Machine Learning for an Adaptive Rule Base

Michal Jalůvka(✉) and Eva Volna

Department of Informatics and Computers,
University of Ostrava, 30. Dubna 22, 70103 Ostrava, Czech Republic
{michal.jaluvka, eva.volna}@osu.cz

Abstract. This paper deals with a design of an original approach for machine learning, which allows the rule base adaptation. This approach uses a fuzzy inference mechanism for decision making, finite-state machine for the rule base switching, and the teacher Supervisor for creating the most suitable rules for the activity (skill) which is supposed to be learned. The used fuzzy inference mechanism is the integration of LFLCore, which was developed at the Institute for Research and Applications of Fuzzy Modeling. The proposed approach of machine learning was tested in individual experiments, in which the system learns to move with its joints. How the system moves with its joints is given by patterns which are submitted before the beginning of learning. The evaluated results with possible modifications are mentioned at the end of this paper together with a formulated conclusion.

Keywords: Machine learning · Fuzzy inference system · Finite-state machine · Supervisor · Pattern · Rule base

1 Machine Learning

Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (e.g. progressively improve performance on a specific task) with data, without being explicitly programmed. Machine learning objectives vary depending on the approach we use. According to [10], there are 4 approaches.

- The first approach is to model the mechanisms that form the basis of human learning. An example may be a recognition of perceptions from the real world and their integration into different groups (classes).
- Another way to approach machine learning is empirical. This approach aims at discovering general principles that relate to learning algorithms characteristics and general domain principles within which these algorithms operate.
- We can also approach machine learning in general. Here, an emphasis is placed on formulating and proving theorems on the workability of whole classes of learning problems and algorithms proposal to solve these problems.
- The last option to approach machine learning is application. This approach is generally related to algorithm proposal (where we solve problem formulation,

R. Fullér et al. (Eds.): WILF 2018, LNAI 11291, pp. 3–16, 2019.
https://doi.org/10.1007/978-3-030-12544-8_1

solution proposal, implementation). From the point of view of machine learning, we focus on formulating a problem, proposal of representation of training examples (or training knowledge), creating a training set, and generating a knowledge base using machine learning.

2 Machine Learning Approaches to an Adaptative Rule Base

An adaptative rule base is based on finding and assessing the suitability rules. We try to have a rule which provides better decision results (as much accuracy as possible). This is similar to human learning of new skills. They try to find the steps or procedures to best control their skill. Best practices will be retained in memory for use in the future.

None of the above-mentioned machine learning approaches (Table 1) addresses the issue of machine learning to adapt the rule base. This issue is dealt with in the following works: *An Adaptive Fuzzy Controller for Trajectory Tracking of Robot Manipulator* [7], *Adaptive Fuzzy Rule-based Classification Systems* [11], and *Adaptive Fuzzy Controller: Application to the control of the temperature of a dynamic room in real time* [14], whose contribution to the problem is summarised in Table 2 according to the set criteria:

Table 1. An overview of basic machine learning methods

	Algorithms	Principle
Decision trees [8]	TDIDT, ID3, ASSISTANT, C4.5	Tree nodes are evaluated according to the attributes of the instance. The decision-making process starts from the root to the nodes to the leaf. The leaves are valued binary values (YES/NO)
Neural networks [5]	Perceptron, Backpropagation	Choosing the right topology and using the training set to configure the neural network The network consists of layers (input, hidden, output) containing neurons
Bayesian learning [3]	Gibbs algorithm, Bayes classifier, EM algorithm	Classification of hypotheses based on conditional probabilities
Feedback learning [15]	Q-learning	From the set of actions is chosen such an action, thanks to it agent finds himself in a new situation and gain the highest reward represented fair value
Learning with a set of rules [15]	Learn-one-rule FOIL	A tree structure whose nodes contain a description of IF-THEN rules. The goal is to select a node/subtree with the best candidates describing the training examples
Evolutionary algorithms [4]	Genetic algorithms	Search for hypotheses (possible solutions - population) that are expressed numerically (sequence of bits). Iteratively to generate new hypotheses from existing hypotheses using crossover and mutation and maintained in the population according to the fitness function

Table 2. Summary of machine learning approaches to address adaptation of the rule base

	Classification	Presence of the fuzzy inference system	Language description in the form of natural language	Way of adaptation
Work [7]	No	Yes	No	Changing the parameter for the given component (P, D) of a controller
Work [11]	Yes	Yes	No	Changing the height of the fuzzy set, classifying into 2 classes
Work [14]	No	Yes	No	Changing the support of a fuzzy set

- The first criterion was the implementation of decision making (not classifying objects) of the fuzzy inference system based on defined IF-THEN rules.
- The second criterion was a language description that is in the form of a natural language text.
- The third criterion was how to adapt the rules. This criterion is not defined because it is not known how the rules should be adapted. Either there would be a given pattern according to which the rules would be set or the rules would be prioritised based on their frequency of use. If a suitable solution is found, this strategy could be taken into account.

The paper aims at proposing a machine learning approach for an adaptive rule base. Adaptation methods for rule bases that are described in the above-mentioned publications do not meet our defined criteria.

3 Proposal of a Machine Learning Method for Adaptive Rule Base

The proposed machine learning approach allows the adaptation of the rule base according to the training set. The rule base adaptation is represented by changing certain rules according to the required criteria. This change can be seen as deductive learning. If we are to achieve a reinforcement of the right rules, supervised learning or reinforcement learning can be used. When using the supervised machine learning approach, it is important to have a training set (patterns). As a way to get a pattern, an online approach was chosen, i.e. to get a pattern during adaptation from the motion of the monitored joints.

These machine learning approaches (deductive learning, supervised learning, online learning) form the basis of the proposed machine learning method. The system architecture (Fig. 1), which implements the proposed machine learning approach, consists of three parts [6]:

- Fuzzy inference system,
- Supervisor,
- Finite-state machine.

The basis of the proposed approach is LFLCore, which is part of the LFLC application [2], which was developed at the Institute for Research and Applications of Fuzzy Modeling.

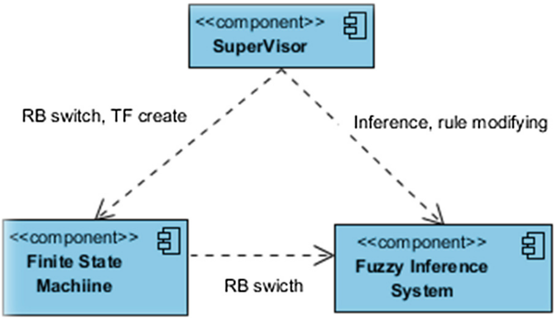

Fig. 1. System architecture

3.1 Linguistic Context

The linguistic context is defined in [13] as follows (1):

$$w = \langle v_L, v_S, v_R \rangle \quad v_L, v_S, v_R \in R \quad v_L < v_S < v_R \tag{1}$$

where v_L denotes the smallest value, v_R is the largest value, and v_S is the usual mean value to be considered in the given situation.

The construction and distribution of fuzzy sets depends on the linguistic context (specifically, on evaluating linguistic expressions). According to the default parameter settings in LFLC [2], the language "one-sided" context is defined as follows (2):

$$w = \langle 0; 0.4; 1 \rangle \tag{2}$$

The proposed machine learning approach uses two language "one-sided" contexts that are symmetric by parameter v_L. These linguistic contexts are uniformly named as language "two-sided" contexts, which are defined as follows (3):

$$\begin{gathered} w = \langle v_{-R}, v_{-S}, v_L, v_S, v_R \rangle \\ v_{-R}, v_{-S}, v_L, v_S, v_R \in \mathrm{R} \\ v_{-R} < v_{-S} < v_L < v_S < v_R \\ |v_{-R}| = v_R \\ |v_{-S}| = v_S \end{gathered} \tag{3}$$

The linguistic context for each input and output variable is set to (4):

$$w = \langle -1; -0.4; 0; 0.4; 1 \rangle \tag{4}$$

3.2 Expressions of Variables

As expressions of variables are used evaluating linguistic expressions [13], which are language expressions representing either a value on an ordered scale (usually a certain number), or a position on it (left/right).

They include atomic expressions (Fig. 2) and fuzzy numbers, which can be complemented by language operators, signatures (+, −), and joined by conjunctions (and, or).

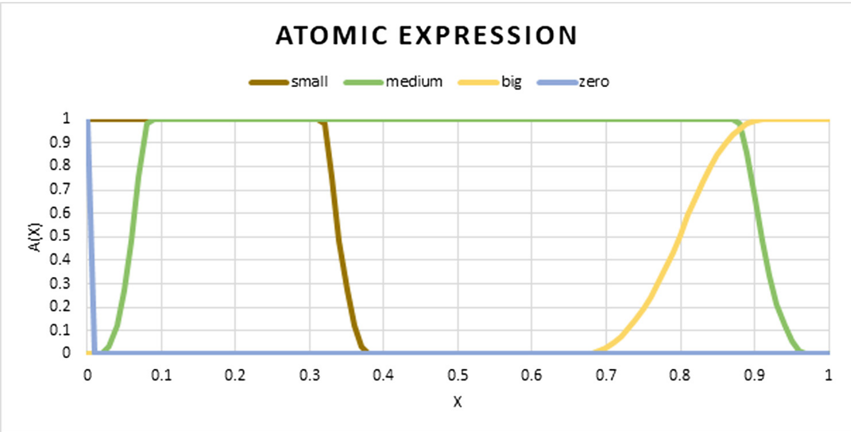

Fig. 2. Atomic expressions

3.3 Inference Method

Since the expressions of input/output variables are evaluating linguistic expressions, Perception Based Logical Deduction (PBLD) is appropriate for working with these expressions as described in [12]. This method handles the language description which is linguistically-logically interpreted. Perception (Fig. 3) means such an evaluating linguistic expression to which a value is assigned in the defined context. According to the perceived perception, an appropriate rule from the linguistic description is subsequently activated, and the result obtained in the given rule is evaluated in the form of evaluating linguistic expression.

The number of activated rules of the inference method corresponds to the number of input variables. This appears to be an advantage over traditional inference methods that process a relational interpreted language description (e.g. Mamdani fuzzy inference [9]) which activate all the defined rules.

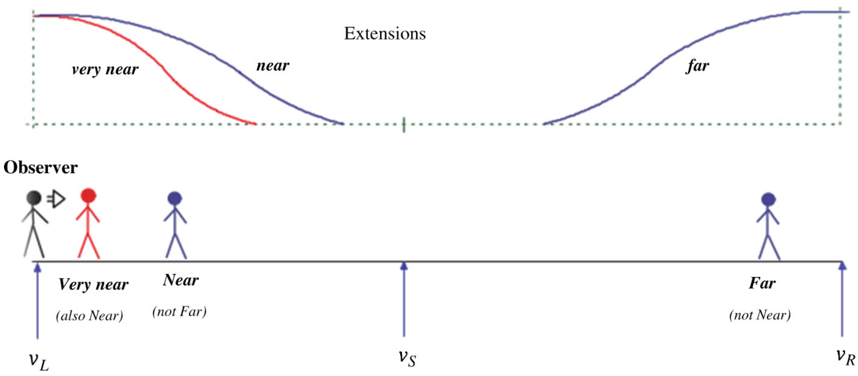

Fig. 3. Perception (adapted from [13])

3.4 Defuzzification Method

According to [12], Defuzzification of Evaluative Expressions (DEE) is recommended for the PBLD inference method. DEE is a collection of methods Last of Maxima (LOM), Mean of Maxima (MOM), First of Maxima (FOM) transferring linguistic expression to a corresponding real number. Generally, DEE is defined as (5):

$$DEE(A) = \begin{cases} LOM(A) & \textit{if A is small or zero} \\ FOM(A) & \textit{if A is big} \\ MOM(A) & \textit{otherwise} \end{cases} \tag{5}$$

4 Finite-State Machine

The finite-state machine is based on a mathematical model of the language grammar, the so-called Chomsky hierarchy [1]. Finite-state machine can recognise a regular language, which is at the lowest level in the hierarchy (Type-3 grammars). The deterministic finite-state machine FA can be defined as follows (6):

$$FA = \left(Q, \sum, \delta, q_0, F\right) \tag{6}$$

where:

Q is a finite, non-empty set of states
$\sum$ is the input alphabet (a finite, non-empty set of symbols)
δ: $Q \times \sum \rightarrow Q$ is the state-transition function
$q_0 \in Q$ is an initial state
$F \subseteq Q$ is the set of final states.

In the proposed machine learning model, states are reflected as the rule bases that are ready to perform operations of a fuzzy inference system.

- Switching the finite-state machine to the next state is decided based on the current state of the counter that acquires the values of the natural numbers.
- The state-transition function δ is reflected as a rule base (states) switching according to the respective value of the counter (symbol).
- The initial state q_0 (initial rule base) is set by the user or supervisor.
- Stop of a run of the final machine occurs when there is no state-transition function for a particular symbol.

5 Supervisor

The last block of the proposed system architecture with machine learning is the supervisor. The supervisor has access to all rules from each rule base. Each rule contained within the database has a parameter fitness. The fitness determines whether the rule is applied when using inferential methods. Rule fitness is reinforced or suppressed during adaptation depending on whether the desired system state is achieved after the performance of the operation (inference). The higher the rule fitness, the better candidate for further decision making the rule is. The fitness of all rules is initialised to 0.

Required states are submitted to the supervisor as patterns. Patterns (or sequence of patterns) are loaded onto a pattern-tape from which the supervisor reads. In addition to information on the required states, the pattern also contains information about:

- the number of a particular step,
- base rules over which the operation (inference) will be performed.

While browsing the patterns, inter-state switching occurs (such a rule base is switched, which is included in the pattern), i.e. a finite-state machine is produced. These inter-state switches are recorded by the supervisor as the transfer functions of the finite-state machine for which is valid (7):

$$\delta_i : Q_i \times \sum\nolimits_i \rightarrow Q_{i+1} \tag{7}$$

where

Q_i is the rule base contained in the i-th pattern, Q_i is a subset of Q,
$\sum_i$ is the number of a particular step (counter value) contained in the i-th pattern,
Q_{i+1} is the base rule contained in the (i + 1)-th pattern.

The course of adaptation is divided into these basic phases (Fig. 4).

1. *Pattern loading*
   Patterns are loaded onto a pattern-tape from which the Supervisor gradually retrieves the pattern that is at the front of the queue. The pattern contains the step number (counter state), the base rule name (on which the operation to be performed), and the required state (the state to be performed).

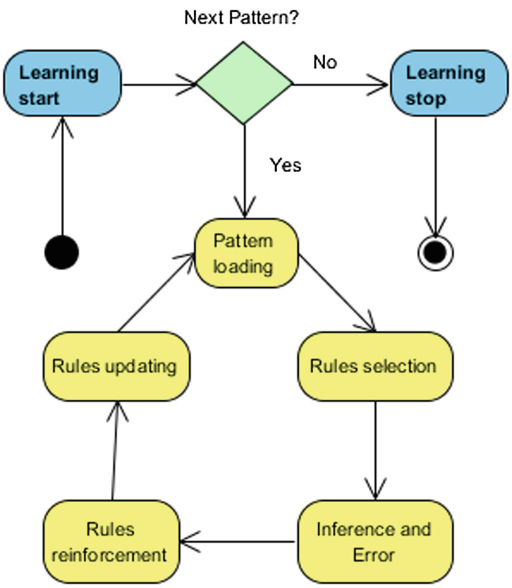

Fig. 4. The course of adaptation

2. *Rules selection*
   In the second phase, the supervisor selects such rules from the rule base that have the given antecedent, which is formed by the required and current state. The desired state value is obtained from the current pattern. The current state value is obtained from the internal state of the given device (from the joint). The values of the desired and current state are assigned to evaluating linguistic expressions to the given context (perception).
3. *Inference and Error*
   In the third phase, the Supervisor performs an inference over the selected rules, which are gradually activated, when the inference is called. After each inference, an error is calculated, e.g. how the current state after the inference differs from the reference value describing the pattern. The error is calculated by (8):

$$E(x, y) = |y - x| \tag{8}$$

   where y is the required value, x is the current value.
   As a result of this phase errors of all rules are calculated and transferred to the next phase.
4. *Rules reinforcement*
   Based on the calculated error, the supervisor will evaluate the rules either as appropriate (+1), inappropriate (−1), or almost acceptable (0). the number of almost appropriate rules *MAX_k* is fixed. If the supervisor identifies an almost appropriate rule, parameter k increases. Rules reinforcement is given by (9):

$$f(E,k) = \begin{cases} 1 & \mathit{if}\ E = 0 \\ 0 & \mathit{if}\ E > 0, k \leq MAX\_k \\ -1 & \mathit{otherwise} \end{cases} \tag{9}$$

5. *Rules updating*
   In the last phase, the resulting value of the function f calculated from the previous phase is added to the fitness of each rule. This fitness determines whether the rule is correct when performing the given activity.

6 Experimental Outcomes

The proposed machine learning approach was tested on several examples in which we teach the system to perform activities consisting of certain steps [6]. Each step represents the movement of an individual joint. In this experiment, the rules for the movement of four joints are adapted. Here, the system should control the joints so that it can take one step.

Joint Specifications:
The input value “Current state” is reflected as the internal state of one joint. The angle of rotation of a given joint can take values from the interval $[-180°; 180°]$. The angle of the joints J1 and J3 is expressed by vectors $\vec{n}$ and $\overrightarrow{v_n}$ and the angle of the joints J2 and J4 is expressed by vectors $\vec{m}$ and $\overrightarrow{v_m}$ (Fig. 5). The output variable “Action hit” is reflected as a change in the state of one joint having values from the interval $[-360°, 360°]$. As required by the fuzzy inference system, these intervals are converted according to a defined context. The input variable “Requested state” reflects the state obtained from a given pattern whose value belongs to the given language context.

If the top joint (J1 and J3) is in action, the bottom joint (J2 and J4) retains the angle between vectors $\vec{m}$ and $\overrightarrow{v_m}$ (Fig. 5).

The structure of the presented pattern is shown in Table 3, where the “desired state” is the measured value corresponding to the angle between the vectors $\vec{n}$ (or $\vec{m}$) and $\vec{v}$. In this case, the Finite-state machine switches the rules of the individual joints according to the defined pattern stored in Table 3.

The structure of rule base is as follows:

- Input/Output Variables:
  - Context: $w = \langle -1; -0,4; 0; 0,4; 1 \rangle$
  - Expressions: (±ro ze, ±vv sm, ±vv me, ±vv bi), where ‘ro ze’ is roughly zero, ‘vv sm’ is very very roughly small, ‘vv me’ is very very roughly medium, ‘vv bi’ is very very roughly big.
- Linguistic description
  - Current status & Desired state - > Action hit.
- Rules:
  - Number of rules: 512.
  - Rule specification: Inconsistent deactivated rules.

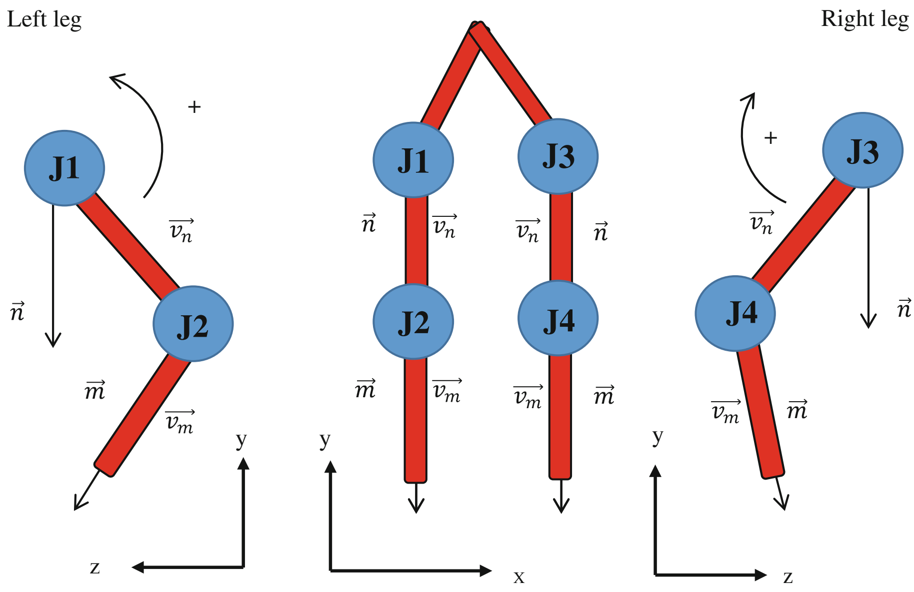

Fig. 5. Joints of left and right legs

Table 3. Sequence for movement of both legs

Step	Rule base regarding the joint	Required state	Step	Bule-base regarding the joint	Required state
1	J1	0	15	J2	−0.02
2	J3	0	16	J4	−0.02
3	J2	0	17	J1	0.138
4	J4	0	18	J3	−0.1
5	J1	0	19	J2	−0.072
6	J3	0.094	20	J4	−0.094
7	J2	0	21	J1	0.1
8	J4	−0.36	22	J3	−0.02
9	J1	−0.072	23	J2	−0.072
10	J3	0.21	24	J4	−0.27
11	J2	0.027	25	J1	0
12	J4	−0.205	26	J3	0
13	J1	−0.1	27	J2	0
14	J3	0.127	28	J4	0

- Inference method:
  – Perception-Based Logical Deduction.
- Defuzzification method:
  – Defuzzification of Evaluative Expressions.

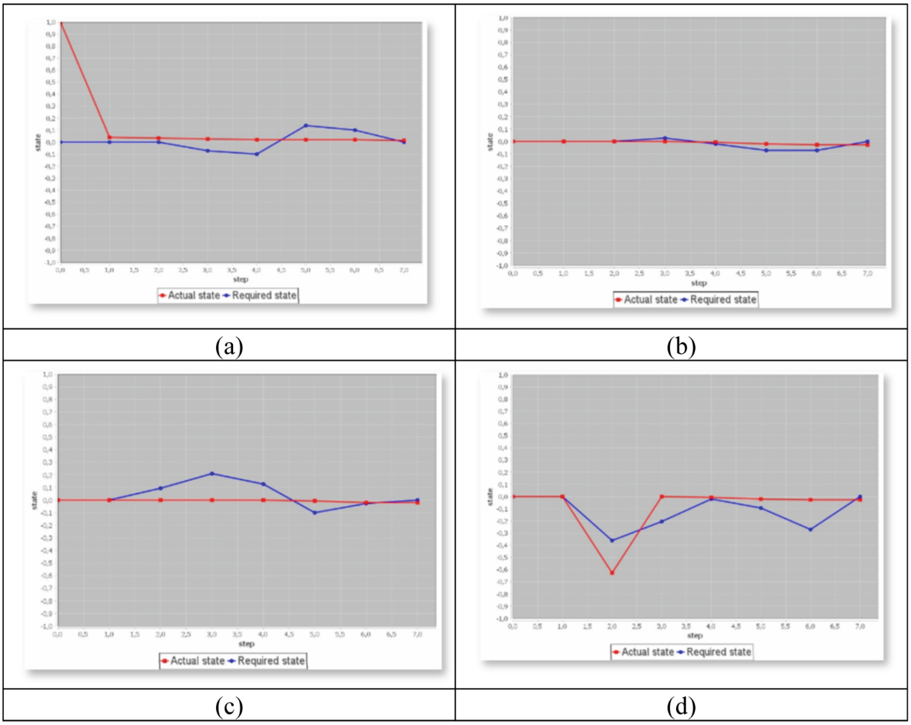

Fig. 6. (a) Continuous change in the motion of the left hip (J1). (b) Continuous change in the motion of the left knee (J2). (c) Continuous change in the motion of the right hip (J3). (d) Continuous change in the motion of the right knee (J4). (Color figure online)

Adaptation of the rule base is done in 21 iterations. This number is determined by setting the initial states for the upper joints J1 and J3 (Fig. 5). The initial value of the lower joints J2 and J4 is always set to 0°. The number of steps in one iteration of adaptive learning is set to 28. This sequence of patterns includes seven movements for each joint. A continuous change of state in one iteration is shown in Fig. 6. The initial value of the left hip is set to 180° and the initial state of the other joints is set to value 0°.

6.1 Evaluation of Experiments

During the adaptation in each experiment, we managed to adapt the individual rule base and select the most appropriate rules for the given activity. These rule bases can be uploaded to an expert system or LFLC [2] (compatibility is preserved). However, it must be noted that each iteration occurs for setting different initial values and reinforces rules that have the same antecedent and different consequence. As a result, these rules are evaluated in such a way that one of them has a positive fitness value and others have negative fitness values.

The adaptive system was tested. Figure 7 shows a continuous change in the movement of one joint - left hip (J1). The initial joint condition was set to −180°. The aim of the experiment was to make the condition of the joint condition comparable with the condition of the joint that was detected during the process of adaptation.

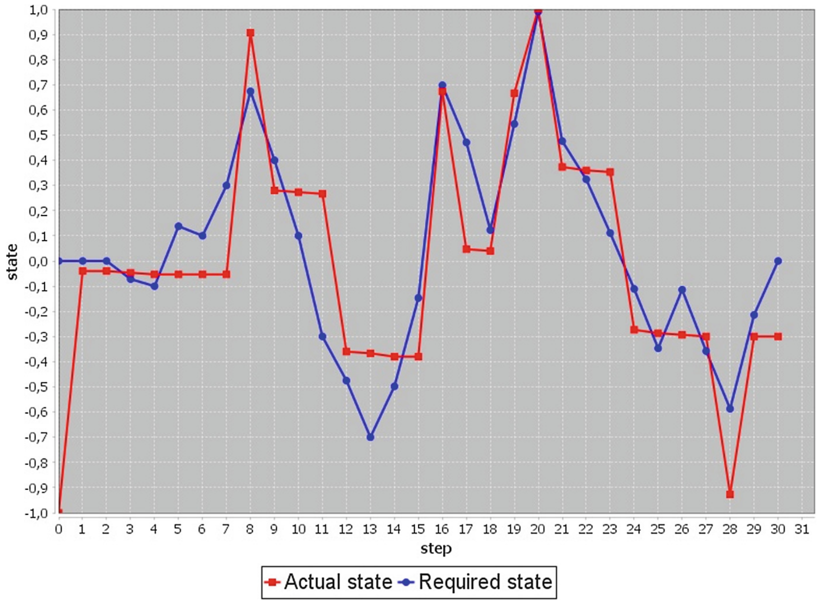

Fig. 7. Continuous change in the movement of one joint.

The red curve showing the current state of the joint at each step approaches, after adaptation, the blue curve showing the state that has to occur in a given step. In some steps, you can see it stay on a place, even if it is to move a bit. It is because when using the defined evaluating linguistic expressions, the adaptation of the system does not register small movements and evaluates the best solution to stay in a place (Fig. 6, steps 1−7).

The benefits of this approach to adaptation of the rule bases are the following:

- The user does not intervene to the rule bases during adaptation, because the supervisor solves everything in the proposed system.
- The boundaries of the language context are fixed, the transformation of the language context to the desired interval and vice versa can be done in the fuzzy inference system interface.
- Simple adaptations when changing the desired allowable value interval for a given joint.
- Simple adaptations when changing patterns of the training set.

7 Conclusions

The aim of the paper was to design a machine learning approach to adapt rule bases. Machine learning's own approach includes the initial setting of the rule bases, the inter-state switching proposal, and the proposal of the method of their adaptation. Each rule base, under which the fuzzy inference system determines, has a defined limited number of evaluating linguistic expressions, type of a fuzzification and defuzzification method, and a randomly generated set of inconsistent of rules. For the inter-state switching, a simple mechanism of the finite-state machine was proposed to allow the rule bases to be switched. Thanks to the finite-state machine, the system avoided using only one rule base that would be defined by multiple input/output variables. We have also proposed how to adapt the rule bases, e.g. what the most appropriate rules should contain according to the pattern. This proposed approach to machine learning adapting the rule bases was tested. The subject of the experimental study was the joint movement according to the presented patterns. The course of these experiments was recorded and subsequently evaluated.

The proposed system will be further developed because we have identified the following shortcomings in the adaptive learning, which will be gradually eliminated.

- The system does not react to small movements (the best option is to remain in a place). This is evident from Fig. 6.
- Redundancy of rule bases. If an antecedent has more consequences, the proposed system prefers such consequence, which fitness is the greatest.
- Evaluation of the best rule is calculated based on the difference between the desired and the current state, see Eqs. (8) and (9).
- It does not cover all combinations of rules, but the proposed rule base is sufficient for this experimental study.

This proposed approach of machine learning can be used in systems such as a humanoid robot. This robot can learn a few activities consisting of simple activities. These activities include, for example, the movement of a robot's joint. Knowledge of "joint movement" could expand the knowledge of moving one leg or both legs. If the system knows to control its legs, it could learn to walk, run, squat, jump. In doing so, these rules to adapt to learning new activities/skills would be used. After deploying the system into operation, it can perform a newly learnt activity. It can be stated that the more activities the system can do, the larger the area of the linguistic description in the rule base it will cover.

Acknowledgments. The research described here has been financially supported by University of Ostrava grant SGS04/PřF/2018. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of the sponsors.

References

1. Chomsky, N.: Three models for the description of language. IRE Trans. Inf. Theory **2**(3), 113–124 (1956)
2. Dvorak, A., Habiballa, H., Novak, V., Pavliska, V.: The software package LFLC 2000-its specificity, recent and perspective applications. Comput. Ind. **51**, 269–280 (2003)

3. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. **29**(2–3), 131–163 (1997)
4. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. Mach. Learn. **3**(2), 95–99 (1988)
5. Gurney, K.: An Introduction to Neural Networks. CRC Press, Boca Raton (2014)
6. Jaluvka, M.: The Machine learning for the adaptive rule base (in Czech). Master thesis, University of Ostrava, Czech Republic (2017)
7. Khalate, A.A., Leena, G., Ray, G.: An adaptive fuzzy controller for trajectory tracking of robot manipulator. Intell. Control Autom. **2**(4), 364–370 (2011)
8. Lior, R.: Data Mining with Decision Trees: Theory and Applications, vol. 81. World Scientific, Singapore (2014)
9. Mamdani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. an experiment in linguistic synthesis with a fuzzy logic controller. In: Readings in Fuzzy Sets for Intelligent Systems, pp. 283–289 (1993)
10. Mitchell, T.M.: Machine Learning. McGraw-Hill, Boston (1997)
11. Nozaki, K., Ishibuchi, H.: Adaptive fuzzy rule-based classification systems. IEEE Trans. Fuzzy Syst. **4**(3), 238–250 (1996)
12. Novák, V., Perfilieva, I.: On the semantics of perception-based fuzzy logic deduction. Int. J. Intell. Syst. **19**(11), 1007–1031 (2004)
13. Novák, V., Perfilieva, I., Mockor, J.: Mathematical Principles of Fuzzy Logic. Springer, Heidelberg (2012)
14. Rojas, I., et al.: Adaptive fuzzy controller: application to the control of the temperature of a dynamic room in real time. Fuzzy Sets Syst. **157**(16), 2241–2258 (2006)
15. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education Limited, London (2016)