

Proceeding Paper

Synthesis of a Feedback Controller by the Network Operator Method for a Mobile Robot Rosbot in Gazebo Environment †

Elizaveta Shmalko ^{*,‡} and Yuri Rumyantsev ‡

Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, 119333 Moscow, Russia; urock@fastsense.tech

* Correspondence: e.shmalko@gmail.com

† Presented at the 15th International Conference “Intelligent Systems” (INTELS’22), Moscow, Russia, 14–16 December 2022.

‡ These authors contributed equally to this work.

Abstract: The article presents an approach based on machine learning with symbolic regression for the synthesis of a stabilization system for a mobile robot. This approach is universal and allows you to numerically solve the synthesis problem in a general setting without the need to form a training sample, instead relying only on the value of the functional. The synthesis is implemented to stabilize the mobile robot Rosbot in the Gazebo simulation environment. The feedback stabilization system is received by the network operator method. The advantage of the method is that it can be applied to a control object of any complexity and linearity.

Keywords: control synthesis; stabilization; machine learning; symbolic regression; Gazebo mobile robot

1. Introduction

Synthesis of a feedback stabilization system is one of the key tasks in applied robotics. Feedback is needed to level out the differences between the model and the real object, as well as other possible uncertainties and noise. From a mathematical point of view, this problem belongs to the class of problems for the synthesis of optimal control systems, where it is necessary to find a control vector function that depends on the state vector of the object and delivers a minimum to the quality functional (see Figure 1).

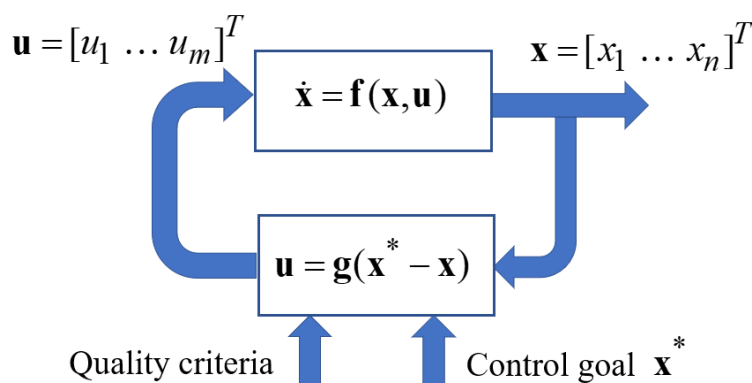


Figure 1. The problem of synthesis of the stabilization system.

There are two main strategies for the synthesis of a control system: parametric and structural-parametric.

Parametric synthesis includes all methods in which the control structure is specified, and only the parameters are optimally tuned from the point of view of the functional. This synthesis strategy is by far the most common. It also includes the most popular various PID



Citation: Shmalko, E.; Rumyantsev, Y. Synthesis of a Feedback Controller by the Network Operator Method for a Mobile Robot Rosbot in Gazebo Environment. *Eng. Proc.* **2023**, *33*, 6. <https://doi.org/10.3390/engproc2023033006>

Academic Editors: Askhat Diveev, Ivan Zelinka, Arutun Avetisyan and Alexander Ilin

Published: 16 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

controllers [1,2], other controllers based on neural networks [3,4], and fuzzy logic [5,6], etc. In all these approaches, some preliminary knowledge of the developer about the object is required in order to set the controller structure as correctly as possible; nevertheless, there is still no reason to consider the chosen structure as optimal, as only the parameters are adjusted according to the optimality criterion.

With the structural-parametric approach, not only are the parameters optimized, but the optimal structure of the feedback control function is also sought. Among the analytical methods for solving the problem of stabilization system synthesis, the most popular are methods based on solving the Riccati equation [7], but, for linear systems only, as well as more modern analytical approaches of backstepping [8] and analytical design of aggregated controllers [9,10], they also depend on the types of right-hand sides of nonlinear differential equations that describe the control object model. This challenging task for nonlinear systems can be addressed by a dynamic programming (DP) algorithm assisted by the Hamilton–Jacobi–Bellman (HJB) equation. However, it always results in difficulty due to the well-known curse of dimensionality. Recently, modern techniques such as adaptive DP (ADP) [11–14] and reinforcement learning (RL) [15–17] have received increasing attention as powerful machine learning and optimization strategies for addressing the control problems of nonlinear systems regarding the use of neural networks for numerical approximating solutions of the HJB equation. However, these approaches also have many computational difficulties, primarily related to the definition and training of the neural networks used.

Thus, the application of machine learning methods opens up broad prospects, but it is necessary to develop novel control methods addressing general mathematical statements of the control synthesis problem in order to satisfy the requirement of optimal performance in control synthesis tasks. This motivates our research.

In this paper, we apply symbolic regression methods to solve the problem of synthesis of a stabilization system. These methods also belong to the class of machine learning methods, but unlike neural networks, they allow us to search for not only for parameters but also for the optimal structure of the control function. These methods use evolutionary optimization algorithms for the structural-parametric search for the control function, basing it directly on the value of the quality functional.

A wheeled mobile robot is considered as a control object.

The paper develops an applied software implementation of the robot control system. To create it, the most popular robotic operating system ROS is used today. It provides the ability to work with all aspects of the control system, including hardware abstraction, low-level control, message passing between processes, and package management. The developed software systems were tested in the Gazebo simulation environment, which is one of the most popular robotic simulators due to its compatibility with ROS. Gazebo is a 3D simulator that aims to model a robot in a way that gives you a close substitute for how the robot would behave in a real physical environment. Gazebo has a fairly reliable simulation of physics and various physical phenomena, takes into account the influence of forces, and also has a large number of plug-ins for simulating the operation of sensors, such as lidars or cameras. Due to these facts, most developers of control systems for robotic systems around the world use this ROS/Gazebo bundle as a standard for testing the developed control algorithms [18,19].

In this work, we used the ready-made Gazebo model ROSbot 2.0 integrated into ROS [20]. As a position source, a plugin was used that gives the true coordinates of the robot. The robot model in the simulator is shown in Figure 2.

For the selected object, the problem of synthesizing the stabilization system was successfully solved by machine learning based on symbolic regression via the network operator method [21].

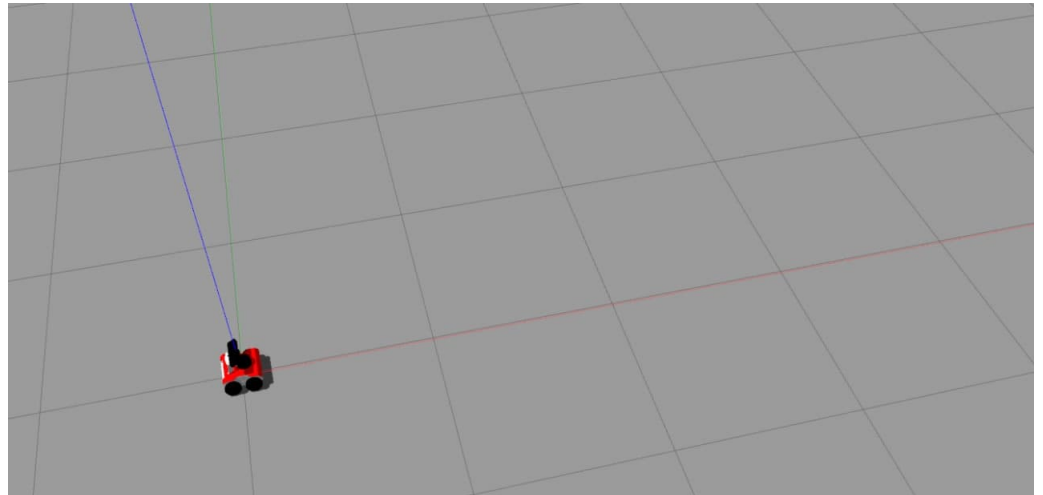


Figure 2. ROSbot in Gazebo simulator.

2. Problem Statement of the Stabilization System Synthesis

The main goal of the introduction of the stabilization system is to provide a stability property for the object in some domain $X_0 \subseteq \mathbb{R}^n$.

Let us be given a mathematical model of a control object. This model can be derived from physical laws or identified by some machine learning technique [22]. Generally, this model is described by a system of ordinary differential equations with a free control vector on the right hand side

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

where the state of the object is described by $\mathbf{x} \in \mathbb{R}^n$, and control by $\mathbf{u} \in U \subseteq \mathbb{R}^m$; U is a compact set, $m \leq n$,

$$\begin{aligned} \mathbf{x} &= [x_1 \dots x_n]^T, \\ \mathbf{u} &= [u_1 \dots u_m]^T, \quad m \leq n, \\ \mathbf{f}(\mathbf{x}, \mathbf{u}) &= [f_1(\mathbf{x}, \mathbf{u}) \dots f_n(\mathbf{x}, \mathbf{u})]^T. \end{aligned} \tag{2}$$

An area of initial conditions is given

$$X_0 \subseteq \mathbb{R}^n. \tag{3}$$

It is necessary to find a control function in the form

$$\mathbf{u} = \mathbf{h}(\mathbf{x}^* - \mathbf{x}), \tag{4}$$

where \mathbf{x}^* is a fixed point in the state space, which becomes an equilibrium point of the differential equation

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

where control function $\mathbf{h}(\mathbf{x}^*, \mathbf{x}) = [h_1(\mathbf{x}^* - \mathbf{x}) \dots h_m(\mathbf{x}^* - \mathbf{x})]^T : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ has the following properties:

$$\begin{aligned} \mathbf{h}(\mathbf{x}^*, \mathbf{x}) &\in U \subseteq \mathbb{R}^m, \quad \forall \mathbf{x}^* \in X_0, \exists \tilde{\mathbf{x}}(\mathbf{x}^*) \text{ such that} \\ \mathbf{f}(\tilde{\mathbf{x}}(\mathbf{x}^*), \mathbf{h}(\mathbf{x}^*, \tilde{\mathbf{x}}(\mathbf{x}^*))) &= 0, \\ \det(\mathbf{A} - \lambda \mathbf{E}) &= (-1)^n (\lambda - \lambda_1) \cdot \dots \cdot (\lambda - \lambda_n) = \prod_{j=1}^n (\lambda - \lambda_j) = 0, \\ \lambda_j &= \alpha_j + i\beta_j, \quad j = 1, \dots, n, \\ \alpha_j &< 0, \quad j = 1, \dots, n, \quad i = \sqrt{-1}, \\ \mathbf{A} &= \frac{\partial \mathbf{f}(\tilde{\mathbf{x}}(\mathbf{x}^*), \mathbf{h}(\mathbf{x}^*, \tilde{\mathbf{x}}(\mathbf{x}^*)))}{\partial \mathbf{x}}, \\ \mathbf{E} &= \text{diag}(\underbrace{1, \dots, 1}_n). \end{aligned} \tag{6}$$

The properties (6) indicate that $\forall \mathbf{x}^* \in X_0$ for the system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^*, \mathbf{x}))$ means that there is always a stable equilibrium point $\tilde{\mathbf{x}}(\mathbf{x}^*) \in \mathbb{R}^n$. Additionally, the equilibrium point possesses attractor properties, since near this point all solutions converge.

Computationally, to provide a stability property to the equilibrium point $\tilde{\mathbf{x}}$, the synthesis problem (1)–(4) is solved with the terminal point $\mathbf{x}^f = \tilde{\mathbf{x}}$, the initial domain $X_0 \subset X$, and the quality criterion

$$J = \max\{t_{f,1}, \dots, t_{f,K}\} + a_1 \sum_{i=1}^K \Delta_{f,i} \rightarrow \min, \tag{7}$$

where a_1 is a weight coefficient,

$$\Delta_{f,i} = \left\| \mathbf{x}^f - \mathbf{x}(t_{f,i}, \mathbf{x}^{0,i}) \right\|, \tag{8}$$

$t_{f,i}$ is a time of achievement of the terminal position from the initial condition $\mathbf{x}^{0,i}$ of the set of initial conditions $X_0 = \{\mathbf{x}^{0,1}, \dots, \mathbf{x}^{0,K}\}, i \in \{1, \dots, K\}$,

$$t_{f,i} = \begin{cases} t, & \text{if } t < t^+ \text{ and } \Delta_{f,i} \leq \varepsilon \\ t^+ & \text{otherwise} \end{cases}, \tag{9}$$

t^+ and ε are given positive values, and $\mathbf{x}(t, \mathbf{x}^{0,i})$ is a partial solution of the system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{h}(\mathbf{x}^f - \mathbf{x})), \tag{10}$$

for initial conditions $\mathbf{x}(t_0) = \mathbf{x}^{0,i}, i \in \{1, \dots, K\}$,

$$\left\| \mathbf{x}^f - \mathbf{x} \right\| = \sqrt{\sum_{i=1}^n (x_i^f - x_i)^2}. \tag{11}$$

Since we are solving the problem of synthesizing a stabilization system using machine learning, we need machine confirmation of the achievement of the desired properties. Let us introduce the following definition of a machine criterion for a differential equation system to obtain some property. To define the property of the whole system (1), it is enough to set a quantity K of partial solutions that obtain this property.

Definition 1. *If D experiments are carried out, and in every i experiment K_i partial solutions of the differential equation perform the required property from any $M_i \geq K_i$ randomly selected initial conditions from the initial domain,*

$$\lim_{D \rightarrow \infty} \sum_{i=1}^D \frac{K_i}{M_i} \rightarrow 1, \tag{12}$$

and so the existence of this property for the differential equation in this domain is proven by a machine.

In other words, as the number of experiments increases, the probability of a “bad” event, when the system does not have the desired property, tends to zero. From a mathematical point of view, this means that all private solutions for a domain of initial conditions have this property except for solutions for a subset of a zero measure.

Based on the proposed formulation of the problem statement, let us consider in the next section a solution of the stabilization system synthesis problem via a machine learning approach of symbolic regression for a mobile robot in a Gazebo simulation environment.

3. Stabilization System Synthesis for ROSbot in Gazebo

We consider the ROSbot 2.0 virtual robot implemented in the Gazebo physical simulation environment. The robot is a platform on four non-rotating wheels around the vertical axis. An electric motor is attached to each wheel. A differential control scheme is used: the robot moves forward and backward by applying the same voltage to all four electric motors; turning the robot to the right or left is carried out by supplying more voltage to the electric motors of the left or right wheels, respectively.

The robot motion model is described by the following system of differential equations [23]

$$\begin{aligned}\dot{x}_1 &= 0.5(u_1 + u_2) \cos(x_3), \\ \dot{x}_2 &= 0.5(u_1 + u_2) \sin(x_3), \\ \dot{x}_3 &= 0.5(u_1 - u_2),\end{aligned}\quad (13)$$

where $\mathbf{x} = [x_1 \ x_2 \ x_3]^T$ is a vector of state, $\mathbf{u} = [u_1 \ u_2]^T$ is a control vector.

Physical control of the robot in Gazebo is implemented using two signals: u^v — the desired linear speed; and u^ω — the desired angular velocity. As we are aiming to stabilize the system, it can be assumed that control signals are completely directly transmitted to the system $v = u^v$, $\omega = u^\omega$. In this case, the (13) equations are converted to the following:

$$\begin{cases} \dot{x}_1 = u^v \cos(x_3), \\ \dot{x}_2 = u^v \sin(x_3), \\ \dot{x}_3 = u^\omega. \end{cases}\quad (14)$$

The stabilization system is synthesized in advance and then is programmed into an onboard computer. For the solution, a machine learning approach based on symbolic regression was chosen. Symbolic regression allows you to search for a solution to the problem without training data, simply according to the formal statement, relying in the search process on the criterion of minimizing the functional. Moreover, this approach is universal and can be equally applicable to models of any kind, including non-linear models or models in the form of neural networks.

The network operator method [21] was used in the calculations. This symbolic regression method is good because it uses the principle of variation of the basic solution, which significantly speeds up the process of finding a solution that is close to optimal. The method encodes possible solutions as a square upper triangular matrix. It is in the form of a matrix that describes the sequence of calculation of the control function that the resulting stabilization system is placed in the on-board computer.

In the calculations, the following parameters were set.

The control values were constrained $-10 \leq u_i \leq 10$, $i = 1, 2$.

The initial domain was defined by 26 elements:

$$\begin{aligned}\bar{X}_0 = \{ & [-2.5 \ -2.5 \ -5\pi/12]^T, [-2.5 \ -2.5 \ 0]^T, [-2.5 \ -2.5 \ 5\pi/12]^T, \\ & [-2.5 \ 0 \ -5\pi/12]^T, [-2.5 \ 0 \ 0]^T, [-2.5 \ 0 \ 5\pi/12]^T, [-2.5 \ 2.5 \ -5\pi/12]^T, \\ & [-2.5 \ 2.5 \ 0]^T, [-2.5 \ 2.5 \ 5\pi/12]^T, [0 \ -2.5 \ -5\pi/12]^T, [0 \ -2.5 \ 0]^T, \\ & [0 \ -2.5 \ 5\pi/12]^T, [0 \ 0 \ -5\pi/12]^T, [0 \ 0 \ 5\pi/12]^T, [0 \ 2.5 \ -5\pi/12]^T, \\ & [0 \ 2.5 \ 0]^T, [0 \ 2.5 \ 5\pi/12]^T, [2.5 \ -2.5 \ -5\pi/12]^T, [2.5 \ -2.5 \ 0]^T, \\ & [2.5 \ -2.5 \ 5\pi/12]^T, [2.5 \ 0 \ -5\pi/12]^T, [2.5 \ 0 \ 0]^T, [2.5 \ 0 \ 5\pi/12]^T, \\ & [2.5 \ 2.5 \ -5\pi/12]^T, [2.5 \ 2.5 \ 0]^T, [2.5 \ 2.5 \ 5\pi/12]^T \}.\end{aligned}\quad (15)$$

The stabilization point was chosen as

$$\mathbf{x}^* = [x_1^* \ x_2^* \ x_3^*]^T = [0 \ 0 \ 0]^T.\quad (16)$$

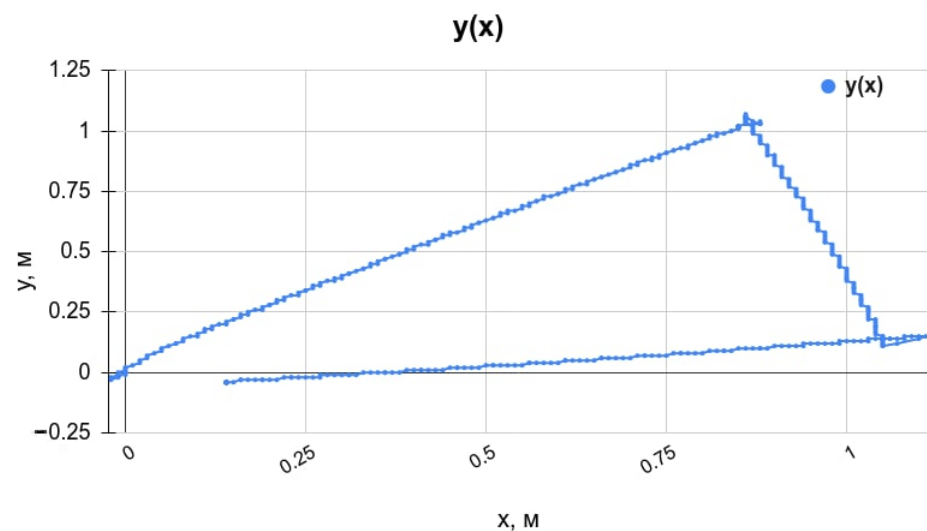


Figure 3. Rosbot trajectory on XY plane under network operator control.

In general we noticed that the robot's movements were stable and predictable.

5. Discussion

In this paper, a Rosbot controller synthesized by a computer algorithm using the network operator method has been verified to work in a Gazebo simulated environment. The synthesis algorithm required only a mathematical model of the robot as an input. The mathematical structure of the control function has been found automatically without any human input. The synthesized function has been verified to be able to control Rosbot in a stable and predictable way.

The presented numerical approach to the synthesis of a stabilization system by a robot, implemented on the Gazebo simulator robot, is a universal machine learning approach for the synthesis of control systems, and opens up broad prospects for its use in various technical problems. The main advantage of the approach is that it is not tied to the type of control object model, and allows you to find the feedback control function in automatic mode using the symbolic regression algorithm.

Author Contributions: Conceptualization, E.S.; methodology, E.S.; software, Y.R.; validation, Y.R.; formal analysis, E.S. and Y.R.; investigation, Y.R.; data curation, Y.R.; writing—original draft preparation, E.S.; writing—review and editing, E.S.; visualization, Y.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Qi, Z.; Shi, Q.; Zhang, H. Tuning of Digital PID Controllers Using Particle Swarm Optimization Algorithm for a CAN-Based DC Motor Subject to Stochastic Delays. *IEEE Trans. Ind. Electron.* **2020**, *67*, 5637–5646. [[CrossRef](#)]
2. Castillo-Zamora, J.J.; Camarillo-Gómez, K.A.; Pérez-Soto, G.I.; Rodríguez-Reséndiz, J. Comparison of PD, PID and Sliding-Mode Position Controllers for V-Tail Quadcopter Stability. *IEEE Access* **2018**, *6*, 38086–38096. [[CrossRef](#)]
3. Hu, F.; Zeng, C.; Zhu, G.; Li, S. Adaptive Neural Network Stabilization Control of Underactuated Unmanned Surface Vessels with State Constraints. *IEEE Access* **2020**, *8*, 20931–20941. [[CrossRef](#)]
4. Li, S.; Ahn, C.K.; Guo, J.; Xiang, Z. Neural Network-Based Sampled-Data Control for Switched Uncertain Nonlinear Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 5437–5445. [[CrossRef](#)]

5. Wang, T.-Y.; Chang, C.-D. Hybrid Fuzzy PID Controller Design for a Mobile Robot. In Proceedings of the 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 13–17 April 2018; pp. 650–653. [CrossRef]
6. Chwa, D.; Boo, J. Adaptive Fuzzy Output Feedback Simultaneous Posture Stabilization and Tracking Control of Wheeled Mobile Robots with Kinematic and Dynamic Disturbances. *IEEE Access* **2020**, *8*, 228863–228878. [CrossRef]
7. Kolmanovskii, V.B.; Lafay, J.-F.; Richard, J.-P. Riccati equations in stability theory of difference equations with memory. In Proceedings of the 1999 European Control Conference (ECC), Karlsruhe, Germany, 31 August–3 September 1999; pp. 3630–3636. [CrossRef]
8. Khalil, H.K. *Nonlinear Systems*; Prentice Hall: Hoboken, NJ, USA, 2002.
9. Kolesnikov, A.A.; Kuz'menko, A.A. Backstepping and ADAR Method in the Problems of Synthesis of the Nonlinear Control Systems. *Mekhatronika Avtom. Upr.* **2016**, *17*, 435–445. [CrossRef]
10. Podvalny, S.L.; Vasiljev, E.M. Analytical synthesis of aggregated regulators for unmanned aerial vehicles. *J. Math. Sci.* **2019**, *239*, 135–145. [CrossRef]
11. Bellman, R.E. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
12. Werbos, P.J. Approximate dynamic programming for real-time control and neural modeling, In *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*; White, D.A., Sofge, D.A., Eds.; Van Nostrand Reinhold: New York, NY, USA, 1992; Chapter 13.
13. Liu, D.; Wei, Q.; Wang, D.; Yang, X.; Li, H. *Adaptive Dynamic Programming with Applications in Optimal Control*; Springer: Cham, Switzerland, 2017.
14. Al-Tamimi, A.; Lewis, F.L.; Abu-Khalaf, M. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Trans. Syst. Man Cybern. B Cybern.* **2008**, *38*, 943–949. [CrossRef]
15. Si, J.; Wang, Y.-T. Online learning control by association and reinforcement. *IEEE Trans. Neural Netw.* **2001**, *12*, 264–276. [CrossRef] [PubMed]
16. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
17. Modares, H.; Lewis, F.L. Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning. *Automatica* **2014**, *50*, 1780–1792. [CrossRef]
18. Lavrenov, R.; Magid, E.; Matsuno, F.; Svinin, M.; Suthakorn, J. Development and Implementation of Spline-based Path Planning Algorithm in ROS/Gazebo Environment. *SPIIRAS Proc.* **2019**, *18*, 57–84. [CrossRef]
19. Zhang, B.; Liu, P. Control and benchmarking of a 7-DOF robotic arm using Gazebo and ROS. *Peer J. Comput. Sci.* **2021**, *7*, e383. [CrossRef] [PubMed]
20. Rosbot_description. Available online: https://github.com/husarion/rosbot_description (accessed on 14 May 2023).
21. Diveev, A.; Shmalko, E. Symbolic Regression Methods. In *Machine Learning Control by Symbolic Regression*; Springer: Cham, Switzerland, 2021. [CrossRef]
22. Shmalko, E.; Rumyantsev, Y.; Baynazarov, R.; Yamshanov, K. Identification of Neural Network Model of Robot to Solve the Optimal Control Problem. *Inform. Autom.* **2021**, *20*, 1254–1278. [CrossRef]
23. Diveev, A.I.; Shmalko, E.Y. Machine-Made Synthesis of Stabilization System by Modified Cartesian Genetic Programming. *IEEE Trans. Cybern.* **2022**, *52*, 6627–6637. [CrossRef] [PubMed]
24. Network Operator. Available online: https://github.com/urock/network_operator (accessed on 14 May 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.