Koc. J. Sci. Eng., 7(1): (2024) 30-41 https://doi.org/10.34088/kojose.1278657

Kocaeli University

Kocaeli Journal of Science and Engineering

http://dergipark.org.tr/kojose

Controlling a Single Tank Liquid Level System with Classical Control Methods and Reinforcement Learning Methods

Murat Erhan ÇİMEN [1], Zeynep GARİP [2,*]

[1] Department of Electric Electronic Engineering, Sakarya University of Applied Sciences, Sakarya, 54050, Turkey, **ORCID:** 0000-0002-1793-485X
[2] Department of Computer Engineering, Sakarya University of Applied Sciences, Sakarya, 54050, Turkey, **ORCID:** 0000-0002-0420-8541

Article Info

Research paper

Received : *April 6, 2023*
Accepted : *August 19, 2023*

Keywords

Tank System
Classical Controllers
Reinforcement learning

Abstract

In this study, the control of the single tank liquid level system used in control systems has been carried out. The control of the single tank liquid level system has been performed with the classic PI, modified PI, state feedback with integrator action, and Q learning algorithm and SARSA algorithms, one of the artificial intelligence methods. The tank system to be modelled was carried out using classical physics, namely Newton's laws. Then, the mathematical model obtained of the system that are continuous model in time is acquired. The originality of the study; the non-linear liquid tank system is controlled by classical controllers and reinforcement methods. For this purpose, the system was firstly designed to model the system, then the system has been linearized at a specific point in order to design classic PI, modified PI, and state feedback with integral. After that, agents of the Q Learning algorithm and SARSA algorithms were trained for the system. Then the agents have controlled the single-level tank system. The results of the classic controllers and supervised controllers are contrasted with regard to performance criteria such as rising time, settling time, overshoot and integral square error. Consequently, Q learning method has produced 0.0804-sec rising time, 0.943 sec settling time and 0.574 integral square errors. So, Q learning algorithm has produced and exhibited more thriving and successful results for controlling single liquid tank system than PI, Modified PI, state feedback controllers and SARSA.

1. Introduction

Machine learning is used in artificial intelligence, with the development of today's computer technology, artificial intelligence has entered many different application areas from health [1–3], and logistics [4, 5] to chemistry, finance [6], [7] to education [8] to computer game [9, 10] and industry [11– 13]. The main cause is that machine learning methods can evaluate and interpret data faster and more accurately today and make the most appropriate and correct decisions. However, according to the place of use, machine learning can be divided into branches such as supervised, unsupervised and reinforcement learning. But reinforcement learning has gained in significance since it can adapt to changing environmental conditions. An agent can interact with the environment, then it can learn what to do, depending on a specific reward. its application to many different fields has begun to be developed [13, 14]. Reinforcement learning, which has the capacity to learn its environment, could also be implemented to control nonlinear systems and industrial processes [12].

Reinforcement learning could be traced back to Bellman's work on optimal control theory in the 1950s [15]. Bellman, working on optimal control theory, developed dynamic programming, which is an approach to optimally control dynamic systems over time. In this method, a function value of a state for the system is specified by calculating the control signal. According to this value and according to the function value of the next state, iteratively discrete optimal control signal was tried to be calculated. This approach can be expressed with the Bellman Equation. Therefore, the approach that enables the calculation of the control signal using the Bellman Equation can be called dynamic programming [16]. This method can be implemented to randomly run Markovnian decision processes and systems can be controlled via MDP without knowing the model of the system. MDP generally

* Corresponding Author: zbatik@subu.edu.tr

refers to processes that depend on the current and next state. It can control MDPs with optimal control methods using dynamic programming, but it requires a processing load [17]. In addition, they have the problem of "the curse of dimensionality" as they operate for every situation [18]. In their study, Farley and Clark proposed a trial-and-error method as well as dynamic programs that evaluate each situation. They have proposed artificial neural networks that learn by trial and error. Later, Farley and Clark implemented the trial-and-error method to pattern recognition [15, 16]. Michie has used the trial-error learning system for tic tac toe game [19]. The temporal difference learning method has also been applied in playing the tic tac toe game. In fact, the basis of temporal difference learning is based on the learning psychology of animals. Minsk first noticed this in 1954 and predicted that it could be significant for learning methods [20, 21]. In the same period, Samuel developed the temporal difference method independently, influenced by chess games [22]. However, when we look at the 1980s, the trial error method, which was transferred into the temporal difference method, is mostly known as actor-critic architecture. Different versions of the temporal difference method have been developed. Later, Chris Watkins developed the Q learning method in 1989 by using the temporal difference learning method with optimal control [23, 24]. In the 1990s Tesauro developed a backgammon playing program [25]. In this method, training has been conducted using a version of the Temporal difference method and artificial intelligence [26]. In fact, the Q learning method has methods such as dynamic programming, temporal difference, and Monte Carlo [10, 27, 28]. Today we live through the information age, especially the use of reinforcement learning with methods such as deep learning [13, 28–30], which continues to be up-to-date. Moreover, it is open to development and open to application in many different fields.

The tank system has been used since Hellenistic times [31]. In particular, there are studies carried out to measure time by keeping the water level constant in the tank. In the 17th century, applications such as pressure, temperature, or speed control of the rotor were carried out in tanks. Examples of these are mechanical applications such as temperature control for furnaces or speed control for windmills. However, the development of the real industry was realized with the invention of steam engines and the industrial revolution took place. The widespread use of variables such as pressure, temperature, mixing, amount, flow rate and level in tank systems used in industrial processes, especially in sectors such as drink, beverage, chemistry, pharmaceutical and petroleum, has made it important to control. Many studies on this topic have been undertaken in recent years [32, 33]. Mizumoto et al. have proposed employing a PID controller design to control the tank liquid system [34]. Taler et al. have put forward and applied a method to control the hot fluid with PID [35]. Samin et al. realized the control of the liquid tank system with PID with PLC. At the conclusion of the research, the values for different parameters were compared and interpreted [36]. Fatih et al. have used to genetic algorithm to determine PID and LQR for controlling of level of the liquid tank system [37]. Selamet et al. have implemented the control of the liquid tank system with the most optimal controller parameters by using the PSO algorithm to specify the parameters of the PID and LQR methods [38]. Sastry et al. have performed the control of the single tank system using a nonlinear PID controller [39]. Kum et al. have carried out the tank system with a sliding mode controller. Wei et al. have realized the control of the liquid tank system with the back-stepping method [40]. Xiao et al. have used fuzzy logic in the control of the tank system in their study. In their study, fuzzy logic can adapt the PID controller. They have obtained successful results [41]. Esakkiappan, on the other hand, has performed the control of the liquid tank system with the PI controller they designed with cuckoo optimization [42]. Then, Son performed the control of the tank system using an adaptive inverse evolutionary algorithm [43]. Urrea et al. Again, using the liquid tank system with PID, Gain Scheduling, Internal Mode Control and fuzzy logic, they have implemented the control of the system comparatively and presented the results [44].

Essentially, in this study, a nonlinear system is controlled using classical controllers and reinforcement learning techniques. In particular, the single-level tank system was controlled using a modified PI controller as well as the classic PI controller. In addition, using the linearized model, the design and control of the State feedback controller with an integrator has been carried out. Finally, the control of the liquid tank system was carried out using SARSA (State-action-reward-state-action) and Q learning methods. The obtained results have been compared with respect to ISE (Integral square error) performance value and overshoot and settling time. In addition, learning-based SARSA and Q learning methods have been compared with regard to performance criteria and control signals. It has seen that Q learning algorithm produced better results.

The main contributions of this paper are as follows:

- PI, Modified PI and State Feedback with integral action controller have been designed to control the single tank liquid system.
- SARSA and Q Learning Methods are applied to control the single tank liquid system.

In this study, the first part of the content is the introduction and information about artificial intelligence, reinforcement learning, tank system and control, and a literature review have been given. In Section 2, there is the Material and Method section, the methods realized in the study have been explained and the designs for the system have been made. Section 3 is the result and the control of the single-level tank system and the results obtained are demonstrated by comparing them in tables and graphics. Chapter 4 is the conclusion and the results are evaluated.

2. Materials and Methods

In reinforcement learning, there are two main system blocks seen in Figure 1. One of these blocks is an agent and the other is an environment. The agent's learning and recognition of the environment via interactions with it, and the experiences that have been obtained as a result of these interactions with the environment, recognizes the environment and begins to respond in a way that achieves the maximum reward it aims [7, 10, 15, 27, 30]. In this method, the agent constantly tries to learn the environment and thus develops the next step.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha \left[R(s_t, a_t) + \gamma \max_{a} Q(s_{t+1}, a) - Q(s_t, a_t)\right] \quad (1)$$

In addition, the epsilon greedy method is applied to determine the q values [45]. This method allows the agent to visit all possible states during learning. Thus, the agent acquires better knowledge of the environment and determines the q values that can maximize the reward. The pseudocode of Q learning has been presented in Algorithm 1. This is an off-policy strategy in which the learning agent learns the value function depending on the current action obtained from the policy currently in use [28].

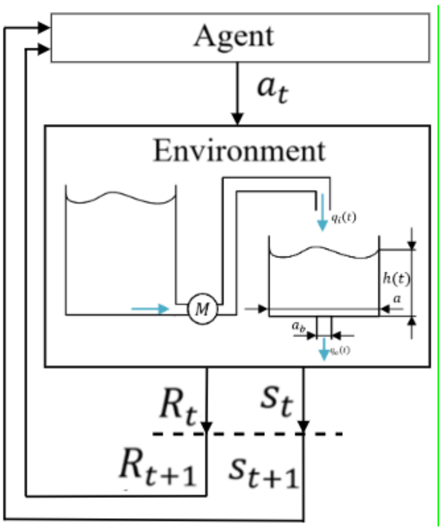

Figure 1. Interaction between Agent and Single Level Tank System Environment.

Algorithm 1 . Q learning Pseudo code.

Input:
State (s)
Action (a_t)
Learning rate (α)
Discount factor (γ)
Reward $R(s_t, a_t)$
Updated table $Q(s_t, a_t)$
Output:
Selected action according to updating table $Q(s_t, a_t)$
For episode 1, $Iter_{max}$ **do**
Initialise state s_t
$done = FALSE$
While $done == FALSE$
Choose a_t with ϵ greedy probability
Execute a_t and observe state s_{t+1}, reward r_t and $done$
Update table using Equation 1
End While
End For

2.1. Q learning Algorithm

Q learning method and SARSA methods are model-independent or model-free reinforcement learning methods. Reinforcement learning also has a feature that learns more from behavior. This occurs during an agent's interaction with the environment. Agent, which is being learned, interacts with environment and evaluates the outputs of the environment to produce an output called a reward [31-32, 50–52]. The agent learns the environment and begins to operate in a way that receives a better reward at each step based on the agent's current state, interaction with environment, and agent's next state. It can be preferred in applications that are difficult to model, especially since they can learn by experiencing the results of actions rather than the model. Especially with the Bellman equation that Richard Bellman recommends, and Watkin uses in reinforcement learning, it is provided to learn the actions and outputs performed depending on the situations. In this learning method, the system is learned in terms of situations and action and reward value rather than a specific model. The simplest version of the Bellman equation used in Q learning is given in Equation 1. Equation 1 used s_t, state at time t, action a_t at time t, s_{t+1} state at time t+1, $R(s_t, a_t)$ reward value at s_t and a_t, $Q(s_t, a_t)$ state of being value at s_t and a_t, α learning factor, and γ discount factor. $\max_{\alpha}\left(Q(s_{t+1}, a_{t+1})\right)$ is the value at which the maximum q value is produced according to the action α in the case of s_{t+1}. In each iteration, this operation is performed, and the q values are updated

2.2. SARSA Algorithm

SARSA, which is a method used in learning Markovinian processes, is a method used in reinforcement learning in machine learning. This method was proposed by Rumble and Niranjan as an alternative to the Q learning algorithm [46]. This method is an on-policy and the learning agent tries to learn the value function depending on the action derived from another policy [28]. That is, it does not need to be a value generated from within itself. The equation used in SARSA is given in Equation 2. The notations s_t, a_t, s_{t+1}, $R(s_t, a_t)$, $Q(s_t, a_t)$, α, γ, which has been used in Equation 1, is the same as the Q learning algorithm. There is only a single difference: the value of $Q(s_{t+1}, a_{t+1})$ represents the Q value obtained when the action a_{t+1} is applied in the case of s_{t+1}. For this value to be produced, the algorithm's action must be applied once, and the Q value produced by this applied action must be determined. In the SARSA algorithm, each action is applied to the a_t system, and $R(s_t, a_t)$ and $Q(s_{t+1}, a_{t+1})$ are determined and $Q(s_t, a_t)$ values are updated. The psoudecode of the SARSA algorithm is given in Algorithm 2.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[R(s_t, a_t) + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{2}$$

Algorithm 2. SARSA Pseudo code.

Input:
State (s)
Action (a_t)
Learning rate (α)
Discount factor (γ)
Reward $R(s_t, a_t)$
Updated table $Q(s_t, a_t)$
Output:
Selected action according to updating table $Q(s_t, a_t)$
For episode 1, $Iter_{max}$ **do**
Initialise state s_t
$done = FALSE$
While $done == FALSE$
Choose a_t with ϵ greedy probability
Execute a_t and observe state s_{t+1}, reward r_t and $done$
Choose a_{t+1} with ϵ greedy probability using state s_{t+1} from Q
Update table using Equation 2
End While
End For

2.3. Single Tank Liquid Level System

Single tank liquid level system, which is a very common system in the industry, is widely used in process control [37, 49]. It is mostly encountered in places where liquid filling and discharge are made such as medicine, food, and agriculture. In Figure 1, the structure of the tank level system is demonstrated. There is a main tank, motor and discharge tank in the system in Figure 1. Liquid level in the discharge tank is tried to be controlled by controlling a motor connected to the main tank. So, a single tank liquid level system has a single input which is the motor and a single output that is liquid level. There are parameters used in the single tank liquid level system, surface area of the tank a, the surface area of the outlet of the discharge tank a_b, and the velocity coefficient c_{db}, which differs due to the outlet structure for fluid [50]. The modeling of the single liquid tank system is presented in equation 3 $q_i(t)$ is the fluid's flow rate sent by the engine into the discharge tank, $q_o(t)$ is fluid's flow rate leaving the discharge tank, and $h(t)$ is the liquid's height in the discharge of tank. Consequently, the fluid supplied by the engine is the system's input, and the fluid's height in the discharge tank is the system's output.

$$q_i(t) - q_o(t) = a\frac{dh(t)}{dt} \tag{3}$$

$$q_o(t) = c_{db} a_b \sqrt{2gh(t)} \tag{4}$$

In Table 1, the parameters of the Single Tank Liquid Level System have been displayed.

Table 1. Parameters of Single Tank Liquid Level System.

Parameters	Unit	Value
c_{db}	-	0.62
a_b	m^2	0.00314
a	m^2	0.00314
g	$m/_{s^2}$	9.81

After the values of the parameter of the system have been substituted in their place, the nonlinear model becomes like as in Equation 5. Then, Equation 6 is obtained by editing Equation 5. After that, Equation 7 is obtained when the $x(t)$ control sign is written instead of $h(t)$ height variable and the $u(t)$ control signal is written instead of $q_i(t)$. Controller design will be realized by linearizing this equation at a certain point.

$$\frac{dh(t)}{dt} = 31.847q_i(t) - 0{,}62\sqrt{2 \times 9.81 \times h(t)} \tag{5}$$

$$\dot{h}(t) = -2.746\sqrt{h(t)} + 31.847q_i(t) \tag{6}$$

$$\begin{aligned} \dot{x}(t) &= -2.746\sqrt{x(t)} + 31.847u(t) \\ y &= x(t) \end{aligned} \tag{7}$$

For linearization, by making $\dot{x}(t) = f(x, u)$ the linear model for a single tank liquid level system has been calculated according to the equilibrium point $x_0 = 1, u_0 = 1$. For this, in the linearization of the state, Equation 8 is first made and linearization is performed around the equilibrium point. Then the values of x_0 and u_0 are written in place to get Equation 9. Equation 10 is obtained when the mathematical operations in Equation 9 are performed. Equation 11 is obtained when this expression in the time dimension is moved to the Laplace dimension. After that, by using the output-to-input ratio to determine the system's transfer function, Equation 12 is obtained.

$$\begin{aligned}\Delta\dot{x}(t) &= \frac{df(x,u)}{dx}\bigg|_{x=x_0}\Delta x(t) \\ &\quad + \frac{df(x,u)}{du}\bigg|_{u=u_0}\Delta u(t) \\ y &= \Delta x(t) + x(t)\end{aligned} \tag{8}$$

$$\begin{aligned}\Delta\dot{x}(t) &= -2.746\left(\frac{1}{2}(x_0)^{-0.5}\right)\Delta x(t) \\ &\quad + 31.847\Delta u(t) \\ y &= \Delta x(t) + x(t)\end{aligned} \tag{9}$$

$$\begin{aligned}\Delta\dot{x}(t) &= -1.373\Delta x(t) + 31.847\Delta u(t) \\ y &= \Delta x(t) + x(t)\end{aligned} \tag{10}$$

$$s\Delta x(s) = -1.373\Delta x(s) + 31.847\Delta u(s) \tag{11}$$

$$G(s) = \frac{\Delta x(s)}{\Delta u(s)} = \frac{31.847}{s + 1.373} \tag{12}$$

2.4. PI and Modified PI

The transfer function of a single tank liquid level system has been obtained in Equation 12. When the PI controller has been designed according to Zeigler Nichol's step response for this system, the obtained controller has is given in Equation 13.

$$PI(s) = K_p + K_i\frac{1}{s} = 0.3887 + 2.9905\frac{1}{s} \tag{13}$$

Since this tank system has been controlled in discrete time, the controller has to be discretized for the system to be implemented. The discrete structure of the discretized controller at T=0.05 sec has been obtained as $PI(z) = \frac{19{,}82z-7.43}{z-1}$. Then by using this controller, controlling of single-level tank system has been conducted by using PI as in Figure 2 and modified PI as in Figure 3.

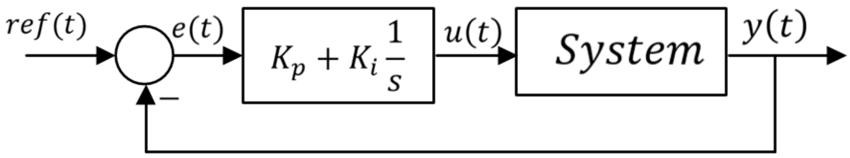

Figure 2. Control Structure with Single Tank Liquid Level System with PI.

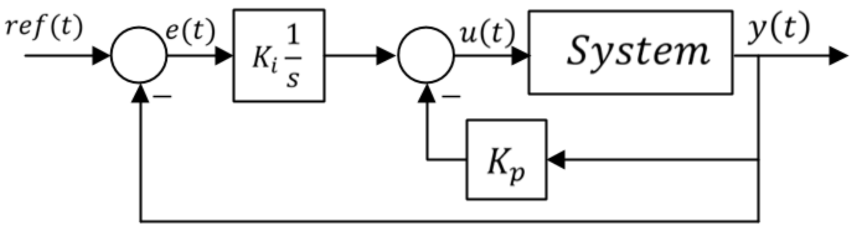

Figure 3. Single Tank Liquid Level System Control Structure with Modified PI.

2.5. State Feedback Controller Design

State feedback controllers with integral action have been designed for tank system control, and the tank system has been controlled. The structure of state feedback with integral action has been demonstrated in Figure 4. The controller has been realized by adding an integral action to a system in the structure designed in the state space. By adding an integrator to controller, steady state error between reference and system response would be eliminated. In Figure 4, each signal namely error $(e(t))$, control signal $(u(t))$, and state of the system $(\dot{x}(t))$ is designed step by step as in Equation 14. As a result, and augmented state space containing the state of the system $(\dot{x}(t))$ and integral of the error $(\dot{x}_i(t))$ is obtained in Equation 15. In Equation 15, the Control signal of the system has become $ref(t)$ in the augmented state. After the parameters of the tank system are substituted in their place, the structure of the system's model to be utilized in the design has been obtained in Equation 16.

$$\begin{gathered} e(t) = r(t) - y(t) = r(t) - Cx(t) \\ e(t) = \dot{x}_i = e(t) = r(t) - y(t) = r(t) - Cx(t) \\ u(t) = x_i - Kx(t) \\ \dot{x}(t) = Ax(t) + Bu(t) \\ \dot{x}(t) = Ax(t) + B\big(x_i - Kx(t)\big) \\ \dot{x}(t) = (A - BK)x(t) + Bx_i \end{gathered} \tag{14}$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix} = \begin{bmatrix} A-BK & B \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} ref(t) \tag{15}$$

$$\begin{bmatrix} \Delta\dot{x}(t) \\ \dot{x}_i(t) \end{bmatrix} = \underbrace{\begin{bmatrix} -1{,}373 - 31{,}847K & 31{,}847 \\ -1 & 0 \end{bmatrix}}_{A_{cl}} \begin{bmatrix} \Delta x(t) \\ x_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} ref(t) \tag{16}$$

The system matrix of the system in which pole placement will be made is $A = -\frac{\sqrt{2\times 9.81}}{2}$. B=1 is taken as C=1. The pole's place of the tank system will be determined with respect to the K parameter. But the value of K parameter will be determined with respect to places of the poles of the tank system. Then, the closed loop characteristic equation is obtained as $det(Is - A_{cl}) = \big(s - (A - BK)\big)(s) + BC = \big(s - (-1{,}373 - 31{,}847K)\big)(s) + 31{,}847$. When this equation is expanded, it is arranged as $det(Is - A_{cl}) = s^2 + (s + 1{,}373 + 31{,}847K)s + 31{,}847$. Normally the characteristic equation of the second order system is $s^2 + 2\zeta w_n s + w_n^2 = 0$. When these equations are equalized, the natural frequency of the system becomes $w_n = 5{,}643$. For the design, the $T_{settling}$ value was chosen as 1 sec, 20 times the sampling time. For this design, the damping coefficient of the system becomes $T_{settling} = \frac{4}{\zeta w_n} = 1sec \rightarrow \zeta w_n = 4$. In the characteristic equation, $\zeta = 0{,}708$ is obtained from the equation $1{,}373 + 31{,}847K = 2\zeta w_n = 8 \rightarrow K = 0.208 = 2$. Since $0 < \zeta < 1$, the system will be controlled as oscillating damped.

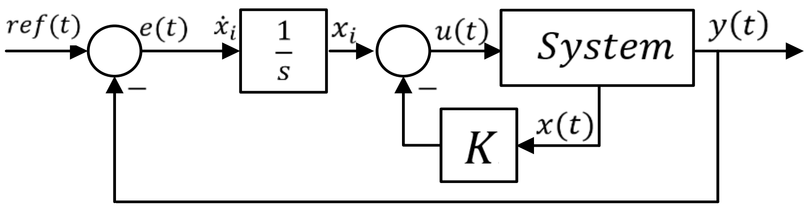

Figure 4. Control Structure with Single Tank Liquid Level System integrator with state feedback.

2.6. Adaptation of Single Tank Liquid Level System to Q Learning and SARSA Controller

When the single-tank system is considered an environment for reinforcement learning, expressions or symbols could be matched each other. For instance, action a(t) applied to the system, that is control signal u(t) or states s_t of Q learning and SARSA might be matched to the states of the system to be controlled. Therefore, the states are given in Equation 17 so that the method can be applied to algorithms. In addition, the reward function obtained from the environment is specified in Equation 18. Nevertheless, it is useful to know that these states and reward functions could be changed as to designer.

$$s(t) = \big(ref, x(t)\big) \tag{17}$$

$$reward = \begin{cases} 15 - 200 \times |ref - x(t)| - a(t)^2 & |ref - x(t)| \\ 5 - 200 \times |ref - x(t)| - a(t)^2 & other \end{cases} \tag{18}$$

Due to the controller running on the Q table, the Q table of the system was created according to the actions and states. The reference range was chosen between $[0{,}5]$ and increased in steps at certain step lengths. $x(t)$ state is selected between $[0{,}2]$ and increased with a certain step length and the q table is created. A small part of the created table is presented in Table 2. At first, before the learning process, a 0 value was assigned to all variables of the Q table. As seen in Table 2, some values are still 0 as seen in the Q table of the agent trained for 200000 steps. Therefore, the agent recognizing the environment shows that it does not undergo every state, that is, not every state occurs in the environment. In addition, in Table 3, common parameters for SARSA and Q learning have been presented. Also, the maximum number of iterations $(Iter_{max})$ for which the algorithms are run is. The parameters of the ε variable used in the epsilon greedy algorithm are presented in Table 3 as the initial value (ε_{start}), the decreased value $(\varepsilon_{decrement})$ and the minimum value (ε_{min}). In addition, the minimum$(ref_{min}, x(t)_{min})$, maximum $(ref_{max}, x(t)_{max})$ and step lengths $\big(ref_{step\ size}, x(t)_{step\ size}\big)$ of discretized $ref(t)$ and $x(t)$ state values have been presented in Table 3. Reward values of Q learning and SARSA algorithm trained to control the single-level tank system are given in Figure 5 by plotting according to iterations. In the beginning, it could be seen that the reward value fluctuated and then increased due to the epsilon decreased in the greedy algorithm during iteration. At last, reward values have been fixed in both algorithms.

Table 2. A sample Q table with Q learning trained over 200000 iterations for controlling the Single Tank Liquid Level System.

State		Action				
$ref(t), x(t)$		0	0.05	0.1	7.95	8
Real	Discrete					
$0, -0.1$	0, 0	0	0	0	0	0
$0, -0.079$	0, 1	0	0	0	0	0
0.1,0.005	0, 5	-115.91	74.56	97.84	-550165.34	-539846.01
0.1,0.026	1, 6	0	0	0	0	0
0.1,0.047	1,7	0	0	0	0	0
0.3,0.383	3,23	349.43	-0.13	-0.31	-3215.11	-3184.24
0.3,0.425	3,24	345.86	-0.58	-0.57	-3223.79	-3268.76
0.8,0.782	8,42	13.13	18.39	23.42	-3653.06	-2483.30
0.8,0.782	8,43	-0.56	-0.39	-0.32	0	0
0.9,0.782	9,43	0	0	0	0	0
0.9,0.782	8,43	-0.56	-0.39	-0.32	0	0

Table 3. Parameters for SARSA and Q learning.

Parameters	Value
$Iter_{max}$	195000
ε_{start}	1.0
ε_{min}	0.001
$\varepsilon_{decrement}$	$\varepsilon_{start}/Iter_{max}$
ref_{min}	0.
ref_{max}	5
$ref_{step\ size}$	0.1
x_{min}	-0.1
x_{max}	2.0
$x_{step\ size}$	0.0209

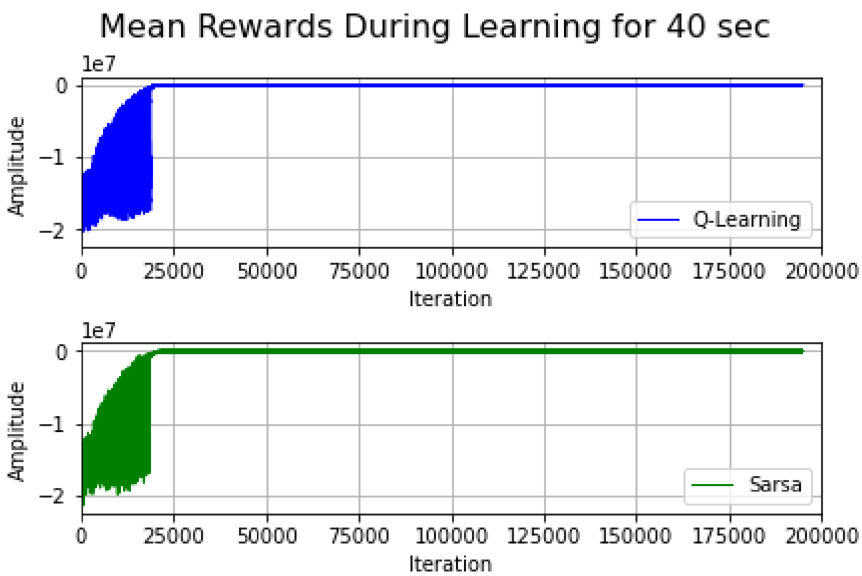

Figure 5. Reward values of Single Tank Liquid Level System by PI, modified PI, state feedback with integral action, Q Learning and SARSA.

3. Simulation Results

The simulation studies carried out in this study have been conducted on a computer with Intel(R) Core (TM) i5-9400 CPU @ 2.90GHz, 64-bit, 8GB RAM. The software language in which the study is carried out is Python and the interface is the Anaconda program. The single-level tank system to be controlled has been built as an environment in this program. Then, the designs were made to control the system with PI, modified PI, state feedback with integral action, Q learning and SARSA. After that system cosntrol has been carried out, and then the results have been obtained. The results obtained have been presented in the table in terms of performance values. In addition, the results obtained in terms of system response and control responses have been given in graphics.

The results of the single-level liquid tank system controlled by PI, modified PI, state feedback, Q learning and SARSA methods were evaluated in terms of T_{rising}, $T_{settling}$, overshoot and ISE performance criteria. The numerical results, in Table 4, for single-level liquid tank system have been demonstrated. The best results among methods are written in bold. When the results are examined, the Q learning algorithm has produced better results than the others with regard to T_{rising}, $T_{settling}$ and ISE. However, when examined in terms of overshoot, it can be seen that SARSA algorithm indicated better results than classical controllers and the Q learning methods. In addition, the step response graph, control signal and reward values of the controlled system have been depicted in graphics. In Figure 6, system responses of the single-level tank system to the step reference input have been depicted. When the responses of the Q learning and SARSA methods have been examined, it can be noticed that although the system has reached the desired reference value faster, it has not remained at the desired reference value or have fluctuated in its responses compared to the classical control methods, that is, it has deviated and come back again. The prime cause for this is that the system is continuous and the controller that controls the system is discretized according to certain step lengths. In addition, control signs have been depicted in Figure 7. Fluctuating in the control signal could be seen more easily. Another reason could be that there is epsilon greedy in the structure that controls the system in Q learning and SARSA algorithms. This method, which is used for discovery, could sometimes lead to the selection of a different control signal within the solution pool. On the other hand, the reward values, that have been obtained after the single tank liquid level system have been controlled, have been indicated in Figure 8 according to time.

Table 4. Performance results of controlling Single Tank Liquid Level System

	T_{rising} (sec)	$T_{settling}$ (sec)	$Overshoot$	ISE
PI	0.0875	0.4283	%25,40	0,1968
Modifiye PI	0.1676	0.5035	%11.36	0.1168
State Feedback	0,4167	0.8180	%6,10	0.2893
Q learning	**0.0804**	**0,0943**	%1,2	**0.0574**
Sarsa	0.0874	0.0943	**%0,7**	0,0590

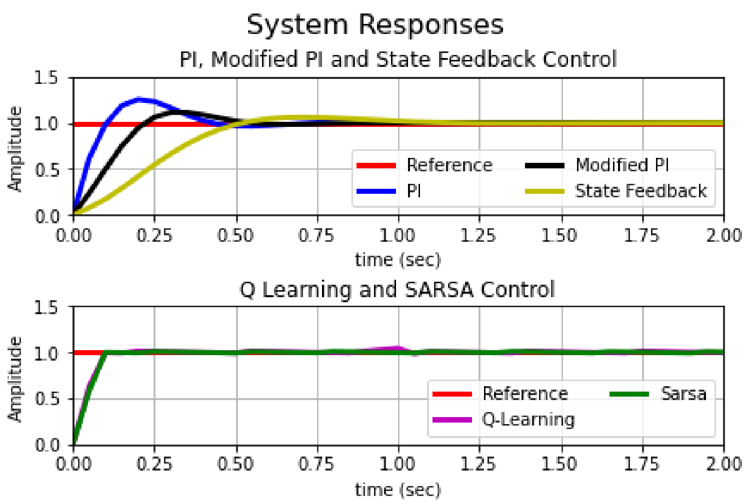

Figure 6. System responses of single tank liquid level System by PI, modified PI, state feedback with integral action, Q Learning and SARSA.

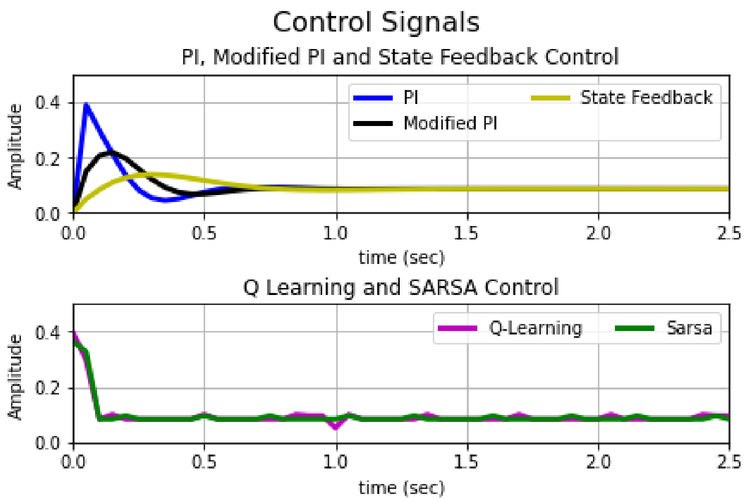

Figure 7. Control signals of Single Tank Liquid Level System by PI, modified PI, state feedback with integral action, Q Learning and SARSA.

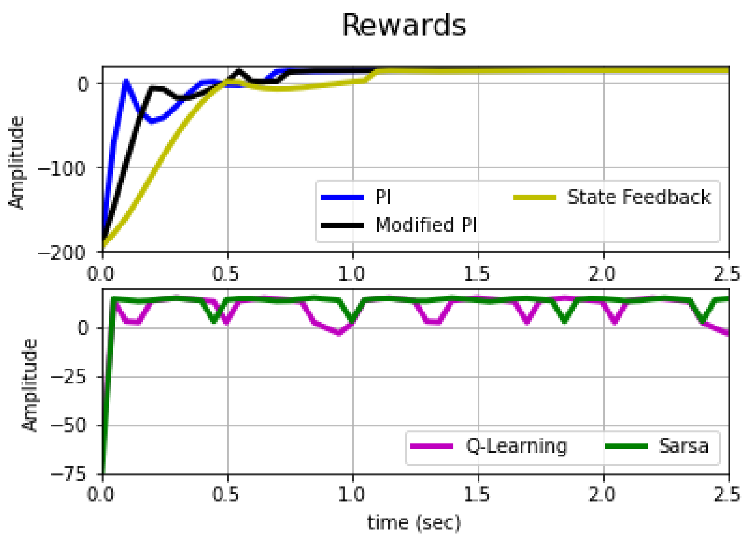

Figure 8. Reward values of Single Tank Liquid Level System by PI, modified PI, state feedback with integral action, Q Learning and SARSA.

The system has been controlled at several reference values in order to better see the performances of controllers for the Single Tank Liquid Level System. As can be seen, Q learning and SARSA methods have generally produced faster results than PI, Modified PI and State Feedback with integral action. However, looking at results for SARSA, it can be noticed that there are chattering or fluctuations when the reference value is 0.5. To prevent this, agents could be trained more or the size of Q table would be increased by reducing the step lengths of the states while creating the Q table. However, the decrease in step length is a tradeoff that increases memory and processing load. In this study, since the Q learning and SARSA methods work discrete, system response and control signal can be aggressive when step length is large. In addition, since it is operated discretely, the performances of the controllers may vary at reference signals at different points. The biggest constraint at this point is memory and processing time. However, in future studies, soft actor-critic, deep q learning, double deep q learning methods that work continuously with artificial neural networks will be used to overcome memory size constraints.

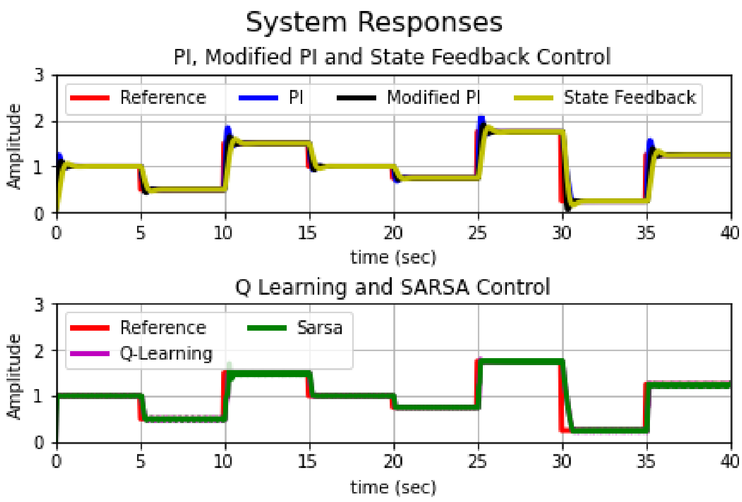

Figure 9. System responses of Single Tank Liquid Level System by PI, modified PI, state feedback with integral action, Q Learning and SARSA for different reference signals.

4. Conclusions

In this study, classical and reinforcement learning-based controllers have been designed to control the nonlinear single-level tank system. First, the system has been linearized and the PI controller has been designed with the classical Ziegler Nichols method. Then, Modified PI has been applied and state feedback with integral action controllers were designed with pole assignment. Then, agents have been trained for SARSA and Q learning algorithms on the Single Tank Liquid Level System which is an environment and also this is a nonlinear system. After that, trained agents have applied to the system to be controlled. The results of the nonlinear single-level liquid tank system controlled by these methods have been assessed about rising time (T_{rising}), settling time ($T_{settling}$), overshoot and ISE performance indexes. When the results have been examined, it can be noticed that the Q learning algorithm has produced better results with regard to rise time (T_{rising}), settling time ($T_{settling}$) and ISE. Furthermore, it has been noticed and concluded that, due to SARSA and the Q learning used in the study being discrete, there might be memory size difficulty that caused chattering or fluctuating problems of system responses in this method. In future studies, softened control signals could be improved in transitions between states, and deep learning-based reinforcement learning methods, which are popular topics, are going to be used.

Declaration of Ethical Standards

If the study does not require an Ethics Approval, the following declaration can be used:

"The author(s) of this article declare that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission."

Otherwise, please declare the board name granting ethics approval, approval date and approval number.

Conflict of Interest

All conflicts of interest should be declared. If there is not any conflict of interest, the content of this section can be arranged as follows.

"The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper."

Acknowledgements

Acknowledgements should be expressed in a separate section before the references.

References

[1] M. A. Pala, M. E. Çimen, Ö. F. Boyraz, M. Z. Yildiz, and A. Boz, 2019. Meme Kanserinin Teşhis Edilmesinde Karar Ağacı Ve KNN Algoritmalarının Karşılaştırmalı Başarım Analizi. *Acad. Perspect. Procedia*, **2**(3). doi: 10.33793/acperpro.02.03.47.

[2] S. Tajjour, S. Garg, S. S. Chandel, and D. Sharma, 2023. A novel hybrid artificial neural network technique for the early skin cancer diagnosis using color space conversions of original images. *Int. J. Imaging Syst. Technol.*, **33**(1), pp. 276–286.

[3] D. Şengür, 2021. EEG, EMG and ECG based determination of psychosocial risk levels in teachers

based on wavelet extreme learning machine autoencoders. *Politek. Derg.*, **25**(3), pp. 85–989, 2021, doi: 10.2339/politeknik.886593.

[4] F. Bayram, 2020. "Derin öğrenme tabanlı otomatik plaka tanıma," *Politek. Derg.*, **23**(4), pp. 955–960.

[5] J. Lisowski, 2023. Artificial Intelligence Methods in Safe Ship Control Based on Marine Environment Remote Sensing. *Remote Sens.*, **15**(1), p. 203. doi: https://doi.org/10.3390/rs15010203.

[6] A. Kurani, P. Doshi, A. Vakharia, and M. Shah, "A comprehensive comparative study of artificial neural network (ANN) and support vector machines (SVM) on stock forecasting," *Ann. Data Sci.*, vol. 10, no. 1, pp. 183–208, 2023, doi: https://doi.org/10.1007/s40745-021-00344-x.

[7] M. S. Ünlü, "Teknik Analiz Ve Derin Pekiştirmeli Öğrenme İle Kriptopara Alım-Satımı," Okan Üniversitesi, 2019.

[8] X. Huang, D. Zou, G. Cheng, X. Chen, and H. Xie, "Trends, research issues and applications of artificial intelligence in language education," *Educ. Technol. Soc.*, vol. 26, no. 1, pp. 112–131, 2023, doi: 10.30191/ETS.202301_26(1).0009.

[9] K. Souchleris and G. A. Sidiropoulos, G. K. Papakostas, "Reinforcement Learning in Game Industry—Review, Prospects and Challenges," *Appl. Sci.*, vol. 13, no. 4, p. 2443, 2023, doi: 10.3390/app13042443.

[10] F. Candan, S. Emir, M. Doğan, and T. Kumbasar, "Takviyeli Q-Öğrenme Yöntemiyle Labirent Problemi Çözümü Labyrinth Problem Solution with Reinforcement Q-Learning Method," in *TOK2018 Otomatik Kontrol Ulusal Toplantısı*, 2048.

[11] G. Mbuwir, B., Ruelens, F., Spiessens, F., & Deconinck, "Reinforcement learning-based battery energy management in a solar microgrid," *Energy-Open*, vol. 2, no. 4, p. 36, 2017.

[12] M. Harmon and S. Harmon, "Reinforcement Learning: A Tutorial," 1997. [Online]. Available: https://apps.dtic.mil/sti/pdfs/ADA323194.pdf

[13] S. A. Ğ. Reyhan and Z. H. Tuğcu, "Akıllı Şebeke Uygulamalarında Derin Öğrenme Tekniklerinin Kullanımına İlişkin Kısa Bir İnceleme," *EMO Bilim. Dergi*, vol. 13, no. 1, pp. 41–61, 2022, [Online]. Available: https://dergipark.org.tr/en/pub/emobd/issue/75563/1196333

[14] I. Tunc and M. T. Soylemez, "Fuzzy logic and deep Q learning based control for traffic lights," *Alexandria Eng. J.*, vol. 67, pp. 343–359, 2023, doi: 10.1016/j.aej.2022.12.028.

[15] A. Leite, M. Candadai, and E. J. Izquierdo, "Reinforcement learning beyond the Bellman equation: Exploring critic objectives using evolution," in *Artificial Life Conference Proceedings 32*, 2020, pp. 441–449.

[16] I. C. Dolcetta and M. Falcone, "Discrete dynamic programming and viscosity solutions of the Bellman equation," in *In Annales de l'Institut Henri Poincaré C, Analyse non linéaire*, 1989, pp. 161–183.

[17] C. Boutilier, R. Reiter, and B. Price, "Symbolic dynamic programming for first-order MDPs," in *IJCAI International Joint Conference on Artificial Intelligence*, 2001, vol. 1, pp. 690–697. doi: 10.1609/aaai.v24i1.7747.

[18] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley, 2007.

[19] D. Michie, "Experiments on the mechanization of game-learning Part I. Characterization of the model and its parameters," *Comput. J.*, vol. 6, no. 3, pp. 232–236, 1963.

[20] M. L. Minsky, *Theory of neural-analog reinforcement systems and its application to the brain-model problem*. Princeton University, 1954.

[21] J. Karlsson, "Learning to solve multiple goals," University of Rochester, 1997.

[22] A. L. Samuel, "Some studies in machine learning using the game of checkers. II—Recent progress," *Annu. Rev. Autom. Program.*, vol. 6, pp. 1–36, 1969.

[23] C. J. C. H. Watkins, "Learning from delayed rewards," King's College UK, 1989.

[24] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992.

[25] G. Tesauro, "Neurogammon: A neural-network backgammon program," in *JCNN international joint conference on neural networks IEE*, 1990, pp. 33–39. doi: 10.1109/IJCNN.1990.137821.

[26] G. Tesauro, "Practical issues in temporal difference learning," *Adv. neural Inf. Process. Syst.*, vol. 4, 1991.

[27] C. Ozan, "İyileştirilmiş pekiştirmeli öğrenme yöntemi ve dinamik yükleme ile kentiçi ulaşım ağlarının tasarımı," Pamukkale Üniversitesi, 2012.

[28] M. K. Çalışır, S., & Pehlivanoğlu, "Model-free

reinforcement learning algorithms: A survey," in *27th Signal Processing and Communications Applications Conference (SIU)*, 2019, pp. 1–4.

[29] A. O. Köroğlu, A. E. Edem, S. N. Akmeşe, Ö. Elmas, I. Tunc, and M. T. Soylemez, "Agent-Based Route Planning with Deep Q Learning," in *13th International Conference on Electrical and Electronics Engineering (ELECO)*, 2021, pp. 403–407.

[30] Y. Li, "Deep reinforcement learning: An overview," *arXiv Prepr. arXiv1701.07274*, 2017, doi: https://doi.org/10.48550/arXiv.1701.07274.

[31] A. Bir and M. Kacar, *Pioneers of Automatic Control Systems*. 2006.

[32] M. E. Çimen, Z. Garip, M. Emekl, and A. F. Boz, "Fuzzy Logic PID Design using Genetic Algorithm under Overshoot Constrained Conditions for Heat Exchanger Control," *J. Inst. Sci. Technol.*, vol. 12, no. 1, pp. 164–181, 2022, doi: 10.21597/jist.980726.

[33] ME, Cimen, and Y. Yalçın, "A novel hybrid firefly–whale optimization algorithm and its application to optimization of MPC parameters," *Soft Comput.*, vol. 26, no. 4, pp. 1845–1872, 2022, doi: 10.1007/s00500-021-06441-6.

[34] Z. Mizumoto, I., Ikeda, D., Hirahata, T., & Iwai, "Design of discrete time adaptive PID control systems with parallel feedforward compensator," *Control Eng. Pract.*, vol. 18, no. 2, 2010, doi: https://doi.org/10.1016/j.conengprac.2009.09.003.

[35] D. Taler, T. Sobota, M. Jaremkiewicz, and J. Taler, "Control of the temperature in the hot liquid tank by using a digital PID controller considering the random errors of the thermometer indications," *Energy*, 2022, doi: https://doi.org/10.1016/j.energy.2021.122771.

[36] R. E. Samin, L. M. Jie, and M. A. Zawawi, "PID implementation of heating tank in mini automation plant using Programmable Logic Controller (PLC)," in *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE)*, 2011.

[37] G. Yüksek, A. N. Mete, and A. Alkaya, "PID parametrelerinin LQR ve GA tabanlı optimizasyonu: sıvı seviye kontrol uygulaması," *Politek. Derg.*, vol. 23, no. 4, pp. 1111–1119, 2020, doi: 10.2339/politeknik.603344.

[38] N. A. Selamat, F. S. Daud, H. I. Jaafar, and N. H. Shamsudin, "Comparison of LQR and PID Controller Tuning Using PSO for Coupled Tank System," in *11th International Colloquium on Signal Processing & Its Applications (CSPA)*, 2015.

[39] D. Sastry, K. Mohan, M. Naidu, and N. M. Rao, "An Implementation of Different Non Linear PID Controllers on a Single Tank level Control using Matlab," *Int. J. Comput. Appl. (*, vol. 54, no. 1, 2012.

[40] and Y. S. Wei, Le, Fang Fang, "Adaptive backstepping-based composite nonlinear feedback water level control for the nuclear U-tube steam generator.," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 1, 2013, doi: 10.1109/TCST.2013.2250504.

[41] Q. Xiao, D. Zou, and P. Wei, "Fuzzy Adaptive PID Control Tank Level," in *International Conference on Multimedia Communications*, 2010. doi: 10.1109/MEDIACOM.2010.10.

[42] C. Esakkiappan, "Soft Computing Based Tuning of PI Controller With Cuckoo Search Optimization For Level Control of Hopper Tank System," *Res. Sq.*, 2021, doi: https://doi.org/10.21203/rs.3.rs-920228/v1.

[43] N. N. Son, "Level Control of Quadruple Tank System Based on Adaptive Inverse Evolutionary Neural Controller," *Int. J. Control. Autom. Syst.*, vol. 18, no. 9, 2020, doi: 10.1007/s12555-019-0504-8.

[44] C. Urrea and F. Páez, "Design and Comparison of Strategies for Level Control in a Nonlinear Tank," *Processes*, vol. 9, 2021, doi: 10.3390/pr9050735.

[45] R. S. Sutton and G. A. Barto, *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 1998.

[46] M. Rummery, G. A., & Niranjan, *On-line Q-learning using connectionist systems*. Cambridge, UK: University of Cambridge, 1994.

[47] C. J. Watkins and P. Dayan, "Q-Learning," *Mach. Learn.*, vol. 8, pp. 279–292, 1992, doi: 10.1007/BF00992698.

[48] A. Wang, H., Emmerich, M., & Plaat, "Monte Carlo Q-learning for General Game Playing," *arXiv Prepr. arXiv1802.05944.*, doi: 10.48550/arXiv.1802.05944.

[49] K. L., R. Bars, J. Hetthéssy, and C. Bányász, *Control Engineering*. Springer, 2019. doi: 10.1007/978-981-10-8297-9.

[50] J. Cimbala and Y. Cengel, *Fluid mechanics: fundamentals and applications*. McGraw-Hill Higher Education, 2006.