1. Szymon BRZEZIŃSKI, 2. Krzysztof STEBEL
ORCID: 2. 0000-0003-2912-6742

DOI: 10.15199/48.2025.05.10

Q-Learning algorithm for PI controller autotuning

Algorytm Q-Learning do automatycznego dostrajania regulatora PI

Abstract. *This paper presents an approach for PI controller autotuning using Q-Learning algorithm. Gains obtained from Q-Learning are tested by simulation on the validated mathematical model of the real electric flow heater implemented in LabView and compared with conventional method for tuning PI controller.*

Streszczenie. *W artykule przedstawiono podejście do autostrojenia regulatora PI z wykorzystaniem algorytmu Q-Learning. Nastawy uzyskane dzięki Q-Learning są testowane poprzez symulację na zweryfikowanym modelu matematycznym rzeczywistego elektrycznego podgrzewacza wody zaimplementowanego w LabView i porównane z konwencjonalnymi metodami strojenia regulatora PI.*

Keywords: Q-learning, reinforcement learning, process control, PI controller
Słowa kluczowe: Q-learning, uczenie ze wzmocnieniem, sterowanie procesami, regulator PI

Introduction

Over the years, various control algorithms have been developed, but the most commonly used algorithm is the PID algorithm due to the low number of tuning parameters, easy implementation, and relatively low cost. Despite its widespread use, the PID controller has limitations in handling complex, nonlinear systems or processes with significant time delays. Advanced control techniques such as model predictive control (MPC) and fuzzy logic control have emerged to address these challenges. These modern control strategies offer improved performance and flexibility, particularly in industries with stringent control requirements like chemical processing and robotics. The most frequently used is the PI structure of the controller [1]. Although using derivative term enhances quality of control but for processes with high levels of noise the use of a PI controller is a better choice. Simple tuning methods are based on simplified models that use some kind of approximation of process dynamics for example FOPDT and SOPDT models [2]. These simplified models are obtained by processing response data of the process. In the industry, QDR rules are often applied for first order systems with dead time. Unfortunately these identification experiments take time and can cause significant bottlenecks in production not to mention economic aspect of conducting these experiments and the involvement of experts. Because of that industrial controllers are far from being properly tuned or operating with default settings. These simplified tuning methods, while widely used, have limitations in capturing the full complexity of process dynamics, especially for higher-order systems. Alternative approaches, such as model-based tuning or adaptive control strategies, can potentially offer more precise and robust control for complex industrial processes. However, the implementation of these advanced techniques frequently requires specialized knowledge and resources, which may not always be readily available in industrial circumstances. In result of that process control performance decreases which can lead to increase of energy consumption and production efficiency. In this paper, Q-learning algorithm is proposed for autotuning PI controller. Different applications of Q-learning in process control have been proposed. Musial et al.[3], propose to use Q- learning as self-improving controller and in [4] implementation aspects of Q-learning controller. Lam et al. [5] proposed an adaptive proportional-integral-derivative controller based on Q-learning algorithm to balance the cart-pole system. Syafiie et al. [6]. implementation of Q-learning algorithm for neutralisation control. This work was important in terms of applying this methodology to continuous systems. Examples of implementation of Q-learning can be found in robotics and avionics [7].

Q-learning: A brief overview

Machine learning has a wide range of applications from the very general to the very specific [9, 10]. Q-learning is a model-free reinforcement learning algorithm widely used in machine learning [11] and optimisation problems. It is based on trial and error learning. Algorithm revolves around the reward/punishment policy which provides optimal solution even for dynamical problems for which accurate model is unknown. The problem can be defined as agent (controller) and environment (plant) interaction shown in figure 1.

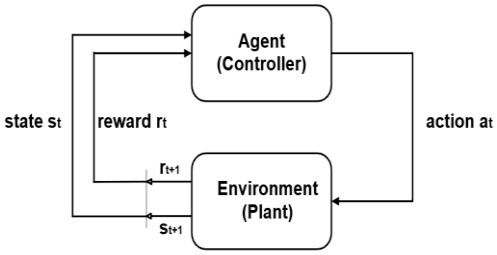

Fig. 1. Schematic diagram of Q-learning algorithm [10].

Q-learning learns directly from this interaction. The Q-learning algorithm is an off-policy value-based learning algorithm. The learned action value function Q, directly approximates the optimal action-value. In general policy can be described as iterative formula using equation (1):

$$Q_\pi(s_t,a_t) \leftarrow Q_\pi(s_t,a_t) + \alpha_t\left[r_{t+1} + \gamma \max_{b \epsilon A_{s_{t+1}}} Q_\pi(s_{t+1},b) - Q_\pi(s_t,a_t)\right] \quad (1)$$

Where t is discrete time, s and a respectively denote state and action that should be taken at a given state, $Q(s,a)$ is the value of Q-matrix that represents reward for taking action a when the system is in the states. In Equation (1), $Q_\pi(s_t,a_t)$ is the value before update, $Q(s_{t+1},a_{t+1})$ is the state to which the system will move from $Q(s_t,a_t)$, α [0,1] is the learning rate, γ [0,1] is the discount factor. Parameters α and γ are considered as tuning parameters of Q-learning

algorithm. Reward *R* is assigned if the system is at the goal state or not. In this implementation if the goal state is reached then *R* = 1 is applied otherwise *R* = -1. Random actions are taken to force exploration and check if actions taken so far can be considered as optimal should be updated. In case of reinforcement learning general scenario follows these procedures shown in figure 2:

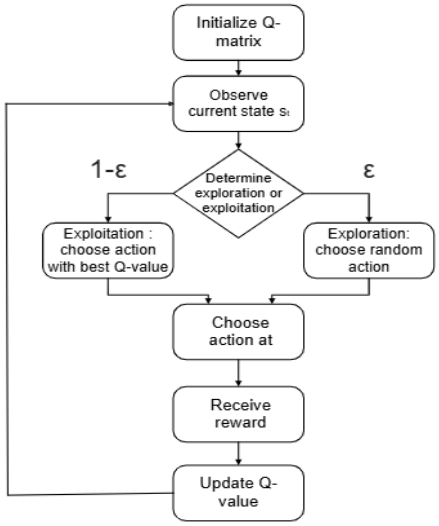

Fig. 2. Schematic diagram of Q-learning algorithm.

Electric flow heater

In order to test and verify gains of PI controller obtained from Q-learning algorithm mathematical model of the real electric flow heater was used shown in figure 3.

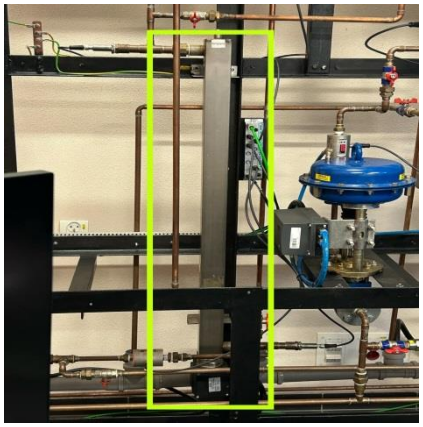

Fig. 3. Picture of the electric flow heater.

Water flows through the heater with the flow F_1 [L/min] and inlet temperature T_{IN} [℃]. The power supply P_H [%] can be manipulated in range from 0-100 [%], nominal power P_{NOM} of the heater is equal to 12 [kW]. Hot water flows out with the same flow and the outlet temperature T_{OUT} [℃]. The volume of the heating chamber is constant and equals V = 1.6 [L]. Model was derived on the balance of mass and flow based on the article [2]. T

$$\frac{dT_{OUT}(t)}{dt} = \frac{F_1(t)}{60 \cdot V} \cdot \left(T_{IN}(t) - T_{OUT}(t)\right) + \frac{P_{NOM} \cdot P_h(t-T_0)}{100 \cdot V \cdot \rho \cdot c_w} \tag{2}$$

where : c_w– specific heat capacity J/kg℃, ρ – density, kg/l, T_0 – dead time.

In order to simulate dynamics of the equation (2) Numerical method was used with sampling period *h* = 1 [s]. Before tuning PI controller using QDR rules it is necessary to approximate important process parameters such as :

$$T_0 = 0.1 \cdot T, \ s \tag{3}$$

$$T = \frac{60 \cdot V}{F_1(t)}, \ s, \tag{4}$$

$$k = \frac{\Delta y}{\Delta u}, \ \frac{℃}{\%} \tag{5}$$

Where (4) is time constant expressed in seconds and (5) is process gain. Both parameters are dependent on flow $F_1(t)$. Simulation experiment was conducted to obtain these parameters. During the initial experiment flow F_1 was equal to 2 [L/min] and power supply was set to 50% of the nominal power. Inlet temperature T_{IN} is set in the simulator to 15 [℃]. Dynamics of the plant are shown in figure 4.

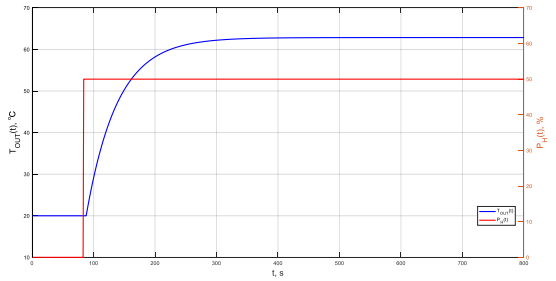

Fig. 4. Time response of the plant.

Using equations (3), (4) and (5), it is possible to estimate parameters of the plant. Dead time is equal to 5 [s], process gain is equal to 0,8335 [℃/%] and time constant equals 51 [s].

Control system

In control theory control performance of the closed loop system usually is evaluated by control error $e = Y_{sp} - Y$ and other indexes such as maximum overshoot, settling time or control signal trajectory. The last one is particularly important because oscillatory behaviour of control signal can be the reason of damaging process instrumentation and it is unwanted A simple way to tune PI parameters is based on step response of the system using FO (first-order) model. Equation (6) provides the output of the controller in discrete time form :

$$U(i) = K_R \cdot \left(e(i) + \frac{1}{T_I} \cdot \sum_{i=0}^{i=\infty} e(i) \cdot h\right) \tag{6}$$

Where: K_R – proportional gain, T_I – integration time, *e(i)* – control error, *U(i)* – control signal, *h* – sampling period. In order for control loop working as intended sampling period of controller must be the same as sampling period of simulator of the heating unit.

Statement of the problem

Because of poorly tuned control systems it is necessary to implement autotuning for industrial controllers for example algorithms based on reinforcement learning. As a novelty, this paper proposes a Q-learning algorithm modified in such a way that it is able to automatically tune the PI controller applied for control of the heating unit. The control goal is to keep process output *Y* (in this case T_{OUT})

equal to the desired setpoint T_{SP} by adjusting manipulating variable P_h in the presence of disturbance *d*. The problem of accurate tuning is time-consuming and over time properties of the system can change. It is possible to distinguish two major causes. One is change of operating point - this is fast change. Second one is the aging of the system instrumentation for example sediment on the heating unit. In this case process can be disturbed by a poorly tuned flow control system as mentioned before time constant (3) and process gain (4) are dependent on flow therefore any change of flow implies a change in the system dynamics. *A schematic* diagram of the proposed control system with Q-learning is shown in figure 5

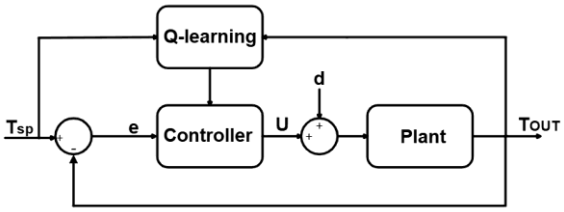

Fig. 5. Diagram of proposed control system with Q-learning.

Goal state is defined as $S = (|T_{OUT}| \leq T_{SP} +/- 0.1 \cdot T_{SP})$. But this does not ensure good control performance because over-regulations and oscillations can occur in transient states to prevent this situation reward function was modified so that not only one sample is taken into the condition of reward function but number of next samples. In general reward function is defined as equation (7) below:

$$R = \begin{cases} 1, & |T_{OUT}| \leq T_{SP} +/- 0.1 \cdot T_{SP} \\ -1, & otherwise \end{cases} \tag{7}$$

Potential implementation of suggested approach was tested by simulation for two examples. One for PI controller tuned with QDR method and one autotuned with Q-learning.

Simulations results

Both simulation experiments were conducted in the same way meaning: power supply is turned off, flow F1(t) is set to 2 [L/min] and inlet temperature is set to 15 [℃] all these inputs are set in the simulator. Setpoint is set to 45 [℃]. After setting these values controller was switched to automatic mode thus power supply turns on to minimize control error.

It is possible that during learning phase algorithm can reach constraints of the control signal. In order to avoid this situation algorithm lowers the value of K_R when the control signal saturates.

Before autotuning PI controller gains were initialized as follows: $K_R = 0.5$, $T_I = 20$ [s]. These gains are also initial conditions from which algorithm will start learning. Gains obtained using the QDR method: $K_R = 3.8$, $T_I = 46.9$ [s]. These gains are the reference to compare performance between conventional and known tuning methods and auto-tuning with Q-learning. In order for algorithm to learn and tune controller it was necessary to define two intervals for PI parameters. One for K_R (8) and one for T_I (9).

$$K_R \in [0.5, 0.7, \ldots, 4.3, 4.5], \tag{8}$$

$$T_I \in [20, 20.2, \ldots, 59.8, 60]. \tag{9}$$

Instead of exploring through entire intervals algorithm chooses values of K_R and T_I which are close to current values. Therefore three actions can be defined as :

- Increase value
- Stay at the current value
- Decrease value

Increasing or decreasing value means changing current table index by 1 or -1 this is equivalent to increasing or decreasing values of K_R and T_I by 0.2. Values of K_R and T_I are changed at the same time meaning if increase or decrease action is selected then both values are changed. This approach introduces local exploring, this also prevents jumps between values. Gives algorithm a chance to stop at the current values if it decides that these values are optimal. It also improves optimization process, because it avoids unnecessary changes. States in this study are defined as current values of K_R and T_I. The Q values are typically initialized as zeros or for example are based on the controller in operation [1], [5]. Other and often effective way for some environments might benefit from a small random start value to encourage initial exploration [11]. In this study of Q-learning the Q matrix is initialized with zeros because algorithm has no knowledge about process at the beginning [12]. During learning process Q-matrix stores Q-values for sets of K_R and T_I. In this implementation optimal values of K_R and T_I are chosen for the maximum value in the matrix Q. If the state of the process is not a goal state some action (increase the value, decrease the value, do not change the value) must be taken by the algorithm. At the beginning this action is random because Q-matrix stores zeros. After taking a random action Q-matrix is modified meaning that agent was rewarded or punished for this action. The procedure is repeated until the goal state is reached, this is one episode of learning. It is also crucial to briefly describe exploration and exploitation phases. Exploration means that the algorithm takes an action, therefore checking new combinations of control parameters. Exploration is crucial in the early stages of learning where the agent has to adapt to changes. During the exploitation phase algorithm focuses on utilizing known information to make decisions that yield the highest reward according to the policy. Proposed algorithm follows procedures shown in figure 6.

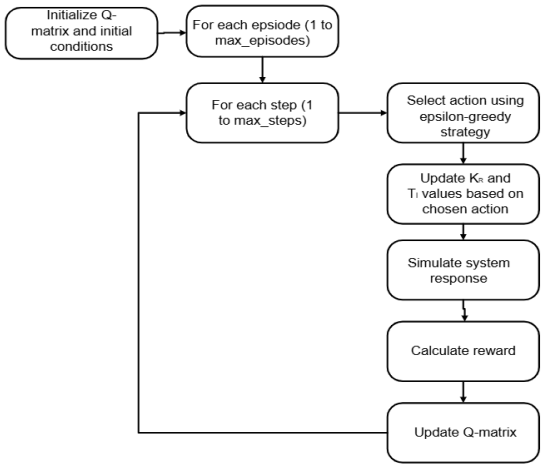

Fig. 6. Schematic diagram of proposed algorithm.

It is also important to mention three important parameters: learning rate α , discount factor γ and

probability ε. Learning rate determines to what extent newly acquired information overrides old information. Learning rate equal to 0 makes agent learn from only prior knowledge while learning factor equal to 1 makes agent learn from the most recent data ignoring prior knowledge to explore possibilities. Probability ε helps the agent decide which action to take based on the current Q-values.

Value of learning rate is dependent on environment and problem. The discount factor γ determines the importance of future reward. Discount factor of 0 will make the agent consider only current rewards while a factor closer to 1 will make the agent to aim for long term high reward. Unfortunately there are no deterministic methods on how to optimally tune Q-learning algorithm. Therefore the values are based on gradual learning and long term reinforcement [6]. The influence of parameters on control performance was tested for three different values and is presented in figure 7.

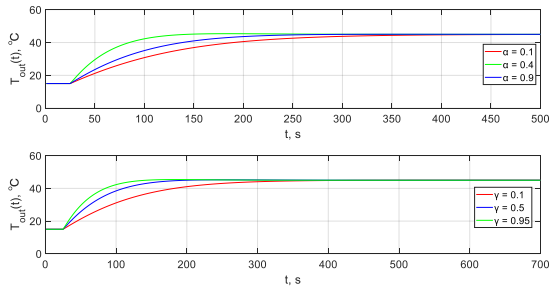

Fig. 7. The influence of parameters on control performance.

The most significant impact on control performance has γ parameter. For γ = 0.1 control performance deteriorated very quickly – settling time is much longer this could mean that the learning process is slower. This means that low values of γ are not desirable. Learning rate α also impacts control performance but not as drastically as discount factor γ. All these parameters influence learning process in a different way because of that for further experiments the parameters in Q-learning are designed as follows: α = 0.4, γ = 0.95. The ε-greedy policy with the probability ε = 0.3 is utilized. Learning process for 50 episodes is shown in figure 8.

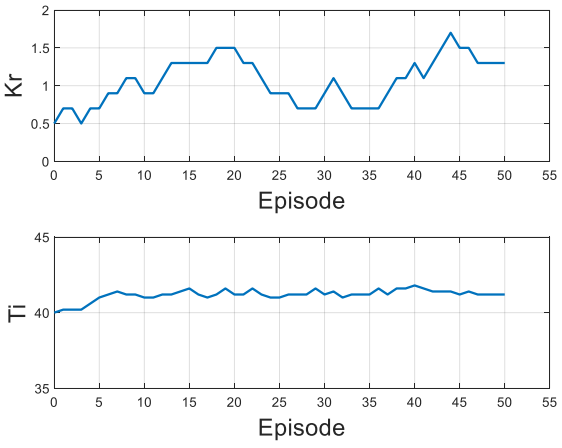

Fig. 8. The update process of control parameters for 50 episodes.

It is significant that to point out that the proposed tuning method tries to increase and decrease the initial values of K_R and T_I before 10-th episode. Algorithm then learned that the lower values K_R and T_I of mean worst performance. Optimization process of T_I values is stable and the value of T_I increases. After 46 learning periods values settle and remain constant until the end of simulation. In total three sets of parameters were obtained from Q-learning algorithm each set for different number of episodes. Parameters shown in table 1.

Table 1. Parameters of the PI controller

Lp.	K_R	T_I, s
QDR	3.8	46.9
Initial tuning (episode = 0)	0.5	20
50 episodes	1.3	41,2
500 episodes	4,1	45,2
5000 episodes	2,5	43

Using modified IAE performance index with weights equation (10) comparison of obtained gains was conducted. The change of modified IAE performance index for twelve number of learning periods is presented in figure 9.

$$IAE = w_1 \cdot \sum_{i=0}^{i=N} e(i)^2 + w_2 \cdot \sum_{i=0}^{i=N} \Delta u(i)^2 \tag{10}$$

where: $e(i)$ – control error, $\Delta u(i)$ – control signal difference, $w_1 = 1$, $w_2 = 0.5$ – weights.

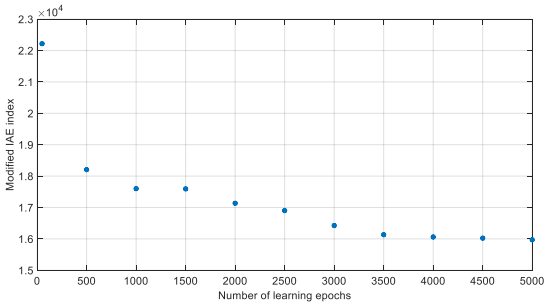

Fig. 9. The change of performance index in function of number of episodes.

For initial tuning modified IAE index is the highest meant the control performance is optimally tuned and needs to be improved. Increasing number of learning episodes improves control performance as index decreases. After 5000 episodes performance index is lower than index for QDR method this shows that Q-learning can be used for auto-tuning controllers and produce better results in comparison to conventional methods. Figures 10 and 11 present trajectories o process variable T_{OUT} and manipulated variable P_h for optimized parameters for 50, 500, 5000 episodes, QDR method and initial tunings.

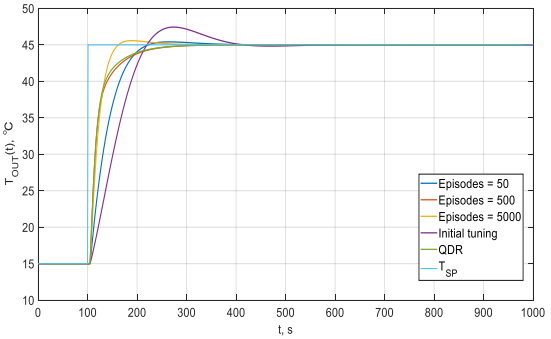

Fig. 10. Process variable T_{OUT} trajectory.

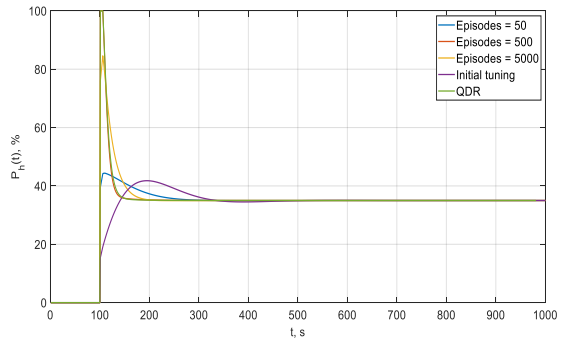

Fig. 11. Manipulated variable P_h trajectory.

Control performance gradually increases with number of learning episodes. Overshoot of 5.4% is present for initial tuning. For 500 episodes process variable trajectory is almost equivalent to trajectory obtained from QDR method. Trajectories for 50 and 500 episodes can be also deemed satisfactory as there is no overshoot and learning time is much shorter compared to 5000 episodes. It is also important to mention disturbance rejection for Q-learning algorithm. Disturbance rejection is presented in figure 12.

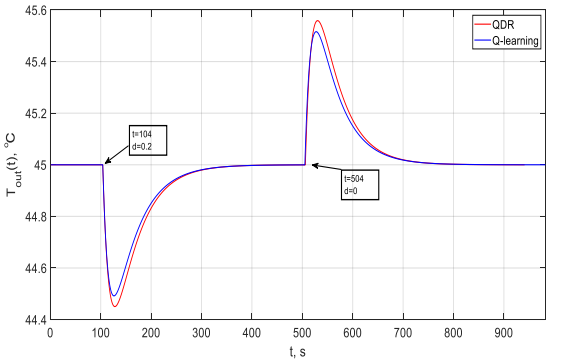

Fig. 12. Process variable T_{OUT} under process disturbance d.

Step changes of disturbance were made in steady state. The amplitude of disturbance is equal to 0.2. The disturbance was present for 400 [s]. Under process disturbance Q-learning also has advantage over conventional tuning method settling times are similar but the overshoot after step change of the process disturbance is lower for Q-learning. This fact works in favour of Q-learning for autotuning industrial controllers. Q-learning algorithm showed that for step changes of setpoint and process disturbance can yield better results than conventional tuning methods.

Conclusions

In this paper, it was shown that Q-learning algorithm can be effectively used for autotuning PI controller applied in industrial control loops. The performance of the autotuning of the PI controller was tested on model of the heating unit with comparison of conventional method of tuning industrial controllers. According to the simulation results Q-learning algorithm was able to stabilize outlet temperature even for low number of episodes with no overshoot this could mean that reward function properly defined and implemented. Another aspect is that the autotuning was performed under disturbance which present in real systems this means that Q-learning algorithm has adaptive properties.
This property can be used for example to help with tuning controllers with gain-scheduling.

However there are some drawbacks of autotuning controllers using Q-learning algorithm. First of all Q-learning algorithm is limited due to relatively long time required for effective learning for faster processes algorithm is not able to follow. Second of all further research should be conducted for adjusting parameters of Q- learning algorithm. On real systems process of autotuning can be performed. The advantage of that is the reflection of real operating conditions but it can be much more time-consuming. The most noticeable difference with existing tuning methods and Q-learning based tuning method is that the proposed Q-learning tuning method is model-free and data-driven which is capable of optimizing the controller parameters despite disturbance and complicated physical models.

***Authors**: inż. Szymon Brzeziński, Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki, ul. Akademicka 16, 44-100 Gliwice, E-mail: szymbrz356@student.polsl.pl ; dr hab. inż. Krzysztof Stebel, Politechnika Śląska, Katedra Automatyki i Robotyki, ul. Akademicka 16, 44-100 Gliwice, E-Mail: krzysztof.stebel@polsl.pl.*

REFERENCES

[1] Musiał, J., Stebel, K., Czeczot, J., "Implementation aspects of Q-learning controller for a class of dynamical processes," 2022 26th International Conference on Methods and Models in Automation and Robotics (MMAR), Międzyzdroje, Poland, (2022), pp. 382-387.
[2] Laszczyk P., Czubasiewicz R., Czeczot J., "LabView based implementation of Balance-Based Adaptive Control technique", 17th International Conference on Methods & Models in Automation & Robotics (MMAR), (2012), Miedzyzdroje, Poland, pp. 516-521.
[3] Musiał, J., Stebel, K., Czeczot, J., Nowak P., Gabrys B., Application of self-improving Q-learning controller for a class of dynamical processes: Implementation aspects, Applied Soft Computing, (2024) Vol. 152, pp.1-21.
[4] Musiał, J., Stebel, K., Czeczot, J. Self-improving Q-learning based controller for a class of dynamical processes. Archives of Control Sciences, (2021). 31, no 3, pp. 527-551.
[5] Lam H.-K., Shi Q., Xiao B., Tsai S.-H., "Adaptive PID Controller Based on Q-learning Algorithm", CAAI Transactions on Intelligence Technology, (2018), vol. 3, no. 4, pp. 235-244.
[6] Shadi M. Sargolzaei M. ”Application of reinforcement learning to improve control performance of plant”. IEEE Int. Conf Computational Intelligence for Measurement Systems and Applications (2008), Istanbul.
[7] Bagnell J.A. Schneider J.G.: “Autonomous helicopter control using reinforcement learning policy search methods”. Proc. IEEE Int. Conf. on Robotics and Automation, Seoul, South Korea. (2001), vol. 2, pp. 1615–1620.
[8] Mazurowski Ł. Komponowanie algorytmiczne-wybrane aspekty. Przegląd Elektrotechniczny, 10b/2012 pp. 243.
[9] Kutyło M., Pluciński M, Laskowska M. Application of the reinforcement learning for selecting fuzzy rules representing the behavior policy of units in RTS-type games. Przegląd Elektrotechniczny, R. 91 NR 2/2015.
[10] Sutton R.S., Barto A.G. “Reinforcement learning: An Introduction, MIT Press, (1998).
[11] Syafiie S., Tadeo F., Martinez E., “Model-free learning control of neutralization processes using reinforcement learning”, Engineering Applications of Artificial Intelligence, Volume 20, Issue 6, (2007), pp. 767-782.
[12] Stebel K., "Practical aspects of the model-free learning control initialization," 2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR) (2015), Międzyzdroje, Poland pp.96-101.