotic Arm Control

To solve this problem, the MSIVP method of order six is employed to approximate the state variable as

$$x_6(t) = \sum_{i=1}^{6} a_i 2^{(i-1)s}(z)^{i-1},$$

where s=1$s = 1$ and $T = 1$. The unknown coefficients are determined by solving a nonlinear programming problem subject to the boundary conditions. The optimal parameter vector is obtained as:

$$a_1 = 1,$$
$$a_2 = -0.692898043157134,$$
$$a_3 = 0.249438742839363,$$
$$a_4 = -0.056389173902395,$$
$$a_5 = 0.008969735170030,$$
$$a_6 = -0.000705039267015.$$

Substituting these coefficients yields the approximate state and control functions

$$x(t) = -0.0226t^5 + 0.1435t^4 - 0.4511t^3 + 0.9978t^2 - 1.3858t + 1.$$
$$u(t) = -0.0226t^5 + 0.0307t^4 + 0.1229t^3 - 0.3556t^2 + 0.6097t - 0.3858.$$

with an optimal cost value of $J = 0.192909335341236$.

In addition, the problem is solved using higher approximation orders through an extended incomplete series representation. The numerical accuracy is evaluated using the L_2 norm and theL_∞ norm for both the state and control variables over the interval $t \in [0,1]$. The corresponding error values are summarized in Tables 5 and 6, while the approximate solutions for different values of n are illustrated in Figures 6 and 7 while Figure 8 Statistical Reading of cost Functional Error.

Table 5: Errors for state of robotic arm problem.

n	x(t)	
	L_2	Infinity norm
6	6.48474317e-4	1.43087492e-4
7	6.46349909e-4	1.44111405e-4
8	6.46349909e-4	1.44160392e-4

Table 6: Errors for control of robotic arm problem.

n	u(t)*10^{-4}		J
	L_2	Infinity norm	E
6	2.744482616	5.69549097	8.642e-10
7	2.697338464	4.31755836	2.539e-12
8	2.710904281	4.39874067	7.993e-15

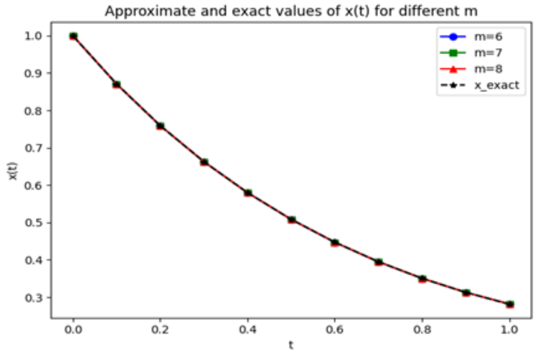

Figure 6: Approximate for state of robotic arm problem.

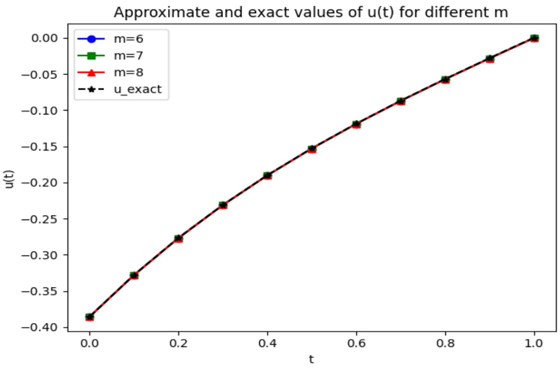

Figure 7: Approximate for control of robotic arm problem.

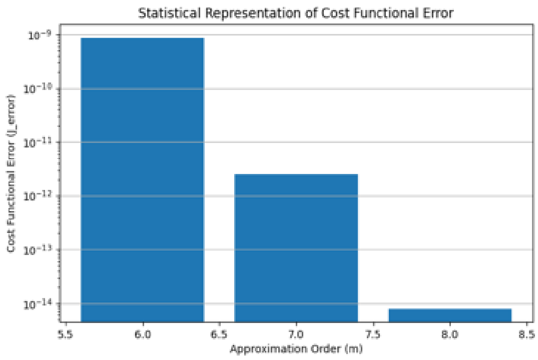

Figure 8: Statistical reading of cost functional error of robotic arm problem.

The simulation results show that the MSIVP method produces smooth and accurate trajectories for both the state and control signals. Error analysis indicates that the deviations in the state and control remain small throughout the entire time interval, confirming the effectiveness and numerical stability of the method.

6 CONCLUSIONS

This paper introduced an efficient computational approach based on modified shifted incomplete Vieta-Pell polynomials for solving optimal control problems over general time intervals. The proposed method transforms the original control problem into a numerically manageable optimization problem through simple and effective operational matrices. From an information technology standpoint, the MSIVP algorithm provides a flexible and computationally efficient tool suitable for implementation in software-based control systems, intelligent robotic platforms, and digital automation frameworks. Numerical results demonstrate fast convergence, high accuracy, and reduced computational effort, making the method appropriate for real-time and simulation-based IT applications. The effectiveness of the proposed method is further validated through robotic motion control examples, confirming its applicability to intelligent systems and IT-driven control environments. Future work may focus on real-time implementation and integration with advanced robotic software architectures.

REFERENCES

[1] A. Diveev and E. Sofronova, “Optimal control problem and its solution in class of feasible control functions by advanced model of control object,” Mathematics, vol. 13, 2025, doi: 10.3390/math13040674.

[2] D. Michael, “Applications of optimal control theory in economics,” Synthese, vol. 31, no. 2, pp. 271–288, 1975.

[3] S. P. Gatyeni, C. W. Chukwu, F. Chirove, F. Nyabadza, and Fatmawati, “Application of optimal control to the dynamics of COVID-19 disease in South Africa,” Scientific African, vol. 16, 2022, doi: 10.1016/j.sciaf.2022.e01268.

[4] G. Abass and S. Shihab, “Operational matrix of new shifted wavelet functions for solving optimal control problem,” Mathematics, vol. 11, p. 3040, 2023, doi: 10.3390/math11143040.

[5] A. Norouzzadeh, N. Nariman-Zadeh, and R. Ansari, “A novel unified variational finite difference (VFD) solution method for optimal control problems,” Measurement and Control, vol. 58, no. 5, pp. 682–695, 2024, doi: 10.1177/00202940241269618.

[6] A. Yıldırım, D. N. Hào, and G. Yagub, “Finite difference method for an optimal control problem for a nonlinear time-dependent Schrödinger equation,” Numerical Functional Analysis and Optimization, vol. 38, no. 6, pp. 799–817, 2017, doi: 10.1080/01630563.2016.1266656.

[7] D. Baleanu, M. Hajipour, and A. Jajarmi, “An accurate finite difference formula for the numerical solution of delay-dependent fractional optimal control problems,” IJOCTA, vol. 14, no. 3, pp. 183–192, 2024, doi: 10.11121/ijocta.1478.

[8] W. Shuo, Z. Xiangcheng, and D. Ning, “Finite element method for an optimal control problem governed by a time fractional wave equation,” Computers & Mathematics with Applications, vol. 164, pp. 45–66, 2024, doi: 10.1016/j.camwa.2024.03.034.

[9] M. Khaksar and M. Shamsi, “An adaptive wavelet collocation method for solving optimal control of elliptic variational inequalities of the obstacle type,” Computers & Mathematics with Applications, vol. 75, no. 2, pp. 470–485, 2018, doi: 10.1016/j.camwa.2017.09.026.

[10] M. Yarahmadi and S. Yaghobipour, “Chebyshev wavelet-based method for solving various stochastic optimal control problems and its application in finance,” Iranian Journal of Numerical Analysis and Optimization, vol. 14, no. 1, pp. 1–19, 2024, doi: 10.22067/ijnao.2023.82445.1265.

[11] A. Ebrahimzadeh, “Solving the optimum control problem constrained with Volterra integro-differential equations using Chebyshev wavelets and particle swarm optimization,” AUT Journal of Mathematics and Computing, in press, 2024, doi: 10.22060/ajmc.2024.23406.1254.

[12] R. Kumari and M. Mehra, “Müntz–Legendre wavelet collocation method for the solution of optimal control problems with fractional-loaded differential equations,” SN Computer Science, vol. 6, p. 461, 2025, doi: 10.1007/s42979-025-03997-0.

[13] K. Maleknejad and A. Ebrahimzadeh, “The use of rationalized Haar wavelet collocation method for solving optimal control of Volterra integral equations,” Journal of Vibration and Control, vol. 21, no. 10, pp. 1958–1967, 2013, doi: 10.1177/1077546313504977.

[14] C. Hsiao and J. Wang, “Optimal control of linear time-varying systems via Haar wavelets,” Journal of Optimization Theory and Applications, vol. 103, pp. 641–655, 1999, doi: 10.1023/A:1021740209084.

[15] H. Eman, F. Samaa, and N. Shihab, “Boubaker wavelets functions: Properties and applications,” Baghdad Science Journal, vol. 18, no. 4, pp. 1226–1233, 2021, doi: 10.21123/bsj.2021.18.4.1226.

[16] S. Al-Rawi, F. Al-Heety, and S. Hasan, “New computational method for optimal control problem with B-spline polynomials,” Engineering and Technology Journal, vol. 28, no. 18, pp. 5711–5718, 2010.

[17] A. Fatema, “Solution of problems in calculus of variations using parameterization technique,” Engineering and Technology Journal, vol. 32, no. 6, 2014.

[18] N. Shihab and J. Al-Eleiwy, “Chebyshev polynomials and spectral method for optimal control problem,” Engineering and Technology Journal, vol. 27, no. 14, 2009.

[19] D. Tasci and F. Yalcin, “Vieta-Pell and Vieta-Pell-Lucas polynomials,” Advances in Difference Equations, p. 224, 2013, doi: 10.1186/1687-1847-2013-224.

[20] S. Wanna, S. Somnuk, and N. Peesiri, “Vieta–Fibonacci-like polynomials and some identities,” Annales Mathematicae et Informaticae, vol. 54, pp. 97–108, 2021, doi: 10.33039/ami.2021.09.002.

[21] B. Souhila, S. Nabiha, and B. Ali, “Construction of generating functions of the products of Vieta polynomials with Gaussian numbers and polynomials,” International Journal of Nonlinear Analysis and Applications, vol. 12, no. 1, pp. 649–668, 2021, doi: 10.22075/ijnaa.2020.20152.2131.

[22] S. Sabermahani, P. Rahimkhani, and Y. Ordokhani, “Application of generalized Lucas wavelet method for solving nonlinear fractal-fractional optimal control problems,” Chaos, Solitons & Fractals, vol. 170, 2023, doi: 10.1016/j.chaos.2023.113348.

[23] M. Taghipour and H. Aminikhah, “Application of Pell collocation method for solving the general form of time-fractional Burgers equations,” Mathematical Sciences, vol. 17, pp. 183–201, 2023, doi: 10.1007/s40096-021-00452-y.

[24] P. Singh and S. Saha, “An efficient numerical method based on Lucas polynomials to solve multi-dimensional stochastic Itô-Volterra integral equations,” Mathematics and Computers in Simulation, vol. 203, pp. 826–845, 2023, doi: 10.1016/j.matcom.2022.06.029.

[25] P. Singh and S. Ray, “A numerical approach based on Pell polynomial for solving stochastic fractional differential equations,” Numerical Algorithms, vol. 97, pp. 1513–1534, 2024, doi: 10.1007/s11075-024-01760-9.

[26] A. Waleed, F. Abdullah, S. Abu Mohammed, and A. Alharbi, “Spectral tau technique via Lucas polynomials for the time-fractional diffusion equation,” AIMS Mathematics, vol. 9, no. 12, pp. 34567–34587, 2024, doi: 10.3934/math.20241646.

[27] N. F. Yalc, D. Tas, and E. Erkus-Duman, “Generalized Vieta-Jacobsthal and Vieta-Jacobsthal-Lucas polynomials,” Mathematical Communications, vol. 20, pp. 241–251, 2015.

[28] K. Bahar, O. Engin, and G. Anthony, “Incomplete generalized Vieta–Pell and Vieta–Pell–Lucas polynomials,” Notes on Number Theory and Discrete Mathematics, vol. 27, no. 4, pp. 245–256, 2021, doi: 10.7546/nntdm.2021.27.4.245-256.

[29] U. Şükran, H. Karatas, and H. Aytar, “Notes on generalization of Vieta-Pell and Vieta-Pell-Lucas polynomials,” International Journal of Mathematics Research, vol. 12, no. 1, pp. 5–22, 2020.