Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 19, SEPTEMBER 2024

DOI: 10.37943/19OXFC5347

Askhat Diveev
PhD, Associate Professor of the Department of Computer Science, Institute of Engineering and Technology
aidiveev@mail.ru, orcid.org/0000-0003-2057-9016
Korkyt Ata Kyzylorda University, Kazakhstan

Nurbek Konyrbaev
PhD, Associate professor, head of the Department of Computer Science, Institute of Engineering and Technology
n.konyrbaev@mail.ru, orcid.org/0000-0002-8788-4149
Korkyt Ata Kyzylorda University, Kazakhstan

Zharasbek Baishemirov
PhD, Associate Professor of Mathematics and Mathematical Modeling
zbai.kz@gmail.com, orcid.org/0000-0002-4812-4104
Abai Kazakh National Pedagogical University, Kazakhstan
Associate Professor of School of Applied Mathematics
Kazakh-British Technical University, Kazakhstan

Asem Galymzhankyzy
Master, Teacher of the Department of Computer Science, Institute of Engineering and Technology
asem.galymzhankyzy@gmail.com, orcid.org/0009-0004-4624-8797
Korkyt Ata Kyzylorda University, Kazakhstan

Oralbek Abdullayev
Master, Teacher of the Department of Computer Science, Institute of Engineering and Technology
oralbek.abdullaev@mail.ru, orcid.org/0009-0001-5649-6805
Korkyt Ata Kyzylorda University, Kazakhstan

CONTROL SYSTEMS SYNTHESIS FOR ROBOTS ON THE BASE OF MACHINE LEARNING BY SYMBOLIC REGRESSION

Abstract: This paper presents a novel numerical method for solving the control system synthesis problem through the application of machine learning techniques, with a particular focus on symbolic regression. Symbolic regression is used to automate the development of control systems by constructing mathematical expressions that describe control functions based on system data. Unlike traditional methods, which often require manual programming and tuning, this approach leverages machine learning to discover optimal control solutions. The paper introduces a general framework for machine learning in control system design, with an emphasis on the use of evolutionary algorithms to optimize the generated control functions. The key contribution of this research lies in the development of an algorithm based on the principle of small variations in the baseline solution. This approach significantly enhances the efficiency of discovering optimal control functions by systematically exploring the solution space with minimal adjustments. The method allows for the automatic generation of control laws, reducing the need for manual coding, which is especially beneficial in the context of complex control systems, such as robotics. To demonstrate the applicability of the method, the research applies symbolic regression to the control synthesis of a mobile robot.

Received: 09.09.2024 Accepted: 26.09.2024 Published: 30.09.2024

The results of this case study show that symbolic regression can effectively automate the process of generating control functions, significantly reducing development time while improving accuracy. However, the paper also acknowledges certain limitations, including the computational demands required for symbolic regression and the challenges associated with real-time implementation in highly dynamic environments. These issues represent important areas for future research, where further optimization and hybrid approaches may enhance the method's practicality and scalability in real-world applications.

Keywords: control synthesis, machine learning control, symbolic regression, evolutionary algorithm.

Introduction

This research presents an innovative approach to control system synthesis based on machine learning methods, specifically symbolic regression. Unlike traditional control system development methods, where programming and tuning are performed manually, the proposed approach automates the process of generating control functions by utilizing evolutionary algorithms to search for optimal solutions. Symbolic regression allows for the creation of mathematical expressions that describe control functions based on data obtained during the operation of the system.

The main contribution of the research lies in the development and application of a machine learning algorithm based on the principle of small variations of the baseline solution. This method significantly improves the efficiency of solution discovery and reduces the time required for the synthesis of complex control systems. The example of control system synthesis for a mobile robot demonstrates that symbolic regression can automate the process of developing control functions, eliminating the need for manual programming, which is particularly important in the context of robotics, where the complexity of control systems increases as the functional capabilities of control objects grow.

Modern control systems, regardless of their complexity—from simple regulators to intelligent systems—are essentially programs executed on the onboard processor of the controlled object. These programs are manually written by programmers with specialized knowledge in control systems, which imposes significant constraints on the development process. The code for complex robotic systems can reach millions of lines, and while the technological process of writing such programs can be effectively organized, automating the process itself remains a challenging task. To minimize programmer effort, code repositories like GitHub are increasingly being used, where ready-made program fragments or similar solutions can be found and adapted for specific control systems. However, this approach does not address the issue of growing code complexity as robots are tasked with solving more intricate problems.

The increasing complexity of programs is inevitably linked to growing demands on the control system. The rise in the number of sensors, interactions between components and external systems, as well as the need for adaptation to dynamically changing environments, leads to an exponential increase in the volume of code. To address similar challenges, AlKhlidi et al. proposed the Modified Fuzzy Particle Swarm Optimization (MPSO) algorithm for path planning, which ensures the calculation of optimal, collision-free paths for mobile robots in complex environments, improving both speed and resource efficiency.[1] This process reaches a point where the length and complexity of the software may become major obstacles in the development of more advanced intelligent systems, including those aspiring to implement artificial intelligence (AI). Thus, one of the key barriers to creating fully-fledged AI lies not only in the complexity of the mathematical models and algorithms but also in the automation of the system development process itself.

Programs that implement control systems are essentially expressions in programming languages that mathematically describe control functions. Therefore, these mathematical expressions could theoretically be derived from solving mathematical problems that describe the dynamics of control objects and their interaction with the environment [2], [3].

However, the complexity of modern mathematical models and the lack of universal methods for solving them complicate the automation of control synthesis processes. In control theory, these problems are typically solved for individual, simplified models, allowing for the development of control theory methods, but excluding external disturbances and uncertainties. As a result, while a mathematical solution can be found, a new challenge arises—how to adapt this solution to real-world conditions. The implementation process for specific control systems becomes particularly difficult when the model cannot account for all possible impacts or environmental changes.

Moreover, traditional approaches to developing control systems are constrained by existing programming methods, where each solution is manually adapted to a specific task and control object. Despite significant advances in software tools and technologies, the question of fully automating control synthesis remains unresolved. In this context, one of the promising areas for further research is the application of machine learning methods, particularly symbolic regression, to automate the synthesis of control systems. In our work, we employ principles of genetic programming, as detailed in [4], to search for optimal control solutions using symbolic regression.

Proposed Methodology

Symbolic regression, as a machine learning method, is capable of constructing mathematical expressions for control functions based on data obtained during the system's operation. This enables the automatic discovery of optimal solutions for complex control tasks, eliminating the need for manual coding. The main advantage of symbolic regression lies in its use of evolutionary algorithms to search for solutions, significantly reducing the time required for solution discovery and improving solution quality. However, the challenge of generating real-time solutions remains, as complex control synthesis tasks still demand substantial computational resources. Willms and Yang addressed similar real-time computational challenges by proposing an efficient dynamic system for robot-path planning, which reduces the time complexity and enhances the system's real-time responsiveness [5].

For obtaining the results presented in the figures, symbolic regression based on genetic programming was used. Data for training were collected using models of robots in dynamic conditions. Symbolic regression automatically generated mathematical expressions that describe the control functions. Evolutionary search algorithms were applied to optimize the obtained solutions, and small variations of the baseline solution were utilized to enhance the efficiency of discovering optimal trajectories. This approach allowed the rapid generation of control laws, significantly reducing the need for manual intervention, particularly in complex systems such as robotic control.

The research methodology is based on a combination of symbolic regression and evolutionary algorithms. Unlike traditional approaches that require complex mathematical modeling and manual tuning, symbolic regression automatically generates control functions by optimizing them based on system data. The advantage of the proposed method lies in its flexibility and ability to adapt to various types of control systems, which is especially crucial for tasks with high dynamics and environmental variability.

Various approaches are used in modern control systems, including classical numerical methods, Model Predictive Control (MPC), gradient methods, and evolutionary algorithms. Each has its strengths and weaknesses, and analyzing them helps highlight how the symbolic

DOI: 10.37943/19OXFC5347

regression-based approach can improve or complement these methods.

Classical numerical methods, such as Linear-Quadratic Regulation (LQR) and optimal control, provide precise solutions for linear and some nonlinear systems. However, as noted by Kwakernaak and Sivan (1972) [6] and Anderson and Moore (1990) [7], these methods rely on accurate mathematical models, limiting their flexibility in dynamic environments with uncertainties. In contrast, symbolic regression can adapt to changing conditions by automatically generating control functions based on system data, making it more suitable for nonlinear systems.

Model Predictive Control (MPC) is popular for real-time control because it uses system models to predict future behavior and compute optimal control actions. However, as Camacho and Bordons (2004) [8] and Mayne et al. (2000) point out [9], MPC faces high computational demands, particularly in fast-changing systems. Symbolic regression offers a way to reduce computational costs by generating control functions from data without requiring complex real-time calculations, while still adapting to system dynamics.

Gradient methods are commonly used to tune controller parameters and perform well for smooth, differentiable systems. However, as Boyd and Vandenberghe (2004) [10] highlight, they can get stuck in local minima and require significant time to converge, especially in high-dimensional systems. Symbolic regression, by using evolutionary algorithms, avoids these pitfalls, providing a more reliable global optimization process.

Evolutionary algorithms, such as genetic programming, have long been used for optimization in control systems, particularly for solving complex, multi-dimensional problems. However, as noted by Goldberg (1989) [11], these methods can be slow due to random fluctuations in the search process during crossover and mutation operations. Symbolic regression mitigates this by applying the principle of small variations to the baseline solution, reducing fluctuations and accelerating the optimization process.

The symbolic regression approach offers several advantages over existing methods:

Automation of Synthesis: Unlike manual tuning and programming, symbolic regression automates the creation of control functions from data, significantly reducing time and resource requirements for complex control systems.

Flexibility and Adaptability: It works effectively with dynamic systems where traditional analytical models may be challenging to develop and automatically adapts to environmental changes, making it ideal for robotics and other complex systems.

Handling Nonlinear Problems: Symbolic regression is particularly suited for nonlinear systems, where traditional methods often struggle.

Reduced Computational Costs: Unlike computationally intensive methods like MPC, symbolic regression minimizes resource demands by automating the search for optimal solutions without requiring continuous real-time predictions.

The primary limitation of symbolic regression is its computational complexity during the solution search phase, which can hinder real-time applications in highly dynamic systems. However, further research into optimizing symbolic regression techniques, such as hybrid approaches, could help overcome this challenge.

Thus, the use of symbolic regression methods in machine learning offers promising opportunities for automating the creation of control systems, which is particularly relevant for robotic systems where control complexity increases as the functional capabilities of the controlled objects expand. Kala et al. improved path planning algorithms by integrating dynamic programming with accelerating nodes, significantly boosting computational efficiency in dynamic robotic environments [12]. Methods presented in [13] have been used to describe the dynamics of nonlinear systems, such as mobile robots, enabling better modeling of complex robot behavior and control in real-world environments. The machine learning framework for the control system using symbolic regression methods is shown in Fig. 1.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 19, SEPTEMBER 2024

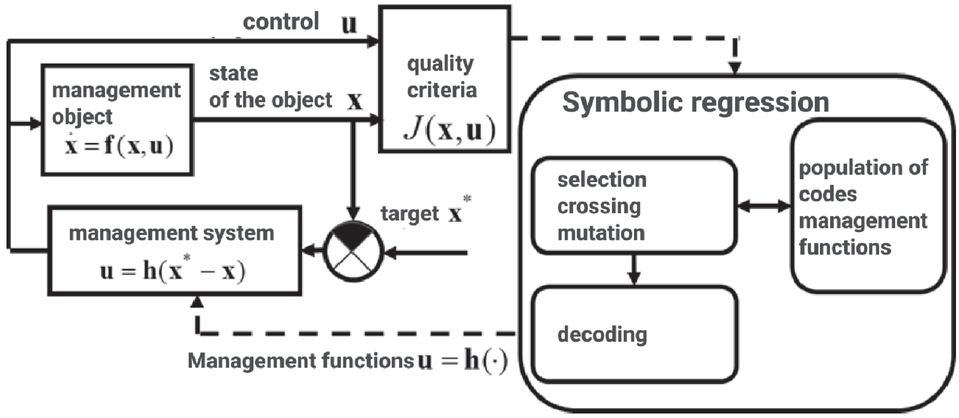

Figure 1. Machine learning framework for the control system

The classical optimal control problem

Machine learning enables solving mathematical problems of any complexity by reducing the requirements for finding a solution. For example, any mathematical problem in control, optimal control, control synthesis, identification, filtering, etc., is always formulated as an optimization problem, and optimization algorithms are always used in machine learning. However, these algorithms never find a strict optimum in machine learning. The solution found is deemed satisfactory by the researcher based on the value of the performance criterion. Another characteristic of solving problems using machine learning is the lack of strict proof that the obtained solution meets the requirements of the problem. Instead, this proof is validated through examples. If, for most examples, the found solution satisfies the researcher in terms of accuracy and performance criterion, the conclusion is made that the solution is acceptable.

Consideration is given to the classical optimal control problem.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}),\ \mathbf{u} \in \mathrm{U} \subseteq \mathrm{R}^m,\ \mathbf{x}(0) = \mathbf{x}^0,\ \mathbf{x}(t_f) = \mathbf{x}^f,\ J = \int_0^{t_f} f_0(\mathbf{x},\mathbf{u})dt \to \min_{\mathbf{u}\in\mathrm{U}}, \quad (1)$$

where $\mathbf{x}$ – state vector, $\mathbf{u}$ – control vector, U – compact set, $\mathbf{x}^0$ – initial state vector, $\mathbf{x}^f$ – terminal state vector, t_f – time to reach the terminal state, it is typically not specified but is bounded, $t_f \leq t^+$, t^+ – positive value, J – control performance criterion value.

Formulation of the optimal control problem

The task is to find the control as a function of time. If the obtained control is substituted into the right-hand side of the system model, the resulting system will have a particular solution that, starting from the given initial state, reaches the terminal state with an optimal value of the performance criterion. However, the time-dependent control function cannot be directly implemented in a real system, as this would result in an open-loop control system. To implement the solution, it is refined the formulation of the optimal control problem. The control is sought in the form of a function of the state vector and time.

$$\mathbf{u} = \mathbf{g}(\mathbf{x},t) \in \mathrm{U}. \quad (2)$$

For the optimal particular solution, there must exist a neighborhood such that if another particular solution of the system, starting from a different initial state, enters this neighborhood at some point, it will not leave the neighborhood but will instead approach the optimal solution. In other words, the optimal trajectory should have an attractive property, meaning the system of differential equations in the region of the optimal solution must be a contracting mapping.

To solve the extended optimal control problem, it is used the solution of the control synthesis problem. In the control synthesis problem, the model of the control object, constraints on the control inputs, terminal state, and the control performance criterion are also given, but the initial conditions are specified as a region in the state space.

$$\mathrm{X}_0 \subseteq \mathrm{R}^n . \tag{3}$$

The control is sought in the form of the following function:

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^f - \mathbf{x}) \in \mathrm{U} . \tag{4}$$

where h(x^*-x) is the control function, structure and parameters that must be found as a result of solving the control synthesis problem. If the obtained control function is substituted into the control system model, we get a system of equations that describes the closed-loop control system.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^f - \mathbf{x})) . \tag{5}$$

A contracting mapping plays a key role in the stability of trajectories, ensuring that the system remains within a small neighborhood of the terminal state (6). This allows the system to move steadily towards the target, even with small deviations from the trajectory, which is particularly important for complex dynamic objects like robots.

To solve this control synthesis problem, machine learning methods based on numerical symbolic regression techniques are employed. Symbolic regression is a method that constructs mathematical expressions to describe control functions. This approach enables efficient automation of the search for optimal solutions, especially for complex systems where traditional analytical methods may not be applicable.

Symbolic regression methods encode mathematical expressions and search for the optimal solution in the space of possible codes. However, one of the main drawbacks of these methods is that during the execution of genetic algorithm operations, significant changes to the codes can occur. This leads to operations like crossover and mutation drastically altering the structure of the solution, making the search process akin to random search.

By its nature, random search does not guarantee finding an optimal solution within a reasonable time frame. Recent advances in symbolic regression, such as the work on controllable neural symbolic regression, offer a promising avenue to address these limitations by integrating deep learning and symbolic models to better control the structure of expressions while optimizing for time efficiency [14]. Genetic algorithms tend to experience significant fluctuations in the search process due to the high variability of the codes, making it difficult to find a stable, optimal solution. In such cases, the control synthesis problem becomes extremely challenging to automate. Without applying additional constraints or improved heuristics during the crossover and mutation phases, the solution search process may become suboptimal and prolonged. Masehian and Sedighizadeh's review underscores the importance of heuristic approaches in overcoming the limitations of classic robot motion planning methods [15].

To accelerate the solution search process, the symbolic regression method should employ the principle of small variations of the baseline solution. In this case, the set of possible solutions consists of the symbolic regression code of one baseline solution and ordered sets of codes representing small variations of this baseline solution. The genetic algorithm operation is performed on the ordered sets of codes for small variations according to classical rules. A drawback of the control synthesis problem is that, due to its complexity, it cannot be solved in real-time on the control object's onboard system.

Another approach to solving the presented optimal control problem is the synthesized control method. Initially, the control synthesis problem is solved to ensure the stability of the object relative to a point in the state space. The control is sought in the form (4).

At the second stage, we consider the model of the closed-loop control system.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^* - \mathbf{x})), \tag{6}$$

For this system, it is solved the optimal control problem with the initial performance criterion. It is used the vector as the control input $\mathbf{x}^* = [x_1^* \ \ldots \ x_n^*]^T$,...determining the position of a stable equilibrium point in the state space. In synthesized control, a stable equilibrium point always exists in the neighborhood of the optimal trajectory, towards which the control object tends to move. Khalil et al. expanded on this concept by focusing on distributed path planning for multi-robot systems, enhancing obstacle avoidance and enabling the coordination of multiple robots in real-time environments [16]. As a result, the object's trajectory always lies within the region where the conditions for a contracting mapping are satisfied. In this method, the control synthesis problem for ensuring stability is also solved at the design stage, and its solution is implemented on the control object's onboard system. Regardless of the current situation, this problem does not need to be solved again onboard. The optimal control problem is then solved in real-time on the control object.

As an example, consideration is given to the control synthesis for a car-like robot.

$$\dot{x}_1 = u_1 \cos(x_3)\cos(x_4), \ \dot{x}_2 = u_1 \cos(x_3)\sin(x_4), \ \dot{x}_3 = u_2, \ \dot{x}_4 = \frac{u_1}{L}\sin(x_3). \tag{7}$$

where $L = 0.4$, $0 = u_1^- \le u_1 \le u_1^+ = 2$, $-1 = u_2^- \le u_2 \le u_2^+ = 1$.

For the system (7), the initial conditions are specified $\mathbf{x}(0) = \mathbf{x}^0 = [0 \ 0 \ 0 \ 0]^T$, terminal state $\mathbf{x}(t_f) = \mathbf{x}^f = [5 \ 5 \ 0 \ 0]^T$, $t^+ = 7.0$, $\varepsilon_0 = 0.025$, and the performance criterion

$$J_3 = p_1 \| \mathbf{x}^f - \mathbf{x}(t_f) \| + t_f + p_2 \int_0^{t_f} \sum_{i=1}^{2} \vartheta(\varphi_i(\mathbf{x}))dt \to \min_{\mathbf{u}\in \mathrm{U}}, \tag{8}$$

where p_1, p_2 are penalty coefficients and $p_1 = 1, p_2 = 2$,

$$t_f = \begin{cases} t, \text{ if } t < t^+ \text{ and } \|\mathbf{x}^f - \mathbf{x}(t)\| \le \varepsilon_0 \\ t^+ \text{ - otherwise} \end{cases}, \ \vartheta(\alpha) = \begin{cases} 1, \text{ if } \alpha > 0 \\ 0 \text{ - otherwise} \end{cases}, \tag{9}$$

$$\varphi_i(\mathbf{x}) = r_i - \sqrt{(x_1 - x_{1,i})^2 + (x_2 - x_{2,i})^2} \ . \ i = 1, 2,$$

$r_1 = 1$, $r_2 = 1$, $x_{1,1} = 1$, $x_{1,2} = 4$, $x_{2,1} = 1$, $x_{2,2} = 4$.

The synthesized control method was used to solve the problem. Initially, the control synthesis problem was solved using the symbolic regression method. Machine learning produced the following result.

$$u_i = \begin{cases} u_i^+, \text{ if } \hat{u}_i \ge u_i^+ \\ u_i^-, \text{ if } \hat{u}_i \le u_i^- \\ \hat{u}_i \text{ - otherwise} \end{cases}, \ i = 1, 2, \tag{10}$$

where

$$\begin{gathered} \hat{u}_1 = E + \arctan(F - q_2(x_2^* - x_2)q_4(x_4^* - x_4) + \sqrt[3]{q_3}(x_3^* - x_3)) + \\ + \sqrt[3]{\sqrt[3]{q_3}(x_3^* - x_3)} + x_2^* - x_2 - (x_2^* - x_2)^3, \end{gathered} \tag{11}$$

$$\hat{u}_2 = \hat{u}_1 - \hat{u}_1^{\#} + +\rho_{18}(E) + \operatorname{sgn}(G) - q_2(x_2^* - x_2)q_4(x_4^* - x_4) + \sqrt[3]{q_3}(x_3^* - x_3) + \\ -(-q_2(x_2^* - x_2)q_4(x_4^* - x_4) + \sqrt[3]{q_3}(x_3^* - x_3))^3 + \sqrt[3]{q_2 x_2^* - x_2} + \rho_{18}(x_2^* - x_2) - (x_1^* - x_1), \tag{12}$$

$$E = \rho_{19}(G) + (F + -q_2(x_2^* - x_2)q_4(x_4^* - x_4) + \sqrt[3]{q_3}(x_3^* - x_3))^3 + \\ + \sin(-q_2(x_2^* - x_2)q_4(x_4^* - x_4) + \sqrt[3]{q_3}(x_3^* - x_3)) + \mu(q_2(x_2^* - x_2)) + \sin(x_1^* - x_1),$$

$$G = \rho_{18}(F - q_2(x_2^* - x_2)q_4(x_4^* - x_4) + \sqrt[3]{q_3}(x_3^* - x_3)) \times \\ \times \sin(-q_2(x_2^* - x_2)q_4(x_4^* - x_4))\rho_{17}(q_2(x_2^* - x_2))\rho_{19}(q_4),$$

$$F = \operatorname{sgn}(-q_2(x_2^* - x_2)q_4(x_4^* - x_4)) + q_2(x_2^* - x_2) + q_1(x_1^* - x_1) + \rho_{18}(x_2^* - x_2),$$

$$\rho_{18}(\alpha) = \operatorname{sgn}(\alpha)(\exp(|\alpha|) - 1), \ \rho_{19}(\alpha) = \operatorname{sgn}(\alpha)\exp(-|\alpha|),$$

$$q_1 = 4.87134, \ q_2 = 10.76855, \ q_3 = 12.34106, \ q_4 = 0.90137.$$

At the second stage, the optimal control problem was solved based on approximating the control using a piecewise constant function. In a similar vein, Tang et al. explored how multi-objective particle swarm optimization (PSO) can be utilized for robot path planning, addressing uncertainties and optimizing the robot's trajectory in dynamic environments [17].

Figure 2 shows the optimal trajectory of the mobile robot's movement on the plane. In the same figure, the projections onto the plane are depicted as small black squares $\{x_1, x_2\}$ of the optimal values of the state vectors that were found $\mathbf{x}^* = [x_1^* \, x_2^* \, x_3^* \, x_4^*]^T$ these vectors determine the position of the stable equilibrium point.

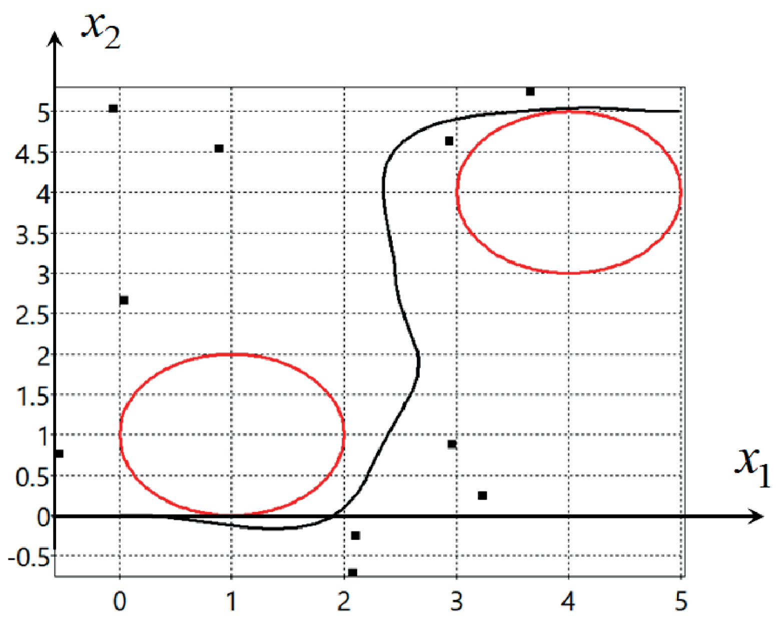

Figure 2. Optimal trajectory of the synthesized control

The optimal trajectory shown in Figure 2 was calculated using symbolic regression methods, which automatically generated control functions based on the robot's current state and surrounding environmental data. The process began by defining a control problem where the robot needed to move from an initial point to a target point while avoiding obstacles. Symbolic regression was used to find optimal control laws, utilizing evolutionary algorithms to iteratively search for and optimize control functions.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 19, SEPTEMBER 2024

Once the optimal control functions were determined, they were applied to the robot's state-space model. This allowed the trajectory to be calculated by solving the system's differential equations, ensuring that the robot followed the optimal path while avoiding obstacles. The small black squares on the figure represent key points where the robot's state reached stable equilibrium, further confirming that the control system successfully maintained the desired trajectory.

From the above, it can be concluded that the symbolic regression methods used for machine learning in control systems allow for the automatic discovery of mathematical expressions for the control function.

The simulation results using symbolic regression methods produced optimal trajectories for the mobile robot, demonstrating high accuracy in control and adaptability to changing environmental conditions. Compared to classical numerical methods such as LQR, our approach does not require an exact mathematical model of the system and can automatically generate control functions based on data. This makes symbolic regression a more flexible and effective method for working with nonlinear systems, where classical methods show limited performance.

Symbolic regression also outperforms Model Predictive Control (MPC), which requires significant computational resources for continuous recalculation of optimal control. In contrast, our method generates control functions from data and conserves computational resources after the solution search phase. Compared to evolutionary algorithms, symbolic regression with small variations helped avoid random fluctuations and sped up the process of finding optimal solutions, making it a more stable and faster method for real-world applications.

Results

The results of the experimental application of symbolic regression methods for solving control system synthesis problems demonstrated the effectiveness of this approach. Machine learning methods, particularly symbolic regression, were successfully employed to address the task of synthesizing control systems. This method enabled the automatic construction of mathematical expressions for control functions based on system data, significantly reducing the need for manual programming. Symbolic regression proved to be effective in solving complex control problems, including optimal control, system synthesis, identification, and filtering tasks, by framing these problems as optimization tasks.

Figure 1 illustrates the machine learning framework for the control system. Based on data about the robot's environment and state, the control function was automatically generated. The results confirm that symbolic regression effectively automates the process of creating control functions, significantly reducing development time.

Optimization algorithms, such as genetic algorithms, were employed in the machine learning process to search for optimal control functions. Although these algorithms did not always find strict optima, the solutions obtained were satisfactory according to the quality criteria set by the researchers. While no strict proof was provided that the solutions fully met the predefined requirements, they performed well in most examples, maintaining acceptable accuracy limits.

In the classical optimal control problem, symbolic regression was applied to determine control as a function of time. The resulting system model produced partial solutions that guided the system from an initial state to a terminal state, achieving optimal values according to the performance criteria. However, directly implementing time-based control functions in real-world systems proved impractical, as it would result in open-loop control systems. Therefore, the control synthesis problem was reformulated, with control functions expressed as a function of both the state vector and time, allowing for practical implementation in closed-loop systems.

Figure 2 shows the optimal trajectory of the robot's movement, obtained using symbolic regression methods. The optimal trajectory minimizes control costs while avoiding obstacles, demonstrating the applicability of the method for real-world robot control tasks.

Symbolic regression successfully generated control functions for systems with predefined regions of the state space and boundary conditions. Other researchers, such as Dovgopolik et al., have extended this concept with algorithms like the Modified Intelligent Bidirectional Random Tree Algorithm, which optimizes path planning in complex environments. The resulting closed-loop system, described by a differential equation, was able to transition from any initial condition within the defined state space to a terminal state while meeting optimal control quality criteria. This approach enabled not just a single optimal trajectory but a set of trajectories from various initial states, demonstrating the system's robustness and adaptability.

In numerical experiments with a mobile robot, the synthesized control functions were applied, and the optimal trajectories generated during these tests closely matched theoretical predictions. The use of symbolic regression allowed for the automatic derivation of control functions, and small variations of base solutions were explored to improve the search process. The experimental results confirmed the effectiveness of this approach, as the projected solutions aligned well with expectations.

The findings suggest that symbolic regression shows potential, for streamlining the development of control systems. It effectively addresses classical control problems by simplifying the synthesis process and reducing complexity. However, despite its advantages, the complexity of the synthesis process limits its current application in real-time control tasks due to significant computational resource demands.

Discussion

In tests, with robots it was shown that symbolic regression can create control functions that mimic paths quite well. This technique streamlines the control synthesis process significantly. Proves beneficial for intricate dynamic systems like robotics. Saving time and reducing errors that often occur with manual programming. The capability to create control functions automatically from system data without needing models offers a notable edge, over conventional methods.

The findings indicate that symbolic regression has the capability to autonomously create control functions - a feature, for dynamic systems like robotics. The capacity to adjust control functions according to data enables adaptability and effectiveness in designing control systems. This becomes particularly vital, in situations where the robots' environment can unexpectedly shift.

A major drawback of this approach is its computational requirements. Finding the solutions through algorithms is indeed efficient but demands significant computational capabilities. This restricts its use, in real time scenarios for embedded systems with constrained processing abilities. The need for resources poses a challenge to the swift integration of symbolic regression, in tasks that necessitate instant responsiveness and high performance.

Despite the obstacles faced in this area of study the application of regression to automate control functions displays potential. This approach streamlines the design phase cuts down on development duration and lessens the need, for involvement in programming control systems. However additional efforts are required to refine the technique, for use cases. Enhancements could include hardware optimization, such as utilizing more powerful processors or implementing parallel computing architectures, to accelerate the search and optimization processes. Additionally, hybrid approaches combining symbolic regression with other machine learning techniques or classical control methods could reduce computational costs while maintaining flexibility.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 19, SEPTEMBER 2024

In future research, improving the efficiency of symbolic regression will be essential to fully exploit its potential. Hybrid methods, which combine symbolic regression with traditional optimization techniques or other machine learning models, could reduce the computational burden and make real-time applications more feasible. Moreover, optimizing symbolic regression algorithms for specific hardware, or leveraging specialized processors like GPUs, could improve performance in dynamic, real-time environments.

In summary, symbolic regression represents a promising tool for automating control system synthesis, especially in scenarios where traditional methods are inadequate due to the complexity of the control systems involved. However, improvements are required to fully leverage its potential in real-time, high-performance environments. Future research should focus on optimizing both the algorithm and hardware resources to ensure that symbolic regression can be deployed in real-world, time-sensitive applications.

Conclusion

The findings of this research underscore the effectiveness of symbolic regression in automating the synthesis of control systems, particularly for complex robotic applications. This study demonstrates that symbolic regression can successfully generate control functions based on data, eliminating the need for manual coding and significantly reducing development time. By framing control problems as optimization tasks, the method efficiently addresses both simple and complex control scenarios. The results validate the potential of symbolic regression to streamline control system development and reduce human error.

However, the study also highlights limitations, specifically the computational demands and the inability of symbolic regression to guarantee strict global optima. These challenges present opportunities for future research focused on optimizing the algorithm for real-time applications and improving the accuracy of solutions through enhanced optimization techniques or hybrid approaches.

In conclusion, symbolic regression represents a significant advancement in the automation of control system synthesis, offering promising solutions for dynamic and complex environments. While further work is needed to address its limitations, the method has the potential to play a key role in the development of intelligent systems and robotics, offering both efficiency and flexibility. Future research should focus on refining this approach to expand its practical applicability, particularly in real-time embedded systems.

Acknowledgement

This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP14869851).

References

[1] Baidaa AlKhlidi, Ahmad T. Abdulsadda, Ali Al Bakri.: Optimal Robotic Path Planning Using Intelligents Search Algorithms. Journal of Robotics and Control (JRC) Volume 2, Issue 6, November 2021 ISSN: 2715-5072 https://www.doi.org/10.18196/jrc.26132

[2] Astolfi A., Marconi L. “Nonlinear and Adaptive Control Design”. Springer, 2003, стр. 95-120.

[3] Kalman R.E. “Contributions to the Theory of Optimal Control”. IEEE Transactions on Automatic Control, 1960, стр. 102-115.

[4] Koza, J.R. (1992). Genetic programming: On the programming of computers by means of natural selection. MIT Press.

[5] A.R. Willms, S.X. Yang, An Efficient Dynamic System for Real-Time Robot-Path Planning, IEEE Transactions on Systems Manand Cybernetics – Part B Cybernetics, 36(4) (2006), 755-766

[6] Kwakernaak, H., & Sivan, R. (1972). *Linear optimal control systems*. Wiley-Interscience

[7] Anderson, B.D.O., & Moore, J.B. (1990). *Optimal control: Linear quadratic methods*. Dover Publications.
[8] Camacho, E.F., & Bordons, C. (2004). *Model predictive control*. Springer. DOI: https://doi.org/10.1007/978-0-85729-398-5
[9] Mayne, D.Q., Rawlings, J.B., Rao, C.V., & Scokaert, P.O.M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789-814. DOI: https://doi.org/10.1016/S0005-1098(99)00214-9
[10] Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press. DOI: https://doi.org/10.1017/CBO9780511804441
[11] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.
[12] Rahul Kala, Anupam Shukla and Ritu Tiwari.: Robot Path Planning using Dynamic Programming with Accelerating Nodes/ From the journal Paladyn, Journal of Behavioral Robotics https://doi.org/10.2478/s13230-012-0013-4
[13] Khalil, H. K. (2002). Nonlinear systems. Prentice Hall.
[14] M. Schlegel, "Controllable Neural Symbolic Regression," arXiv Preprint, 2021, doi: https://www.doi.org/10.48550/arXiv.2107.14351
[15] E. Masehian, D. Sedighizadeh, Classic and heuristic approaches in robot motion planning—a chronological review, in: Proc. World Acad. Sci., Eng. Technol.. 2007, 23. 101–106.
[16] H. Khalil, A. Shukla, and R. Tiwari, "Distributed Multi-Mobile Robot Path Planning and Obstacle Avoidance," Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN), pp. 146–151, 2016, https://www.doi.org/10.1109/CICN.2016.35 .
[17] J. Tang, J. Zhu, Z. Sun, A novel path planning approach based on appART and particle swarm optimization, Lect. Notes Comput. Sci. 3498 (2005) 253–258