Neuromorphic Robot Dream: A Spike Based Reasoning System

Alexander Toschev
Laboratory of Neurobiology, ITIS,
Department of Intellectual Robotics,
Kazan Federal University, Kazan, Russia
E-mail: atoschev@kpfu.ru

Max Talanov
Laboratory of Neurobiology, ITIS,
Department of Intellectual Robotics,
Kazan Federal University, Kazan, Russia
E-mail: max.talanov@gmail.com

Alexander Tchitchigin
Service Science and Engineering Laboratory, Innopolis University and
Laboratory of Neurobiology, Kazan Federal University, Kazan, Russia,
E-mail: a.chichigin@innopolis.ru

Salvatore Distefano
Social and Urban Computing Group, ITIS, Kazan Federal University, Kazan, Russia and
MDSLAB, University of Messina, Messina, Italy
E-mail: s_distefano@it.kfu.ru, sdistefano@unime.it

Abstract

In this position paper we propose the approach to create a reasoning system starting from brain spikes as main information medium to control a (quasi) real-time robotic system. Based on the robot dream architecture, the robot input in form of spike stimulus is provided to a simulated spiking neural network, then elaborated and fed back to the robot as specific rules. The reasoning rule-based system for intelligent spike processing transforming signals into software actions or hardware signals is thus specified.

Keywords: artificial intelligence, robotics, spikes, neurobiology, reasoning, logic.

1. Introduction

This work is a part of the Robot Dream project [1] focusing on the integration of real-time robotic system with spiking neural network simulation of a brain. One of the parts of the Robot Dream project is the translation the simulated neuron spikes into robotic system signals and the management of the robotic system using the reasoning system, based on the TU (thinking and understanding) framework described in [2]. The Robot Dream project integrates the simulated brain via spiking neural network with the robot system.

Figure 1. depicts the “wake-dream” cycle of the Robot Dream project architecture. The idea of the “wake-dream” was inspired by mammalian dream and sleep cycles. We split the computations into two phases: “wake” and “dream”. In the “wake” phase the robotic system actively interacts with the complex social environment in a
(quasi) real-time. Thus system reactions over permanently changing environment should be performed via light-weight rule-based system. The “dream” phase is responsible for processing of experience accumulated during the “wake” phase. During the “dream” phase the “dreaming brain” plays back stored experience over the part of the brain corresponding to the input channel (visual information over visual cortex, haptic information over somatosensory cortex, etc.). We call this process the direct translation. During the reverse translation the simulated neuronal structures are consequentially transformed into the form of the logical rules of the robotic system. The aim of this paper is to describe the system of light-weighted logical rules operating in the real-time robotic system which were obtaining during the reverse translation phase.

![Diagram of wake-dream cycle](image)

**Fig. 1.** The representation of “wake-dream” cycle: during “wake” phase the robotic system accumulates experience, during “dream” phase the accumulated experience is transferred from a robotic system to the “dreaming brain”. Later the “dreaming brain” updates its synaptic connections and translates the neuronal structures into the robotic system rules. Robotic system during “wake” phase operates in real-time using light-weighted rules based system.

2. **Robot lifecycle**

The robotic system operates in the (quasi) real-time mode during the “wake” phase. If the (quasi) real-time requirement is strict the robot should use a lightweight management system. The second requirement is that the robotic system must store the input from sensors and actuation control logs in a format transferable to the “dreaming brain”, according to our bio-inspired approach basically as spikes.

Following a biologic analogy, our solution is based on spikes both in the representation of input information coming from the robot and in controlling the latter through spikes-encoded rules. This way all sensed data is stored in form of tagged spike trains. Later it is transmitted and “played back” over the simulated “dreaming brain”. We propose to use spikes as main information unit for the reasoning system instead of textual or numerical forms. Both approaches are described in detail below.

3. **Rule based system**

Due to the (quasi) real-time requirement we propose to use TU framework [2], based on the “Critic-Selector-Way to think” (T3) model [3]. This framework helps us to implement the rule based system managing the robotic system. The T3 “Critic-Selector-Way to think” triplet is inherited from the works of Marvin Minsky [4]. This approach provides an option to evaluate incoming sensory data over the stored knowledge in the knowledge base. The input information of the TU framework [2] is textual while in this work we propose to use spikes for the representation of input and processing information of the TU framework. The TU framework is based on probabilistic rules and uses logical reasoning with the 6-levels of mental activity and T3 over spikes [4].

3.1. **Probabilistic Critic**

Probabilistic critic is the implementation of critic from the T3 “Critic-Selector-Way to think” triplet. Spikes trigger several critics, that start inbound information processing in parallel on several levels of mental activity. Critics are grouped in contexts based on their level of mental activity and semantics of the processing information (audial, visual, tactile). The activation of one critic of the context increases the probability of triggering corresponding critics on the same context. This way every critic is a temporal probabilistic predicate that contains set of rules that are evaluated not only over the

© The 2017 International Conference on Artificial Life and Robotics (ICAROB 2017), Jan. 19-22, Seagaia Convention Center, Miyazaki, Japan
incoming information, but over current system state and context of recently processed information. The sequence of the logical spiking information processing is depicted in Fig. 2. The incoming set of spikes S1 triggers the pattern P1 of critics on two levels: instinctive (basic reflexes) and learned (simple-trained reactions). The pattern P1 and the next set of spikes trigger two patterns P2 on learned level and P3 on the instinctive levels. Later these two patterns trigger two more patterns one on each level: P4 and P5. This mechanism of spiking reasoning system is inspired by the “Hierarchical temporal memory” HTM approach [5] and in a similar way has predictive mechanism: the activation of the pattern P1 could be used as the indication of the S2 set of spikes. To use the advantages of training we propose to extend logical rules with weight value similar to confidence value in NARS [6].

3.2. Way to think
A way to think is a main activity to change the content of memory and perform an action [4]. A way to think runs a workflow that could trigger the hardware controller of the robotic system or change the data of the information processing context. We propose the following workflow:

1. Spike hits the system;
2. The system creates inbound context for this spike based on its attributes: source channel (visual, auditory, ...), time, number of activated neurons, neurotransmitters, previously processed information, see Data Structures section for the details;
3. The system starts processing the spike;
4. Several Critics are activated by based on resulting probability of their rules;
5. Several spikes are accumulated in the system state and when their number reaches the threshold the motor reaction is triggered;
6. The critic activates a way to think;
7. The way to think generates data for controllers.

3.3. Data structures
According to our approach, the input from robots is encoded by spikes. A spike is an abstract object with following attributes:

1. Source channel;
2. Timing: start time, duration.
3. Semantic tag of the event to be processed.
4. Number of activated neurons.

Fig. 2. Inbound set of spikes S1 trigger pattern of P1 critics CL1, C11, C12, C13, C14, C15 on two levels: instinctive reactions and learned reactions. The pattern P1, and the spikes S2 trigger the patterns: P2 (CL2) and P3 (C16, C17, C18) they in turn trigger P4 (CL3) and P5 (C11, C13, C19). This way, when P1 is triggered it could indicate that the next incoming set of spikes could most probably be S2.

5. Neurotransmitters used to generate this spike.
6. Previously processed information context.

4. The implementation of the model of six
We implement the framework based on "Model of Six" developed by Marvin Minsky [4] as an instance of a rule-based (soft) real-time architecture. Firstly we developed first three layers of the model: instinctive, learned and deliberative thinking. All layers process input data in parallel, the fastest one wins. This means that whatever layer selects a next action first, that action gets executed. This way simple instinctive reactions get executed much faster than long deliberative planning. Later long deliberative planning could override the fast behavior. It's useful for immediate handling of "emergency" situations. On the other hand, in the case no instinctive or learned rules match the situation the system can "take some time" to plan sequence of actions leading to a goal using deliberative thinking.

At the bottom of our layered architecture, closest to the hardware, there is the driver layer. This layer allows to use several drivers for different hardware and robotic platform configurations. Next one is hardware abstraction layer (HAL). This layer defines abstract high-level commands (like "move forward 10 cm") hardware or platform agnostic. The driver is responsible for executing these commands on a given robotic system. The third layer implements actual reasoning and control providing the system with the
commands to be enforced. In our case, as stated above, it implements the Marvin Minsky's "Model of Six" [4]. The specific rules managing the system are tailored on the application running on the robotic system and subject to modifications through the feedback from the “dreaming brain”. The initial rules for an application are not generated by the inference engine ("Model of Six"). This architecture provides a relatively straightforward and flexible enough framework to experiment with and adapt to different tasks and robotic systems.

5. Conclusion

In the paper we propose the mechanism of the translation of neuron spikes into control signals for the robotic system. The TU framework is used to process reasoning life-cycle over the spikes and apply proper transformation of the brain neuronal signals (spikes) into the pseudo-neuronal activity of the robotic system. However, the system should be able to track existing signals and reactions from simulated “sleeping” brain mapping them into the robotic systems.

References

1. A. Tchitchigin and M. Talanov, Robot dream (Agent and Multi-Agent Systems: Technology and Applications: 10th KES International Conference, KES-AMSTA 2016 Puerto de la Cruz, Tenerife, Spain, June 2016 Proceedings 2016).
2. A. Toschev and M. Talanov (eds.), Thinking Lifecycle as an implementation of machine understanding in software maintenance automation domain (Agent and Multi-Agent Systems: Technologies and Applications. 9th KES International Conference, KES-AMSTA 2015 Sorrento, Italy, June 2015, Proceedings, 2015).
3. A. Toschev, Thinking lifecycle as an implementation of machine understanding in software maintenance automation domain (Smart Innovation, Systems and Technologies, pp. 301–310, 2015).
4. M. Minsky, The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind, Simon & Schuster, 2007.
5. J. Hawkins, HIERARCHICAL TEMPORAL MEMORY (Numenta, Inc., 2011), pp. 421–439.
6. H. Wang and M. Morales, Pedunculopontine and laterodorsal tegmental nuclei contain distinct populations of cholinergic, glutamatergic and GABAergic neurons in the rat. (The European journal of neuroscience, 2009).
7. A. Kumar and S. Rotter, Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding (Nature reviews neuroscience, 2010).
8. L. Buesing and W. Maass, Simplified rules and theoretical analysis for information bottleneck optimization and PCA with spiking neurons (Advances in Neural Information Processing Systems, 2008).
9. P. Vincent and H. Larochelle, tacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion (Journal of Machine Learning Research, 2010).

© The 2017 International Conference on Artificial Life and Robotics (ICAROB 2017), Jan. 19-22, Seagaia Convention Center, Miyazaki, Japan