A Comparative Study of Data-Driven Control Methods

Ariel J. Levy
University of Cape Town
Cape Town, South Africa
LVYARI002@myuct.ac.za

ABSTRACT

Robust data-driven control methods are steadily rising in prominence as solutions for tackling complex control problems in the data-rich environment of the Information Age. Machine learning-based approaches, such as reinforcement learning (RL) and neuroevolution (NE), are of particular interest due to their generalizability and adaptability. These methods aim to discover optimal control policies by relying on data, which diverges from the manual model construction inherent in classical controller architectures like Proportional-Integral-Derivative (PID). A thorough analysis of existing literature is necessary to ascertain the most powerful control method among these. As such, this review attempts to analyse the effectiveness of reinforcement learning and neuroevolution, as control methods, in conjunction with PID, as a benchmark. It was found that both RL and NE served as viable control methods and, in most cases, equalled or exceeded the performance of traditional control methods. Among the reinforcement learning algorithms investigated, Proximal Policy Optimization (PPO) demonstrated the highest potential with the best performance relative to training duration. The current literature on NE is limited, and information specifically about Neuroevolution of Augmenting Topologies (NEAT) is even sparser. Consequently, a comprehensive assessment of these methods' effectiveness awaits further experimentation.

1 INTRODUCTION

Control theory is a field that lies at the intersection of control engineering and applied mathematics; it deals with the development of algorithms that, given the current state of a system, serve to guide it towards a desired state. The field, while ever-present in today's technology-dependent society, has roots that stretch back centuries.

1.1 Historical Development

The First Industrial Revolution marked a profound shift in humanity's relationship with technology. Traditional artisanal craftsmanship yielded to the efficiency of mechanised production and, consequently, once rural, agrarian communities evolved into spawling urban societies [14]. The need to control and optimise the performance of these new, widely adopted machines – thereby maximising the throughput of factories – led to the conception of control theory, initially proposed by James C. Maxwell [1867].

Such a field was well-suited to aiding the development of novel technologies throughout the 19^{th} and into the 20^{th} century; with use found in burgeoning industries such as boiler control for steam generation, electric motor speed control, and temperature, pressure, and flow control in the process industries [4]. A large milestone in the progression of control theory came in 1911 when Elmer Sperry created the PID controller to automate ship steering for the US Navy [4]. PID controllers have since gained widespread adoption in industry, due to their versatility, robustness, and high accuracy.

The field continued to undergo major developments throughout the World Wars, specifically regarding the theoretical aspect of control theory – which had long been overshadowed by the focus on the theory's engineering side. Analyses of control systems and their designs were performed by many independent groups, which ultimately resulted in the formalisation of modern control theory in the early 1950's [14].

1.2 Modern Advances

Since the 1960s, modern control theory also referred to as model-based control (MBC), has matured into a useful field with many successful applications, especially in the aerospace industry [7]. MBC theory works by using a model of the system being controlled (referred to as the plant) and a set of assumptions – that should, in theory, represent the true system – to predict and adapt the future behaviour of the controlled system. This methodology is flawed, if the model is inaccurate or the assumptions made about the nature of the system do not hold, then no meaningful conclusions can be drawn and the physical controller will not function well [7]. In practice, this disconnect between the theoretical and practical aspects of MBC presents an obstacle to adopting this controller architecture in certain industries where systems are: novel, highly complex, or uncertain.

MBC's reliance on the plant model being accurate, to function as intended, can be reduced by implementing an even more powerful control framework: data-driven control (DDC). DDC works by leveraging the abundance of information in the Digital Age to fine-tune controllers using learning techniques – irrespective of the framework's dependence on the system's mathematical model. This means that historical data can be used to account for a lacklustre system model, if one even exists, with the controller's performance improving autonomously as more input/output (I/O) data is collected [7, 14]; this means that DDC can support both model-based and model-free control methods – depending on the implementation chosen.

1.3 Data-Driven Control Frameworks

1.3.1 Classical Control. Classical control theory – which governs controllers such as PID – deals with managing a system so that its output follows a desired signal. It is believed that applying a data-driven approach to the concepts of classical control theory could improve the performance of traditional controllers, in particular, this enables the controller to adapt to varying system conditions (such as the introduction of noise), automatically tune parameters, and better handle non-linear systems. However, this control method has, thus far, failed to gain a foothold in industry due to the dominance of model-based frameworks [3].

1.3.2 Machine Learning and Control. As mentioned earlier, the concept of DDC is based on the principle of self-learning. Combined with the substantial volume of I/O data generated, this naturally directs one's attention to the field of machine learning (ML).

More specifically, the subfield of ML that aligns best with these factors is reinforcement learning; RL is a technique that allows a system to learn through trial and error – as humans do – to achieve the most optimal output for a specific task. This enables a controller to teach itself how to control a system (optimally) in an environment, with no knowledge of the underlying dynamics of the system (i.e. it is model-free) [5].

Another distinct, but related, approach in the field of ML is neuroevolution. NE makes use of genetic algorithms (GA) to evolve neural networks (NN) [16, 21]. In the context of controllers, this allows an initial NN – which governs the controller's actions – to evolve and adapt over generations subject to its environment.

This learning method can be improved by applying a technique known as NEAT. NEAT makes slight alterations to neuroevolution, which is said to result in increased efficiency. It should be noted that NEAT – and to a lesser extent NE – are not thoroughly researched learning methods. As such, literature is hard to come by and ascertaining their true effectiveness as learning methods may be difficult.

1.4 Systems Under Consideration

To provide a means of comparison between the chosen control methods – PID, RL, and NE – two dynamical systems are considered. The first is the inverted pendulum, chosen due to its regular use as a benchmark problem in the field of control theory. The second is DC-DC voltage converters, which have found widespread use in both industrial and personal settings.

1.4.1 The Inverted Pendulum. The inverted pendulum is a pendulum system in which the centre of mass lies directly above the pivot point; the system is intrinsically unstable and will fall over without additional assistance due to a gravitational torque. With only one degree of freedom, the pendulum's motion is confined to angular movement along the arc of a circle. It is considered a benchmark problem in the field of control theory due to its: simple structure, non-linearity, and sensitivity to initial conditions [23].

1.4.2 DC-DC Converters. A DC-DC converter is an electronic circuit that converts direct current (DC) from one voltage level to another; these converters are available in two types: boost converters, which raise the voltage of the primary power supply, and buck converters, which reduce the voltage. These power conversion devices have gained widespread use in settings, such as renewable energy generation, electric cars, and small-scale electronic appliances [1, 12]. As such, the precise and efficient control of voltage signal transitions is crucial to industry. Equally important is considering their performance when subject to noise, which arises due to inherent imperfections in input signals.

2 BACKGROUND

This literature review will explore various control methods that can be applied to data-driven-based controllers. Its primary objective is to compare the historical performance and implementation of three distinct techniques: PID, RL, and NE.

2.1 The PID Controller

PID controllers attempt to manage a system so that its output follows a desired signal. This is achieved by taking an error signal, calculated by comparing the system's real-world and desired outputs, and adjusting the system's input to move the real-world output closer towards the desired output. They have three essential terms which regulate a process [3]. The Proportional (P) term adjusts the output relative to the size of the error; a large error means strong corrective action. The Integral (I) term accumulates the error over time; this drives the system to the desired output as time increases even if it consistently falls short. Finally, the Derivative (D) term anticipates future error by considering its current rate of change; if the desired output approaches this term increases to prevent overshoot [2]. Signal analysis is most commonly carried out in the frequency domain – instead of in the time domain – achieved through the use of the Laplace transform [9].

2.2 Reinforcement Learning

At a high level, most reinforcement learning methods are able to self-learn, given three crucial components: a policy, a lot of data, and time [5]; policy refers to the strategy the agent (in this case the controller's logic) uses to act on decisions it makes. RL problems are usually modelled as a Markov Decision Process (MDP) which encompasses the following aspects: the set of internal states, the set of actions available, the transition probability matrix (which quantifies how actions lead to state transitions), and the reward function (which specifies how to reward the agent for actions it takes) [14]; broadly speaking, the policy is a solution to the MDP.

There are a variety of RL methods that exist. However, this review will explore only two such methods: Q-Learning and PPO; both because they are straightforward to implement and are well-researched, especially when considering the systems being investigated.

2.2.1 Q-Learning. A model-free learning method proposed by Christopher J. Watkins [1989] that starts with an agent in a state. The agent then selects an action based on its current strategy. After taking the appropriate action, it receives either a reward or a punishment. The agent learns from previous actions and optimises its strategy using this reward information. This repeats until an optimal strategy is found. The value of each state-action pair is stored in a Q-table, which is updated as the agent interacts with the environment and rewards/punishments are distributed [24].

2.2.2 PPO. PPO is a policy gradient method, developed by John Schulman *et al.* [2017], which serves two primary functions: sampling data via interaction with the agent's environment and optimising the agent's policy parameters using stochastic gradient ascent; stochastic methods are used to allow natural exploration of possible solutions [20].

2.3 Neuroevolution

NE makes use of GAs – which simulate the evolutionary process of natural selection observed in biological populations – to evolve neural networks by dynamically adjusting their weights [16, 21]. Only optimal behaviours persist throughout this evolutionary process, leading to an increasingly effective controller.

This learning method can be improved by applying a technique known as NEAT. NEAT makes slight alterations to neuroevolution, which is said to result in increased efficiency. Essentially, NEAT imbues the NN with the ability to change its structure; the number of nodes and types of layers can now vary and evolve across generations – much like the weights in ordinary NE [21]. This introduces more diversity to the NN, leading to a more balanced GA; and thus more evolutionary advanced, better-performing solutions.

3 APPLICATIONS OF CONTROL METHODS

To facilitate a meaningful comparison of the considered control methods, evaluating them within the context of their application to specific dynamical systems is necessary. This analysis is carried out within the specific context of the inverted pendulum and DC-DC voltage converter systems.

3.1 Inverted Pendulum

3.1.1 PID. Attempting to stabilise an inherently non-linear system (inverted pendulum) with a linear controller (PID) may seem counterintuitive. However, the problem can be linearised by applying a first-order Taylor approximation to the awkward trigonometric terms in the system's governing differential equation, making control with a PID controller possible; note this approximation only applies for small deviations from the pendulum's equilibrium position [22, 23].

It was found by Jia-Jun Wang [2011] that the controller was able to find and maintain the (simulated) inverted pendulum at its equilibrium position after 3 seconds – under ideal conditions; this result is subject to the tuning of the PID controller, the pendulum's physical parameters, and the initial angular offset.

A similar experiment was carried out by Ella S. Varghese *et al.* [2017], with two PID controllers used to control a (simulated) moving pendulum-cart system – with one controller for each. They established that the inverted pendulum found and maintained its equilibrium position from 6 seconds onwards – under ideal conditions. When noise was introduced, the pendulum was unable to stabilise, with it only oscillating between ±0.05 rad of the equilibrium position.

3.1.2 Reinforcement Learning. In the case of the inverted pendulum, there are three possible actions that the RL model can take at any one moment: push left, push right, and do nothing.

Sardor Israilov *et al.* [2023] tested a Q-Learning approach for up to 10^7 episodes (equivalent to 6 days of real-time experiments) on a simulated inverted pendulum – controlled by a motorised cart – and found that the RL model was unable to stabilise the system for any meaningful amount of time. They claimed this failure could be attributed to learning inefficiencies resulting from the Q-table's matrix representation. To address this suspected inefficiency, they also carried out a method known as Deep Q-Learning on the inverted pendulum; which involved replacing the Q-table with a neural network. This adjustment did seem to work, as it stabilised the inverted pendulum successfully at its equilibrium position – independent of initial conditions [8].

Mohammad Safeea and Pedro Neto [2024] took a different approach to controlling the pendulum's movements, opting instead, to use a real-world, seven-jointed robot. The Q-Learning policy converged in 10^4 episodes (conducted virtually to save time) before the model was transferred to the real-world system for experimentation. This experiment showed that the controller could keep the inverted pendulum at its equilibrium position for at least 5 seconds within some small angular margin of error [17]. Unfortunately, the system failed after this point because it exceeded the flange's position – a real-world constraint likely not considered in the virtual training environment.

When PPO was applied to OpenAI Gym's inverted pendulum problem [6], Swagat Kumar [2021] found that the RL algorithm was able to be trained to solve the problem in 1000 episodes (with each episode lasting 200 iterations). In this case, the condition for having successfully solved the problem was ensuring that the pendulum maintained its equilibrium position for a considerable time – at least 50 episodes [10]. This was accomplished considerably faster and with more consistency than the Q-Learning-based implementations.

3.1.3 Neuroevolution. In a paper by Christiaan J. Pretorius *et al.* [2017], NE-based techniques – including NEAT – were applied to a real-world, robotic inverted pendulum; note that training of these genetic algorithms was performed in a simulated environment. It was found that the NE models could make predictions with high accuracies. Unfortunately, the NEAT models did not train very accurately compared to the other NE-based models. Further analysis of the NEAT models' evolutionary parameters could have yielded more promising results, but this was not done in the study. The computational efficiency of the methods was quantified in terms of the amount of time a model takes to quantify the fitness of a single controller during the evolution process. It was found that the linearised physics model was the fastest on average taking just 9 ms – due to the existence of a known, closed-form solution – while the NE models took almost twice as long (20 ms). The non-linear physics model performed the worst with a time of 89 ms.

They also found that NE-based models outperformed both physics models in low-noise environments, remaining balanced for 22 seconds on average (compared to around 3 seconds for the physics models). However, when higher amounts of noise were introduced, the NE model struggled to maintain balance – more so than the physics models; the margin of error for balancing was chosen to be within just 6 degrees of the pendulum's equilibrium position [15].

3.2 DC-DC Converters

3.2.1 PID. DC-DC converters exhibit non-linear behaviours, as a result, some aspects of the system or the (linear) PID controller must be adjusted or simplified. Furthermore, due to the intricate nature of the system, this adaptation process is significantly more complex than merely linearising a differential equation. Two such adaptation techniques are explored: one for boost converters and one for buck converters; it should be noted that the methods presented are not exclusive to a particular type of converter, but may require minor adjustments for the other type.

Mirza F. Adnan *et al.* [2017] took a very straightforward, hands-on approach: adjusting the PID controller's parameters through trial and error. Using a (simulated) boost converter they found that this approach was able to step up the source voltage (90 – 110 V) to the correct reference voltage (200 V) with an overshoot error of between 2 – 3%; the converter was found to take around 1000 s to reach the reference voltage – after which steady-state was maintained – for all distinct source voltages used [1]. The manual nature of the controller's tuning suggests that the system lacks robustness in handling any potential irregularities, such as noise or disturbances.

Norsyahidatul F. Nanyan *et al.* [2024] suggested employing a more practical approach to a buck converter system. That is, applying the Sine Cosine Algorithm (SCA) to the system to automatically tune and optimise the parameters of the PID controller, allowing the controller to adjust to any system variations. The SCA is a procedure that iteratively adjusts candidate solutions using sine and cosine functions, exploring the search space efficiently, and allowing them to converge towards optimal solutions [12]. When this algorithm (SCA) and an improved version presented in the paper (ISCA) – which employs additional checks to prevent potential trappings within local optima – were implemented experimentally, they were both able to step down the source voltage to the correct reference value (3 V) with five decimal places of precision within the order of 10^{-6} seconds; steady-state was maintained thereafter. This was quicker and far more precise than any of the results produced by alternate methods. Additionally, when disturbances were applied to the system – by altering the capacitance and inductance of components in the circuit – the results mirrored those seen (for both ISCA and SCA) in the ideal environment; which implies that this implementation is resilient to any irregularities.

3.2.2 Reinforcement Learning. While no literature was found on Q-Learning being used to control DC-DC converters, an approach involving Deep Q-Learning was found. B. Nishanthi and J. Kanakaraj [2023] applied this neural-network-based RL method to control a bi-directional DC-DC converter, which is compatible with both buck and boost topologies. They established that the Deep Q-Learning controller outperformed a PI controller, and reached within 6% of the reference voltage in a time of $9 \cdot 10^{-4}$ s. They also tested the system under different and changing loads to demonstrate robustness. This showed that, under challenging conditions, the Deep Q-Learning controlled converter attained stability faster and with less error than the PI-controlled one [13].

Similarly, Utsab Saha *et al.* [2023] applied PPO to a DC-DC Boost converter. The input voltage was set at 24 V, while the reference voltage was set between 48 V and 60 V. Across all reference voltages, the error was found to be at most 0.37%; steady state was reached within 0.3 seconds. The input voltage was also slightly varied to test the performance of the RL controller under dynamic system conditions [18]. The controller performed well under these conditions, learning from the system's behaviour, and adjusting its actions accordingly; this indicated that this PPO controller was well-suited to handling variations in source voltage.

3.2.3 Neuroevolution. Fredy H. Sarmiento *et al.* [2012] attempted to optimise NE-based architectures to control a boost converter. The model was initially evolved and optimised virtually, before being run physically. The model was tested under two conditions: changing loads and different source voltages. To test both decreases and increases in loads, whilst in steady state, the resistance of the circuit was lowered, with this change reverted shortly thereafter. It was found that the decrease in load resulted in an overshoot of 18% – stabilising in 48 ms – while the subsequent increase in load triggered an overshoot of 22% – stabilising after 75 ms. When the source voltage was altered from 120 V to 80 V – under a constant load – the output voltage was seen to drop by 20% before recovering to its value in 130 ms [19].

4 CONCLUSIONS

The landscape of control theory is evolving in response to the data-rich environment of the Information Age. This progression has triggered a surge in efforts to implement and master data-driven techniques, with cutting-edge techniques like reinforcement learning and neuroevolution at the forefront of development. These methods are believed to surpass traditional control techniques, such as PID, particularly when handling non-linear processes, due to their unconstrained and model-free nature.

Both studies that explored PID's application to the inverted pendulum were successful within a reasonable time frame, which is unsurprising due to PID's longstanding position as one of the most reliable traditional controllers. The controller's inability to deal with noise was expected, as it is not a linear process. This drawback could potentially be fixed by investing in data-driven techniques to augment the existing PID controller's architecture, as already discussed above.

Concerning the reinforcement learning-based control methods for the inverted pendulum, it was found that PPO models were substantially faster to train and better at stabilising the pendulum than Q-Learning techniques. Both attempts at applying Q-Learning resulted in some system failure, indicating that a change in approach is probably needed for this specific problem. Deep Q-Learning also outperformed regular Q-Learning but is the most computationally expensive to implement.

Various neuroevolution-based methods were applied to the stabilisation of the inverted pendulum. They were all found to outperform linearised and non-linearised physical models; they also proved to be very computationally efficient, with their performance closely resembling the linearised physics model, which entails solving an analytical system of linear equations. The NE models also performed significantly better than the physical models in low-noise environments – with the NE-based controller struggling to balance the pendulum when subject to higher noise levels.

The application of PID control to DC-DC converters found that one method, the SCA, was successful and the other, trial-and-error, was less so. The trivial approach of manually tuning the PID controller for a boost converter worked but took very long to implement. Conversely, the more complex method of applying the SCA to buck converters produced successful results in both ideal and noisy environments in a significantly shorter time; with the results being more precise and processed faster than the trivial method. It seems that the burden of implementing ISCA is not worthwhile, as it produced very similar results to the regular SCA and required more work and processing.

When considering the application of RL to DC-DC converters, it was established that both Deep Q-Learning and PPO were able to control the converter successfully. Both methods offered more flexibility and adaptability for handling dynamic scenarios than traditional control methods – like a PI controller. Furthermore, PPO demonstrated lower errors relative to the reference voltage, whereas Deep Q-Learning exhibited faster adaptation to voltage changes and a quicker return to steady state.

The robustness of NE-controlled DC-DC converters was tested under two conditions: changing loads and different source voltages. The experimental results were moderately successful and showed, to an extent, the potential viability of the scheme.

It remains to be seen whether data-driven, machine learning-based control methods truly represent the logical evolution of the field of control theory. However, the literature does show promise in this direction.

REFERENCES

[1] Mirza F. Adnan, Mohammad A. Oninda, Mirza M. Nishat, and Nafiul Islam. 2017. Design and Simulation of a DC-DC Boost Converter with PID Controller for Enhanced Performance. *International Journal of Engineering Research & Technology* 06 (2017). https://doi.org/10.17577/IJERTV6IS090029

[2] Karl J. Astrom and Tore Hagglund. 1995. *PID Controllers Theory, Design and Tuning* (second ed.). Instrument Society of America, 59–70.

[3] Alexandre S. Bazanella, Lucíola Campestrini, and Diego Eckhard. 2023. The data-driven approach to classical control theory. *Annual Reviews in Control* 56 (2023), 100906. https://doi.org/10.1016/j.arcontrol.2023.100906

[4] Stuart Bennett. 1996. A brief history of automatic control. *IEEE Control Systems* 16, 3 (1996), 17–25. https://doi.org/10.1109/37.506394

[5] Alain Bensoussan, Yiqun Li, Dinh P. Nguyen, Minh-Binh Tran, Sheung C. Yam, and Xiang Zhou. 2020. Machine Learning and Control Theory. (2020). arXiv:2006.05604 [cs.LG]

[6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540* (2016).

[7] Zhong-Sheng Hou and Zhuo Wang. 2013. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences* 235 (2013), 3–35. https://doi.org/10.1016/j.ins.2012.07.014

[8] Sardor Israilov, Jesús Fu, Sánchez-Rodríguez, Franco Fusco, and Guillaume Allibert. 2023. Reinforcement learning approach to control an inverted pendulum: A general framework for educational purposes. *PLOS ONE* 18, 2 (2023), e0280071. https://doi.org/10.1371/journal.pone.0280071

[9] Reto B. Keller. 2023. *Time-Domain and Frequency-Domain.* Springer International Publishing, 305–306. https://doi.org/10.1007/978-3-031-14186-7_5

[10] Swagat Kumar. 2021. Controlling an Inverted Pendulum with Policy Gradient Methods-A Tutorial. (2021). arXiv:2105.07998 [cs.LG]

[11] James C. Maxwell. 1867. On Governors. *Proceedings of the Royal Society of London* 16 (1867), 270–283. http://www.jstor.org/stable/112510

[12] Norsyahidatul F. Nanyan, Mohd A. Ahmad, and Baran Hekimoğlu. 2024. Optimal PID controller for the DC-DC buck converter using the improved sine cosine algorithm. *Results in Control and Optimization* 14 (2024), 100352. https://doi.org/10.1016/j.rico.2023.100352

[13] B. Nishanthi and J. Kanakaraj. 2023. Enactment of Deep Reinforcement Learning Control for Power Management and Enhancement of Voltage Regulation in a DC Micro-Grid System. *Electric Power Components and Systems* 52, 4 (2023), 555–565. https://doi.org/10.1080/15325008.2023.2227200

[14] Krupa Prag, Matthew Woolway, and Celik Turgay. 2022. Toward Data-Driven Optimal Control: A Systematic Review of the Landscape. *IEEE Access* 10 (2022), 32190–32212. https://doi.org/10.1109/access.2022.3160709

[15] 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, 3–4 (2017), 419–445. https://doi.org/10.1007/s10846-017-0465-1

[16] Risto Miikkulainen. 2010. *Neuroevolution.* Springer US, Boston, MA, 716–720. https://doi.org/10.1007/978-0-387-30164-8_589

[17] 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

[18] Utsab Saha, Shakib Shahria, and A. B. M Harun-Ur Rashid. 2023. Proximal Policy Optimization-Based Reinforcement Learning Approach for DC-DC Boost Converter Control: A Comparative Evaluation Against Traditional Control Techniques. (2023). arXiv:2310.02945 [eess.SY]

[19] Fredy H. Sarmiento, Diego F. Molano, and Mariela C. Ortiz. 2012. Optimization of a neural architecture for the direct control of a Boost converter. *Tecnura* 16 (2012), 41–49. https://www.redalyc.org/pdf/2570/257024143004.pdf

[20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]

[21] 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

[22] Elisa S. Varghese, Anju K. Vincent, and V. Bagyaveereswaran. 2017. Optimal control of inverted pendulum system using PID controller, LQR and MPC. *IOP Conference Series: Materials Science and Engineering* 263, 5 (2017), 052007. https://doi.org/10.1088/1757-899X/263/5/052007

[23] Jia-Jun Wang. 2011. Simulation studies of inverted pendulum based on PID controllers. *Simulation Modelling Practice and Theory* 19, 1 (2011), 440–449. https://doi.org/10.1016/j.simpat.2010.08.003

[24] Christopher J. Watkins. 1989. *Learning From Delayed Rewards.* Ph. D. Dissertation. University of Cambridge. https://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf