Implementation of Reinforcement Learning Tools for Real-Time Intelligent Decision Support Systems

Eremeev A.P.,
Kozhukhov A.A.
Institute of Automatics and Computer Engineering
Moscow Power Engineering Institute
Moscow, Russia
Email: eremeev@appmat.ru
Email: saaanchezzz@yandex.ru

***Abstract*—The paper describes implementation of multi-agent reinforcement learning tool based on temporal differences. The possibilities of combining learning methods with statistical and expert methods of forecasting for subsequent integration into the forecasting subsystem for use in long-term intelligent decision support system of real-time were considered. The work is supported by RFBR and BRFBR.**

***Keywords*—artificial intelligence, intelligent system, real time, reinforcement learning, forecasting, decision support.**

I. Introduction

Reinforcement learning methods (RL) [1], based on the using large amount of information for learning in arbitrary environment, is one of the most rapidly developing areas of artificial intelligence, related with the development of advanced intelligent real-time systems (RT IS) typical example of which is a real-time intelligent decision support system (RT IDSS) [2,3].

One of the most promising in terms of use in RT IS, related to the class of dynamic intelligent systems [4,5], is a learning, based on the temporal differences (TD) [1]. TD-learning process is based directly on experience without preliminary knowledge about the environmental behavior of the model environment. TD-methods intending for multi-dimensional time series can update the estimates, including other received estimates without waiting for the final result. Thus, TD-methods are adaptive. The latter property is very important for the IS of semiotic type abled to adapt to changes in the controlled object and environment [3].

Using the multi-agent approach for dynamic distributed control systems and data mining systems, including RT IDSS, that are capable of improving the efficiency and reliability of such systems is the fastest growing and promising approach [6].

When modern RT IDSS are developing, important consideration should be given to means of forecasting the situation at the object and the consequences of decisions, expert methods and learning tools [7]. These resources are necessary for modification and adaptation of RT IDSS with regard to changes in object and external environment, and for enhancing the application field and improving system performance.

II. Principles of reinforcement learning

We assume, that the uncertainty of information entering the RT IDSS‘s database about the problem area current state of the object and environment, mainly associated with the erroneous operation of the sensor (sensors) or errors from dispatching personnel. The functions of the RL-learning contains of the non-Markov decision model‘s adaptation to the situation by analyzing the history of decision and improving their quality [1,8,9].

The RL-learning decision-making module adjusting the decision-making strategy by interacting with the environment and the analyzing the evaluation function (payment function), called the agent. Agent target is to find an optimal (for a Markov process) or acceptable (for not-Markov process) decision-making strategies, also called the policies, in the process of learning. Intelligent agents must be able to support multiple learning paths and adapt to experience changes in the environment.

The target of RL-learning is to maximize the expected benefits R_t, which is determined as a function defined on the sequence of rewards:

$$R_t = r_{t+1} + r_{t+2} + ... + r_{t+T}, \quad (1)$$

where T - the final time step, r_t - reward at the time step t. This approach can be used in applications where the final step can be defined by natural way, from the kind of the problem, i.e., when the interaction of the agent-environment can be divided into a sequence, called episodes.

The main problem of RL-learning, is to find compromise between learning and application by agent. Agent must prefer actions that he had already applied and found that they are effective in terms of getting more reward. On the other hand, to detect such actions, the agent must try to perform actions that were not previously performed. Thus, agent should be use the actions that are already known, and explore new actions to be able to have the best choice in the future. An important characteristic of RL-learning is getting delayed compensations that occur in complex dynamic systems. This means that the action produced by the agent can affect not only the current award, but also all subsequent.

In terms of the use in RT IDSS, TD-methods can solve several problems: the problem of prediction the values of cer-

tain variables within a few time steps and management problem based on the how RL-agent's learning affects the environment. Thus, the agent should predict the future environment state and use these values to change the environment in order to maximize the received reward.

In despite of the problem of finding a compromise between learning and application, RL-learning has a number of important advantages for use in RT IDSS:

- using simple feedback on the basis of scalar payments;
- rapid response in supporting mode when the agent needs to adapt to changes in the environment quickly;
- interactivity and the ability to change (replenishment) analyzed data (history);
- effectiveness in non-deterministic environments;
- efficiency in conjunction with the temporal models for problems of finding consistent decisions;
- openness to modification and the comparative simplicity of inclusion in intelligent systems for various purposes (planning, management, training, etc.).

III. Scheme of reinforcement learning methods based on temporal differences

Let's consider RL-methods based on temporal differences (TD-methods) in terms of their use in the RT IDSS [1, 10, 11]. TD-methods using experience to solve the prediction problem. Given some experience following a policy TD-methods update their estimates. If nonterminal state S_t is visit at time t, then methods update their estimates $V(S_t)$, based on what happened after that visit, i.e. for the valuation adjustment to wait only the next time step is necessary. Directly at time $t+1$ target evaluation value is formed and necessary adjustment to existing reward r_{t+1}, and evaluation $V(S_{t+1})$ is produced. The simplest TD method, known as TD(0), is:

$$V_{S_t} \leftarrow V(S_t) + \alpha[r_{t+1} + \gamma V(S_{t+1}) - V(S_t)], \qquad (2)$$

where γ - the valuation of the terminal state. The target value will be $r_{t+1} + \gamma V(S_{t+1})$ during adjustment. TD-methods are partly based on current estimates in their adjustments so they are self-adjusting. One of the advantages of TD-methods is that they do not require knowledge of the environment model with its rewards and the probability distribution of the next states.

While using the TD-methods, estimate (benefit) becomes known at next time step. This advantage of TD-methods often crucial when using in RT IDSS, because as episodes in some situations can be so lengthy that the learning process delay related to the necessity to complete the episodes are too large.

TD-methods are learning on the basis of each transition, regardless of the ongoing actions in the future and, accordingly, are not sensitive to situations where it is necessary to ignore episodes or reduce the significance of the episodes which can greatly slow down the training. TD-methods can be divided into two main categories - on-policy and off-policy approaches. In the on-policy approach, the strategy used to control, similar to the evaluation strategy that improved during learning. In the off-policy approach, control strategy has no relationship with the evaluation strategy.

SARSA – on-policy TD control method. For an on-policy method we must estimate $Q^{\pi(s,a)}$ for the current behavior policy π and for all states s and actions a:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)], \qquad (3)$$

where α - constant step length; γ - the valuation of the terminal state. This update is done after every transition from a nonterminal state s_t. If s_{t+1} is terminal, then $Q(s_{t+1}, a_{t+1})$ is defined as zero. This rule uses every element of the quintuple of events, $(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$, that make up a transition from one state-action pair to the next. As in all on-policy methods, we continually estimate Q^π for the behavior policy π, and at the same time change π toward greediness with respect to Q^π .

Q-learning - off-policy TD control method that finds the optimal value of Q function to select the follow-up actions and at the same time determines the optimal strategy. Similarly to method TD(0) in each iteration there is only knowledge of the two states: s and one of its predecessors. Thus, the values of Q allow to get a glimpse of the future actions quality in the previous states and to make the decision task easier.

For this method, we need to evaluate the function of the value of the action $Q^\pi(s, a)$, for the current policy π and for all states s and actions a, where the episode consists of a sequence of alternating states and state-action pairs. One-step Q-learning is characterized by the following relationship:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max \alpha Q(s_{t+1}, a) - Q(s_t, a_t)], \qquad (4)$$

where α - constant step length; γ - the valuation of the terminal state. In this case, the desired function Q directly approximates Q^π - optimum function values of action, regardless to the applied strategies. The strategy determines which state-action pairs are visited and adjusted. In Q-learning update rule is always based on the greedy and deterministic strategy which improved. At the same time, the actions selected for control, based on a different strategy (not depending on $Q(s, a)$). For example, for generating action, the strategy can be used with uniform distribution in space operations.

To ensure the necessary convergence all couples must continue to adjust. This is the minimum requirement in the sense that each method is guaranteed to finding the optimum course of action. It was found [1] that function converges to Q^π with probability 1 in case of stochastic approximation for the step length sequence of values Q_t.

TD (λ) - method in which a time difference contains of the n-steps. There is an additional memory variable associated with each state, its eligibility trace. The eligibility trace for state s at time t is denoted $e^t(s)$. On each step, the eligibility traces for all states decay by $\gamma\lambda$, and the eligibility trace for the one state visited on the step is incremented by 1:

$$e^t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if} \quad s \neq s_t \\ \gamma\lambda e_{t-1}(s) + 1 & \text{if} \quad s = s_t \end{cases} \qquad (5)$$

where γ – the valuation of the terminal state, λ – trace-decay parameter. These traces show the acceptability of each state at

the changes taking place in learning if there is corroborating event. Thus, for method TD (λ):

$$V_{S_t} \leftarrow V(S_t) + e(s)\alpha[r_{t+1} + \gamma V(S_{t+1}) - V(S_t)], \quad \forall s \in S : e(s) \neq 0 \tag{6}$$

While using eligibility traces, all states must be updated at each step (choosing of actions a at the state (s_t) and receive reward r at the state (s_{t+1})). At the same time, current reward information is propagated back to the states with higher values of eligibility traces. It can be shown that with value $\lambda = 0$, the algorithm becomes similar to the algorithm TD (0), updating only state s_t at step $t + 1$. With a value $\lambda = 1$, the algorithm finds a solution equivalent to full-passing method that makes sense only to occasional problems in the assessment of value after getting all rewards and counting final benefit.

IV. Multi-agent case in reinforcement learning

It is known that multi-agent systems are groups of autonomous interacting entities (agents) having a common integration environment and capable to receive, store, process and transmit information in order to address their own and corporate (common to the group of agents) analysis tasks and synthesis information [9]. The structure of the multi-agent system for RL-training is influenced by several agents at the same time and accordingly the action of each agent may depend on the actions of other agents system.

The advantages of multi-agent systems in the RL-learning:

- the possibility of parallel computing, as it uses the distributed type of the agents interaction in order to increase system performance;
- exchange of experience between the agents, by means of learning and simulation, that allows to help RL-agents with similar tasks to learn faster and achieve higher productivity;
- resiliency - system continues to operate even after failure of one or more agents;
- scalability - inclusion or exclusion of the agent from the system does not affect the operation of the system;

In addition, there are few disadvantages of multi-agent systems:

- complexity setting of learning targets;
- unsteadiness of learning problems arising from the fact that all agents are learning at the same time and each agent faces the challenge of changing the learning targets. So the basic strategy can vary with changes in strategies of other agents. Thus, RL-agent needs to find a compromise between the use of current knowledge and research environment to gather information and improve this knowledge;
- necessity of coordination;
- exponential growth in discrete state-action space. The basic Q-learning algorithm estimates the values of all possible pairs of the state-action, which leads respectively to an exponential increase in computational complexity;

V. Implementation of reinforcement learning tool for the forecasting subsystem

On the basis of statistical and expert methods of forecasting was suggested combined (integrated) prediction method [7,10], which contains of an averaging the results obtained on the basis of the moving average method and the Bayesian approach, based on weighting coefficients. Then, resulting prediction corrected by values of series obtained by the method of exponential smoothing. After that, forecast adjusted by results of the expert methods: ranking and direct evaluations. The probability of each outcome acquired by statistical methods, increased or decreased depending on the expert assessment values for these outcomes.

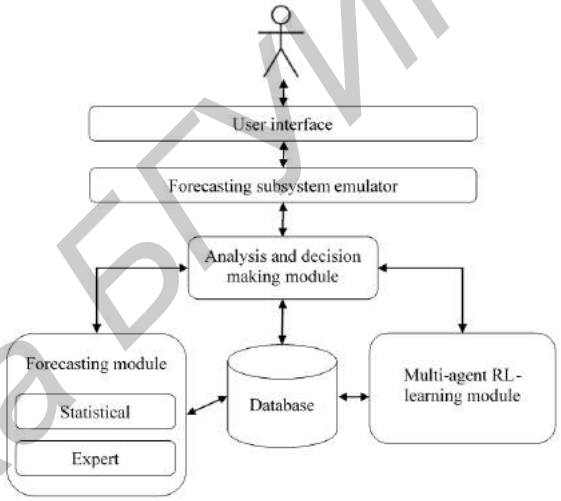

Fig. 1. Architecture of forecasting subsystem

Proposed architecture (Fig. 1) of prediction subsystem includes:

- emulator, which simulates the state of the environment with using of various system parameters change algorithms (linear and random) in the online database;
- prediction module based on statistical methods (extrapolation method of moving average, exponential smoothing and the Bayesian approach) and forecasting expert methods (ranking and direct evaluation);
- multi-agent RL-learning module consist of the group of independent agents each of which is trained on the basis of a developed TD-methods (TD (0), TD (λ), SARSA, Q-learning) as well as used for the accumulation of knowledge of the environment and able to adapt and modify this knowledge. The generalized scheme of multi-agent reinforcement learning module (MARL) is shown on Fig. 2;
- decision-making module designed for the data analysis coming from the prediction module and multi-agent RL-learning module, making decisions on follow-up actions and adjusting management strategies;

Software implementation of a subsystem prototype of forecasting using statistical and expert modules to meet the challenges of the expert diagnosis of complex technological

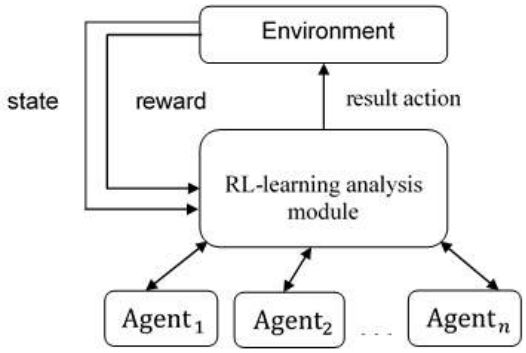

Fig. 2. Structure of multi-agent reinforcement learning module

objects (one of the nuclear power plant subsystem) were made [10,11]. The test results revealed that there is necessity to raise additional methods: RL-learning methods based on temporal differences that reveal the existing regularity by analyzing the history of the process and reduce the influence of random phenomena.

Various algorithms of TD-methods (TD (0), TD (λ), SARSA, Q-learning) have been developed and examined in order to analyze the possibility of their application in an integrated tool in order to comparing the prediction results using different techniques and their combinations [11] and maximize rewards from the environment. The main research problem of the paper is designing multi-agent RL-learning analysis module based on temporal differences methods, its integration into the forecasting subsystem, finding the most preferred methods of RL-learning and prediction in order to inclusion in the RT IDSS and evaluation of the efficiency of multi-agent systems in terms of use in RT IDSS.

VI. Conclusion

In this paper were analyzed various methods of RL-learning and implemented the relevant algorithms in terms of their further integration into the forecasting unit of RT IDSS. Particular attention was paid to the methods based on temporal differences (TD-methods). A combined forecasting method based on statistical and expert methods was proposed and algorithms for the combined method were implemented. The architecture of forecasting subsystem consist of forecasting module (includes statistical and expert sub-modules), the multi-agent RL-learning module (comprises of different agents that communicates with RL-learning analysis sub-module which collect result from the agents, find optimal result action, send overall result to the environment and pass current reward and state to agents from the environment) and the module of analysis and making decisions (which collect data from other modules, analyse received information and form final effect on the environment) were suggested.

Currently, the multi-agent RL-learning module and separate agents, that working with different algorithms based on temporal differences are developing in terms of creating an integrated tool and its following inclusion in an integrated environment that focuses on the use in RT IDSS of semiotic type, in order to expand the scope, improve productivity and efficiency of the modern RT IDSS.

References

[1] R.S. Sutton, A.G.Barto. Reinforcement Learning. – London, The MIT Press, 2012, - 320 p. (Russ. Ed. Moscow: BINOM., 2011).

[2] Vagin V.N., Yeremeyev A.P. Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making // Journal of Computer and Systems Sciences International. - 2001. - N 6. Pp. 953-961.

[3] Vagin V.N., Eremeev A.P. Scientific school of artificial intelligence at MPEI at the department of Applied Mathematics: formation and results // Vestnik MPEI. – 2015. – N 2. – Pp. 29-37 (in Russian).

[4] Rybina G.V., Parondjanov S.S. The technology of building dynamic intelligent systems. – Moscow.: MEPHI, 2011. (in Russian)

[5] Osipov G.S. Methods of artificial intelligence. – 2nd edition. – Moscow.: FIZMATLIT, 2015 (in Russian).

[6] L. Busoniu, R. Babuska, and B. De Schutter, ≪Multi-agent reinforcement learning: An overview≫ Chapter 7 in Innovations in Multi-Agent Systems and Applications – 1 (D. Srinivasan and L.C. Jain, eds.), vol. 310 of Studies in Computational Intelligence, Berlin, Germany: Springer, 2010. - Pp. 183–221.

[7] Varshavsky P.R., Kozhukhov A.A. Development of forecasting subsystem using statistical and expert methods // VII International Scientific and Practical Conference "Integrated models and soft computing in artificial intelligence". – M.: Fizmatlit, 2015. – Pp. 174-180 (in Russian).

[8] Eremeev A.P., Podogov I.U. Generalized method of hierarchical reinforcement learning for intelligent decision support systems // Software and Systems – 2008. - N 2. - Pp. 35-39 (in Russian).

[9] Eremeev A.P., Podogov I.U. Reinforcement learning methods for real-time intelligent decision support systems // Vestnik MPEI. – 2009. - N 2. - Pp. 153-161 (in Russian).

[10] Eremeev A.P., Kozhukhov A.A. Development of an integrated environment based on forecasting and reinforcement learning for intelligent systems of real time// Proceedings of the Congress on Intelligent Systems and Information Technology ≪IS&IT'16≫. Scientific edition in 3 volumes.– Taganrog: YUFU, 2016.. – V. 1. - Pp. 140-149 (in Russian).

[11] Eremeev A.P., Kozhukhov A.A. Analysis and development of reinforcement learning methods based on temporal differences for real time intelligent systems // KII-2016. – Smolensk: Universum, 2016. – Pp. 323-330 (in Russian).

РЕАЛИЗАЦИЯ ИНСТРУМЕНТОВ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ

Еремеев А.П., Кожухов А.А.

В статье описываются алгоритмы методов обучения с подкреплением на основе темпоральных различий. Оцениваются преимущества мультиагентной технологии в рамках применения в интеллектуальных системах реального времени. Рассматривается реализация многоагентного инструмента обучения с подкреплением на основе темпоральных различий. Представлены способы комбинирования методов обучения со статистическими и экспертными методами прогнозирования. А так же рассматриваются возможности их последующей интеграции в подсистему прогнозирования для использования в интеллектуальных системах поддержки принятия решений реального времени.