Comparative Analysis of Traditional Control Methods, Reinforcement Learning, and Neuroevolution in Optimizing Binary Control Systems

Matthew Fleischman
FLSMAT002@myuct.ac.za
University of Cape Town
South Africa

Krupa Prag*
krupa.prag@uct.ac.za
University of Cape Town
South Africa

ABSTRACT

The increasing complexity of modern nonlinear control systems has made traditional model-based control insufficient. Development of machine learning and big data in recent years has facilitated the emergence of model-free control which does not rely on mathematical models of the system's dynamics. Model-free methods allow for more optimal control of nonlinear systems when compared to model-based methods. This research reviews the current literature on three model-free control methods: Q-learning, Proximal Policy Optimisation, and Neuroevolution of Augmenting Topologies. These three methods, as well as traditional PID control, are considered in the context of optimising the inverted pendulum problem and DC-to-DC voltage conversion. A side-by-side comparison of the methods is also drawn.

KEYWORDS

Neuroevolution, NEAT, Q-Learning, Proximal Policy Optimization, Inverted Pendulum, DC-to-DC Voltage Conversion, Binary Control Systems, Data-Driven Control,

1 INTRODUCTION

Humans possess the ability to manipulate their environment. The field of control theory was developed to command, manage, direct, and regulate the behaviour of systems. The evolution of control methods throughout history underpins many of Humanity's technological advancements [9].

One of the first automatic control devices, the water clock, was invented in Alexandria in the third century BC. A controlled stream of water was used to fill a reservoir at a constant rate, allowing the ancient Greeks to keep track of time. During the Industrial Revolution, engines, boilers, and furnaces needed to be controlled to operate efficiently; therefore, control devices such as pressure valves and temperature regulators were created. James Watt developed the governor for the steam engine, which is thought to be one of the first negative feedback devices. [15]. In 1868, J.C. Maxwell formulated a mathematical model for Watt's flyball governor, effectively giving birth to control theory [4].

In the past, control systems were model-based; a physical model describing the system's dynamics was required to control it. Model-based control works well for systems that rely on linear differential equations; however, when faced with nonlinear systems, simplifying approximations of the system's dynamics are required. These approximations reduce the accuracy of the model and the performance of the control method [17]. The technological advancements of modern society have led to increasingly complex industrial processes. As a result, many systems now rely on nonlinear dynamics. Fortunately, the emergence of computers, machine learning, and big data in recent decades has enabled the development of data-driven control (DDC) [12]. Not only can DDC be used to enhance model-based control, but it has also led to the development of model-free control — an application of the DDC framework that does not require a mathematical model of the system dynamics [17]. Today, there is a plethora of literature on model-free control methods, of which this review considers three.

To compare different control methods, they should be applied to control problems, and the performance and behaviour of the systems should be observed. This paper considers the application of model-based and model-free control methods to nonlinear binary control problems. Binary control uses devices that can exist in one of two distinct states. These states are often represented as "on" or "off", "true" or "false", "left" or "right", etc. The most popular benchmark application for nonlinear binary control is the inverted pendulum problem. The aim of this problem is to balance an inverted pendulum in an upright position for a sufficiently long time. The inverted pendulum problem has complex nonlinear dynamics; however, it is well-researched and is easy to understand and simulate. For more information regarding the mathematical modelling of the inverted pendulum, see [5]. The second considered application is DC-to-DC voltage conversion. This involves using buck-boost converters to transform the input voltage to a desired higher or lower output voltage. The raising and lowering states of the buck-boost converters represent the binary states. Voltage conversion is a practical and beneficial application of modern control theory, as it is widely used in power electronics and energy systems [16].

2 LITERATURE REVIEW

This literature review first analyses traditional control methods, focusing on PID controllers in section 2.1. Next, three model-free DDC methods are considered in section 2.2: Q-learning, Proximal Policy Optimization (PPO), and Neuroevolution of Augmenting Topologies (NEAT). Each control method has its basic history and methodology discussed. They are then observed and evaluated in the context of optimal inverted pendulum control and DC-to-DC voltage conversion. Finally, a side-by-side comparison of the control methods is made in section 2.3.

2.1 Traditional Control Methods

Proportional-integral-derivative (PID) controllers lie at the heart of traditional control methods. Aspects of PID control have existed

*Supivisor

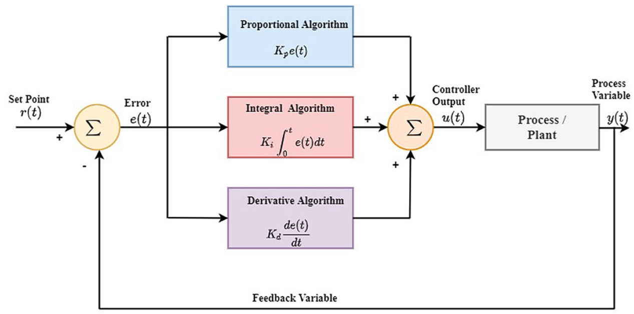

Figure 1: Diagram of PID control [4]

since the mid-1800s. In 1911, Elmer Sperry developed the first real PID-type controller, which was used to design automatic steering systems for the US Navy. A century later, PID controllers constitute around 90-95% of all control loops. Their popularity is owed to their simple nature and comprehensive literature [4].

PID control is traditionally model-based and consists of three contributions to the controlling mechanism. The proportional term adjusts the output proportionally to the error between the desired setpoint and the actual process value, acting promptly to match input changes. The integral term responds to accumulated error over time, gradually eliminating any remaining deviation between the setpoint and the process value. Finally, the derivative term reacts to the rate of change of error, helping stabilise the system by anticipating future deviation and adjusting the output accordingly [4]. Figure 1 illustrates the three contributions of PID control. The gain coefficients for the proportional, integral, and derivative terms are represented by K_p, K_i, and K_d, respectively. The calculated error at time t is represented by *e(t)*.

The PID control methodology was implemented in [18] to stabilise an inverted pendulum on a moving cart. The objective of the cart was to travel a distance of 0.1 m. Two PID controllers were used. A cart PID controller managed the cart position x, and an angle PID controller managed the pendulum angle θ. The inverted pendulum on a moving cart was simulated for both perfect and noisy environments. For the ideal environment, the PID controllers stabilised the inverted pendulum quickly and smoothly. The PID controllers created minor oscillations in the noisy environment but performed well. These results testify to the robustness and effectiveness of PID control. The experiment then combined PID control with a linear quadratic regulator (LQR). LQR control is another example of traditional model-based control. This review does not discuss its details; however, it should be noted that the PID+LQR control method was more effective than PID alone. It can be said that combining traditional control methods has the potential to improve model-based control. Lastly, the paper proposed using genetic algorithms (GAs) to enhance the PID controllers. Considering that GAs can be regarded as a form of DDC, it is implied that the hybrid of DDC and traditional control methods might provide optimal inverted pendulum control.

A step-by-step guide on the design of PID controllers for inverted pendulum control is given in [24]. This paper demonstrates that when creating PID controllers for inverted pendulum control, a lot of time needs to be spent on understanding the mathematics of the system. It should be noted, however, that the resultant PID controllers showed robustness and good dynamic performance.

The efficiency of a simulated DC-to-DC boost converter, which used a closed-loop PID controller, was analysed in [2]. A boost converter steps up an input voltage to a required output voltage. The simulation was done using MATLAB/Simulink. Using the PID controller, the system obtained an efficiency of over 90%; therefore, the DC-to-DC converter circuit, controlled by the PID controller, provided good efficiency for voltage conversion.

PID methods, although model-based, still show relevance in modern nonlinear control systems. PID controllers can offer adequate performance in inverted pendulum control and DC-to-DC voltage conversion as discussed above; however, the majority of literature believes that PID control is limited by uncertainties, nonlinearities, process dynamics and disturbances [27].

2.2 Model-Free Data-Driven Control Methods

Given the limitations of model-based control systems, this section explores model-free DDC methods that do not rely on a mathematical model of the system. These should allow for better performance when faced with nonlinearities and disturbances (noisy environments). DDC uses input and output data collected during the operation of a system to design controllers. DDC learns from data rather than relying on a predefined system model, which makes DDC particularly useful for complex systems where creating an accurate mathematical model is challenging [17]. Model-free control is an application of the data-driven approach, which learns from observed data and makes decisions based on the system's behaviour without explicitly knowing the system's dynamics. This is achieved using algorithms that can learn and adapt over time as more data is collected [13].

Reinforcement Learning (RL) is an example of a popular learning-based framework often used in model-free control [17]. This section considers two model-free RL algorithms, Q-learning and PPO. RL refers to a category of machine learning problems where an autonomous agent interacts with an environment. An agent is anything that can act on its environment. The behaviour of the agent is determined by its policy. Policies map the perceived states of the environment to the possible actions that the agent can take. The agent receives information, updates its policy, and takes action accordingly. In response, the environment provides a reward signal, which can be positive or negative. The primary objective for the agent is to maximise the expected cumulative reward throughout its interactions [6].

RL shows promise in the field of model-free control by directly earning from its interactions with the environment; however, applying RL control methods to systems with large state spaces, or systems where only partial observations are possible proves challenging. Neuroevolution has the potential to handle large state spaces by evolving populations of neural networks to find those best suited to the task. Neuroevolution can also deal with environments where only partial observations are possible by selecting networks that perform well under these conditions [8]. Within the broader field of neuroevolution, there are many different evolutionary methods. NEAT is considered in subsection 3.3.

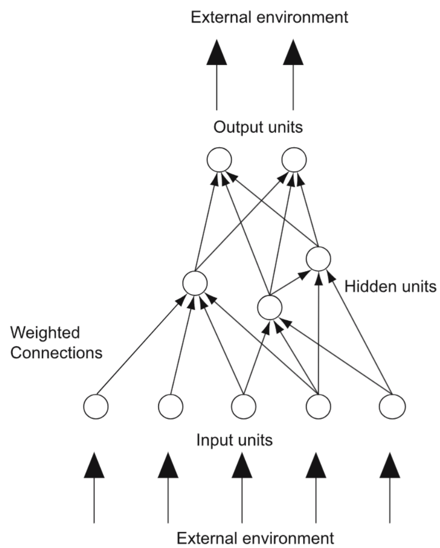

Figure 2: A generic neural network architecture [26]

Neural networks lie at the heart of neuroevolution and are inspired by the functioning of the human brain. They are composed of interconnected nodes, mimicking the neurons. A neural network consists of three types of layers: An input layer, one or more hidden layers, and an output layer. Figure 2 shows the basic structure of the three layers of a neural network. Each node has an associated threshold and weight. If the output of a node exceeds its specified threshold value, that node is activated, transmitting data to the next layer of the network [26]. It should be noted that neural networks are often applied to RL algorithms to increase performance. The combination of neural networks and RL is known as deep RL, which is effective in control systems. PPO is inherently a form of deep RL; however, classical Q-learning does not use neural networks. Deep Q-learning is not covered in this review.

2.2.1 ***Q-Learning****.* Q-learning is one of the most popular RL approaches. In 1989, Chris Watkins proposed Q-learning, one of the first model-free RL algorithms, in his thesis "Learning from Delayed Rewards" [25]. Q-learning is versatile and has many applications in various artificial intelligence problems. It uses an off-policy method, which means that two policies are used: The greedy policy that the agent evaluates and updates, and the ϵ-greedy policy used to perform actions on the system. This separation provides flexibility and allows the agent to learn from suboptimal actions and explore different strategies, mitigating the dilemma of making wrong choices. Like other RL methods, Q-learning uses a reward system to encourage actions that result in positive environmental changes. The expected cumulative reward is determined by the Q-value function [14]. Q-learning selects actions that greedily maximise the Q-value. The update equation is fundamental to Q-learning's methodology and is as follows:

$$Q(s,a) \leftarrow Q(s',a') + \alpha[r + \gamma maxQ(s,a) - Q(s',a')] \tag{1}$$

In equation 1,

- $Q(s',a')$ is the Q-value for the previous state and action.
- α is the learning rate.
- r is the reward.
- γ is the discount factor (reduction rate of reward over time).
- $maxQ(s,a)$ is the maximum Q-value for the new state s and its possible actions.

Q-learning was used in two steps to solve the inverted pendulum problem in [7]. First, the pendulum was swung up into the inverted position. Second, the upright position was controlled and maintained. They found that splitting the problem into these two tasks was the main reason for their success in solving the inverted pendulum problem. The agent learned from two environments and retained the most valuable information from each. It was also found that exploration (trying and testing different actions) was most important when the pendulum was far from the upright position, and exploitation (using actions that are known to yield reward) was most important when the pendulum was adequately close to the upright position. The entirety of this experiment was performed in simulation. Another paper also used simulation to apply a Q-learning approach to balance an inverted pendulum; however, the simulation was intended to train the control policy, which was then transferred to a real-world robot arm that balanced a real-world inverted pendulum [20]. Using a simulation mitigated the risk of damage to hardware, reduced the time needed for the training process, and allowed the authors to experiment with different initial starting positions of the pendulum. The robot arm could balance the pendulum for 5 seconds before failing due to a limitation of the range of motion of the flange of the arm.

Using simulations of the inverted pendulum to train Q-learning policies may seem contradictory. Being model-free is one of the main advantages of Q-learning; however, a model of the system's dynamics is required to simulate the environment in which the Q-learning policies are trained. This should not discredit the combined use of simulation and Q-learning. The simulations can be thought of as giving the Q-learning policies a headstart. Approximations about the system's nonlinear dynamics can afford to be made in simulation. Then, once the policies are transferred to real-world systems, Q-learning methods can again be used to iron out the inaccuracies caused by the linearisation of the dynamics.

It was attempted to improve the power efficiency of a dual-active-bridge (DAB) DC-to-DC converter through the use of Q-learning in [23]. An agent was first trained offline to optimise the modulation strategy and was then trained online to provide real-time control decisions to the DAB DC-to-DC converter. Power efficiency was used as a metric for success. The use of Q-learning removed the inconvenient process of manually choosing the optimal operation mode. The control decisions provided by the Q-learning algorithm enabled excellent efficiency during voltage conversion. When contrasted with the traditional SPS modulation, there was a noticeable increase in the measured maximum efficiency by 1.6% at approximately 600 W. Under light load conditions, an efficiency enhancement of 7.5% was observed.

2.2.2 ***Proximal Policy Optimization (PPO)****.* PPO, a deep RL method, was introduced by Schulman *et al.* in their 2017 paper "Proximal Policy Optimization Algorithms" [21]. Standard policy gradient methods often suffer from unfavourable policy fluctuations. PPO is a gradient policy method that employs a proximal objective function to control policy changes during updates. PPO performs multiple passes (epochs) over the given data, optimising the policy parameters to maximise the expected reward. A defining characteristic of PPO is the incorporation of a clipping function into its objective function. The clipping function restricts the update step size, discouraging updates that deviate too much from the current policy, helping to mitigate bad policy fluctuations and maintain a balance between exploration and exploitation [21].

A rotary inverted pendulum (RIP) was controlled in both simulation and hardware environments using PPO methods in [3]. The control approach comprised four parts: modelling the pendulum, interfacing with hardware, defining the environment, and designing the agent. Two agents were used. Firstly, a Deep Deterministic Policy Gradient (DDPG) agent, which does not require deep knowledge of conventional control theory, was proposed for the swing-up control action of the pendulum. DDPG is another example of a model-free RL algorithm. Secondly, a PPO agent was used to train the mode selection operation. The paper noted that the DDPG-PPO algorithm was sensitive to the choice of hyperparameters, such as learning rates and discount factors, and that the process of choosing these hyperparameters was time-consuming. DDPG-PPO provided good control of the RIP in both simulation and real-world environments and outperformed classical PID controllers. The blended use of the PPO and DDPG model-free algorithms motivates the idea of combining the advantages of various methods to achieve optimal results.

PPO was used to optimise buck-boost converters in [16]. The time it took to reach the desired voltage level, known as the settling time, and the controller's capacity to sustain a steady output voltage were used as metrics to gauge the buck-boost converter's quality. The performance of the PPO controller was compared to the performance of a traditional PI controller, along with hybrids of PPO and PID controllers. The shortest settling time for the boost cases was achieved by the PPO controller, which was around half the setting time of the PI controller. The hybrid methods showed the greatest stability in sustaining a steady output voltage. Another paper, [11], optimised a pre-existing feedback controller's settings by using PPO. The DC-DC buck-boost converter showed excellent performance during transient states in voltage responses due to the adaptiveness of the methods. Once again, PPO was used in conjunction with other algorithms and controllers, motivating the idea of optimal performance through hybrid methods.

2.2.3 ***Neuroevolution of Augmenting Topologies (NEAT)***. NEAT follows the basic principles of Neuroevolution; however, it is unique because of its ability to evolve and retopologise its neural networks. NEAT was first introduced in a paper titled "Evolving Neural Networks through Augmenting Topologies" by Kenneth Stanley and Risto Miikkulainen in 2002 [22]. The aim was to improve the efficiency and effectiveness of Neuroevolution by allowing the evolution of both the topology and weights. It was argued by Gruau *et al.* (1996) that this approach would reduce the need for human intervention and would save time [10]. NEAT starts with a simple network, which becomes increasingly complex over the generations. Additional nodes and connections are created, which allows NEAT to navigate the search space more effectively. One can think of the neural network as being self- optimising. A key feature of NEAT is its ability to divide a population of networks into species based on the similarity of their structures. This speciation protects innovation and allows topologies of the neural networks to be optimised without the risk of being outcompeted by one another [22].

There is little research and documentation on the application of NEAT in inverted pendulum control compared to RL methods. Literature pertaining to the use of NEAT in DC-to-DC voltage conversion is not available. These observations suggest that NEAT may be underrepresented in the field of control problems compared to other methodologies. Further research and exploration are needed to understand the potential of NEAT in this domain fully.

Multiple neuroevolution methods, as well as a model-based method, were compared in optimising inverted pendulum control in [19]. One of the neuroevolution methods was NEAT. A real-world inverted pendulum was used, and noise was injected into the system. Along with NEAT, two other neuroevolution methods were considered: Feed-Forward Neural Networks (FFNNs) and Elman Recurrent Neural Networks (ERNNs). An advantage of NEAT was that a fixed network topology did not have to be decided upon before network training. NEAT showed similar performance to FFNNs and was outperformed by ERNNS. The NEAT algorithm did not perform as well as expected; however, it was noted that the networks did not train as well as they could have. Another paper that only used NEAT to stabilise an inverted pendulum observed that the stability and regularity of the controller were highly dependent on the sophistication of the NEAT algorithm's fitness function [1]. The paper also noted a very steep learning curve for the population of agents. The lacklustre performance of NEAT in both of the papers mentioned above gives reason for NEAT's lack of application in the context of inverted pendulum control and DC-to-DC voltage conversion.

2.3 Comparison of Control Methods

The following should be considered when comparing different control methods: Performance, tuning, maintenance, and ease of use [27]. An obstacle to making comparisons of control methods through reviewing the literature is that different papers use different metrics for quality evaluation; therefore, it is necessary to consult research that directly compares two or more methods. The majority of literature uses model-based methods as benchmarks for model-free methods. There exists less research comparing model-free methods to one another in the context of binary control systems. This gap in literature should be filled to understand model-free control better.

Based on the current literature, PPO seems to be the most promising of the three model-free methods. PPO is newer than Q-learning and NEAT and already possesses a wealth of research. NEAT has the least literature and shows mediocre results compared to the other control methods in its application to inverted pendulum control and DC-to-DC voltage conversion. It is easy to understand why PID control has retained its popularity. The model-based nature of PID restricts its effectiveness and efficiency, but when combined with modern model-free methods, PID methods are still applicable in the control of nonlinear systems.

Hybrid methods were mentioned in many of the papers discussed in this review [5, 11, 18]. Hybrid methods combine two or more model-free or model-based methods to achieve optimal control of the considered system. The literature surrounding hybrid control methods is not discussed in this review in order to maintain a reasonable scope; however, it is worth noting that hybrid control often provides promising results and should be considered when seeking optimal control of a nonlinear system.

3 CONCLUSIONS

This research provided a comprehensive literature review on various control methods and their applications to the inverted pendulum problem and DC-to-DC voltage conversion. A traditional model-based method, PID, and three model-free methods, Q-learning, PPO, and NEAT, were examined. Q-learning is a value-focused RL method, and PPO is a policy-focused RL method. NEAT is a type of neuroevolution algorithm that optimises its neural networks. PPO was considered to be the most promising method in terms of optimal binary control of nonlinear systems, and NEAT was considered to be the least promising. PID was found applicable to nonlinear control when combined with other model-free methods. NEAT requires further research in the field of binary control theory.

REFERENCES

[1] Awjin Ahn and Caleb Cochrane. 2014. Using NEAT to Stabilize an Inverted Pendulum. (2014).

[2] R Arulmurugan and N Suthanthira Vanitha. 2012. Optimal design of DC to DC boost converter with closed loop control PID mechanism for high voltage photovoltaic application. *International Journal of Power Electronics and Drive Systems* 2, 4 (2012), 434.

[3] Rajmeet Singh Bhourji, Saeed Mozaffari, and Shahpour Alirezaee. 2024. Reinforcement learning ddpg–ppo agent-based control system for rotary inverted pendulum. *Arabian Journal for Science and Engineering* 49, 2 (2024), 1683–1696.

[4] Rakesh P Borase, DK Maghade, SY Sondkar, and SN Pawar. 2021. A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control* 9 (2021), 818–827.

[5] Olfa Boubaker. 2013. The inverted pendulum benchmark in nonlinear control theory: a survey. *International Journal of Advanced Robotic Systems* 10, 5 (2013), 233.

[6] Damien Ernst and Arthur Louette. 2024. Introduction to reinforcement learning. (2024).

[7] Alessio Ghio and Oscar E. Ramos. 2019. Q-learning-based Model-free Swing Up Control of an Inverted Pendulum. In *2019 IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*. 1–4. https://doi.org/10.1109/INTERCON.2019.8853619

[8] Faustino Gomez, Jürgen Schmidhuber, and Risto Miikkulainen. 2006. Efficient non-linear control through neuroevolution. In *European Conference on Machine Learning*. Springer, 654–662.

[9] Graham Clifford Goodwin, Stefan F Graebe, Mario E Salgado, et al. 2001. *Control system design*. Vol. 240. Prentice Hall Upper Saddle River.

[10] Frederic Gruau, Darrell Whitley, and Larry Pyeatt. 1996. A comparison between cellular encoding and direct encoding for genetic neural networks. In *Proceedings of the 1st annual conference on genetic programming*. 81–89.

[11] Mojtaba Hajihosseini, Milad Andalibi, Meysam Gheisarnejad, Hamed Farsizadeh, and Mohammad-Hassan Khooban. 2020. DC/DC Power Converter Control-Based Deep Machine Learning Techniques: Real-Time Implementation. *IEEE Transactions on Power Electronics* 35, 10 (2020), 9971–9977. https://doi.org/10.1109/TPEL.2020.2977765

[12] Zhongsheng Hou, Ronghu Chi, and Huijun Gao. 2016. An overview of dynamic-linearization-based data-driven control and applications. *IEEE Transactions on Industrial Electronics* 64, 5 (2016), 4076–4090.

[13] Zhongsheng Hou and Shangtai Jin. 2013. *Model free adaptive control*. CRC press Boca Raton, FL, USA:.

[14] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim. 2019. Q-Learning Algorithms: A Comprehensive Classification and Applications. *IEEE Access* 7 (2019), 133653–133667. https://doi.org/10.1109/ACCESS.2019.2941229

[15] Frank L Lewis. 1992. *Applied optimal control and estimation*. Prentice Hall PTR.

[16] Krupa Prag, Matthew Woolway, and Turgay Celik. 2021. Data-driven model predictive control of DC-to-DC buck-boost converter. *IEEE Access* 9 (2021), 101902–101915.

[17] Krupa Prag, Matthew Woolway, and Turgay Celik. 2022. Toward data-driven optimal control: A systematic review of the landscape. *IEEE Access* 10 (2022), 32190–32212.

[18] Lal Bahadur Prasad, Barjeev Tyagi, and Hari Om Gupta. 2014. Optimal control of nonlinear inverted pendulum system using PID controller and LQR: performance analysis without and with disturbance input. *International Journal of Automation and Computing* 11 (2014), 661–670.

[19] Christiaan J Pretorius, Mathys C Du Plessis, and John W Gonsalves. 2017. Neuroevolution of inverted pendulum control: a comparative study of simulation techniques. *Journal of Intelligent & Robotic Systems* 86 (2017), 419–445.

[20] Mohammad Safeea and Pedro Neto. 2024. A Q-learning approach to the continuous control problem of robot inverted pendulum balancing. *Intelligent Systems with Applications* 21 (2024), 200313. https://doi.org/10.1016/j.iswa.2023.200313

[21] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[22] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation* 10, 2 (2002), 99–127. https://doi.org/10.1162/106365602320169811

[23] Yuanhong Tang, Weihao Hu, Jian Xiao, Zhangyong Chen, Qi Huang, Zhe Chen, and Frede Blaabjerg. 2021. Reinforcement Learning Based Efficiency Optimization Scheme for the DAB DC–DC Converter With Triple-Phase-Shift Modulation. *IEEE Transactions on Industrial Electronics* 68, 8 (2021), 7350–7361. https://doi.org/10.1109/TIE.2020.3007113

[24] Jia-Jun Wang. 2011. Simulation studies of inverted pendulum based on PID controllers. *Simulation Modelling Practice and Theory* 19, 1 (2011), 440–449.

[25] Christopher John Cornish Hellaby Watkins. 1989. Learning from delayed rewards. (1989).

[26] Xin Yao. 1999. Evolving artificial neural networks. *Proc. IEEE* 87, 9 (1999), 1423–1447.

[27] K.J. Åström and T. Hägglund. 2001. The future of PID control. *Control Engineering Practice* 9, 11 (2001), 1163–1175. https://doi.org/10.1016/S0967-0661(01)00062-4 PID Control.