 doi:10.20944/preprints202011.0482.v1

Article

Design and simulation of adaptive PID controller based on fuzzy Q-learning algorithm for a BLDC motor

Reza Rouhi Ardeshiri [1], Nabi Nabiyev [2], Shahab S. Band [3,4,*] and Amir Mosavi [5,6,*]

1 University of Michigan-Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, P. R. China; r.rouhi.a@sjtu.edu.cn
2 Azerbaijan State Oil and Industry University, Baku, Azerbaijan; nabinabiyev600@gmail.com
3 Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam shamshirbandshahaboddin@duytan.edu.vn
4 Future Technology Research Center, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan
5 Faculty of Civil Engineering, Technische Universität Dresden, 01069 Dresden, Germany ; amir.mosavi@mailbox.tu-dresden.de
6 School of Economics and Business, Norwegian University of Life Sciences, 1430 As, Norway
* Correspondence: shamshirbands@yuntech.edu.tw

Abstract: Reinforcement learning (RL) is an extensively applied control method for the purpose of designing intelligent control systems to achieve high accuracy as well as better performance. In the present article, the PID controller is considered as the main control strategy for brushless DC (BLDC) motor speed control. For better performance, the fuzzy Q-learning (FQL) method as a reinforcement learning approach is proposed to adjust the PID coefficients. A comparison with the adaptive PID (APID) controller is also performed for the superiority of the proposed method, and the findings demonstrate the reduction of the error of the proposed method and elimination of the overshoot for controlling the motor speed. MATLAB/SIMULINK has been used for modeling, simulation, and control design of the BLDC motor.

Keywords: Q-learning; Fuzzy logic; Adaptive controller; BLDC motor

1. Introduction

BLDC motors have many advantages, including high efficiency, higher torque-waited ratio, high dynamic response, and long operating life. Hence the BLDC motor was employed in many industrial applications, including electric automotive and robotics [1]. PID controller is a popular used controller in various industries because of no need for an in-depth understanding of the system dynamics and its simple structure. On the other hand, it suffers from the inadequate adjustment of gains. During the use of the PID controller, determining the appropriate PID coefficients is probably challenging in the case of the existence of various nonlinearities and uncertainties, including hysteresis, friction, payload variations, etc. The desired speed tracking and control system performance could be affected by all these factors [2]. This issue can be addressed through applying intelligent control approaches, including fuzzy control, adaptive control, and neural networks. Many tuning approaches and discussions were presented in [3]. Moreover, it is possible to discover countless research articles focusing on auto-tuning PID control [4], APID control [2,5], auto-tuning PID-type fuzzy control [6], etc. in the existing literature. The parameters of the controller have been automatically tuned in auto-tuning PID and APID control according to the alterations of process parameters [4,5].

Approaches based on neural networks (NNs) for motor control system does have a powerful ability to overcome system uncertainty and disturbance, while more computational capacity and data storage space is required. These methods usually do not include transforming the continuous time

 doi:10.20944/preprints202011.0482.v1

domain plant into the equivalent of its neural network [2]. Fuzzy logic control is a powerful, intelligent control approach, primarily according to control of rules and the knowledge of professional staff and specialized skills is the key source of the rule. The fuzzy control can therefore attain the effect of simulating the automatic control mechanism [7–10]. However, the summarization of the rules of fuzzy control is challenging, and it is hard to tune online once the control rules are set, and also there is not enough capacity for the new rules [11]. In summary, it is difficult to understand the characteristics of the motor control by one particular type of intelligent control, such as the single PID modification or individual fuzzy control; however, the combination of the traditional system and modern control may possess a reasonable control impact following complementing [12].

A PID compensator with combining of model reference adaptive control (MRAC) has been proposed by El-Samahy et al. [1]. The findings indicate that the proposed method is robust, as well as the MRAC performance seems to be much higher compared to the self-tuning fuzzy PID controller, particularly for the systems exposed to abrupt disturbances. However, the choice of the appropriate value of adaptation rate gain can affect the speed response. In [2], an adaptive interaction method theory-based self-tuning algorithm for PID control is presented, which is properly applied in artificial neural networks. This modern technique would not need the continuous time domain plant to transform into its corresponding neural network. This approach is implemented on the BLDC motor. In [13], the author built and evaluated the performance of an APID controller on the basis of an additional error of inversed-control signal for BLDC motor drives. It is shown that in the case that there are discrepancies in fault tolerance, process parameters, and learning capabilities, the proposed method provides stronger solutions. The core principle of this approach is, however, focused on a neural network that needs higher computation and time usage, making it impossible to implement in real-time.

This article combines conventional PID control and fuzzy Q-learning to achieve the self-tuning PID-type fuzzy Q-learning control strategy which is utilized to the BLDC motor device and has the same benefits of an ordinary PID structure and are easy to omit the system error and overshoot as well as possesses excellent robustness to parameter interference and alteration.

This article has the following structure: the BLDC mathematical model is presented in Section 2. BLDC speed controls were built for control techniques, including conventional adaptive PID and self-tuning PID-type fuzzy Q-learning adaptive control in 3. In Section 4, the numerical results dependent on the simulated procedure are presented and discussed, and the conclusion is shown in 5.

2. Mathematical model of BLDC motor

BLDC motors are a class of synchronous motors with permanent magnets. They are operated by DC voltage, and without a brush, so electronic commutation is used to rotate the rotor. There is a need for proper commutation of the information about the rotor position for electronic commutation. In addition, Hall-effect sensors are applied for the purpose of detecting the rotor position with a 120-degree location as sensor-less technologies. The BLDC motor commutation is accomplished with 6 phases having this distance [13–15].

Based on the voltage equation of the BLDC motor, the voltage can be defined by the equation as given in Ref. [16]. The full-bridge driving in the two-phase conduction mode is as presented in Fig. 1 as well [17]. The BLDC motor parameters are shown in Table 1.

3. Self-tuning PID-type fuzzy Q-learning adaptive control

3.1. Reinforcement Learning

Reinforcement Learning is an approach aimed at solving problems such as control systems [18], energy management systems [19–21] and is one of the methods of machine learning. The essence of learning influenced this approach because only by communicating with the environment can the control policy produce without understanding the underlying system model. Agent-environment

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

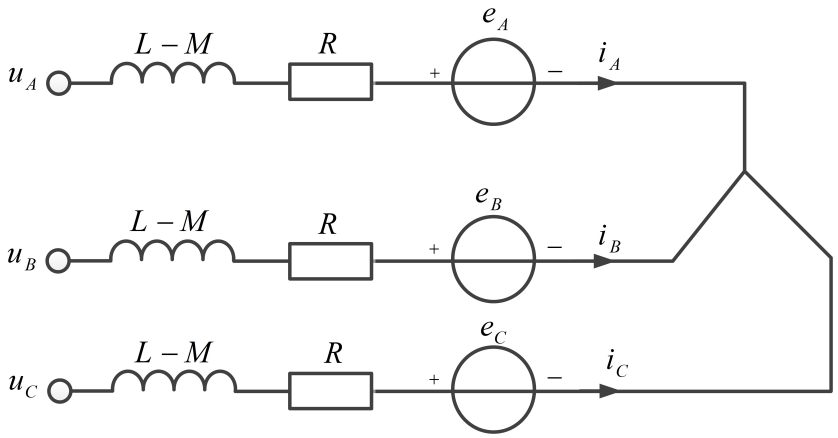

Figure 1. Equivalent circuit of the of the three-phase BLDC motor.

Table 1. The parameters of BLDC motor for per phase.

Rating and Symbol	Units	Quantity
Rated current (I)	Amp.	5
Friction coefficient (B_v)	N.m.s.	$1e^{-3}$
DC resistance (R)	Ω	2.8750
Torque constant (K_t)	$\frac{N.m}{A}$	1.4
Number of poles (P)	–	4
Rated voltage (V)	Volt.	36
Peak torque (T_p)	N.m	0.42
Rotor inertia (J)	$\frac{kg}{m^{-2}}$	$0.8e^{-3}$
Rated speed (Ω)	RPM	4×10^3
Inductance (L)	m H	8.5

interaction shown in Fig. 2 can be described as the RL problem. The agent monitors $S_t \in S$ in this interaction and takes measures $A_t \in A$. The environment gets actions, updates new states $S_{t+1} \in S$ and produces a scalar reward $R_{t+1} \in R \subset \mathbb{R}$. Afterward, the agent gets the award, remains in new states $S_{t+1} \in S$, and selects new measures $A_{t+1} \in A$. This interaction persists until the terminal state $S_T \in S$ is reached. The purpose of the agent is to extract the optimum control strategy to optimize the discounted accumulated rewards, called as expected discounted return G_t in the long term, the governing equation of which is given in Ref. [18].

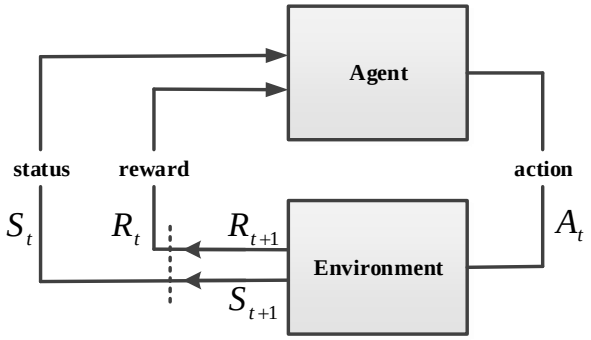

Figure 2. The structure of agent-environment interaction in RL.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

3.2. Q-Learning

The Q-learning algorithm [22] is an RL value-based off-policy learning algorithm, in which the learned action-value function Q, rather than following the existing policy, estimates the optimal action-value function, q^*, in a direct manner. The updating rule equation is as given in Ref. [22].

In general, the Q-learning agent selects action A_t for the current state S t and determines the next step S_t. It earns a reward from this transition and updates the pair (S_t, A_t) value on the condition that it carries out the optimal policy from the state St. An exploration/exploitation scheme is used by a Q-learning agent to measure a policy which optimizes the value of future discounted rewards (Rew), which is defined as given in Ref. [23]. The general procedures of the Q-learning algorithm are demonstrated in an algorithm as given in Ref. [24].

Algorithm 1 Q-learning algorithm.

Initialize $Q(s, a), \forall a \in A, \forall s \in S,$ arbitrarily;
For every episode;
repeat
  Initialize state S_t
  repeat
    Perform action A_t, observe S_{t+1} and R_{t+1};
    Update q-values;
    $S_t \leftarrow S_{t+1}$
  until S_t reaches terminal state S_T;
until episode ends;

3.3. Fuzzy Q-Learning

The q-values in Q-learning for every state action pair are stored and updated. In cases where there is a big number of state action pairs, it may not be feasible to apply it [25]. Fuzzy inference systems benefit from the Q-function in order to achieve good approaches [26] and, at the same time, provide the applicability of Q-learning in continuous state-space problems (fuzzy Q-learning) [27]. **x** implies the crisp set of inputs in fuzzy Q-learning, which describes the agent's state. This is converted to varying values as well as every fuzzy rule fits a state. This means that the degree to which the agent is in a state is defined by each rule's strength [28,29]. In other words, no fixed (predefined) state-action pair is possible; however, the results from each rule (state-action pair) are generated via the exploration/exploitation algorithm. The FIS, therefore, has competition acts with respect to any rule; these rules are in the form that is presented in Ref. [30].

Specifically, the fuzzy Q-learning algorithm and its equations are broadly outlined in 7 stages, which are listed in the following [23].

1. Observe the state **x**
2. Perform the action A_t for each rule t based on the exploration/exploitation algorithm.
3. Compute the global output $a(\mathbf{x})$ by its equation (see Ref. [23]).
4. Compute the related value of $Q(S_t, A_t)$ according to its equation (see Ref. [23]).
5. Obtain the new state information ($\mathbf{x}'$).
6. Compute reward R_{t+1}
7. Update q-values based on the updating rules equation (see Ref. [22]).

3.4. Control Strategy

An automatic self-tuning of the fuzzy inference system is provided by fuzzy Q-Learning only on the basis of reinforcement signals. Fuzzy reasoning handles continuous states and produces continuous actions. Previous information can be integrated into fuzzy rules that can greatly minimize learning time. The need for a professional is virtually unrealistic in requiring a wide spectrum of rules to navigate stochastic environments. This middleware is applied to try to provide an adaptive layer for adjusting both the reference knowledge and rules.

Fig. 3 indicates the proposed method structure, which is a combined PID controller as the main control and fuzzy q-learning approach as an intelligent algorithm for controller parameter tuning. The

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

control signal of the conventional PID controller can be defined as given in Ref. [31] (See equation (7) in mentioned reference, considering $\lambda = \mu = 1$).

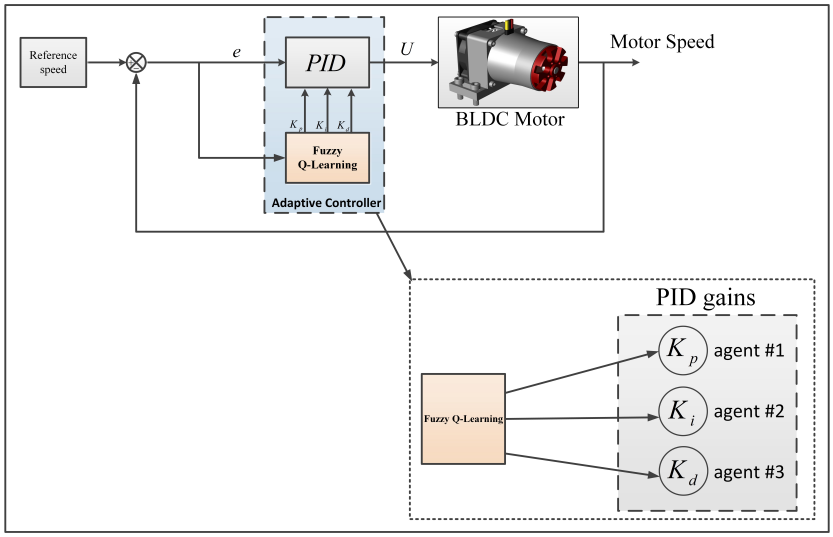

Figure 3. The structure of the self-tuning PID-type fuzzy Q-learning adaptive controller.

In Fig. 3, K_p, K_i, K_d are the gains of PID controller and e is the system error. The gains of PID controller is considered as multi-agent system and each gain including K_p, K_i, and K_d operate as an agent. The set point (Reference speed) is the considered rotor angular velocity. The calculated rotor velocity is subtracted from the set point via the feedback control loop, and the error value (e) emerges. The error is the input value of both the fuzzy Q-learning agent and the PID controller. The motor's speed is the control procedure that possesses the angular velocity of the rotor as output and the control signal of the PID controller (voltage) as input.

The value of only one state variable of the fuzzy Q-learning agent, conditioning in the range of [-1, 1], is defined by the input signal (error). Seven membership functions (MFs) are employed for fuzzy input with triangular-shape which seven states depicted by an equal number of fuzzy rules resulted from the one input with seven membership functions. For fuzzy output, The five MFs are employed with triangular-shape as well.

As previously stated, the interaction of the environment is a factor for agent learning. The exploration/exploitation algorithm should be applied for determining whether the agent is to carry out exploitation or exploration to obtain knowledge through its own experience. Agent scan for 500 rounds/state to carry out a high degree of scan in a new state. The agent then verifies and carries out the activity that was not carried out (if there are any), that provides all probable actions to be carried out for every state at least once. Then for a given state, the agent conducts 1% exploration and 99% exploitation.

The agent reward (R) is described as presented in Ref. [23]. Moreover, the amount of future discounted rewards is described as given in mentioned reference.

4. The numerical simulation of BLDC motor

MATLAB runs the simulation on a PC with an Intel Core $i7-6500U$ processor (6 MB cache, up to 3.18 GHz), 8 GB RAM.

The purpose of this simulation is the design of a controller for a motor speed of 3000 RPM with eliminating the overshoot. Since overshoot can inflict too much current on the motor at the first moment and cause serious damage to the motor, it is, therefore, necessary to consider overshoot as a serious issue. In the first step, an APID controller is implemented due to a comparison with the proposed method and using its PID coefficients to adjust PID gains' initial value for the FQL method. However, selecting the PID gains' initial value by means of try and error can achieve acceptable results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

Because the property of the Q-learning method is that over time, it can improve its coefficients by the actions it receives. Nevertheless, an adaptive method has been used to enhance the controller performance. In the second step, the fuzzy Q-learning method was proposed for BLDC motor speed control. The simulation time is set to 0.5s (case 1) and 3s (case 2) with a simulation step time of 0.001s seconds. The FQL agent round is equal to 0.1s seconds. The discount factor and the learning rate values equal to $\gamma = 0.9$ and $\eta = 0.1$ respectively. In time 0.2 s, a sudden load with torque 10N.m is applied to the motor due to verify the robustness and highlight performance.

As can be seen, Fig. 4 and Fig. 8 show a comparison between the self-tuning fuzzy Q-learning PID (FQLPID) controller and APID controller in case 1 and case 2, respectively. The comparison indicates the maximum efficiency of the FQLPID. At the commencement of the simulation, whereby the FQLPID primarily investigates, it can be seen that the adaptive PID has a huge overshoot and then achieve the steady-state, In the event that the fuzzy Q-learning PID does not have overshoot in the first moment (see Fig. 4 and Fig. 8). By adding a sudden load in 0.2s, it is observed that the fuzzy Q-learning PID is more resilient to adaptive PID and has returned to the normal state. However, adaptive PID has oscillation during the simulation time (see Fig. 5 and Fig. 8). Details of the performance response for both controllers at two different time points can be found in Table 2 for two cases, including at 3s (exploitation phase) and 0.5s (exploration phase). The values of the integral time absolute error (ITAE) and the integral absolute error (IAE) also showed the better performance of the fuzzy Q-learning PID after the exploration phase. The IAE and the ITAE are defined as given in Ref. [32].

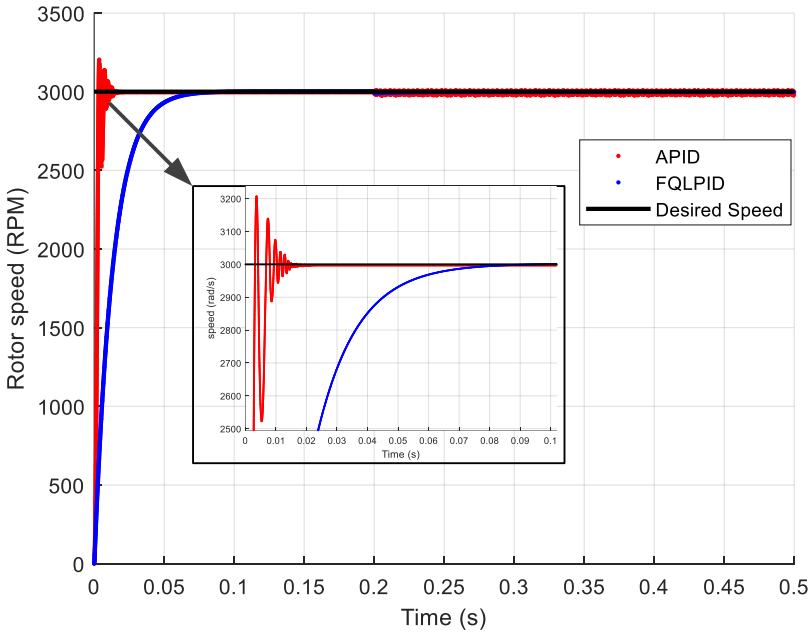

Figure 4. Speed response of controllers.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

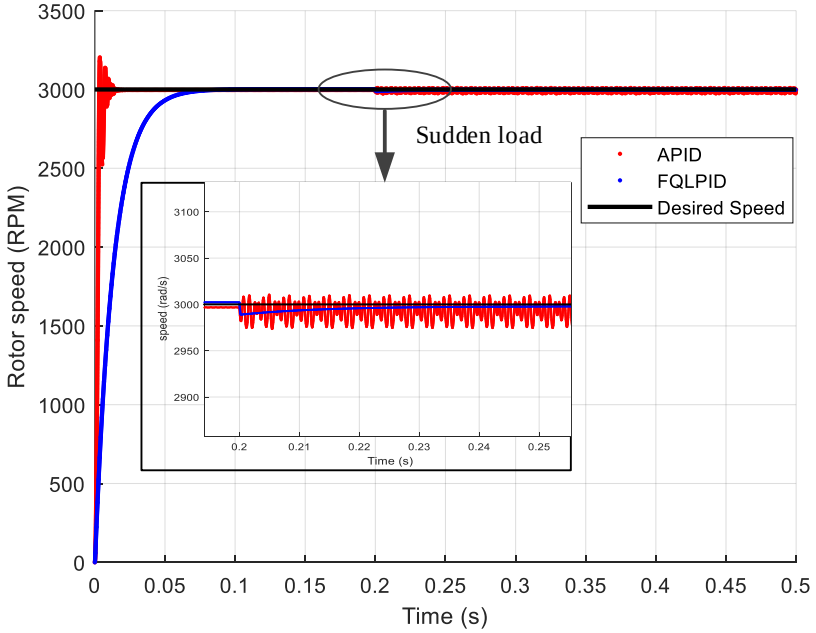

Figure 5. Speed response of controllers with sudden load for case 1

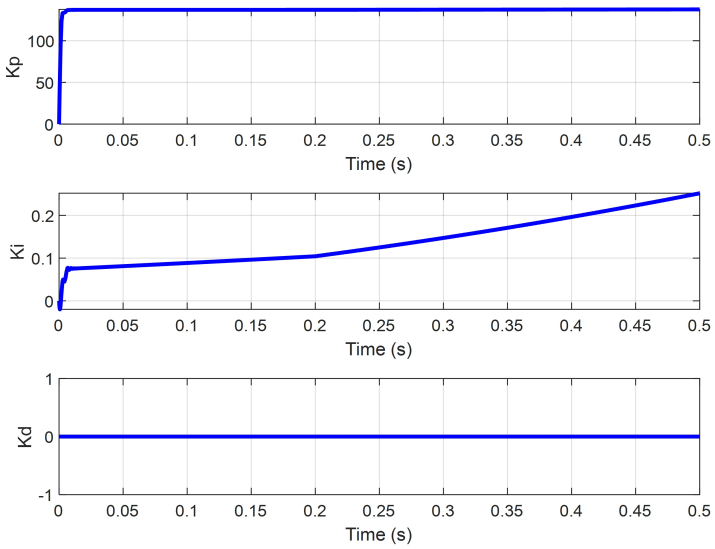

Figure 6. PID gains of adaptive PID controller for case 1.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

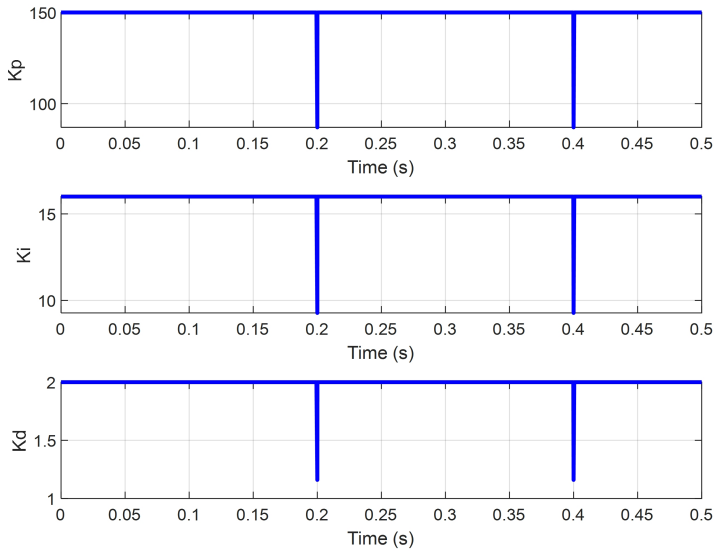

Figure 7. PID gains of fuzzy Q-learning PID controller for case 1.

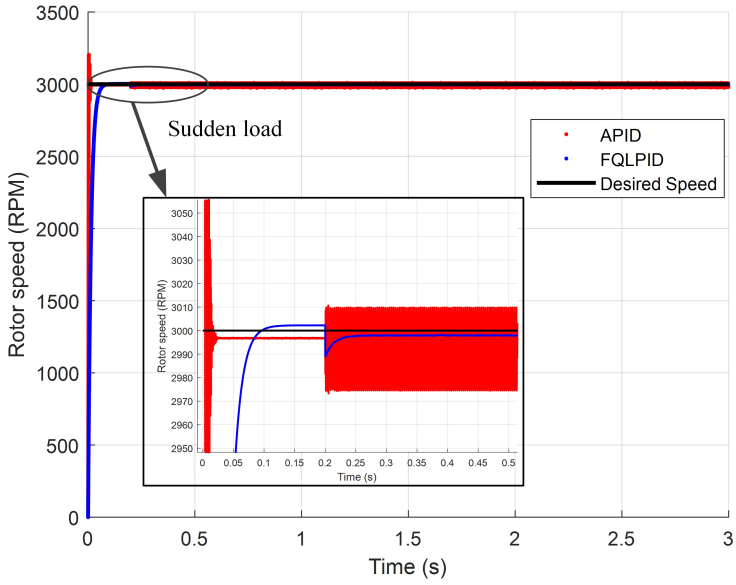

Figure 8. Speed response of controllers with sudden load for case 2

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 19 November 2020 doi:10.20944/preprints202011.0482.v1

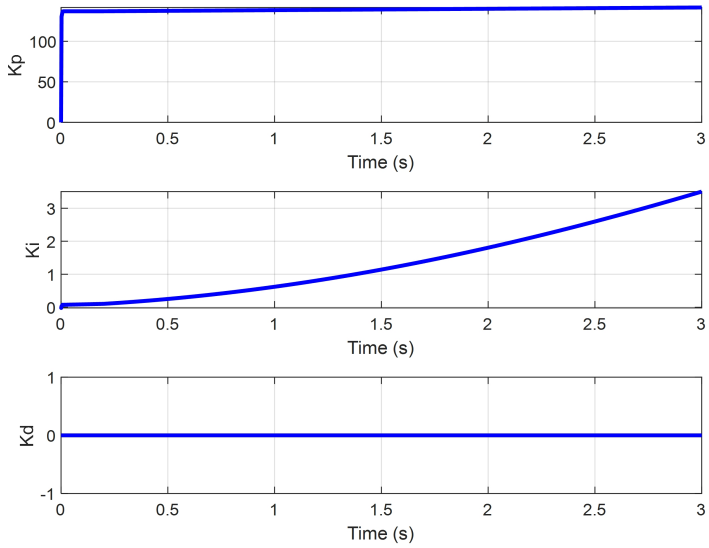

Figure 9. PID gains of adaptive PID controller for case 2.

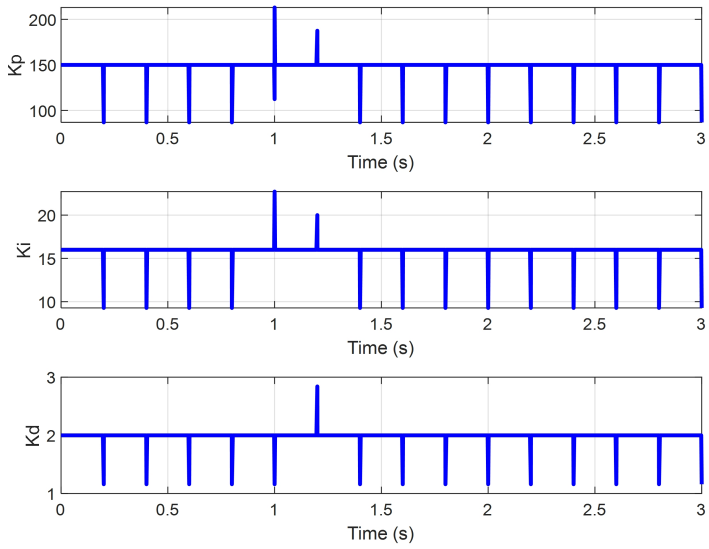

Figure 10. PID gains of fuzzy Q-learning PID controller for case 2.

Table 2. Characteristics of response curves (APID controller and FQLPID controller).

	Case 1 (0.5s)		Case 2 (3s)	
	APID	FQLPID	APID	FQLPID
Settling time (s)	0.0100	0.0514	0.0100	0.0514
Rise time (s)	0.0020	0.0291	0.0020	0.0291
Overshoot (%)	6.8432	0.1407	6.8432	0.1407
Error	2.6530	1.9691	9.2049	1.3738
RMS	165.8617	349.3006	68.7247	142.6103
Ratio (%)	0.0884	0.0656	0.3068	0.0457
IAE	10.8247	41.0772	36.9507	45.4053
ITAE	1.1897	0.7950	46.7698	8.1293

5. Conclusions

The numerical simulation of the BLDC motor has been realized to the desired reference speed. The simulation of the system applying the adaptive PID (APID) as well as the self-tuning PID-type fuzzy Q-learning adaptive controller (FQLPID) was performed by MATLAB/SIMULINK software. The resulted figures illustrate the best performance of the self-tuning PID-type fuzzy Q-learning adaptive controller among the conventional adaptive PID with regard to both the amplitude of the oscillations and steady-state error. The proposed controller results in the desired system without the overshoot. It can be inferred from the obtained results that the proposed controller is more effective than others.

References

1. El-Samahy, A.A.; Shamseldin, M.A. Brushless DC motor tracking control using self-tuning fuzzy PID control and model reference adaptive control. *Ain Shams Engineering Journal* **2018**, *9*, 341–352.
2. Gundogdu, T.; Komurgoz, G. Self-tuning PID control of a brushless DC motor by adaptive interaction. *IEEJ Transactions on Electrical and Electronic Engineering* **2014**, *9*, 384–390.
3. Cominos, P.; Munro, N. PID controllers: recent tuning methods and design to specification. *IEE Proceedings-Control Theory and Applications* **2002**, *149*, 46–53.
4. Cameron, F.; Seborg, D. A self-tuning controller with a PID structure. In *Real Time Digital Control Application*; Elsevier, 1984; pp. 613–622.
5. Howell, M.; Gordon, T.; Best, M. The application of continuous action reinforcement learning automata to adaptive PID tuning **2000**.
6. Soyguder, S.; Karakose, M.; Alli, H. Design and simulation of self-tuning PID-type fuzzy adaptive control for an expert HVAC system. *Expert systems with applications* **2009**, *36*, 4566–4573.
7. Zhang, P.J.; Guo, J.H.; Ma, H.J.; Ma, H.M. Design on Fuzzy Controller of DC Motors Speed Control Based on DSP. Advanced Materials Research. Trans Tech Publ, 2013, Vol. 722, pp. 357–360.
8. Yin, Y.H.; Zheng, B.; Zheng, H.X. A method for modeling and simulation of brushless DC motor control system based on matlab [J]. *Journal of System Simulation* **2008**, *2*.
9. Kofinas, P.; Dounis, A.; Mohamed, E.S.; Papadakis, G.; others. Adaptive neuro-fuzzy model for renewable energy powered desalination plant. *Desalination and Water Treatment* **2017**, *65*, 67–78.
10. Ardeshiri, R.R.; Khooban, M.H.; Noshadi, A.; Vafamand, N.; Rakhshan, M. Robotic manipulator control based on an optimal fractional-order fuzzy PID approach: SiL real-time simulation. *Soft Computing* **2019**, pp. 1–12.
11. Zheng, Y.; Dai, R.; Zhou, Z. Design of torque system for electric bicycle based on fuzzy PID. *The Journal of Engineering* **2019**.
12. Ardeshiri, R.R.; Hoda Nikkhah, K.; Reza-Ahrabi, A. Design and simulation of self-tuning fractional order fuzzy PID controller for robotic manipulator. *Int J Autom Control* **2019**, *13*, 595–618.
13. Rif'an, M.; Yusivar, F.; Kusumoputro, B. Adaptive PID controller based on additional error of an inversed-control signal for improved performance of brushless DC motor. 2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering. IEEE, 2017, pp. 315–320.
14. Xia, C.l. *Permanent magnet brushless DC motor drives and controls*; John Wiley & Sons, 2012.
15. Gamazo-Real, J.C.; Vázquez-Sánchez, E.; Gómez-Gil, J. Position and speed control of brushless DC motors using sensorless techniques and application trends. *Sensors* **2010**, *10*, 6901–6947.
16. Jigang, H.; Jie, W.; Hui, F. An anti-windup self-tuning fuzzy PID controller for speed control of brushless DC motor. *Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije* **2017**, *58*, 321–335.
17. Krause, P.C.; Wasynczuk, O.; Sudhoff, S.D.; Pekarek, S. *Analysis of electric machinery and drive systems*; Vol. 2, Wiley Online Library, 2002.
18. Shi, Q.; Lam, H.K.; Xiao, B.; Tsai, S.H. Adaptive PID controller based onQ-learning algorithm. *CAAI Transactions on Intelligence Technology* **2018**, *3*, 235–244.

19. Kofinas, P.; Doltsinis, S.; Dounis, A.; Vouros, G. A reinforcement learning approach for MPPT control method of photovoltaic sources. *Renewable Energy* **2017**, *108*, 461–473.
20. Kofinas, P.; Vouros, G.; Dounis, A.I. Energy management in solar microgrid via reinforcement learning. Proceedings of the 9th Hellenic Conference on Artificial Intelligence, 2016, pp. 1–7.
21. Ardeshiri, R.R.; Balagopal, B.; Alsabbagh, A.; Ma, C.; Chow, M.Y. Machine Learning Approaches in Battery Management Systems: State of the Art: Remaining useful life and fault detection. 2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems (IESES). IEEE, 2020, Vol. 1, pp. 61–66.
22. Watkins, C.J.C.H. Learning from delayed rewards **1989**.
23. Kofinas, P.; Dounis, A.I. Fuzzy Q-learning agent for online tuning of PID controller for DC motor speed control. *Algorithms* **2018**, *11*, 148.
24. Sutton, R.S.; Barto, A.G.; others. *Introduction to reinforcement learning*; Vol. 135, MIT press Cambridge, 1998.
25. Shamshirband, S.; Anuar, N.B.; Laiha, M.; Kiah, M.; Misra, S. Anomaly detection using fuzzy Q-learning algorithm. *Acta Polytechnica Hungarica* **2014**, *11*, 5–28.
26. Castro, J.L. Fuzzy logic controllers are universal approximators. *IEEE transactions on systems, man, and cybernetics* **1995**, *25*, 629–635.
27. Glorennec, P.Y.; Jouffe, L. Fuzzy Q-learning. Proceedings of 6th international fuzzy systems conference. IEEE, 1997, Vol. 2, pp. 659–662.
28. Kofinas, P.; Dounis, A.; Vouros, G. Fuzzy Q-Learning for multi-agent decentralized energy management in microgrids. *Applied energy* **2018**, *219*, 53–67.
29. Kofinas, P.; Vouros, G.; Dounis, A.I. Energy management in solar microgrid via reinforcement learning using fuzzy reward. *Advances in Building Energy Research* **2018**, *12*, 97–115.
30. Bonarini, A.; Lazaric, A.; Montrone, F.; Restelli, M. Reinforcement distribution in fuzzy Q-learning. *Fuzzy sets and systems* **2009**, *160*, 1420–1443.
31. Ardeshiri, R.R.; Ghadami, S.M.; Reza-Ahrabi, A. Compare performance of fractional-order fuzzy PID controller for tow-link robotic manipulator via evolutionary algorithms. *Int J Mechatron Electr Comput Technol* **2017**, *7*, 3235–3245.
32. Kofinas, P.; Dounis, A.I. Online tuning of a PID controller with a fuzzy reinforcement learning MAS for flow rate control of a desalination unit. *Electronics* **2019**, *8*, 231.