Received: December 31, 2023. Revised: March 18, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Unsupervised Machine Learning Control Techniques for Solving the General Synthesis of Control System Problem

Karrar Sahib Nassrullah[1,2]* **Ivan Viktorovich Stepanyan[2,3]** **Haider Sahib Nasrallah[2,4]**
Neder Jair Mendez Florez[2] **Abdelrrahmane Mohamed Zidoun[2]** **Shahad Raouf Mohammed[1]**

[1]*Department of Mechanical Engineering, Collage of Engineering, University of Kerbala, Karbala 56001, Iraq*
[2]*Department of Mechanics and Control Processes, Academy of Engineering, Peoples' Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya Street, Moscow, 117198 Russia*
[3]*Mechanical Engineering Research University of the Russian Academy of Sciences, Moscow, 101000 Russia*
[4]*Department of Computer, Collage of Education, University of Kerbala, Karbala 56001, Iraq*
* Corresponding author's Email: almousawi_karrar@uokerbala.edu.iq

Abstract: The focus of this study is a relatively new development in the field of control: machine learning control. This study offers a mathematical framework for machine learning control and explores three symbolic regression-based techniques for supervised and unsupervised learning. One of the challenges associated with machine learning control pertains to the control general synthesis. This entails figuring out a control function contingent upon the object's state, guaranteeing the attainment of the control objective while optimizing the quality criterion value across all possible initial states within a permissible zone where finding a satisfactory solution occurs within the space of codes. The implementation of the small variations principle within the basic solution is suggested as a viable technique for developing the algorithms of search. This paper extensively discusses three symbolic regression techniques, including Cartesian genetic programming (CGP), synthesized genetic programming (SGP) and parse-matrix evolution (PME). Notably, synthesized genetic programming, being a novel technique, and PME get utilized for the first time to address the general synthesis of control problems. The mathematical expression's SGP code is a six-row integer matrix; the first row of the matrix represents the functions that take two arguments, while the second and fourth rows represent the functions that take one argument. The third and fifth rows represent the arguments of the mathematical expression, and the sixth row represents the priority. The computational example demonstrates the potential of symbolic regression approaches as unsupervised machine learning control techniques for addressing the machine learning control challenge of general synthesis of control in order to achieve the stability of a mobile robot system. Likewise, practical experience shows that synthesized genetic programming has faster efficiency than Cartesian genetic programming and parse-matrix evolution in discovering solutions, about 2.33 and 2.11 times on average, respectively.

Keywords: Unsupervised machine learning control, Symbolic regression, Cartesian genetic programming, Synthesized genetic programming, Parse-matrix evolution, Mobile robot.

1. Introduction

One of the most significant challenges in control theory is the general synthesis of control systems. The primary characteristic of the challenge concerning the general synthesis of the control system pertains to acquiring a control function under a range of initial conditions. The arguments of this control function structure consist of the various components that make up the state vector of the object. When the discovered control function is replaced in the model's right part of the control object, an ordinary differential equation system that does not have a vector of free control is produced. This kind of model is usually referred to as the model of closed-loop control. The primary distinction and intricacy of the synthesis problem, in contrast to the optimal control problem, lies in the presence of initial conditions within either the whole state space or some particular area within it. Additionally, the synthesis problem involves determining the control as a multidimensional function derived from the coordinates of the state space for multiple initial conditions. In contrast, the optimal control problem seeks to find the control as a function of time for a single initial condition.

Although the control synthesis problem is crucial, precise techniques for tackling it have yet to be available. Analytical solutions for the control synthesis problem are only available for simple objects of limited size. Analytical solutions for most applied control objects are generally unattainable.

In the 1950s [1], R. Bellman proposed the Dynamic Programming (DP) approach to tackle the control synthesis problem. He derived the dynamic programming equation, a partial differential equation system referred to as the optimality equation, the Bellman equation, or the Hamilton-Jacobi-Bellman (HJB) equation for continuous-time optimal control problems [2-4].

A general solution to the Bellman equation is necessary to address the problem of control system synthesis analytically. Obtaining an analytical solution employing this technique is nearly impossible for most practical problems. Existing solutions only analyze the Bellman equation for systems of low order.

When dealing with nonlinear systems, the DP approach can be used in conjunction with the HJB equation to find a numerical solution. Yet, the primary limitation of dynamic programming methods remains the computational complexity needed to define the value function and the restriction to a single initial condition because of the “curse of dimensionality”. Consequently, this numerical approach is not employed to address the general synthesis problem of control. Present-day methods like Adaptive Dynamic Programming (ADP) [5, 6] and Reinforcement Learning (RL) [7, 8] are gaining popularity as effective methodologies of machine learning for tackling control problems in nonlinear systems. These techniques involve using neural networks to numerically approximate solutions to the HJB equation. Nevertheless, these methods encounter numerous computational challenges, mainly concerning establishing and training the neural networks employed.

Not long after the formulation of the problem of optimal control [9], V.G. Boltyanski dubbed the problem of general synthesis of control this name and came up with its formulation in the late 1960s [10]. Using Pontryagin’s Maximum Principle (PMP), Boltyanskii and Pontryagin addressed numerous general synthesis problems for plain models of control objects. They yielded control functions based on logical deductions by analyzing the optimal trajectories’ set. Extending this technique to control higher dimensions was not feasible.

The control synthesis problem is currently being solved using many analytical techniques, such as integrator backstepping [11, 12], analytical design of aggregated regulators (ADAR) [13, 14], and control Lyapunov functions [15, 16].

The researcher manually implements the backstepping technique. Applying it yields good results for systems of low order. The primary limitation of this control technique is that it pre-demands a strict feedback form-based dynamic system in order to be controlled. Another flaw in this control design is the presence of an unavoidable phenomenon called “explosion of complexity” on top of these disadvantages [12].

The ADAR technique is effective for specific tasks. It is essential to consider that the control vector typically has fewer dimensions than the state vector, resulting in multiple solutions for the nonlinear equations’ system regarding control. Additionally, the ADAR technique is manual and not easily automated, similar to backstepping. Previous instances of this technique effectively solving the synthesis problem have only included low-order systems. Generally speaking, almost all analytical techniques for solving control synthesis problems have an identical fundamental drawback: they are type-dependent concerning the nonlinear differential equations describing the model of the control object.

Popular techniques such as neural network controllers [17, 18], fuzzy logic controllers [19, 20], and PID controllers [21, 22] are predominantly utilized to address the control synthesis problem. Each of these techniques specifies the control structure and optimally adjusts merely the parameters based on the functional perspective. Consequently, prior knowledge about the object is necessary for the developer to establish the controller structure accurately.

Machine learning control represents an innovative approach in the control realm [23]. The main distinctive feature of this novel methodology for establishing a control system lies in using symbolic regression techniques to explore the proper structure and parameters of the desired control function.

Symbolic regression techniques employ an automated process to seek out the mathematical expressions that represent the control functions. Symbolic regression employs a predetermined set of basic functions, such as *cos*, *minus*, multiplication, and others, to design an effective structure for the control function. At the same time, it optimizes the necessary parameters for the controller. These techniques differ based on the encoding type of the mathematical expressions and use a special kind of genetic algorithm. Consequently, a researcher is instantly provided with the effective structure of the controller and the suitable parameters.

This article primarily examines the utilization of symbolic regression in the context of control general synthesis problem-solving. Genetic programming (GP) was the technology predominantly that employed to showcase effective solutions for applied control challenges [24].

The most important contribution that this article makes is to:

- The invention of a novel symbolic regression technique, known as “synthesized genetic programming” (SGP), which involved the introduction of innovative coding concepts such as “priority” and “pivot,” which had not been employed in earlier symbolic regression techniques.
- Employing the novel technique of “synthesized genetic programming” (SGP) for the first time to solve the general synthesis problem in the control system.
- Likewise, the traditional technique of “parse-matrix evolution” (PME) was utilized for the first time to address the problem of control system general synthesis.
- Utilizing the small variations principle of the basic solution in symbolic regression techniques led to the development of further techniques such as ‘‘variational synthesized genetic programming’’ (VSGP) and ‘‘variational parse-matrix evolution’’ (VPME).
- The acquired results validate the efficacy of the employed symbolic regression techniques (CGP, PME and SGP) as unsupervised machine learning control techniques, which can serve as universal numerical techniques for addressing the control general synthesis problem.
- The solutions to the control general synthesis problem were used for the first time to compare symbolic regression techniques. The results indicated that the SGP method outperforms other methods in terms of speed when discovering solutions.

The rest of this work is structured as follows: In Section 2, a mathematical framework of the control general synthesis problem as supervised and unsupervised machine learning control is offered. Section 3 describes the principle of the small variation of the basic solution. The three symbolic regression techniques are explained in Section 4. The results and discussion are presented for a mobile robot in Section 5, and Section 6 focuses on the conclusions of this work.

2. The general synthesis of control as a machine learning-based control problem

The control general synthesis problem in control theory refers to the challenge of determining a control function, also known as a control law, which is contingent upon the state of the controlled object. The design challenge of a feedback controller can be classified as a control synthesis problem, as these controllers generate control signals by considering the object’s state. The seeking for a control function of this nature needs to become seen as a machine learning control problem.

Let us contemplate the formal articulation of the control synthesis problem.

The mathematical model of the control object is expressed as a system of ordinary differential equation

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

with $\boldsymbol{x}$ being a vector representing the state space, $\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}$, $\boldsymbol{u}$ denotes a vector representing the control, $\boldsymbol{u} \in \boldsymbol{U} \in \mathbb{R}^{\boldsymbol{m}}$, and $\boldsymbol{U}$ representing a compact set, $m \leq n$.

The set of initial conditions is provided as the domain

$$\mathbf{X}_0 \subseteq \mathbb{R}^n \tag{2}$$

The terminal condition is provided

$$\boldsymbol{x}(t_f) = \boldsymbol{x}^f \in \mathbb{R}^n \tag{3}$$

where t_f represents the unidentified time required to move from any arbitrary initial condition $\boldsymbol{x}^0 \in \mathbf{X}_0$ to the terminal one in Eq. (3).

The finishing time is limited

$$t_f \leq t^+ \tag{4}$$

where t^+ represents a predetermined positive value.

It is vital to identify a control function in the prescribed form

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x}) \in \boldsymbol{U} \tag{5}$$

where $\boldsymbol{g}(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R}^m$, that guarantees that the dynamic model's expressed object

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{x})) \tag{6}$$

will reach the terminal condition in Eq. (3) from any initial state while optimizing the specified quality criterion

$$J = \int \dots \int_{\mathbf{X}_0} \int_{t_0}^{t_f} f_0(\boldsymbol{x}(t, \boldsymbol{x}^0), \boldsymbol{u}(t)) dx_1^0 \dots dx_n^0 dt \to \min_{\boldsymbol{u} \in \boldsymbol{U}} \tag{7}$$

The control synthesis problem in machine learning is possible to address through two approaches: supervised machine learning control and unsupervised machine learning control.

2.1 Synthesis-based control as supervised machine learning control

This methodology involves the utilization of a training set for the purpose of learning. In order to generate a set for training regarding the control general synthesis problem, this makes it is imperative to address the optimal control problem for various points inside the offered initial set. This process will yield optimal controls sets

$$\boldsymbol{U}_0 = \{\boldsymbol{h}(t, \boldsymbol{x}^{0,1}), \dots, \boldsymbol{h}(t, \boldsymbol{x}^{0,L})\} \tag{8}$$

and optimal trajectories

$$\widetilde{\mathbf{X}} = \{\widetilde{\boldsymbol{x}}(t, \boldsymbol{x}^{0,1}), \dots, \widetilde{\boldsymbol{x}}(t, \boldsymbol{x}^{0,L})\} \tag{9}$$

Hence, in order to address the control general synthesis problem and determine the control function in the specified form as in Eq. (5), it suffices for approximating the training set in Eq. (9) based on the following criterion

$$J_1 = \sum_{i=1}^{L} \sum_{j=0}^{K_i} \left\| \boldsymbol{x}(t_j, \boldsymbol{x}^{0,i}) - \widetilde{\boldsymbol{x}}(t_j, \boldsymbol{x}^{0,i}) \right\| \to \min_{\boldsymbol{g}(\boldsymbol{x}) \epsilon \boldsymbol{U}} \tag{10}$$

where $t_j = 0$, $\boldsymbol{x}(t_j, \boldsymbol{x}^{0,i})$ is deemed to be a partial solution to Eq. (6) given the initial conditions $\boldsymbol{x}^{0,i}$. Similarly, $\widetilde{\boldsymbol{x}}(t_j, \boldsymbol{x}^{0,i})$ is regarded as a partial solution to Eq. (9), $i \in \{1, \dots, L\}$.

Symbolic regression is utilized as well, as a means to address the approximation problem. The utilization of control general synthesis based on the optimal trajectories' approximation enables the determination of a control function in Eq. (5) that achieves optimal control with a level of precision corresponding to the training set approximation.

2.2 Synthesis-based control as unsupervised machine learning control

The other technique directly seeks the control function by minimizing a quality criterion.

The system's initial conditions in Eq. (2) are represented by a set of points corresponding to the initial states

$$\boldsymbol{X}_0 = \{\boldsymbol{x}^{0,1}, \dots, \boldsymbol{x}^{0,L}\} \tag{11}$$

The terminal state is defined as a specific point within the state space as in Eq. (3).

The quality criterion is presented in the subsequent form

$$J_2 = \sum_{i=1}^{L} t_{f,i} + c_1 \sum_{i=1}^{L} \left\| \boldsymbol{x}(t_{f,i}, \boldsymbol{x}^{0,i}) - \boldsymbol{x}^f \right\| \to \min_{\boldsymbol{u} \in \boldsymbol{U}} \tag{12}$$

where $t_{f,i}$ is a period characterized by the attainment of the terminal goal in Eq. (3) starting from the initial one $\boldsymbol{x}^{0,i}$, $i = 1, \dots, L$, c_1 represents a weight coefficient.

$$t_{f,i} = \begin{cases} t, & \text{if } t < t^+ \text{ and } \left\| \boldsymbol{x}(t, \boldsymbol{x}^{0,i}) - \boldsymbol{x}^f \right\| \le \varepsilon \\ t^+, & \text{otherwise} \end{cases} \tag{13}$$

$i = 1, \dots, L$, t^+ and ε are provided positive values.

It is important to determine a control function in the subsequent form with the objective of minimizing functional in Eq. (12)

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x}^f - \boldsymbol{x}) \tag{14}$$

where this control function represents a stabilization system for the control object in Eq. (1).

The challenge for this methodology is navigating a vast and intricate search space within a non-numerical area of functions' codes where a specific singular metric does not exist. Perhaps this can elucidate the observation that symbolic regression techniques, despite their extensive range of capacities, have not yet established themselves as an effective tool for addressing the challenge of machine learning control. It is probable that the introduction of supplementary processes is necessary to make searching for the proper solutions easier and faster. An example of such a process could be the small variations principles of the basic solution.

3. Small variations principles of the basic solution

The complexity of seeking an optimal solution in the code space is attributed to the categorization of this work within the realm of non-numerical problems related to optimization. In the context of these particular search spaces, the utilization of evolutionary algorithms involving arithmetic operations is unfeasible. Hence, the genetic algorithm serves as a prominent search algorithm inside a space of codes, employing a methodology that does not rely on arithmetic operations during its iterative process.

Research on this problem has resulted in the development of the small variations principle of the basic solution. This technique enables the discovery of near-effective solutions within an acceptable timeframe. In order to implement the principle, small variations are first made within the code of the symbolic regression technique. A small-dimensional integer vector represents the small variation. The provided code includes the necessary information to execute the small variation. Hence, one possible solution, referred to as a basic solution, can be encoded using the symbolic regression technique. The researcher picks up the basic solution and represents the solution that matches up closest with the effective solution of the problem. A small variation refers to a negligible alteration in the code that results in the emergence of an additional possible solution. Based on this principle, other possible solutions can be represented by coded sets of small variations of the proposed basic solution. In order to generate different possible solutions, it is vital to application a small variations vector to the basic solution. The new name of variational genetic algorithm (VarGA) is employed because the genetic algorithm utilizes genetic operations on a variation vector to search for the effective solution. This VarGA focuses on exploring the ordered sets space of vectors with small variations in order to get the effective solution. Continuing the search process and reaching a specified number of generations, the basic solution is substituted with the most effective solution discovered up to that point. This strategy proves to be highly advantageous for the search for control systems, as it capitalizes on the expertise of numerous control specialists who possess the ability to intuitively devise effective control systems or draw upon their experiential knowledge. The control system in question can be regarded as a basic solution. In this scenario, the researcher has the ability to intervene in the machine's search process and provide guidance on the specific areas to explore in order to identify the most effective solution.

In order to elucidate the concept of variation, it is necessary to introduce a vector denoting the extent of variation

$$\mathcal{W} = [\mathcal{w}_1 \dots \mathcal{w}_d]^T \tag{15}$$

where d represents the dimension of the variation vector, specified by the information necessary to execute a small variation. This dimension is contingent upon the symbolic regression technique employed. As an illustration, let $\mathcal{w}_1$ represents an index denoting a small variation. Similarly, $\mathcal{w}_2$ and $\mathcal{w}_{d-1}$ can be understood as indices indicating the element position in the code or the indexes of the element in a vector or matrix that define the variable element. Finally, $\mathcal{w}_d$ represents the updated value of the defined element.

Possible solutions can be obtained from the sets of variations' vectors

$$S = \{\boldsymbol{W}_1, \dots, \boldsymbol{W}_H\} \tag{16}$$

where

$$\boldsymbol{W_i} = \{\mathcal{W}^{i,1}, \dots, \mathcal{W}^{i,D}\} \tag{17}$$

where D is a variations depth or length.

The genetic algorithm facilitates evolutionary processes on sets of small variation vectors. During the crossover operation, two sets of variations vectors are chosen either randomly or based on approaches often employed in the field of Genetic Algorithm (GA)

$$\boldsymbol{W_i} = \{\mathcal{W}^{i,1}, \dots, \mathcal{W}^{i,D}\}$$
$$\boldsymbol{W_j} = \{\mathcal{W}^{j,1}, \dots, \mathcal{W}^{j,D}\} \tag{18}$$

Specify a crossover point $p \in \{1, \dots, D\}$. Swap the variation vectors beyond the crossing point inside the chosen sets. Consequently, we get two additional sets of vectors representing variations

$$\boldsymbol{W_{H+1}} = \{\mathcal{W}^{i,1}, \dots, \mathcal{W}^{i,p}, \mathcal{W}^{j,p+1}, \dots, \mathcal{W}^{j,D}\} \tag{19}$$

$$\boldsymbol{W_{H+2}} = \{\mathcal{W}^{j,1}, \dots, \mathcal{W}^{j,p}, \mathcal{W}^{i,p+1}, \dots, \mathcal{W}^{i,D}\} \tag{20}$$

Two additional sets of vectors representing variations refer to two newly introduced codes within the vicinity of the basic solution

$$G_{H+1} = \boldsymbol{W_{H+1}} \circ G_0 = \mathcal{W}^{H+1,D} \circ \dots \circ \mathcal{W}^{H+1,1} \circ G_0 \tag{21}$$

$$G_{H+2} = \boldsymbol{W}_{H+2} \circ G_0 = \mathcal{W}^{H+2,D} \circ \ldots \circ \mathcal{W}^{H+2,1} \circ G_0 \quad (22)$$

where G_0 represents the basic solution code.

In order to execute the mutation process on the acquired sets in Eqs. (19) and (20), it is necessary to employ a random selection to choose one vector from the sets. This selected vector will then be substituted with a randomly produced vector that exhibits variations.

This strategy may appear to involve a duplication of coding of the possible solutions, but it offers two advantages. Initially, the utilization of the basic solution provides a framework for exploration inside an intricate area of functions. It effectively expedites the search process by restricting the scope of the search. Furthermore, by applying the techniques of the genetic algorithm to the vectors of variations rather than directly to the codes for possible solutions, we may consistently obtain accurate codes of possible solutions without the necessity of incorporating further checks.

The utilization of this principle was initially employed within the framework of the network operator method. The term "variational" is incorporated into the nomenclature of the symbolic regression technique, which leverages the concept of this principle. The application of the principle can be utilized in conjunction with any existing symbolic regression method as a means to address the difficulties associated with resolving the control general synthesis problem.

4. Numerical techniques of symbolic regression for the synthesis of control problem

Symbolic regression is a commonly employed numerical approach for addressing the control synthesis problem. Symbolic regression techniques have shown significant advancements in the last ten years. Moreover, the broader scientific community has lastly acknowledged the significance of interpretable machine learning. Currently, a multitude of symbolic regression techniques have been known. Several designations can be identified, including genetic programming (GP) [24], Cartesian GP [25], and network operator [26]. Symbolic regression approaches encode the mathematical expression being searched using a specific code form. These methods then employ a specialized genetic algorithm that looks for the effective solution within the code space. Various approaches exhibit differences in their code form.

For instance, let us think about the subsequent mathematical expression

$$y = \exp(q_3 x_2^2 + q_1 x_3^2)\sin(q_2 x_1) + \cos(-q_3 x_3 + x_1) \quad (23)$$

where q_1, q_2 and q_3 exemplify the parameters, x_1, x_2 and x_3 exemplify variables, and both exemplify arguments of the mathematical expression.

In order to represent a mathematical expression in a coded form, it qualifies as satisfactory to utilize the subsequent sets of arguments and primary functions:

- The arguments' set

$$\mathrm{F}_0 = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = x_3, f_{0,4} = q_1, f_{0,5} = q_2, f_{0,6} = q_3\} \quad (24)$$

- The functions set that is characterized by one argument

$$\mathrm{F}_1 = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = z^2, f_{1,4}(z) = \sin(z), f_{1,5}(z) = cos(z), f_{1,6}(z) = exp(z)\} \quad (25)$$

- The functions set that is characterized by two arguments

$$\mathrm{F}_2 = \left\{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\right\} \quad (26)$$

where the first component of the indexes stands for the number of arguments while the second one exemplifies the function number, knowing that the argument of a mathematical expression is signified when the first component equals zero.

4.1 Variational cartesian genetic programming

Cartesian genetic programming (CGP) employs a non-graphical representation for expressing codes of expressions. This technique involves the integration of the two sets of fundamental functions into a unified set.

$$F = F_1 \cup F_2 = \{f_1(z) = z, f_2(z) = -z, f_3(z) = z^2, f_4(z) = \sin(z), f_5(z) = \cos(z), f_6(z) = exp(z), f_7(z_1, z_2) = z_1 + z_2, f_8(z_1, z_2) = z_1 z_2\} \quad (27)$$

CGP codes for mathematical expressions typically take the form of a three- or four-row integer matrix. The initial row of the matrix denotes the indexes of functions obtained from the set of fundamental functions in Eq. (27). The fundamentals functions' set in Eq. (27) consists of functions that have a maximum of two arguments. Consequently, encoding the matrix requires only three rows. It should be noted that the number of rows in the matrix varies depending on the number of arguments used for the available functions, where in the case of using the one-argument function, the third element in the column does not have any practical application. The remaining rows stand for the argument indices in Eq. (24). When a column's calculation is complete, its result should be appended to the arguments in Eq. (24). So, after each computation, there will be more arguments. As a result, the total number of arguments will grow with each new calculation.

Towards encoding the expression x_2^2 as the first column of the CGP matrix for the example in Eq. (23), determining the square's function in the set of fundamental functions in Eq. (27) is the first step where its number is 3, $f_3(z) = z^2$. Next, locate the variable x_2 in the arguments' set in Eq. (24), and its number is 2. The third item in this column serves no use and is postulated as a number of 1. Therefore, the expression x_2^2 that represents the first column of the matrix is encoded as $[3\ 2\ 1]^T$. Once the value of this column has been calculated, it will have been appended to the arguments in Eq. (24) at position 7, $(|F_0| + 1 = 6 + 1 = 7)$. The subsequent expression, denoted as $q_3x_2^2$, will be assigned to the second column. The index of the multiplication function in the set in Eq. (27) is 8. The parameter q_3 and the expression x_2^2 (the first column) of the set in Eq. (24), correspond to indices 6 and 7, respectively. Consequently, the code assigned to the second column is $[8\ 6\ 7]^T$. Additionally, it has been included as the eighth argument, together with the existing arguments in Eq. (24).

For this reason, the CGP final code of the example's mathematical equation in Eq. (23) looks like this:

$$R_{CGP} = \begin{bmatrix} 3 & 8 & 3 & 8 & 7 & 6 & 8 & 4 & 2 & 8 & 7 & 5 & 8 & 7 \\ 2 & 6 & 3 & 4 & 8 & 11 & 5 & 13 & 6 & 15 & 16 & 17 & 12 & 19 \\ 1 & 7 & 2 & 9 & 10 & 3 & 1 & 4 & 5 & 3 & 1 & 6 & 14 & 18 \end{bmatrix} \tag{28}$$

A small variation to the CGP code involves altering an element within the matrix. In order to implement a small variation, a three-element integer vector will do the trick

$$\mathcal{W} = [\mathcal{w}_1\ \mathcal{w}_2\ \mathcal{w}_3]^T, \tag{29}$$

in the context of a matrix, $\mathcal{w}_1$ represents the column number, $\mathcal{w}_2$ represents the row number inside that column, and $\mathcal{w}_3$ represents the updated value of the element. When $\mathcal{w}_2$ equals 1, the subsequent value ($\mathcal{w}_3$) is required to be selected from the set of fundamental functions in Eq. (27). In all other instances, the following value is required to be less than the aggregate of the set of arguments in Eq. (24) plus the number of columns (i.e., the sum of the number of arguments and the number of columns minus 1).

In order to implement subsequent variations to the matrix in Eq. (28),

$$\begin{aligned} \mathcal{W}^1 &= [2\ 2\ 1]^T, \\ \mathcal{W}^2 &= [9\ 1\ 4]^T, \\ \mathcal{W}^3 &= [7\ 3\ 2]^T, \\ \mathcal{W}^4 &= [13\ 1\ 7]^T \end{aligned} \tag{30}$$

The CGP matrix will be updated to

$$\mathcal{W}^1 \circ \mathcal{W}^2 \circ \mathcal{W}^3 \circ \mathcal{W}^4 \circ R_{CGP} = \begin{bmatrix} 3 & 8 & 3 & 8 & 7 & 6 & 8 & 4 & \mathbf{4} & 8 & 7 & 5 & \mathbf{7} & 7 \\ 2 & \mathbf{1} & 3 & 4 & 8 & 11 & 5 & 13 & 6 & 15 & 16 & 17 & 12 & 19 \\ 1 & 7 & 2 & 9 & 10 & 3 & \mathbf{2} & 4 & 5 & 3 & 1 & 6 & 14 & 18 \end{bmatrix} \tag{31}$$

and the corresponding mathematical expression for this updated matrix is

$$y = \exp(x_1x_2^2 + q_1x_3^2) + \sin(q_2x_2) + \cos(\sin(q_3) * x_3 + x_1) \tag{32}$$

4.2 Variational synthesized genetic programming

This technique is invented by our team. This technique is brand new, being the first instance in which it has been applied to address the control synthesis problem. Synthesized genetic programming (SGP) eschews the utilization of graphical representations for expressing codes of mathematical expressions. In general, the mathematical expression's SGP code is a six-row integer matrix. The first row of the matrix denotes the indexes of functions belonging to the functions set that is characterized by two arguments in Eq. (26). The indexes of functions from the functions set that is characterized by one argument in Eq. (25) are represented by the second and fourth rows. The third and fifth rows represent the indexes of arguments from the arguments set in Eq. (24). The sixth row represents the priority, which will thereafter be elucidated to elucidate its role. Within each column of the matrix, the second element (the one-argument function) and the third element (the argument) represent the first argument for the first element of the column (the two-argument function). Additionally, the fourth and fifth elements represent the second argument for the first element of the column. The term of the pivot for each column means either the first argument (the second and third elements) or the second argument (the fourth and fifth elements) of this column. The pivot can be determined by assigning the priority (the sixth element in the column) of 1 or 2 to opt for the desired pivot of the column; nevertheless, in most contexts, its number is 1. It is important to acknowledge that the number of rows in the SGP matrix is contingent upon the number of arguments employed in the available functions. Specifically, when utilizing a three-argument function such as the if function, the number of rows is going to be 8. This is due to each argument being allocated two elements in the column, combined with the first element representing the three-argument function and the final element denoting the priority. As with the CGP approach, after completing the calculation for each column in the matrix of the SGP, the result of this column should be appended to the set of arguments in Eq. (24), progressively increasing the total number of arguments with each calculation.

In order to implement the example in Eq. (23) by this technique, let us get started by coding the expression $q_3x_2^2$ as the first column of the SGP matrix. For the first element in this column, determine the index of multiplication function in the functions set that is characterized by two arguments in Eq. (26); it is the number of 2, $f_{2,2}(z_1, z_2) = z_1z_2$. For the second element, the function of the parameter q_3 is the identity function, $f_{1,1}(z) = z$, from the functions set that is characterized by one argument in Eq. (25); the index of this function is 1. For the third element, the location of the parameter q_3 in the arguments set in Eq. (24) is 6. For the fourth element, the variable x_2 function is the square $f_{1,3}(z) = z^2$, and its index is 3 in the set in Eq. (25). For the fifth element, the location of the variable x_2 in the arguments set in Eq. (24) is 2. The sixth element is the priority, and its number is 1. As a result, the code of the expression $q_3x_2^2$ that represents the first column in the matrix is $[2\ \ 1\ \ 6\ \ 3\ \ 2\ \ 1]^T$. After calculating this column, it will have been appended to the arguments set in Eq. (24) as the seventh element, denoted as $(|F_0| + 1 = 6 + 1 = 7)$. The following expression $q_1x_3^2$ will be the second column, which is the same idea as the first column, and its code is $[2\ \ 1\ \ 4\ \ 3\ \ 3\ \ 1]^T$. Consequently, it will be appended as the eighth element to the arguments set in Eq. (24). The third column will demonstrate the amalgamation of the preceding two columns, specifically represented as $q_3x_2^2 + q_1x_3^2$. For the first element of the third column, the index of the addition function in the set in Eq. (26) is 1. The identity function will be the primary function in this case for the first column ($q_3x_2^2$) and the second column ($q_1x_3^2$); therefore, the second and fourth elements will get number 1 as the identity function in the set in Eq. (25). The indexes of the first column ($q_3x_2^2$) and the second column ($q_1x_3^2$) in the arguments set in Eq. (24) are 7 and 8, respectively. So, the code of the third column is $[1\ \ 1\ \ 7\ \ 1\ \ 8\ \ 1]^T$ and will have been appended to the arguments set in Eq. (24) as the ninth element.

The final code of the SGP matrix, for example in Eq. (23), can be expressed as:

$$R_{SGP} = \begin{bmatrix} 2 & 2 & 1 & 2 & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 6 & 5 \\ 6 & 4 & 7 & 5 & 6 & 11 & 9 & 12 \\ 3 & 3 & 1 & 1 & 1 & 1 & 4 & 1 \\ 2 & 3 & 8 & 1 & 3 & 1 & 10 & 13 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{33}$$

In order to keep track of a small variation, this technique resembles the CGP technique, wherein an identical integer vector is employed. Specifically, w_1 represents the column index inside the matrix, w_2 corresponds to the row index within the column w_1, and w_3 signifies the updated value of the element. If w_2 equals 1, the subsequent number (w_3) must either be zero or modified based on the functions set that is characterized by two arguments in Eq. (26). If w_2 is equal to either 2 or 4, then w_3 will be modified to either zero or selected from the functions set that is characterized by one argument in Eq. (25). If w_2 is equal to either 3 or 5, then w_3 can either be set to zero or can only be determined by the combination of the number of arguments in Eq. (24) and the number of columns minus one. Certain conditions dictate the implementation of small variations to the SGP matrix based on the pivot and priority. These requirements can be elucidated by implementing the following variations to the matrix in Eq. (33):

$$\begin{aligned}\mathcal{W}^1 &= [3\ \ 6\ \ 2]^T,\\ \mathcal{W}^2 &= [5\ \ 2\ \ 0]^T,\\ \mathcal{W}^3 &= [4\ \ 1\ \ 1]^T,\\ \mathcal{W}^4 &= [6\ \ 5\ \ 0]^T,\\ \mathcal{W}^5 &= [3\ \ 1\ \ 0]^T,\\ \mathcal{W}^6 &= [8\ \ 2\ \ 3]^T,\\ \mathcal{W}^7 &= [6\ \ 6\ \ 2]^T\end{aligned} \tag{34}$$

The updated matrix of the SGP will look like:

$$\mathcal{W}^1 \circ \mathcal{W}^2 \circ \mathcal{W}^3 \circ \mathcal{W}^4 \circ \mathcal{W}^5 \circ \mathcal{W}^6 \circ \mathcal{W}^7 \circ R_{SGP} = \begin{bmatrix} 2 & 2 & \mathbf{0} & \mathbf{1} & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 6 & \mathbf{3} \\ 6 & 4 & 7 & 5 & 6 & 11 & 9 & 12 \\ 3 & 3 & 1 & 1 & 1 & 1 & 4 & 1 \\ 2 & 3 & 8 & 1 & 3 & \mathbf{0} & 10 & 13 \\ 1 & 1 & \mathbf{2} & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{35}$$

The first variation $\mathcal{W}^1$ has changed the third column's priority (the 6th element) from 1 to 2. Consequently, it has changed the pivot from the first argument (the 2nd and 3rd elements) to the second argument (the 4th and 5th elements) of this column. It is worth noting that the second variation $\mathcal{W}^2$ did not affect the 5th column since it is not possible to alter any element of the pivot to zero in each column of the matrix, where the pivot of the 5th column is the first argument (the 2nd and 3rd elements) because of the number of priority is 1 in this column. In contrast, the fourth variation $\mathcal{W}^4$ can be accomplished since the 5th element of the 6th column is not an element of the pivot in this column, where the expression of this column was $-q_3x_3 + x_1$, then the variable x_1 has been neglected. In this case, the unit element of the function is used as the second argument where the unit element of the addition function is 0 and for the multiplication function is 1, so the expression has become $(-q_3x_3 + 0)$. Interestingly, the fifth variation $\mathcal{W}^5$ has fulfilled a primary change, where the expression of the third column was $q_3x_2^2 + q_1x_3^2$. This variation cancelled the addition function (changed the first element to 0). The number of priority turned out to be 2 as a result of the first variation $\mathcal{W}^1$, so the new expression of this column has got the expression of the pivot (the 2nd argument), whose code is $[1\ \ 8]^T$ that represents $q_1x_3^2$ (the identity function and the expression of the second column). The seventh variation $\mathcal{W}^7$ was not accomplished since the fourth variation $\mathcal{W}^4$ has changed the 5th element in the 6th column to 0, and changing the priority means changing the pivot of the column, and the pivot element is not allowed to be zero. Eventually, the third and sixth variations $\mathcal{W}^3$ and $\mathcal{W}^6$ can be performed directly.

The new matrix can be expressed mathematically as

$$y = \exp(q_1x_3^2)\sin(q_2 + x_1) + (-q_3x_3)^2 \tag{36}$$

As mentioned above, the analysis highlights the crucial importance of the priority, which may be summed up as follows: The main task of the priority is to pick the pivot for each column. Moreover, it effectively avoids zero values in the pivot elements due to small variations. Additionally, it has the likelihood of decreasing the length of mathematical expressions.

4.3 Variational parse-matrix evolution

Parse-Matrix Evolution (PME) utilizes parse-matrix as its chromosomal representation. It can be readily executed using any kind of programming language and is freely controllable [27].

The mathematical expression's PME code is usually an integer matrix, where the number of columns depends on the number of the used arguments in the functions set in Eq. (27), the number of rows is equal to the number of arguments plus two; for instance, we have functions that take two arguments in Eq. (27), so, the number of columns will be 4. The first element in each row is the function number from the set in Eq. (27); the second and third elements are the numbers of arguments from the set in Eq. (24); if there is a function that take one argument in the first element in any row, then the third element will not be utilized. The fourth element represents the number or sequence of the rows.

The code of this method for any mathematical expression is a parse-matrix

$$C = [c_{i,j}], i = 1, \dots, L,\ \ j = 1, \dots, p+2 \qquad (37)$$

where (p) is the maximal number of arguments and each row in the code matrix in Eq. (15) has the following elements:

$$\text{-}\ c_{i,1} = n(F) - \left\lfloor \frac{|F|}{2} \right\rfloor \qquad (38)$$

where $n(F)$ is the ordinal number of an element in the set F in Eq. (27), and $|F|$ is the total number of elements in the set F.

$$\text{-}\ c_{i,p+2} = n(H) - \left\lfloor \frac{|H|}{2} \right\rfloor \qquad (39)$$

where $n(H)$ is the ordinal number of an element in the set H in Eq. (41), and $|H|$ is the total number of elements in the set H.

$$\text{-}\ c_{i,J} = n(B) - \left\lfloor \frac{|B|}{2} \right\rfloor,\quad j = 2, \dots, p+1 \qquad (40)$$

where $n(B)$ is the ordinal number of an element in the set B in Eq. (42), and $|B|$ is the total number of elements in the set B.

Now, to encode the example in Eq. (23) by the PME technique, the next sets should be prepared:

$$H = \{h_1, \dots, h_{14}\} \qquad (41)$$

$$B = \{b_1 = x_1,\ b_2 = x_2,\ b_3 = x_3,\ b_4 = q_1,\ b_5 = q_2,\ b_6 = q_3,\ b_7 = h_1, \dots,\ b_{20} = h_{14}\} \qquad (42)$$

where H is an ordered set of elements for storing the results of calculations and B represents the sets of arguments in Eqs. (24) plus (41).

The first row of the matrix will be performed the expression x_2^2 and as follows:

$$x_2^2 :\ f_3(b_2), c_{1,1} = n(F) - \left\lfloor \frac{|F|}{2} \right\rfloor = 3 - \left\lfloor \frac{|8|}{2} \right\rfloor = 3 - 4 = -1 \qquad (43)$$

where the ordinal number of the function z^2 in the set F in Eq. (27) is 3 = $n(F)$, and the total number of functions in the set F is 8 = $|F|$.

$$c_{1,2} = n(B) - \left\lfloor \frac{|B|}{2} \right\rfloor = 2 - \left\lfloor \frac{|20|}{2} \right\rfloor = 2 - 10 = -8 \qquad (44)$$

where the ordinal number of the element $b_2 = x_2$ in the set B in Eq. (42) is 2 = $n(B)$, and the total number of elements in the set B is 20 = $|B|$.

$c_{1,3}$ is not utilized and the sign (*) has been put in its place.

$$c_{1,4} = n(H) - \left\lfloor \frac{|H|}{2} \right\rfloor = 1 - \left\lfloor \frac{|14|}{2} \right\rfloor = 1 - 7 = -6 \qquad (45)$$

where the ordinal number of the element h_1 in the set H in Eq. (41) is 1 = $n(H)$, and the total number of elements in the set H is 14 = $|H|$.

The result of calculation of the first row will get $b_7 = h_1$ in the set B in Eq. (42), and the first row in the parse-matrix will be $c_{1,j} = [-1\ \ -8\ \ *\ \ -6]^T$.

The second row of the matrix is the expression $q_3x_2^2$ and will be performed as such:

$$q_3x_2^2 :\ f_8(b_6, b_7), c_{2,1} = n(F) - \left\lfloor \frac{|F|}{2} \right\rfloor = 8 - \left\lfloor \frac{|8|}{2} \right\rfloor = 8 - 4 = 4 \qquad (46)$$

where the ordinal number of the multiplication function in the set F in Eq. (27) is 8 = $n(F)$, and the total number of functions in the set F is 8 = $|F|$.

$$c_{2,2} = n(B) - \left\lfloor \frac{|B|}{2} \right\rfloor = 6 - \left\lfloor \frac{|20|}{2} \right\rfloor = 6 - 10 = -4 \qquad (47)$$

$$c_{2,3} = n(B) - \left\lfloor \frac{|B|}{2} \right\rfloor = 7 - \left\lfloor \frac{|20|}{2} \right\rfloor = 7 - 10 = -3 \qquad (48)$$

where the ordinal number of the element $b_6 = q_3$ in the set B in Eq. (42) is 6 = $n(B)$, and the total number of elements in the set B is 20 = $|B|$, and the ordinal number of element of $b_7 = h_1 = x_2^2$ in the same set is 7 = $n(B)$, and also the total number of elements in the set B is 20 = $|B|$.

$$c_{2,4} = n(H) - \left\lfloor \frac{|H|}{2} \right\rfloor = 2 - \left\lfloor \frac{|14|}{2} \right\rfloor = 2 - 7 = -5 \qquad (49)$$

where the ordinal number of the element h_2 in the set H in Eq. (41) is 2 = $n(H)$, and the total number of elements in the set H is 14 = $|H|$.

The result of calculation of the 2nd row will get $b_8 = h_2$ in the set B in Eq. (42), and the 2nd row in the parse-matrix will be $c_{2,j} = [\,3 \quad -1 \quad 0 \quad -1\,]$.

In the same way, the values of the remaining rows in the parse-matrix are calculated and the final code of the mathematical expression in Eq. (23) by this technique is

$$R_{PME} = \begin{bmatrix} -1 & -8 & * & -6 \\ 4 & -4 & -3 & -5 \\ -1 & -7 & * & -4 \\ 4 & -6 & -1 & -3 \\ 3 & -2 & 0 & -2 \\ 2 & 1 & * & -1 \\ 4 & -5 & -9 & 0 \\ 0 & 3 & * & 1 \\ -2 & -4 & * & 2 \\ 4 & 5 & -7 & 3 \\ 3 & 6 & -9 & 4 \\ 1 & 7 & * & 5 \\ 4 & 2 & 4 & 6 \\ 3 & 9 & 8 & 7 \end{bmatrix} \tag{50}$$

To introduce a small variation, a three-ingredient integer vector suffices

$$\mathcal{W} = [\mathcal{w}_1 \ \mathcal{w}_2 \ \mathcal{w}_3]^T \tag{51}$$

where $\mathcal{w}_1$ represents the row index in the PME code, $\mathcal{w}_2$ represents the column index in that row $\mathcal{w}_1$, and $\mathcal{w}_3$ represents the new ingredient value. If $\mathcal{w}_2$ equals 1, the subsequent number ($\mathcal{w}_3$) must be modified based on the functions set in Eq. (27). If $\mathcal{w}_2$ is equal to either 2 or 3, then $\mathcal{w}_3$ will be selected from the set in Eq. (24) minus one, and there is no variation on the fourth column.

To facilitate the application of the small variations to this technique, firstly we should perform the Eq. (38) for each function in the set in Eq. (27), for example, if we have 30 functions, so the first and last functions will be

$$f_1 = n(F) - \left\lfloor \frac{|F|}{2} \right\rfloor = 1 - \left\lfloor \frac{|30|}{2} \right\rfloor = 1 - 15 = -14$$

$$f_{30} = n(F) - \left\lfloor \frac{|F|}{2} \right\rfloor = 30 - \left\lfloor \frac{|30|}{2} \right\rfloor = 30 - 15 = 15 \tag{52}$$

so, we renumber the functions in the set in Eq. (27) from the first function, which is -14, $f_{-14} = z$, to the last function, which is 15 and so on.

The same thing for Eq. (40), and apply it on the set in Eq. (42), for example, if we have 6 arguments and $H = 10$, then the first and last arguments will be

$$b_1 = n(B) - \left\lfloor \frac{|B|}{2} \right\rfloor = 1 - \left\lfloor \frac{|16|}{2} \right\rfloor = 1 - 8 = -7,$$

$$b_{16} = n(B) - \left\lfloor \frac{|B|}{2} \right\rfloor = 16 - \left\lfloor \frac{|16|}{2} \right\rfloor = 16 - 8 = 8 \tag{53}$$

also, we renumber the arguments in the set in Eq. (42) from the first argument, which is -7, $b_{-7} = x_1$, to the last argument, which is 8 and the calculation of the first row will be -1, and the calculation of the second row will be 0, and so on.

In order to implement the subsequent variations to the matrix in Eq. (50),

$$\mathcal{W}^1 = [6 \quad 1 \ -\mathbf{1}]^T, \qquad \mathcal{W}^2 = [11 \quad 3 \quad -\mathbf{3}]^T \tag{54}$$

The PME matrix will be updated to

$$\mathcal{W}^1 \circ \mathcal{W}^2 \circ R_{PME} = \begin{bmatrix} -1 & -8 & * & -6 \\ 4 & -4 & -3 & -5 \\ -1 & -7 & * & -4 \\ 4 & -6 & -1 & -3 \\ 3 & -2 & 0 & -2 \\ -\mathbf{1} & 1 & * & -1 \\ 4 & -5 & -9 & 0 \\ 0 & 3 & * & 1 \\ -2 & -4 & * & 2 \\ 4 & 5 & -7 & 3 \\ 3 & 6 & -\mathbf{3} & 4 \\ 1 & 7 & * & 5 \\ 4 & 2 & 4 & 6 \\ 3 & 9 & 8 & 7 \end{bmatrix} \tag{55}$$

where $-\mathbf{1}$ represents square function and $-\mathbf{3}$ represents x_2^2 (the calculation of the first row).

The corresponding mathematical expression for this updated PME matrix is

$$y = (q_3 x_2^2 + q_1 x_3^2)^2 \sin(q_2 x_1) + \cos(-q_3 x_3 + x_2^2) \tag{56}$$

5. Results and discussions

Suppose the general synthesis problem's solution involves a mobile robot's control system.

The control object's mathematical model is represented by the following form:

$$\begin{aligned} \dot{x}_1 &= 0.5(u_1 + u_2)\cos(x_3), \\ \dot{x}_2 &= 0.5(u_1 + u_2)\sin(x_3), \\ \dot{x}_3 &= 0.5(u_1 - u_2), \end{aligned} \tag{57}$$

where $\mathbf{x} = [x_1\ \ x_2\ \ x_3]^T$ stands for the state vector and $\mathbf{u} = [u_1\ \ u_2]^T$ stands for the control vector.

The extent of control is limited

$$u^- = -10 \le u_i \le 10 = u^+,\ i = 1{,}2. \quad (58)$$

The initial conditions consist of 8 different states

$$X_0 = \left\{\boldsymbol{x}^{0,1} = \left[3\ \ 3.5\ \tfrac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3\ \ 3.5 - \tfrac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,3} = \left[3\ \ -3.5\ \tfrac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,4} = \left[3\ \ -3.5 - \tfrac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,5} = \left[-3\ \ 3.5\ \tfrac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,6} = \left[-3\ \ 3.5\ -\tfrac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,7} = \left[-3\ \ -3.5\ \tfrac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,8} = \left[-3\ \ -3.5\ -\tfrac{5\pi}{16}\right]^T\right\} \quad (59)$$

The terminal state is

$$x^f = [0\ \ 0\ \ 0]^T \quad (60)$$

It is essential to pinpoint a function of control that is dependent on the state coordinates

$$u_i = g_i(x_1, x_2, x_3), u^- \le g_i(x_1, x_2, x_3) \le u^+,\ i = 1{,}2, \quad (61)$$

for minimizing the criterion

$$J_{syn} = \sum_{l=1}^{8}\left(t_{f,l} + \sqrt{\sum_{i=1}^{3} x_i^2\left(t_{f,l}, \mathbf{x}^{0,l}\right)}\right) \quad (62)$$

where

$$t_{f,l} = \begin{cases} t, & if\ \sqrt{\sum_{i=1}^{3} x_i^2(t, \boldsymbol{x}^{0,l})} < \varepsilon \\ t^+, & otherwise \end{cases} \quad (63)$$

$\varepsilon = 0.01, t^+ = 2.5$ s, $x_i\left(t, \mathbf{x}^{0,l}\right)$ represents a partial solution of the model in Eq. (57) with control in Eq. (61) for initial state $\mathbf{x}^{0,l}$, $l \in \{1, \dots, 8\}$.

The problem was addressed using the three techniques: variational Cartesian genetic programming (VCGP), variational synthesized genetic programming (VSGP) and variational parse-matrix evolution (VPME).

The three techniques yielded the subsequent control law

$$g_i(x_1, x_2, x_3) = \begin{cases} u^-, \text{if}\ \tilde{u}_i < u^- \\ u^+, \text{if}\ \tilde{u}_i > u^+ \\ \tilde{u}_i, \text{otherwise} \end{cases},\ i = 1{,}2, \quad (64)$$

where for the variational Cartesian genetic programming, the found control functions as in Eq. (64) is

$$\tilde{u}_1 = sgn\left(\left(q_1(x_1^f - x_1)\right)^3\right)\sqrt{\left|(q_1(x_1^f - x_1))^3\right|} \quad (65)$$

$$\tilde{u}_2 = \left(\frac{1}{sgn\left(\left(x_3^f - x_3\right)\left(x_2^f - x_2\right)\right)\sqrt{\left|\left(x_3^f - x_3\right)\left(x_2^f - x_2\right)\right|}}\right) + q_2\left(x_1^f - x_1\right) + \left(x_3^f - x_3\right)\left(x_2^f - x_2\right) + (x_3^f - x_3)^3 \quad (66)$$

$q_1 = 4.90261$, $q_2 = 8.79806$. The quality criterion in Eq. (62) for the variational CGP solution is $J_{syn} = 2.07197$.

The found control functions by the variational synthesized genetic programming as Eq. (64) is

$$\tilde{u}_1 = \left(x_2^f - x_2\right)^{\frac{1}{3}}\left(x_1^f - x_1\right)q_1\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right) + \sin\left(\left(x_2^f - x_2\right)^{\frac{1}{3}}\left(x_1^f - x_1\right)q_1\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right)\right), \quad (67)$$

$$\tilde{u}_2 = \rho\left(q_3^2\left(x_1^f - x_1\right)\right) + 0.5 * q_3^2\left(x_1^f - x_1\right), \quad (68)$$

$$\rho(\mu) = \begin{cases} 0, & \text{if}\ \ |\mu| < \delta\ , \\ sgn(\mu), & \text{otherwise} \end{cases}, \quad (69)$$

$q_1 = 2.07946$, $q_2 = 2.63935$, $q_3 = 2.96333$, $\delta = 10^{-8}$. The quality criterion in Eq. (62) for the variational SGP solution is $J_{syn} = 2.26092$.

The found control functions by the variational parse-matrix evolution as Eq. (64) is

$$\tilde{u}_1 = (q_1\left(x_1^f - x_1\right) + \left(x_1^f - x_1\right)\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right))/2, \quad (70)$$

$$\tilde{u}_2 = q_3\left(x_1^f - x_1\right)^3 - q_4\left(x_1^f - x_1\right), \quad (71)$$

$q_1 = 8.98565$, $q_2 = 9.73584$, $q_3 = 6.96312$, $q_4 = 5.39341$. The value of criterion in Eq. (62) for the obtained solution is J_{syn} = 2.28499.

The mobile robot trajectories from 8 different initial states in Eq. (59) to the terminal state in Eq. (60) have been illustrated in Figs. 1-3. When one

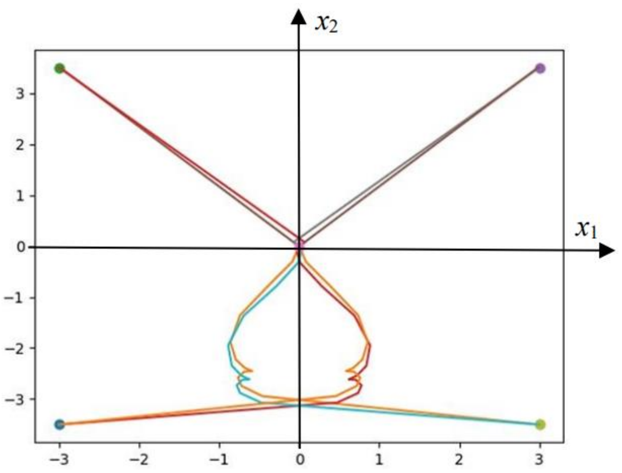

Figure. 1 The formed robot trajectories from multiple initial states to one terminal state using the obtained control function from VCGP

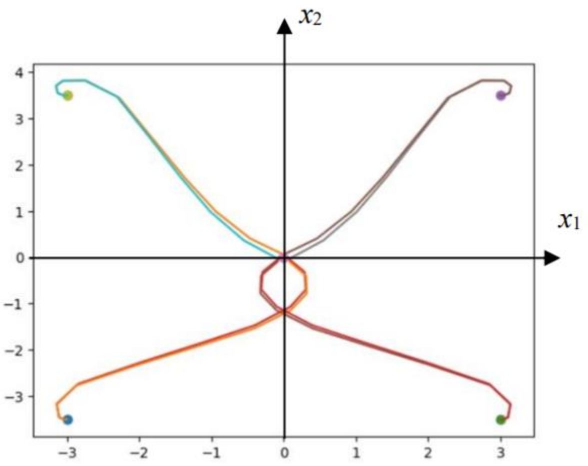

Figure. 2 The formed robot trajectories from multiple initial states to one terminal state using the obtained control function from VSGP

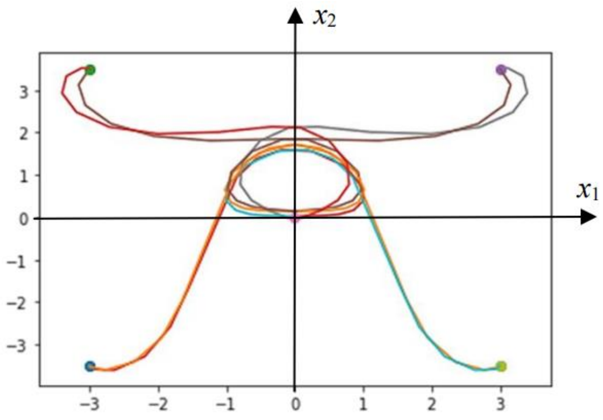

Figure. 3 The formed robot trajectories from multiple initial states to one terminal state using the obtained control function from VPME

mobile robot changes its initial state (multiple initial states) and moves to one terminal state from each initial state to form multiple trajectories and obtain the control function as a feedback control function, this represents the control system general synthesis problem.

It is essential to acknowledge that CGP, SGP and PME, when not implementing the principle of small variations within the basic solution, were unable to address the problem effectively and failed to identify a satisfactory solution using the given search parameters.

The objective of the studies was to demonstrate the efficacy of computationally symbolic regression techniques in obtaining a control function. When incorporated into the right side of a system of differential equations governing the control object, this control function results in the stability of the said object. Consequently, it has been shown that several symbolic regression techniques can effectively address this machine learning problem without the need to create a training set. This is achieved by focusing solely on minimizing the quality functional criterion, which falls under the domain of unsupervised machine learning.

To evaluate the performance of our technique, the synthesized genetic programming (SGP), in comparison to the Cartesian genetic programming (CGP) and parse-matrix evolution (PME), we conducted a series of 10 runs. These runs aimed to solve the control general synthesis problem. This problem is being employed for the first time to compare three techniques by doing a set number of runs. The assessment is conducted by considering the sequence of generation that leads to the attainment of the first best criterion in Eq. (62). The criterion is deemed best when the resulting value is in close range to, equal to, or smaller than the t^+ value, where $t^+ = 2.5\ s$, and the robot's trajectories are smooth. It is important to highlight that the same eight initial conditions in Eq. (59) and terminal condition in Eq. (60) were employed. The variational genetic algorithm utilized a population size of 256 individuals and undergoes 256 generations. The results are presented in Table 1, where (No. of generation) represents the generation sequence at which the first best criterion was obtained, which corresponds to it in the table.

As indicated in Table 1, concerning runs 2, 3 and 6 of the CGP, as well as 5, 6 and 7 of the PME techniques, the generations sequence reached its conclusion, but the best criterion still needs to be obtained. Instead, only the lowest criterion achieved during those runs was recorded. This implies that the obtained criterion value was unsatisfactory, and some or all the trajectories did not accurately approach the terminal condition, as depicted in Fig. 4. It is evident that additional generations or time are required for the trajectories to accurately reach the terminal condition

Table 1. The comparison among the SGP, the CGP and the PME results to solve the control synthesis problem

No. of run	No. of generation for SGP	First best criterion in Eq. (62) for SGP	No. of generation for CGP	First best criterion in Eq. (62) for CGP	No. of generation for PME	First best criterion in Eq. (62) for PME
1	85	2.42091	242	1.27057	100	1.55012
2	55	2.29666	256	2.98215	125	1.88136
3	44	2.14699	256	3.07657	236	1.27184
4	130	2.52565	89	2.58705	33	2.35650
5	40	2.18848	219	2.39783	256	2.69261
6	66	1.95788	256	3.28522	256	3.71335
7	24	1.90595	125	2.5949	256	3.37253
8	68	2.49517	158	2.18277	102	2.38121
9	111	2.39997	28	1.99979	85	2.49953
10	100	2.29468	57	2.09583	79	2.28305
ave-rage	**72.3**	**2.263234**	168.6	2.447268	152.8	2.40021

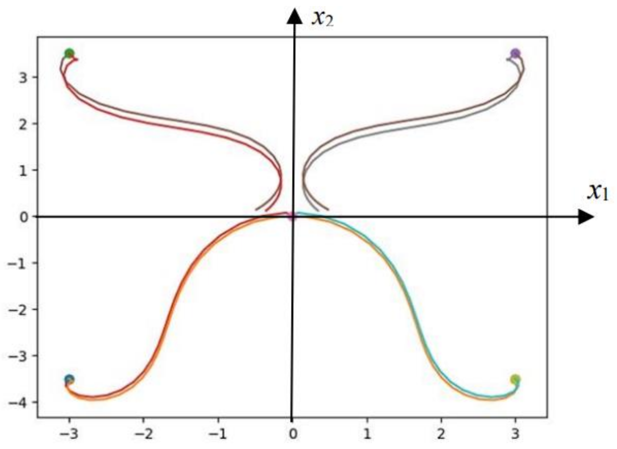

Figure. 4 The robot trajectories do not all reach the terminal condition

and consequently achieve a better criterion value. On the contrary, the results of the SGP technique were deemed satisfactory. As a result of the above, practical evidence has confirmed that synthesized genetic programming exhibits a faster solution-finding rate, about 2.33 and 2.11 times on average, than Cartesian genetic programming and parse-matrix evolution, respectively. This disparity in performance could be attributed to the presence of a more significant number of functions available for each argument within each column inside the matrix of the SGP code.

6. Conclusions

This study introduces a formal mathematical framework for the control synthesis problem, both in a direct manner (unsupervised machine learning) and by employing a training set (supervised machine learning). When one control object changes its initial state (multiple initial states) and moves to one terminal state from each initial state to form multiple trajectories and obtain the control function as a feedback control function, this represents the control system general synthesis problem. The utilization of machine learning techniques, namely symbolic regression approaches, for control purposes, presents an opportunity to address the intricate challenge of control general synthesis inside control theory. The computational instance regarding a mobile robot showcases the potential and future aspirations of symbolic regression techniques like Cartesian genetic programming, synthesized genetic programming and parse-matrix evolution for solving the control general synthesis problem as unsupervised machine learning control techniques. This marks the first use of parse-matrix evolution and synthesized genetic programming in solving the control general synthesis problem, where synthesized genetic programming consists of innovative coding concepts such as the pivot and the priority. On top of that, evidence from our experiments has demonstrated that the efficiency of synthesized genetic programming, being a novel technique, surpasses that of Cartesian genetic programming and parse-matrix evolution in terms of speed of solution discovery for the control system general synthesis problem. This performance discrepancy could be attributed to the greater availability of diverse functions for each argument within the respective columns of the synthesized genetic programming code matrix. Generally, the results elucidated in this work possess significant implications regarding theoretical understanding and practical applications.

The potential applications of different symbolic regression techniques in machine learning control offer novel opportunities in the field of control systems. This shift involves moving away from manual and analytical approaches towards machine

Received: December 31, 2023. Revised: March 18, 2024.

search and machine learning control, thereby expanding the horizons of control-related research.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The contributors to this article are Karrar Sahib Nassrullah (KSN), Ivan Viktorovich Stepanyan (IVS), Haider Sahib Nasrallah (HSN), Neder Jair Mendez Florez (NMF), Abdelrrahmane Mohamed Zidoun (AMZ) and Shahad Raouf Mohammed (SRM), which everyone made the following contributions: Conceptualization, KSN and HSN; methodology, KSN and IVS; software, NMF, KSN and AMZ; validation, HSN, KSN, and NMF; formal analysis, KSN and SRM; investigation, IVS, NMF and AMZ; resources, HSN and SRM; data curation, NMF, AMZ and KSN; writing—original draft preparation, KSN and HSN; writing—review and editing, KSN and SRM; supervision, IVS.

Notations

Symbol	**The meaning**
$\boldsymbol{x}$	The state space vector
$\boldsymbol{u}$	The control vector
$\mathbf{X}_0$	The domain of initial conditions
$\boldsymbol{x}^f$	The terminal condition
t_f	The finishing time from initial condition to terminal one
$\boldsymbol{g}$	Unknown function
$\boldsymbol{U}_0$	The optimal controls set
$\tilde{\mathbf{X}}$	The optimal trajectories set
J_1	The functional for supervised machine learning control
J_2	The functional for unsupervised machine learning control
L	No. of initial points
c_1	A weight coefficient
t^+ , ε	Provided positive values
$\mathcal{W}$	The vector of small variations
S	Possible solution
$\boldsymbol{W}$	The set of variations' vectors
D	A variations depth or length
$\boldsymbol{W_{H+1}}, \boldsymbol{W_{H+2}}$	The new sets of variations' vectors from crossover
G_0	The basic solution code
G_{H+1} , G_{H+2}	The codes within the vicinity of the basic solution code
R_{CGP}	The code of CGP technique
R_{SGP}	The code of SGP technique
H	An ordered set of elements for storing the results of calculations in PME technique
B	the sets of arguments in Eqs. (24) plus (41) in PME technique
R_{PME}	The code of PME technique
u^-, u^+	The lower and upper of the control values
J_{syn}	The functional for unsupervised machine learning control for a mobile robot example
$\tilde{u}_1, \tilde{u}_2$	The control functions
$q_1, \ldots, q_4$	The parameters of the control functions

References

[1] R. Bellman, *Dynamic Programming*, Princeton University Press, 1957.

[2] O. Hernández-Lerma, L. R. Laura-Guarachi, S. Mendoza-Palacios, and D. González-Sánchez, *An introduction to optimal control theory: The dynamic programming approach*, Vol. 76, Springer Nature, 2023.

[3] M. Gerdts, *Optimal control of ODEs and DAEs*, Walter de Gruyter GmbH & Co KG, 2023.

[4] F. Dufour, A. Piunovskiy, and A. Plakhov, "On the equivalence of the integral and differential Bellman equations in impulse control problems", *International Journal of Control*, Vol. 95, No. 1, pp. 173-186, 2022.

[5] X. Yi, B. Luo, and Y. Zhao, "Adaptive dynamic programming-based visual servoing control for quadrotor", *Neurocomputing*, Vol. 504, pp. 251-261, 2022.

[6] Y. Zhang, L. Zou, Y. Liu, D. Ding, and J. Hu, "A brief survey on nonlinear control using adaptive dynamic programming under engineering-oriented complexities", *International Journal of Systems Science*, Vol. 54, No. 8, pp. 1855-1872, 2023.

[7] D. Wang, N. Gao, D. Liu, J. Li, and F. L. Lewis, "Recent progress in reinforcement learning and adaptive dynamic programming for advanced control applications", *IEEE/CAA Journal of Automatica Sinica*, Vol. 11, No. 1, pp. 18-36, 2024.

[8] S. E. Li, *Reinforcement learning for sequential decision and optimal control*, Berlin/Heidelberg, Springer, 2023.

[9] L. S. Pontryagin, V. G. Boltyanski, R. V. Gamkrelidze and E. F. Mishenko, *Mathematical Theory of Optimal Processes*, New York: Interscience, 1969.

[10] V. G. Boltyanski, *Mathematical Methods of Optimal Control*, New York: Holt, Rinehart and Winston, 1971.

[11] M. I. Al-Saedi, "A Modified Block Backstepping Controller for Autonomous Vehicle Lane Changing", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 240-254, 2023, doi: 10.22266/ijies2023.0430.20.

[12] A. J. Humaidi, M. R. Hameed, A. F. Hasan, A. Al-Obaidi, A. T. Azar, I. K. Ibraheem, A. Q. Al-Dujaili, A. K. Al Mhdawi, and F. A. Abdulmajeed, "Algorithmic Design of Block Backstepping Motion and Stabilization Control for Segway Mobile Robot", *Mobile Robot: Motion Control and Path Planning*, pp. 557-607, 2023.

[13] L. T. Thang, T. V. Son, T. D. Khoa, and N. X. Chiem, "Synthesis of sliding mode control for flexible-joint manipulators based on serial invariant manifolds", *Bulletin of Electrical Engineering and Informatics*, Vol. 12, No. 1, pp. 98-108, 2023.

[14] S. P. Kostiukov, A. A. Volkova, and E. V. Churochkina, "Application of the Method of Analytical Design of Aggregated Regulators (ADAR) in the Synthesis of the Laws of Aircraft Motion Control in the Modes of Automatic Refueling in the Air", In: *Proc. of International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 599-603, 2022.

[15] P. Osinenko, G. Yaremenko, and G. Malaniya, "On Stochastic Stabilization via Nonsmooth Control Lyapunov Functions", *IEEE Transactions on Automatic Control*, Vol. 68, No. 8, pp. 4925-4931, 2023.

[16] A. Reed, G. O. Berger, S. Sankaranarayanan, and C. Heckman, "Verified path following using neural control lyapunov functions", In: *Proc. of Conf. on Robot Learning*, pp. 1949-1958, 2023.

[17] F. A. AL-Taie, and A. S. Al-Araji, "Development of Predictive Voltage Controller Design for PEM Fuel Cell System based on Identifier Model", *International Journal of Intelligent Engineering and Systems*, Vol. 16, No. 2, pp. 343-360, 2023, doi: 10.22266/ijies2023.0430.28.

[18] N. Hassan and A. Saleem, "Neural Network-Based Adaptive Controller for Trajectory Tracking of Wheeled Mobile Robots", *IEEE Access*, Vol. 10, pp. 13582-13597, 2022.

[19] M. K. Al-Nussairi, S. D. Al-Majidi, A. J. Mshkil, A. M. Dakhil, M. F. Abbodm, and H. S. Al-Raweshidy., "Design of a Two-Area Automatic Generation Control Using a Single Input Fuzzy Gain Scheduling PID Controller", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, pp. 443-455, 2022, doi: 10.22266/ijies2022.1231.40.

[20] T. T. Thuong, V. T. Ha, and L. N. Truc, "Intelligent Control for Mobile Robots Based on Fuzzy Logic Controller", In: *Proc of The International Conf. on Intelligent Systems & Networks*, pp. 566-573, 2023.

[21] Sh. M. Mahdi, and S. M. Raafat, "Robust Interactive PID Controller Design for Medical Robot System", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, 2022, doi: 10.22266/ijies2022.0228.34.

[22] R. S. Patil, Sh. P. Jadhav, and M. D. Patil, "Review of Intelligent and Nature-Inspired Algorithms-Based Methods for Tuning PID Controllers in Industrial Applications", *Journal of Robotics and Control (JRC)*, Vol. 5, No. 2, pp. 336-358, 2024.

[23] D. Thomas, S. Brunton, and B. Noack, *Machine learning control-taming nonlinear dynamics and turbulence*, Vol. 116, Springer International Publishing, 2017.

[24] J. R. Koza, *Genetic Programming*, MIT Press, 1992.

[25] J. F. Miller, and P. Thomson, "Cartesian Genetic Programming", *Lecture Notes in Computer Science*, p. 121, 2000.

[26] A. Diveev, and E. A. Sofronova, "Numerical method of network operator for multiobjective synthesis of optimal control system", In: *Proc of IEEE International Conference on Control and Automation*, pp. 701-708, 2009.

[27] C. Luo and S.-L. Zhang, "Parse-matrix evolution for symbolic regression", *Engineering Applications of Artificial Intelligence*, Vol. 25, No. 6, pp. 1182-1193, 2012.