Recent Innovations in Mechatronics (RIiM) Vol. 2. (2015). No. 1-2.
DOI: 10.17667/riim.2015.1-2/10.

Rule-base reduction in Fuzzy Rule Interpolation-based Q-learning

Dávid Vincze
Department of Information Technology
University of Miskolc
Miskolc, Hungary
david.vincze@iit.uni-miskolc.hu

Szilveszter Kovács
Department of Information Technology
University of Miskolc
Miskolc, Hungary
szkovacs@iit.uni-miskolc.hu

***Abstract*—The method called Fuzzy Rule Interpolation-based Q-learning (FRIQ-learning for short) uses a fuzzy rule interpolation method to be the reasoning engine applied within Q-learning. This method was introduced previously by the authors along with a rule-base construction extension for FRIQ-learning, which can construct the requested FRI fuzzy model from scratch in a reduced size, implementing an incremental creation strategy. The rule-base created this way will most probably contain only those rules which were significant during the construction process, but have no important role in the final rule-base. Also there can be rules which became redundant (can be calculated by using fuzzy rule interpolation) thanks to another rule in the finished rule base. The goal of the paper is to introduce possible methods, which aim to find and remove the redundant and unnecessary rules from the rule-base automatically by using variations of newly developed decremental rule base reduction strategies. The paper also includes an application example presenting the applicability of the methods via a well known reinforcement learning example: the cart-pole simulation.**

Keywords—FRIQ-learning, reinforcement learning, rule-base reduction, fuzzy rule interpolation

I. INTRODUCTION

Reinforcement learning [13] methods, like Q-learning [18] can come to an aid in various situations, where the solution for the problem is hidden in the feedback gathered from the environment. A function describing the environment gives rewards (positive or negative) for every action performed. Based on the gathered rewards from the environment through the reward function, reinforcement learning methods are approximating the goodness value of each possible action performed in the possible states of the state space. (The functions describing the rewards are designed especially for the current task to be solved).

This way reinforcement learning methods can solve problems where priory knowledge can be expressed in the form what is needed to be achieved, not in how to solve the problem directly. This means that the usage of a method like this allows solving control problems without defining an exact imperative method for solving the problem.

Q-learning [18] is a reinforcement learning method, which can be used for constructing the state-action-value function (where value means the goodness of the action in the corresponding state). The purpose of Q-learning is finding the fixed-point solution Q of the Bellman Equation [3] through iteration. The original Q-learning method works with discrete state and action spaces. Introducing fuzzy reasoning to Q-learning results in a method which is extended to continuous environments. This variation is called Fuzzy Q-learning (FQ-Learning), which traditionally applies the zero-order Takagi-Sugeno fuzzy inference (see details in [1], [4] and [5]).

The Fuzzy Q-learning method can be further enhanced with a capability to use sparse fuzzy rule bases, by the means of Fuzzy Rule Interpolation (FRI). A method which incorporates the 'FIVE' Fuzzy Rule Interpolation (FIVE FRI) technique was introduced by the authors previously in [15]. This latter method, called Fuzzy Rule Interpolation-based Q-learning (FRIQ-learning for short) uses the mentioned fuzzy rule interpolation method to be the reasoning engine applied within Q-learning.

A rule-base construction method was also developed for FRIQ-learning [17], which can construct the requested FRI fuzzy model from scratch in a reduced size, implementing an incremental creation strategy of an intentionally sparse fuzzy rule base. Furthermore this incrementally constructed rule-base possibly contains rules which were only significant during the construction process itself, but meanwhile their importance lowered in the final rule-base. There can be rules which were superseded by other much ‘stronger’ or near equal but different rules. Also there can be rules which are redundant (can be calculated by using fuzzy rule interpolation) in the finished rule base.

The goal of the paper is to introduce possible methods, which aim to find and remove the redundant and unnecessary ‘less important’ rules from the rule-base automatically by using variations of newly developed decremental rule base reduction strategies. The paper also includes an application example presenting the applicability of the methods via a well known reinforcement learning example: the cart-pole (reversed pendulum) simulation.

II. FUZZY RULE INTERPOLATION-BASED Q-LEARNING

Many Fuzzy Rule Interpolation (FRI) techniques exist already, see e.g. [2] for a comprehensive overview on FRI methods. Also a freely available toolbox implementing various FRI methods is presented in [6].

Introducing FRI in Q-learning gives the possibility of omitting rules (action-state values) from the fuzzy rule-base gaining the potentiality of applying the proposed method in larger state dimensions with a reduced rule-base sized action-state space. Rule-base reduction with FRI can also be achieved by sparse fuzzy rule-base identification methods based on input-output data sets, e.g. the RuleMaker Toolbox [7] is a freely available sparse rule-base identification software.

Fuzzy Rule Interpolation-based Q-learning (FRIQ-learning) is different from these kind of identification methods, because it does not require input-output data sets for the construction of a suitable sparse fuzzy rule-base.

This FRIQ-learning method, which was first introduced by the authors in [15] is the result of the substitution of the zero-order Takagi-Sugeno fuzzy model of Fuzzy Q-learning (FQ-learning) with the 'FIVE' FRI model. The 'FIVE' FRI is a fast and application oriented FRI method, for in depth details on the method see [8] [9] and [10].

In this model, the state-action-value function is represented by a fuzzy rule-base, where a fuzzy rule has the form:

If $x_1 = A_{k,1}$ **And** $x_2 = A_{k,2}$ **And** … **And** $x_m = A_{k,m}$ **Then** $y = c_k$,

where x is the observation, A is the fuzzy rule antecedent, y is the conclusion, and c_k is the consequent value.

Applying the FIVE FRI method with singleton rule consequents to be the model of the state-action-value function, we get:

$$\tilde{Q}(s,a)=\begin{cases} q_{i_1 i_2 \ldots i_N u} & \text{if } \mathbf{x}=\mathbf{a}_k \text{ for some } k, \\ \sum_{i_1,i_2,\ldots,i_N,u} \prod_{n=1}^{N}\left(1/\delta_{s,k}^{\lambda}\right)/\left(\sum_{k=1}^{r}1/\delta_{s,k}^{\lambda}\right)\cdot q_{i_1 i_2 \ldots i_N u} & \text{otherwise.} \end{cases} \quad (1)$$

where $\tilde{Q}(s,a)$ is the approximated state-action-value function.

The partial derivative of the model consequent $\tilde{Q}(s,a)$ with respect to the fuzzy rule consequents $q_{u,i}$, required for the applied fuzzy Q-learning method in case of the FIVE FRI model from (1) can be expressed by the following formula (according to [11]):

$$\frac{\partial\tilde{Q}(s,a)}{\partial q_{i_1 i_2 \ldots i_N u}}=\begin{cases} 1 & \text{if } x = a_k \text{ for some k,} \\ \left(1/\delta_{s,k}^{\lambda}\right)/\left(\sum_{k=1}^{r}1/\delta_{s,k}^{\lambda}\right) & \text{otherwise} \end{cases} \quad (2)$$

where $q_{u,i}$ is the constant rule consequent of the k^{th} fuzzy rule, $\delta_{s,k}$ is the scaled distance in the vague environment of the observation, and the k^{th} fuzzy rule antecedent, λ is a parameter of Shepard interpolation (in case of the stable multidimensional extension of the Shepard interpolation it equals to the number of antecedents according to [14]), x is the actual observation, and r means the number of the rules.

Replacing the partial derivative of the conclusion of the 0-order Takagi-Sugeno fuzzy inference with the partial derivative of the conclusion of FIVE (2) with respect to the fuzzy rule consequents $q_{u,i}$ leads to the following equation for the Q-Learning action-value-function iteration:

$$\begin{aligned} &\text{if } \mathbf{x}=\mathbf{a}_k \text{ for some } k: \\ &q_{i_1 i_2 \ldots i_N u}^{k+1}=q_{i_1 i_2 \ldots i_N u}^{k}+\Delta\tilde{Q}_{i,u}^{k+1}= \\ &=q_{i_1 i_2 \ldots i_N u}^{k}+\alpha_{i,u}^{k}\cdot\left(g_{i,u,j}+\gamma\cdot\max_{v\in U}\tilde{Q}_{j,v}^{k+1}-\tilde{Q}_{i,u}^{k}\right) \\ &\text{otherwise}: \\ &q_{i_1 i_2 \ldots i_N u}^{k+1}=q_{i_1 i_2 \ldots i_N u}^{k}+\prod_{n=1}^{N}\left(1/\delta_{s,k}^{\lambda}\right)/\left(\sum_{k=1}^{r}1/\delta_{s,k}^{\lambda}\right)\cdot\Delta\tilde{Q}_{i,u}^{k+1}= \\ &=q_{i_1 i_2 \ldots i_N u}^{k}+\prod_{n=1}^{N}\left(1/\delta_{s,k}^{\lambda}\right)/\left(\sum_{k=1}^{r}1/\delta_{s,k}^{\lambda}\right)\cdot\alpha_{i,u}^{k}\cdot\left(g_{i,u,j}+\gamma\cdot\max_{v\in U}\tilde{Q}_{j,v}^{k+1}-\tilde{Q}_{i,u}^{k}\right) \end{aligned} \quad (3)$$

where $q_{i_1 i_2 \ldots i_N u}^{k+1}$ is the $k+1^{th}$ iteration of the singleton conclusion of the $i_1 i_2 \ldots i_N u^{th}$ fuzzy rule taking action A_u in state S_i, S_j is the new observed state, $g_{i,u,j}$ is the observed reward completing the $S_i \to S_j$ state-transition, γ is the discount factor and $\alpha_{i,u}^{k}\in[0,1]$ is the step size parameter.

The $\tilde{Q}_{j,v}^{k+1}$ and $\tilde{Q}_{i,u}^{k}$ action-values can be approximated by equation (3), which uses the FIVE FRI model. An application example presenting a proof-of-concept implementation can be found in [15].

III. INCREMENTAL RULE-BASE CONSTRUCTION

Our first approach for rule-base reduction was creating a rule-base incrementally constructed from scratch. This method (introduced in [16] and [17]) simply increases the number of the fuzzy rules by inserting new rules in the required positions. Instead of initially building up a full rule base with the conclusions of the rules (Q values) set to a default value, only a minimal sized rule base is created with 2^{N+1} fuzzy rules at the corners of the N+1 dimensional antecedent (state-action space) hypercube. In cases when the action-value function update is considered as high (e.g. greater than a preset limit ε_Q: $\Delta\tilde{Q}>\varepsilon_Q$), and even the closest existing rule to the actual state is farther than a preset limit ε_s, then a new rule is inserted to the closest possible rule position. These possible

rule positions are gained by inserting a new state among the existing ones ($s_{k+1}=s_k$, $\forall k > i$, $s_{i+1}=\frac{s_i+s_{i+2}}{2}$). In case if the update value is relatively low ($\Delta\tilde{Q} \leq \varepsilon_Q$), or the actual state-action point is in the vicinity of an already existing fuzzy rule, then the rule-base remains unchanged (only the conclusions of the corresponding rules will be updated). The next step is updating the *Q* value, performed regarding to the FRIQ-Learning method according to the equation (3) as it was discussed earlier.

This way the resulting action-value function will be modeled by a sparse rule base which contains only the fuzzy rules which seem to be most relevant in the model. Applying the FIVE FRI method, as stated earlier, allows the usage of sparse rule bases which could result in saving a considerable amount of computational resources and reduced state space.

An application example for this method was presented in [17], which resulted in the reduction of the original rule-base with 2268 rules to a rule-base which only contains 182 rules. When this latter rule-base is used for solving the problem of application example the same results and rewards can be achieved.

Further reduction of rule-bases created as introduced previously is presented in the following chapter.

IV. RULE-BASE REDUCTION STRATEGIES

As already mentioned earlier, the previously completed incrementally constructed rule-base possibly contains redundant rules. It is possible to find and remove these rules from the rule base automatically by using various decremental rule-base reduction strategies presented in the followings.

According to the Bellman equation [3] only the highest of the possible Q values for the next $(k+1)^{th}$ iteration step is used in the calculation of the next (currently approximated) Q value:

$$\max_{v \in U} \tilde{Q}_{j,v}^{k+1} \tag{4}$$

In other words this means that high Q values are possibly more significant than lower Q values (when following a greedy strategy), hence rules with higher consequent values possibly have greater impact on the resulting problem solution and on the rewards.

This suggests a strategy to try to omit rules from the previously incrementally constructed rule base (e.g. see Fig. 1.) which have low Q values as their conclusion. Following this strategy (Strategy I.), the selected rules based on their conclusion value (means the absolute Q value) are omitted one by one, as the whole process is evaluated over and over again. If the rewards given by the environment with the truncated rule base remain the same, or near the same (reward difference is within a preset interval), or maybe higher than the rule is considered to be redundant, therefore it will be removed from the rule base. In the other case when the given reward is considerably lower or the evaluation fails, the rule is considered being a cardinal rule, therefore it has to stay in the rule base (see Fig. 2.).

Depending on the actual problem, difference in the cumulative rewards could be allowed to some degree, till the problem is still solved and produces the same or near the same rewards. Various thresholds can be used in defining 'near the same' depending on the task and requirements. Close matches of the rewards should result in approximately the same steps as were the original incrementally constructed full rule base, when using the final reduced rule base. Accepting relatively greater (depending on the exact reward function), but still valid, differences between the rewards could result in a different step-by-step solution, but the overall task will still be solved.

The next strategy (Strategy II.) is very similar to the previously presented strategy, the only difference is, that it first chooses the rule with the highest consequent (*Q* value). This way the probably most important rules are tested first to determine whether they are needed or not.

Another developed strategy (Strategy III.) selects rule groups for removal, hence allowing mass removal of rules, which could result in faster completion of the reduction process. First it calculates the range of the *Q* values, and divides the rule group using the halved range value as a threshold. With the rule group of lower *Q* values removed temporarily, then evaluate the reduced rule-base. If the reduced rule-base seems to be still sufficient for solving the task, permanently remove the rule group. In the other case, when the temporarily truncated rule-base fails to sufficiently solve the problem the removed rule groups is restored. Therefore if the group of rules seems to be too large, then decrease the threshold limit of *Q* by halving again the previously calculated threshold value based on the range of *Q* values. This process is repeated until the group of rules can be removed or if the group contains only 1 rule and the problem still cannot be solved with the removal of this last rule, then mark the rule to be permanent (so this rule will not be selected again into a group during the reduction process), and restart

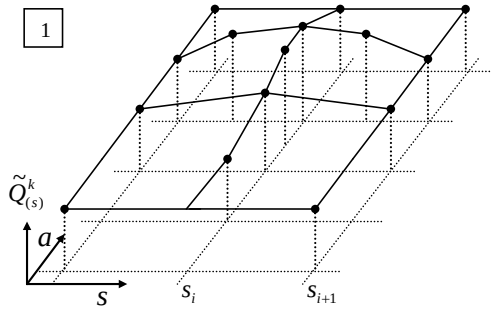

Fig. 1. The incrementally constructed rule-base used as starting point

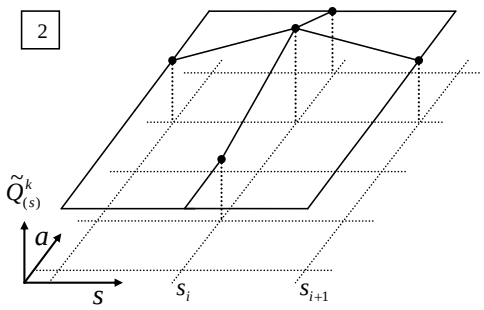

Fig. 2. The final decrementally reduced rule-base providing the same (or approximately the same) results

the process with a newly calculated threshold value, now omitting the value of the previously marked rule.

This process should be repeated until all the remaining rules for possible group forming are marked permanent (meaning that those have been already checked for removal).

It is worth noting that different rule-bases can exist for solving the same problem with equal results and rewards.

After the reduction process is complete, the final rule base will contain only the most significant rules, in other words this method extracts the cardinal rules which are basically operating the FRI-based system.

In the followings the widely used cart-pole example, which was used to present previous FRIQ-learning related examples (in [15] and [17]), will be used to demonstrate the presented rule base reduction strategies to gather a still functional but minimal size, truncated rule base.

A. Application example for presenting the decremental reduction strategies

For the demonstration of the proposed reduction strategies the same cart-pole application is used as for the previous FRIQ-learning application examples in [15] and [17]. The original implementation, which works in discrete space, was developed by José Antonio Martin H. This implementation uses SARSA [12] (a Q-learning method) and is freely available from [19]. The example program runs through episodes, where an episode means a cart-pole simulation run. The goal of the application is to move the cart to the center position while balancing the pole. Maximum reward is gained when the pole is in vertical position and the cart is on the center position mark. An episode is considered to be successfully finished (gains positive reinforcement in total) if the number of iterations (steps) reaches one thousand while the pole stays up without the cart crashing into the walls. Otherwise the episode is considered to be failed (gains negative reinforcement in total). The fuzzy rules are defined in the following form:

If s_1 = $A_{1,i}$ **and** s_2 = $A_{2,i}$ **and** s_3 = $A_{3,i}$ **and** s_4 = $A_{4,i}$ **and** a = $A_{5,i}$ **Then** q = B_i

The rule antecedent variables are the following: s_1 – shift of the pendulum, s_2 – velocity of the pendulum, s_3 – angular offset of the pole, s_4 – angular velocity of the pole, a – compensation action of the cart. The linguistic terms used in the antecedent parts of the rules are: Negative (N), Zero (Z), Positive (P), the multiples of three degrees in [-12,12] degree interval (N12, N9, N6, N3, Z, P3, P6, P9, P12) and for the actions: from negative to positive in one tenth steps (AN10-AP10, Z).

The cart-pole demo reads the previously incrementally constructed rule base as a starting rule base for further reduction. Then the previously introduced strategies are applied for rule and rule group removal. With the truncated rule base a whole episode is evaluated. If the episode is considered successful with the removed rule or rule group, then the rule or rule group is removed permanently, otherwise it will be inserted back into the rule base. This is repeated until every rule is checked for possible removal (see Fig. 3., Fig. 4.,

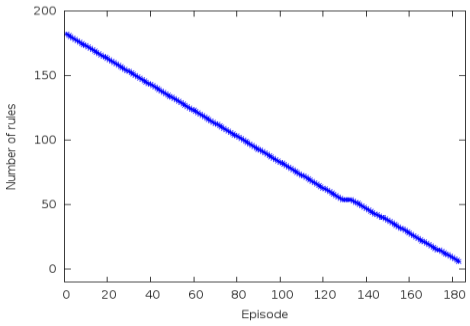

Fig. 3. The number of rules per episode using Strategy I.

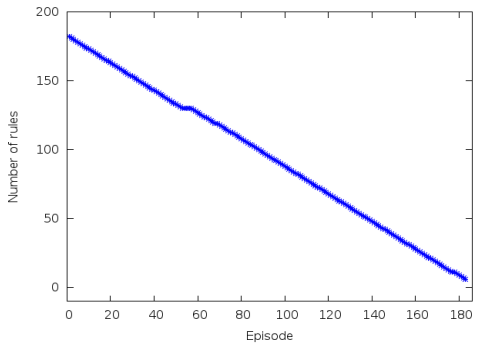

Fig. 4. The number of rules per episode using Strategy II.

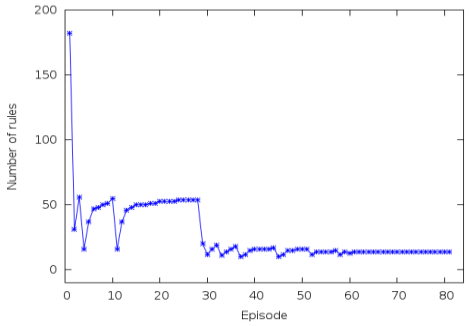

Fig. 5. The number of rules per episode using Strategy III. with exact reward matching.

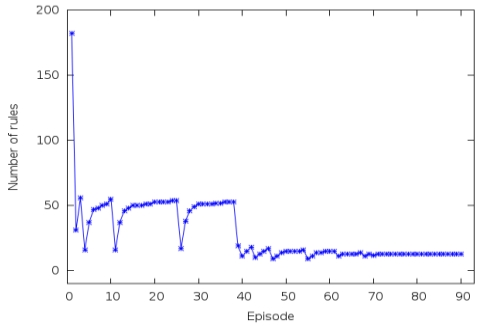

Fig. 6. The number of rules per episode using Strategy III. when differences in the rewards are allowed.

Fig. 5. and Fig. 6. – it can be easily recognized on the curve where the rules were not omitted).

The starting rule base, as the result of the incremental rule base construction method as seen in the previous subchapter, consists of 182 rules. Regarding Strategy I. and Strategy II. this means 183 episode runs (+1 for determining the correct reward without removing any rules) and possibly one rule fewer in each and every episode run, which means that the computational resource needs are possibly (when a rule is sentenced to be removed) decreasing with every episode.

Two different conditions were used in this example application for deciding whether the episode was successful or not. First the rewards were strictly checked to be the same as the previous episode, which means the rewards have to be the same as they were with the incrementally constructed rule base. Then in the second case the matching of the rewards were not strict, the rewards did not have to match exactly, it is enough for the rewards to be positive, meaning that the episode was successful (but the step-by-step solution can differ).

In this demonstration, the previously presented three strategies with both conditions were evaluated. Results are shown in Table III. and also on Fig. 3. through Fig. 6.

The smallest rule-base of only 6 rules was achieved by both Strategy I. and II., but when exact reward matching was not necessary, 5 rules in the final reduced rule base were enough to successfully achieve the required task.

The time taken for the various strategies is also shown in Table III. It can be clearly seen that the group removal strategy is some magnitudes faster, but in spite of being fast, in this very example the smallest rule-base found by this strategy contains more rules than in Strategy I.

TABLE I.
RULES IN THE RULE-BASE AFTER PERFORMING THE DECREMENTAL REDUCTION WHEN AN EXACT REWARD MATCH IS MANDATORY

R#	s_1	s_2	s_3	s_4	*a*	*q*
1	P	Z	Z	P	AP10	1907.33
2	P	Z	N3	N	AN10	1898.73
3	P	Z	Z	N	AN8	1904.22
4	P	Z	N3	P	AP8	1899.27
5	N	Z	N12	N	AP10	-5251.65
6	P	P	Z	N	AN8	-3100.5

TABLE II.

RULES IN THE RULE-BASE AFTER PERFORMING THE DECREMENTAL REDUCTION WHEN AN EXACT REWARD MATCH IS NOT MANDATORY

R#	s_1	s_2	s_3	s_4	*a*	*q*
1	P	Z	Z	P	AP10	1907.33
2	P	Z	N3	N	AN10	1898.73
3	P	Z	Z	N	AN8	1904.22
4	P	P	Z	N	AN8	-3100.5
5	P	Z	P12	P	AP6	-6446.87

TABLE III.

RESULTS OF THE DIFFERENT REDUCTION STRATEGIES

Strategy	*Episodes*	*Rules*	*Time*
I. w/ 0 diff	183	6	≈4180 s
I. w/ ∞ diff	183	5	≈4150 s
II. w/ 0 diff	183	6	≈4175 s
II. w/ ∞ diff	183	6	≈4210 s
III. w/ 0 diff	81	14	≈310 s
III. w/ ∞ diff	90	13	≈231 s

V. CONCLUSIONS

Possible decremental reduction strategies have been developed for the Fuzzy Rule Interpolation-based Q-learning method (FRIQ-learning), which can further reduce the size of the previously incrementally constructed fuzzy rule-bases. The various combinations of the reduction strategies were evaluated via the cart-pole application example. The application example clearly shows the benefit of using such a strategy, instead of the 2268 rules in the original FRIQ-learning example application and the 182 rules in the incrementally constructed FRIQ-learning example application only 5 rules were enough in a certain case using the presented rule-base reduction methods. One of the developed strategies provides an usable fuzzy rule-base in much less time (nearly 20 times faster in the cart-pole example) than the other basic strategies, but the size of final rule-base in this case tends to be somewhat, but not significantly larger (14 rules in the cart-pole example).

This huge drop in the number of rules is significant, which means that not only the amount of computational resources required for processing is greatly reduced, hence allowing for wider state-action spaces with the same computational resources, but the small number of rules in the final fuzzy rule-base could also allow the rules to be easily presented in a human readable form. This can be of great value in the case of problems where the solution of the problem is not known in a rule-base like form (step-by-step directions in the possible cases), but can be composed in a manner which is suitable for a reinforcement learning methods, in this case for the FRIQ-learning (define the desired goal with positive and negative rewards).

Acknowledgment

This research was realized in the frames of TÁMOP 4.2.4. A/2-11-1-2012-0001 „National Excellence Program – Elaborating and operating an inland student and researcher personal support system convergence program” The project

was subsidized by the European Union and co-financed by the European Social Fund.

References

[1] Appl, M.: Model-based Reinforcement Learning in Continuous Environments. Ph.D. thesis, Technical University of München, München, Germany, dissertation.de, Verlag im Internet (2000)

[2] Baranyi P., Kóczy L. T., Gedeon, T. D.: A Generalized Concept for Fuzzy Rule Interpolation, IEEE Trans. on Fuzzy Systems, vol. 12, No. 6, 2004, pp 820-837.

[3] Bellman, R. E.: Dynamic Programming. Princeton University Press, Princeton, NJ (1957)

[4] Berenji, H.R.: Fuzzy Q-Learning for Generalization of Reinforcement Learning. Proc. of the 5th IEEE International Conference on Fuzzy Systems (1996) pp. 2208-2214.

[5] Horiuchi, T., Fujino, A., Katai, O., Sawaragi, T.: Fuzzy Interpolation-Based Q-learning with Continuous States and Actions. Proc. of the 5th IEEE International Conference on Fuzzy Systems, Vol.1 (1996) pp. 594-600.

[6] Johanyák, Z. C., Tikk, D., Kovács, S. and Wong, K. K.: Fuzzy Rule Interpolation Matlab Toolbox - FRI Toolbox, Proc. of the IEEE World Congress on Computational Intelligence (WCCI'06), 15th Int. Conf. on Fuzzy Systems (FUZZ-IEEE'06), July 16--21, 2006, Vancouver, BC, Canada, pp. 1427-1433.

[7] Johanyák, Z. C.: Sparse Fuzzy Model Identification Matlab Toolbox - RuleMaker Toolbox, IEEE 6th International Conference on Computational Cybernetics, November 27-29, 2008, Stara Lesná, Slovakia, pp. 69-74.

[8] Sz. Kovács, "New Aspects of Interpolative Reasoning", Proceedings of the 6th. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, 1996, pp. 477-482.

[9] Sz. Kovács, and L.T. Kóczy, "Approximate Fuzzy Reasoning Based on Interpolation in the Vague Environment of the Fuzzy Rule base as a Practical Alternative of the Classical CRI", Proceedings of the 7th International Fuzzy Systems Association World Congress, Prague, Czech Republic, 1997, pp. 144-149.

[10] Sz. Kovács, and L.T. Kóczy, "The use of the concept of vague environment in approximate fuzzy reasoning", Fuzzy Set Theory and Applications, Tatra Mountains Mathematical Publications, Mathematical Institute Slovak Academy of Sciences, Bratislava, Slovak Republic, vol.12, 1997, pp. 169-181.

[11] Krizsán, Z., Kovács, Sz.: Gradient based parameter optimisation of FRI "FIVE", Proceedings of the 9th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, November 6-8, ISBN 978-963-7154-82-9, pp. 531-538, (2008).

[12] Rummery, G. A., Niranjan, M.: On-line Q-learning using connectionist systems. CUED/F-INFENG/TR 166, Cambridge University, UK., 1994

[13] Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction, MIT Press, Cambridge (1998)

[14] D. Tikk, I. Joó, L. T. Kóczy, P. Várlaki, B. Moser, and T. D. Gedeon (2002). Stability of interpolative fuzzy KH-controllers. Fuzzy Sets and Systems, (125) 1, 105-119.

[15] Vincze, D., Kovács, Sz.: Fuzzy Rule Interpolation-based Q-learning, SACI 2009, 5th International Symposium on Applied Computational Intelligence and Informatics, Timisoara, Romania, May 28-29, 2009, ISBN: 978-1-4244-4478-6, pp. 55-59.

[16] Vincze, D., Kovács, Sz.: Reduced Rule Base in Fuzzy Rule Interpolation-based Q-learning, 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, CINTI 2009, November 12-14, 2009, Budapest Tech

[17] Vincze, D., Kovács, Sz.: Incremental Rule Base Creation with Fuzzy Rule Interpolation-Based Q-Learning, I. J. Rudas et al. (Eds.), Computational Intelligence in Engineering, Studies in Computational Intelligence, Volume 313/2010, Springer-Verlag, Berlin Heilderberg, 2010, pp. 191-203.

[18] Watkins, C. J. C. H.: Learning from Delayed Rewards. Ph.D. thesis, Cambridge University, Cambridge, England (1989)

[19] The cart-pole example for discrete space can be found at: http://www.dia.fi.upm.es/~jamartin/download.htm