Decreasing Induction Motor Loss Using Reinforcement Learning

Mohammad Bagher Naghibi Sistani
Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
Email: mb-naghibi@um.ac.ir

Sadegh Hesari
Bojnourd Branch, Islamic Azad University, Young Researcher and Elite Club, Bojnourd, Iran
Email: assafsyria@hotmail.com

***Abstract*—In this paper, we have tried to reduce the induction motor losses by controlling the magnetic currents in different torque loads. Reinforcement learning is a method where an agent considers the environment state chooses one action among all possible actions, and the environment returns a numerical signal as a reward for that action. The agent aims at finding a policy by trial-and-error method to reach the maximum sum of rewards. The main proposed idea of this paper is implementing Q-Learning algorithm to find the optimal action in every state of the environment. In this method, quantized amounts of electromagnetic Torque and motor speed are considered as states, and magnetic current is considered as action. Simulation results shows that this method can reduce the power loss about 50% in comparison with the standard driver of motor (FOC) when the motor works in low loads.**

***Index Terms*—reinforcement learning, Q-Learning algorithm, induction motor, decreasing loss**

I. Introduction

Electric motors involve about 40 percent of the whole electric power consumption in the world. Widespread using of induction motors show that if their loss decreases to several percentages, it will play an important role in total power consumption in the world. Therefore, the motor efficiency is an inevitable approach of motor development [1]. Motor loss mostly relates to the control strategy and substantially acts in lower load time which is improved through optimized selection of the flux level [2], [3]. Search control (SC) and Loss Model control (LMC) are two strategies of flux level control. LMC is faster and have no torque ripple but it has a disadvantage that is when the parameters change. There are many researches to overcome this problem by use of soft-computing techniques [4], [5]. In this paper, a Reinforcement Learning algorithm was used which was posed on the induction motor drive FOC 5hp.

Reinforcement learning means routing from situation to action in a way that the numerical reward of the action is the maximum. In this method the learner is not told what to do, but it tries the possible actions and discovers the action with maximum reward [6]. One of the reinforcement learning methods with a simple implementation is Q-Learning method, which was presented in 1989 by Watkins [7]. This algorithm is used by the agent to learn through experience or training. Every repetition equals a training course. The aim of training is to create the brain of the agent, which is displayed by Q matrix. More training will lead to a better Q matrix that can be used by the agent to move in the optimal direction. This way, by having a Q matrix, the agent can choose the best state by referring to the state matrix and selecting the maximum choice, rather than doing a lot of exploration and searching [6]. In this paper we have focused on controlling induction motor drive. To designate this system, it is assumed that the agent has no information about the drive system. On the other hand, here we have assumed that the agent can collect states and actions of the system during the real behavior of the motor. Learning through interaction with real system has some advantages such as:

1. There is no need to have initial knowledge about the learner's policy [6].
2. There is no need to have ideas about the controlling law [8].
3. There is no need to have leaning data, which is significantly superior to [9]-[13]. In fact, the agent is not told what to do in each state, and goodness or badness of an action is displayed for the agent by a scalar measure called reinforcement signal. Having gained this information, the agent is required to learn how to find the best action. This characteristic is one of the special advantages of the reinforcement learning algorithm.

This paper is organized as follows. In Section II, the basics of reinforcement learning and Q-Learning and ϵ-greedy algorithms is described briefly. In Section III, the induction motor model with loss is described on the basis of equivalent circuit of the rotor magnetic current. In Section IV, according to Sections II and III, we present some suggestions and use reinforcement learning method to reduce the loss of an induction motor. In Section V, the simulation results are investigated by MATLAB

Manuscript received October 18, 2014; revised December 19, 2014.

doi: 10.12720/joace.4.1.13-17

software. Section VI presents the general conclusions of this paper.

II. The Basics of Reinforcemence Learning

In reinforcement learning, the main objective of learning is to do something to achieve a goal, without feeding any external direct information to the learner agent [6], [14]. In this method, the only information source for the agent is through a reward and/or punishment signal [6], [15]. In this state, the agent aims at maximizing the received reward that is varying in a time interval. This way, the agent learns the mode of action by concentrating on the received reward [16]. The interaction between the agent and the environment in reinforcement learning is shown in Fig. 1.

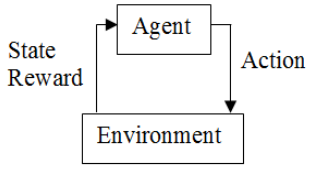

Figure 1. The interaction between the agent and the environment

The learner agent acquires a description of its surrounding environment thorough some sensors. When the agent performs an action, it receives a reward that can be positive or negative, depending on goodness or badness of the action. Equation (1) shows one of the best-known Q-Learning equations in reinforcement learning [6].

$$Q(s,a) \leftarrow (1-\alpha)\, Q(s,a) + \alpha\, [reward + \gamma\, Max_{a'}\, Q(s',a')] \quad (1)$$

where α is a number between 0 and 1, which is called learning rate, and determines the learning speed. It is obvious that the higher is the amount of α, the higher is the learning speed. However, it should be noted that too high amounts of α makes learning instable [17], [18]. Usually, in most applications 0.1 is suggested for learning rate. γ is also a number between 0 and 1, which is called discount factor, and during the learning process prevents the quality function from diverging. In the case of delayed reward γ chooses about 0.9. The Max of Q(s', a') is the quality of the optimal action in the new position of the system, i.e. s'. On the other hand this amount is regarded as the maximum quality in s' position.

A. ϵ-Greedy Algorithm

Greedy method is a well-known selection method [18]. This method recommends that, in every state, an action should be selected whose value function is maximum. In another method, that is known as ϵ-greedy method, a small ϵ number in [0, 1] interval is determined that states the probability of an action to be selected randomly [18]. For example, if ϵ equals 0.2, the possibility for the action to be selected is 0.2 and the possibility for an action with the highest quality to be selected is 0.8. As such, ϵ can vary dynamically to have a high amount at the beginning of learning, which increases the possibility of random selection, and can decrease by learning progress, which increases the possibility of quality-based selection [6], [18].

III. Induction Motor Model

In this paper, we use an equivalent circuit that refers to rotor magnetic current. An iron loss resistor R_f is added in parallel with magnetic inductance in rotor flux reference frame, also we define the field angels in phasor diagram which has been shown in Fig. 2, [1]-[4].

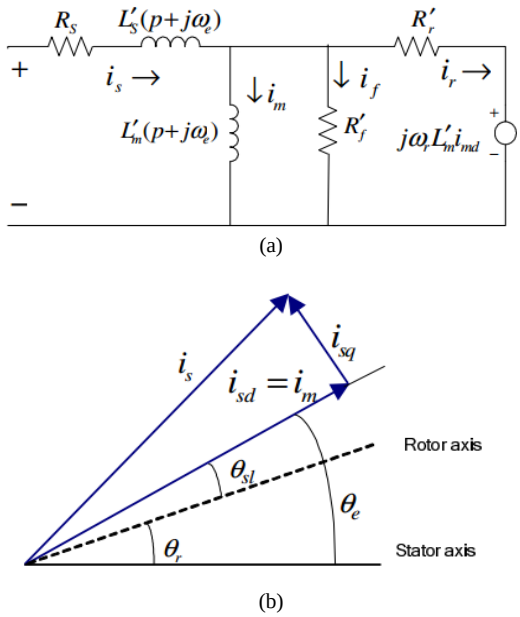

Figure 2. The induction motor model. (a) The induction motor equivalent circuit including iron loss. (b) Phasor diagram of the equivalent circuit and field angels.

In permanent state, there is no leaking inductance on the motor and the motor equivalent circuit is like Fig. 3.

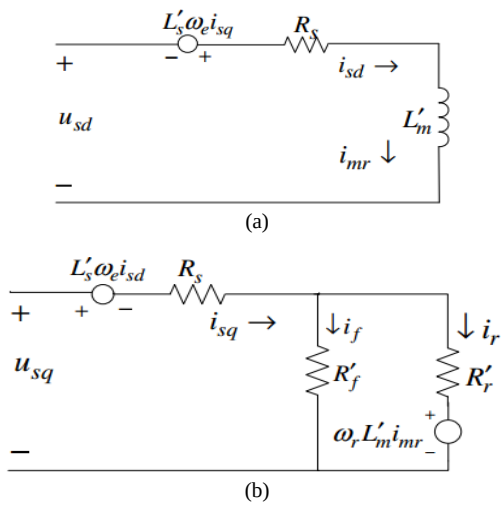

Figure 3. The motor equivalent circuit in permanent state. (a) Equivalent axis of d axis. (b) Equivalent axis of q axis.

In order to develop loss model, a simple and normal method has been used in previous researches [5].

$$P_{total} = R_d i_{mr}^2 + R_q \frac{T_e^2}{K_t^2 i_{mr}^2} \quad (2)$$

where R_d and R_q are defined as:

$$R_d = R_s + \frac{L'^2_m}{R'_f + R'_r}\omega_r^2 \qquad , R_q = R_s + \frac{R'_f R'_r}{R'_f + R'_r}.$$

As can be seen in (2), the total loss of the induction motor is attributed to ω_r, T_e and i_{mr} current. This equation states that by controlling i_{mr} current, motor loss can be controlled.

IV. Proposed Algorithm

In this method ω_r and T_e of electrical motor are measured in all time steps and are considered as states then appropriate i_{mr} is determined as action for each state. After applying the defined i_{mr} in each state a reward or punishment is considered based on the measured power loss specified in Table I. and the Q-table will be updated according to (1). The suggested algorithm of this paper is as follows:

1. First, Q(s,a) amount is assumed zero for all states and actions.
2. The present state of the system (T_e, ω_r) is obtained.
3. According to ϵ-greedy (Fig. 4), algorithm an i_{mr} action is selected.
4. The i_{mr} action is applied to the motor and we await our action's score reward (r).
5. The new state of the system after performing the action i_{mr} is obtained.
6. According to (1), Q(s,a) amount is updated.

The score after performing i_{mr} action in (2) is investigated. In this paper the received reward is considered as follows:

TABLE I. The Received Reward after Performing the Actions

```
If ploss == 0      ,  reward = 100
elseif (ploss < = 10 & ploss > 0) ,   reward = 75
elseif (ploss >10 & ploss <=30)    ,   reward = 50
elseif (ploss > 30 & ploss<=50)    ,   reward=25
else        ,  reward=0
end
```

The agent tries to behave in a way that maximizes the reward function. For electromagnetic torque, rotor speed and rotor magnetic current, we have:

T_e = 0 to 2 Nm ω_r = 150 rad/s i_{mr} = 0 to 5 A

Given the motor presented in Appendix A, the appropriate amount of i_{mr} magnetic current is regulated according to [5].

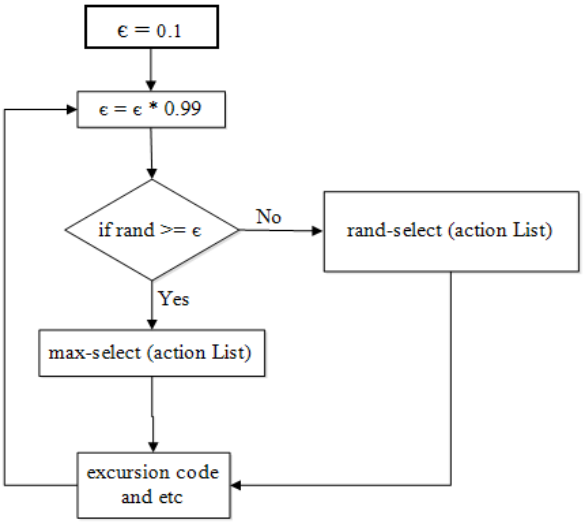

Figure 4. ϵ-greedy algorithm

V. Simulation Results

The proposed algorithm is implemented in 3 steps. In the first step, the applied reference torque is between 0 and 2 Nm, speed is 100 rad/s and i_{mr} current is between 0 and 5 A (this amount of current is considered according to motor model [5]). The electromagnetic torque of output is divided into 1000 separate chunks. As such, i_{mr} current is divided into 50 chunks. In the second step, the Q-table is completed offline using Q-Learning algorithm. The rows of this matrix are the possible states and the columns of this matrix are the possible actions (id). In this step, the agent has no information about the motor. In fact, for each i_{mr} current, the output torque and motor speed is calculated based on the model of motor and P_{total} is calculated using (2) then the reward signal is sent to the agent based on Table I. According to this reward, the agent decides how this i_{mr} has been appropriate. The second step is summarized in Fig. 5, The motor speed is considered constant and equals the speed of stable state.

In the following tables some results related to the second step are presented. Q matrix is obtained here as a 1000×50 matrix. That is it has 1000 states and 50 actions. In each table, the learner agent chooses an action 25000 times among 50 actions that has the maximum amount of Q.

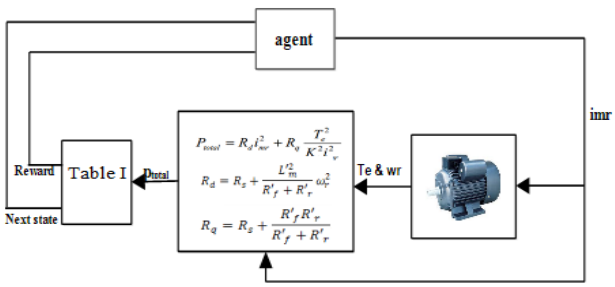

Figure 5. Completing Q table in second step.

In Table II, for the state T_e=0 and ω_r =100, the best current that can be applied is i_{mr}=0.2. In this state the total loss of the motor is obtained 0.0735 W. In Table III, for the state T_e=0.8 and ω_r =100, the most appropriate current is i_{mr}=0.2. In this state the total loss of the motor is obtained 9.5510 W. In Table IV, for the state T_e=1.2 and ω_r =100, i_{mr}=2.5 current is selected as the best current. In this state the total loss of the motor is obtained 14.6534 W. In Table V, in the state T_e=0.8 and ω_r =100, i_{mr}=3.8 current is selected by Q-Learning algorithm. In this state the total loss is obtained 30.3372 W.

TABLE II. CALCULATING Q FOR THIS STATE: T_E=0, Ω_R =100

states \ action	imr=0	imr=0.2	imr=1	imr=2.4	imr=3	imr=4.3	imr=5
Te=0 & Wr=100	Q= 49.99	Q= 199.52	Q= 198.57	Q= 130.96	Q= 171.91	Q= 116.58	Q= 112.49

TABLE III. CALCULATING Q FOR THIS STATE: T_E=0.8, Ω_R =100

states \ action	imr=0	imr=0.2	imr=1	imr=2	imr=3	imr=4.3	imr=5
Te=0.8 & Wr=100	Q= 99.00	Q= 123.95	Q= 167.91	Q= 199.98	Q= 174.60	Q= 149.62	Q= 149.41

TABLE IV. CALCULATING Q FOR THIS STATE: T_E=1.2, Ω_R =100

states \ action	imr=0	imr=0.2	imr=1	imr=2	imr=2.5	imr=4.3	imr=5
Te=1.2 & Wr=100	Q= 124.38	Q= 69.76	Q= 132.79	Q= 148.40	Q= 149.60	Q= 122.43	Q= 122.31

TABLE V. CALCULATING Q FOR THIS STATE: TE=2.0, Ω_R =100

states \ action	imr=0	imr=0.2	imr=1	imr=2.4	imr=3	imr=3.8	imr=5
Te=2 & Wr=100	Q= 69.67	Q= 79.44	Q= 138.07	Q= 143.29	Q= 145.91	Q= 149.36	Q= 122.31

In the third step, using the obtained results from the Q table of the second step, we complete the structural model of Fig. 6. In this structure, motor torque and speed are obtained momentarily from feedback output, and for each speed and torque, Q-Learning algorithm will apply an i_{mr} current to the motor online. Applying such a current to the motor will result in loss reduction and output improvement. The simulation results for this approach are shown in Fig. 8-Fig. 9.

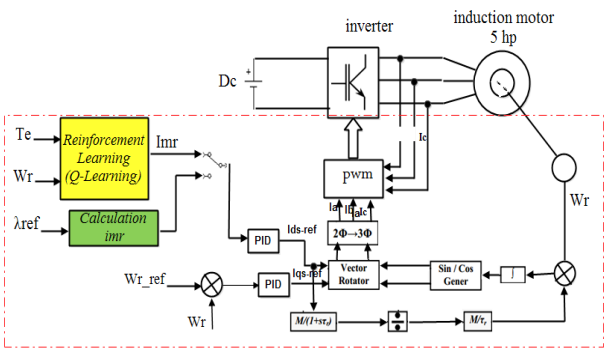

Figure 6. The suggested structure (the third scenario)

Fig. 7 shows the output speed of the motor. From the figure it can be seen that the speed has reached its reference amount, 100 rad/s, in no time, less than 0.02 seconds.

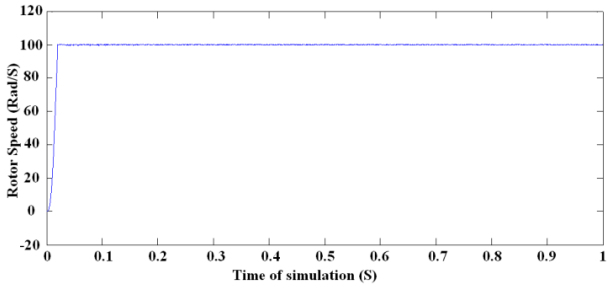

Figure 7. Output speed of the motor

The motor loss is compared in two scenarios, with the FOC controller and with RL method and is shown in Fig. 8.

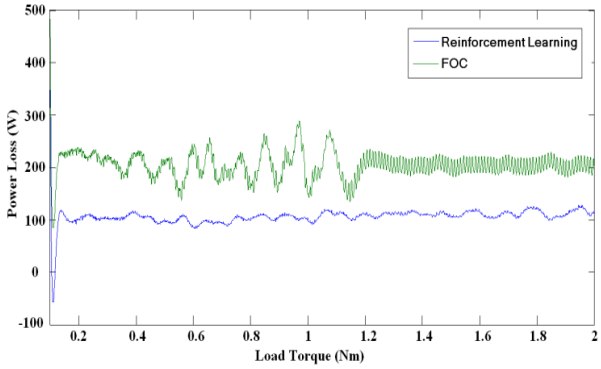

Figure 8. Power loss of motor with FOC and RL controllers.

From the results of Fig. 8, it can be seen that using the reinforcement learning approach has lower loss. We could reduce the loss about 100 watts in 0 to 2 Nm torque. In stable state (permanent motor), rotor magnetic current equals the current of stator *d* axis (i_{mr}=id). The mean current of stator *d* axis can be seen in Fig. 9, this current varies between 2 and 4 Ampere.

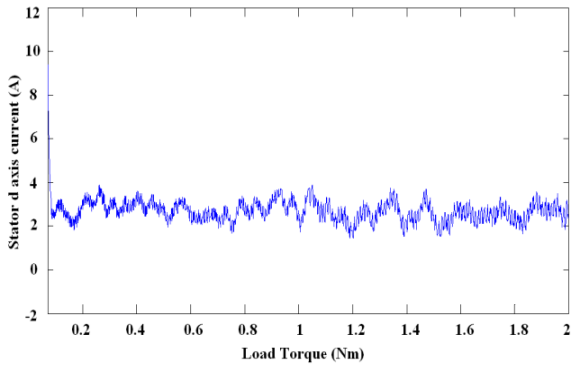

Figure 9. The mean current of stator d axis

VI. CONCLUSION

In this paper, a different approach was conducted on inductive motor drive. The algorithm employed for this purpose was Q-Learning algorithm. The motor used to optimize the loss was a 5hp induction motor with input

torque of 0 to 2 Nm. Since motor has the highest loss in low loads, in this paper we considered the motor loss reduction algorithm in 0 to 2 Nm intervals. The proposed method for motor loss reduction applied in three steps. In the first step drive modeling was performed. In the second step, Q-Learning approach was conducted to complete the optimum Q table according to the model in the first step. And finally, in the third step this Q table was employed in an online manner to reduce the induction motor loss. The simulation results shows that this method can reduce the power loss about 50% in comparison with the standard driver of motor (FOC) when the motor works in low loads.

APPENDIX A CHARACTERISTICS OF MOTOR

motor 5 hp , 1750 rpm , 460 v	
rotor resistance =1.083	Stator resistance =1.115
rotor inductance =0.0059	stator inductance =0.0059
Inertia = 0.02	Mutual inductance= 0.2037
friction factor= 0.005752	pole pairs = 2

REFERENCES

[1] Y. Tai and Z. Liu, "Efficiency optimization of induction motor using genetic algorithm and hybrid genetic algorithm," in *Proc. International Conference on Electrical Machines and Systems*, Aug 2011, pp. 1-4.

[2] M. N. Uddin and S. W. Nam, "Adaptive backstepping based online loss minimization control of an IM drive," *IEEE Trans. Power Engineering Society General Meeting*, IEEE, pp. 1-9, June 2007.

[3] M. N. Uddin and S. W. Nam, "New online loss-minimization-based control of an induction motor drive," *IEEE Trans. Power Electronics*, vol. 23, no. 2, pp. 926–933,Mar. 2008.

[4] T. R. Chelliah and J. G. Yadav, "Optimal energy control of induction motor by hybridization of loss model controller based on particle swarm optimization and search controller," in *Proc. World Congress on Nature & Biologically Inspired Computing*, IEEE, Dec 2009, pp. 1178–1183.

[5] W. Beevi and S. Kumar, "Loss minimization of vector controlled induction motor drive using genetic algorithm," *IEEE Trans. Green Technologies*, pp. 251-257, Dec. 2012.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Bradford Books, MIT, 1998, pp. 1054.

[7] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, Dept. Elect. Eng., King's Univ., May 1989.

[8] S. C. Wang, Z. X. Song, H. Ding, and H. B. Shi, "An improved reinforcement Q-learning method with BP neural networks in robot soccer," in *Proc. Fourth International Symposium on Computational Intelligence and Design*, vol. 1, 2011, pp. 177-180.

[9] A. Tarigoppula, N. Rotella, and J. T. Francis, "Properties of a temporal difference reinforcement learning brain machine interface driven by a simulated motor cortex," in *Proc. 34th Annul. Int. Conf. of the IEEE EMBS*, San Diego, California USA, Sept. 2012, pp. 3284-3287.

[10] R. J. Wai and C. C. Chia "Motion control of linear induction motor via petri fuzzy neural network," *IEEE Trans. on Industrial Electronics*, vol. 54, no. 1, pp. 281-295, Feb. 2007.

[11] Q. Wang, J. Y. Qin, and H. J. Zhou, "Reinforcement learning based self-constructing fuzzy neural network controller for ac motor drives," in *Proc. 6th IEEE Conference on Industrial Electronics and Applications*, June 2011, pp. 913-018.

[12] H. J. Zhang, *et al.* "Multi-objective reinforcement learning algorithm and its application in drive system," *IEEE Trans. 34th Annual Conference of Industrial Electronics*, pp. 274-279, Nov. 2008.

[13] S. T. Hagen and B. Kröse, "Neural Q-learning," *Neural Computing & Applications*, vol. 12, no. 2, pp. 81-88, Nov. 2003.

[14] J. Pazis, and M. G. Lagoudakis, "Reinforcement learning in multidimensional continuous action spaces," in *Proc. IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, April 2011, pp. 97-104.

[15] C. Gaskett, D. Wettergreen, and A. Zelinsky, "Q-Learning in continuous state and action spaces," *Advanced Topics in Artificial Intelligence*, Springer Berlin Heidelberg, 1999, pp. 417-428.

[16] P. F. Dominey, "Complex sensory-motor sequence learning based on recurrent state representation and reinforcement learning," *Biological Cybernetics*, vol. 73, no. 3, pp. 265-274, 1995.

[17] Q. Wang and Z. L. Zhan, "Reinforcement learning model algorithms and its application," in *Proc. Int. Conf. on Mechatronic Science Electrical Engineering and Computer*, Aug. 2011, pp. 1143-1146.

[18] B. Fu, X. C., Y. He, and M. Wu, "An efficient reinforcement learning algorithm for continuous actions," in *Proc. 25th Chinese Control and Decision Conference (CCDC)*, May 2013, pp. 80-85.

Mohammad Bagher Naghibi Sistani is an assistant professor of Ferdowsi University of Mashhad. He received his Ph.D. degree on Control Engineering from Ferdowsi University of Mashhad and M.Sc. degree on Control Engineering from Tehran University. He is interested in Optimal Control, Reinforcement Learning and Soft Computing.

Sadegh Hesari was born in Bojnourd, Iran in 1988. He received the B.S. degree in Electrical Engineering from Sadjad University of Mashhad, Iran, in 2011, and the M.Sc. degree in Electrical Engineering from Azad University of Bojnourd in 2014. Now he is a Ph.D. student of Electrical Engineering at University of Amirkabir. He is interested in Machine Drive Systems, Reinforcement Learning and Soft Computing.