Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 145 (2018) 629–634

Procedia Computer Science

www.elsevier.com/locate/procedia

Postproceedings of the 9th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2018 (Ninth Annual Meeting of the BICA Society)

Adaptive control of multiped robot

Alexander Demin, Evgeniy Vityaev*

Sobolev Institute of Mathematics, pr. Koptyuga 4, Novosibirsk 630090, Russia

Abstract

In the paper, a logical-probabilistic method of adaptive control of modular systems is presented. It base on the functional similarity of modules, the logical-probabilistic algorithm of directed search of rules and joint training of control modules. Starting with the search for common rules for all modules of the control system then it upload them with a more specific ones in accordance with the ideas of semantical-probabilistic inference. With the use of an interactive 3D-simulator, successful experiments were conducted with virtual multiped robot mode. Experimental studies have shown that the proposed approach is quite effective and can be used to manage modular systems with a large number of degrees of freedom.

Keywords: control system; learning control system rules; knowledge discovery; modular robots

1. Introduction

The task of developing control systems for modular robots faces serious difficulties resulting from the hyper-redundancy of such systems. The presence of a large number of degrees of freedom makes it impossible to apply traditional approaches to the creation of control systems by directly setting the sensor-motor functions by the developer. Therefore, the development of methods of automatic generation of a control system based on various training models becomes urgent.

* Corresponding author. Tel.:+7-913-945-4867.
E-mail address: vityaev@math.nsc.ru

1877-0509
10.1016/j.procs.2018.11.071

In the world practice in the field of adaptive control of hyper-redundant and modular systems, solutions based on the use of population methods (evolutionary methods, methods of particle swarm, etc.) are most often proposed in integration with other known methods of machine learning (Reinforcement Learning, neural networks, etc.) [1, 2, 8-10, 13, 14]. However, evolutionary methods have serious limitations associated with the need for a population of robots, which does not allow for training and adaptation in real-time mode [11]. A common drawback of such solutions is the impossibility of training in the regime of real work and weak scalability relative to the increase in the complexity of the system (the number of degrees of freedom). On the whole, it should be noted that at present there is no sufficiently universal solution of the problem of adaptive control of hyper-redundant systems development.

This work proposes a new approach to creating the learning control systems for modular robots, based on the use of the modules functional similarity and the logical-and-probabilistic algorithm of the guided search of rules. The main advantage of the proposed approach is the high learning rate and the teach-and-learn capability in real-time mode based only on the experience of the system's interaction with the environment.

In previous works [3-5, 7], examples of application of the proposed approach for teaching typical representatives of the simplest hyper-redundant modular robots were described: a snake-like robot, a multi-legged robot and a proboscide manipulator. The obtained results demonstrated the main advantages of the approach: training and adaptation in the real work, high training speed and good scalability in relation to the complexity of the system. However, the examples of robot models considered there were limited to identical modules. In this paper, the applicability of the proposed approach considered for different types of modules. To this end, an experiment was set up to teach a virtual model of a simple multi-legged robot with limbs of two different types. Using the example of this model, the applicability of the approach and the effectiveness of the training were assessed.

2. Simulator

For conducting the experiments with the proposed control model an interactive 3D-simulator with graphical user interface was developed. The program has virtual environment visualization capabilities as well as the capability of recording experiments in a video file. As a physics engine, the simulator uses the Open Dynamic Library (ODE) [12], which allows you to simulate the dynamics of solid bodies with different types of joints.

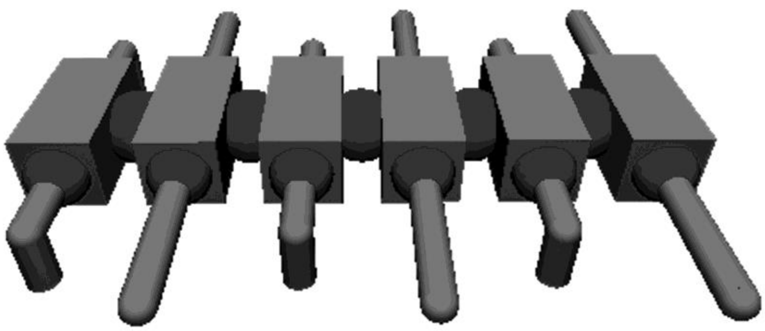

Fig. 1. Multiped robot with two type of limbs.

The main objective of the experiment was to test the capabilities of the proposed model to successfully detect effective control rules for different types of modules. To this end, the simulator model was created a multi-robot, consisting of two repeating types of modules (Fig. 1). Even modules have a pair of L-shaped limbs on the right and left sides, capable of moving only in the horizontal plane. Odd modules have a pair of straight limbs, capable of moving only in the vertical plane. Modules are alternately connected to each other by means of rigid joints. In total, the robot has six modules: three modules with L-shaped limbs and three with straight ones.

The task of the control system was to train an effective way forward for the given robot model. It is obvious that the robot can effectively move forward only by moving the L-shaped limbs back. However, since the L-shaped limb can move only in the horizontal plane, in order to throw it forward for the next step and not to move the robot in the opposite direction, it is necessary to use straight limbs to lift the robot above the ground. As a result, effective movement of the robot is possible only with the coordinated work of modules of different types. Thus, the selected robot design, despite its simplicity, is a good test model for testing the capabilities of the system to detect consistent control rules for different types of modules.

3. Control system

To create the modular robots control system, it is proposed to use neural networks consisting of trainable logic neurons, each of which controls a separate robot module.

Logical neurons operate in a discrete time $t = 0,1,2,...$. Each neuron contains a set of inputs assuming $input_1,...,input_k$ valid values and one output $output$ assuming the values from a predefined set $\{y_1,...,y_m\}$. At any time point t , the neuron inputs are supplied with information by assigning real values of the inputs $input_1 = x_1,...,input_k = x_k$, $x_1,...,x_k \in \mathbb{R}$. The result of the neuron's work is the output signal $output = y$, $y \in \{y_1,...,y_m\}$ assuming one of the possible values $\{y_1,...,y_m\}$.

After all the neurons of the network have completed the work, the reward comes from the external environment. The award function is set depending on the ultimate goal and is used to evaluate the control quality. Control system task is detecting such patterns of neuron functioning that would ensure getting the maximal reward.

Variety of the patterns that define the work of neurons are suggested to search in the form of logical patterns with estimates that are as follows:

$$\forall i\ (P(i), X_1(i),..., X_m(i), Y(i) \to r)\,, \tag{1}$$

where $i = 1,..,n$ – the variable on the neurons.

$X_j(i) \in \mathbb{X}$ – predicates from the specified set of input predicates $\mathbb{X}$ that describe the inputs j of the neuron N_i $(i = 1,..,n)$. For example, in the simplest case, these predicates can be preset as $X_j(i) = (input_k(i) = x_r)$, where x_r – some constants from the range of input signal values that can be preset, for example, by quantization the range of possible values of the corresponding inputs of neurons.

$Y_j(i) \in \mathbb{Y}$ – predicates from a given set of output predicates $\mathbb{Y}$ describing the output of neurons N_i $(i = 1,..,n)$ and looking as $Y_j(i) = (output(i) = y_r)$ where y_r – some constants from the range of output signals.

$P(i) \in \mathbb{P}$ – predicates from a set of predicates $\mathbb{P}$ look as $(i = j)$, where $j = 1,..,n$ the intent of which is to narrow the scope of type rules application (1) to specific neurons.

r – reward whose maximization is the task of a neuron.

These patterns predict that if neuron gets input signals N_i , $i = 1,..,n$ that satisfy the input predicates $X_1(i),..., X_m(i)$ of the rule premises, and the neuron sends output signal specified in the output predicate $Y(i)$, the reward mathematical expectation will be equal to r.

Additionally, we will note that if a neuron N_j has an input specific only to that neuron, it is assumed that the predicate $X(i)$ describing this input will take the value "0" for all $i \neq j$, i.e. for all other neurons. Similarly, if the output of any neuron N_j can take any value y specific only to that neuron, the corresponding output predicate $(output(i) = y)$ will also take the value of "0" for all $i \neq j$.

We will now explain the need for introducing a set of predicates $\mathbb{P}$. Should a rule (1) does not contain predicates from $\mathbb{P}$, it will look like $\forall i (X_1(i),..., X_m(i), Y(i) \to r)$ and will describe patterns common to all neurons N_i , $i = 1,..,n$. Adding a predicate to premises of the rule from $\mathbb{P}$ automatically narrows the scope of the rule application to a specific neuron. Thus, the rules containing predicates from $\mathbb{P}$, describe patterns specific to particular neurons. It should also be noted that the narrowing of the rule scope applicability (1) can takes place not only because of predicates from $\mathbb{P}$, but also because of input or output predicates from $\mathbb{X}$ and $\mathbb{Y}$ describing specific inputs or outputs of particular neurons.

In order to find the patterns of the type (1), it is proposed to use an algorithm based on the semantic probabilistic inference described in the papers [6, 15]. This algorithm helps to analyze the data set that stores the statistics of the neural network operation (input-output of neuron and the reward received) and all the statistically significant patterns of the type (1) are extracted.

4. Locomotion control system of multiped robot

The scheme of the neural circuit chosen to control the robot consisted of six neurons – one neuron per robot module (Fig. 2). Each neuron N_i, $i = 1,...,6$ controlled the movements of the left and right limbs of its module, feeding the activating signals to the corresponding angular motors, rotating the limbs in the joint. To simplify the task, the movements of the left and right limbs of each module were synchronized in such a way that one limb mirrored the movements of the other.

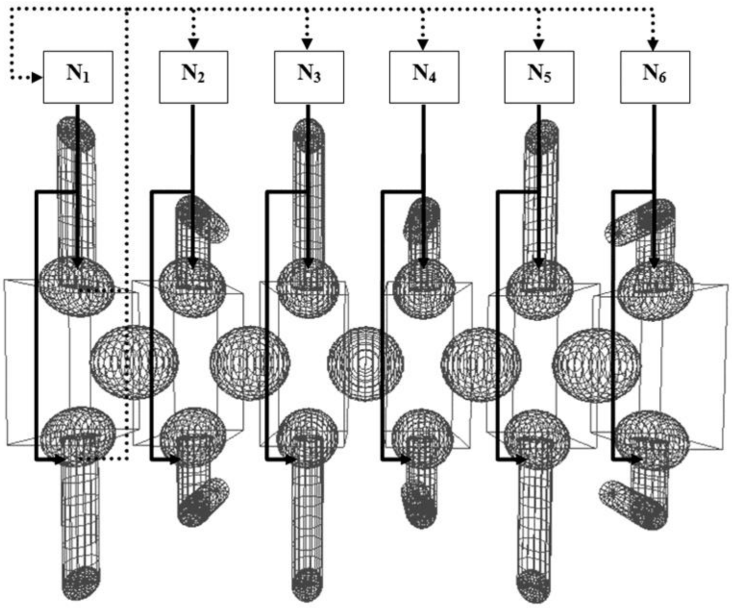

Fig. 2. Diagram of the neural control circuit of the multiped robot.

The first neuron N_1 receives information about the position of the extremities of the first module at its input. The same information goes to the inputs of all other neurons, N_i, $i = 2,...,6$. Thus, in this scheme, the state of the first module, in fact, can be considered as a kind of cycles counter for all other modules.

The reward for the control system was calculated after the completion of the step execution cycle and the return of the limbs of the first module to the starting point. By step is meant the whole sequence of actions that was performed by limbs in the time interval between the current and previous states. As the starting point, the maximum vertical position of the extremities of the first module was chosen.

The award is calculated as follows. Suppose, at the current time t_1, the position of the limbs of the first module corresponds to the initial point of the start for the current step. Let t_0 – the previous point in time, when these limbs were at the starting point. Then all actions in the time interval from $(t_0 + 1)$ to t_1 is the cycle of the execution of the step, and the reward for these moments of time t, where $(t_0 + 1) \le t \le t_1$ and $(t_0 + 1) < t_1$, will be equal to $r = S / (t_1 - (t_0 + 1))$. Where S – is the distance that the robot will overcome in the forward direction in the same time interval (from $(t_0 + 1)$ to t_1). In the case of an "empty" step $t_1 = (t_0 + 1)$, i.e. when the finitenesses of the first module simply stay at the starting point for two consecutive strokes, the reward for the time t_1 is set to 0. This reward function stimulates the control system to find sequences of actions that would overpass as much distance as possible to accomplish as few actions as possible.

Using the 3D-simulator, a number of successful experiments were conducted to teach the considered control system for the modes of movement. In a series of experiments, the control system managed to steadily learn control rules that ensure coordinated movements of the limbs of different types of modules, leading to an effective movement of the robot forward. Fig. 3 gives an example of the optimal sequence of movements found in the learning process.

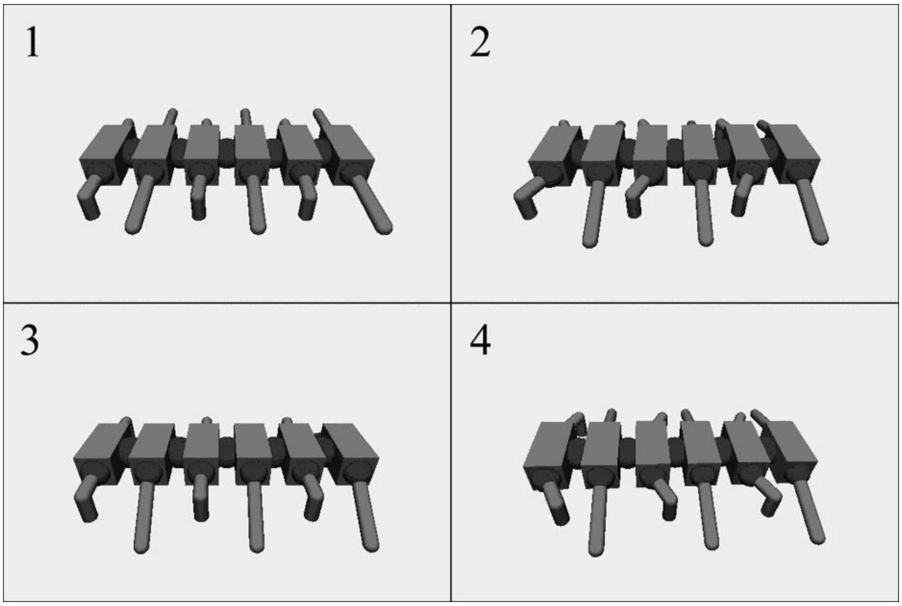

Fig. 3. Sequence of multiped robot movements when moving forward.

5. Conclusion

This paper proposes an adaptive control of modular systems with a large number of degrees of freedom based on joint learning of control modules, starting with finding the common control rules for all modules and finishing with their subsequent specification in accordance with the ideas of the semantic probabilistic inference. The main advantages of the proposed approach are: teach-and-learn capability in real time and the high learning rate which is achieved through the effective use of the module's functional commonality properties and the algorithm of the rule-oriented search.

The results of the experiment have confirmed that the proposed approach can be successfully used to control modular robots, consisting of different types of modules.

Acknowledgements

This work is financially supported by the Russian Science Foundation grant # 17-11-01176.

References

[1] Bongard J.C. (2013) "Evolutionary Robotics." *Communications of the ACM* **56** (**8**): 74-83.

[2] Daoxiong Gong, Jie Yan, Guoyu Zuo. (2010) "A Review of Gait Optimization Based on Evolutionary Computation." *Applied Computational Intelligence and Soft Computing* **2010**: 12.

[3] Demin A.V. (2012) "The model of the adaptive control system and its application for a virtual robot locomotion control." *Young scientist* **11** (**46**): 114-119.

[4] Demin A.V. (2013) "Learning control model of chemotaxis for C.Elegans nematode." *Neiroinformatics* **7** (**1**): 29-41.

[5] Demin A.V. (2014) “Teaching of locomotion ways of the snake-like robot virtual model.” *Young scientist* **19** (**78**): 147-150.

[6] Demin, A. V., Vityaev, E. E. (2008) “Logical model of the adaptive control system.” *Neuroinformatics* **3** (**1**): 79-107.

[7] Demin, A. V., Vityaev, E. E. (2014) “Learning in a virtual model of the C. elegans nematode for locomotion and chemotaxis.” *Biologically Inspired Cognitive Architectures* **7**: 9-14.

[8] Ito K., Matsuno F. (2002) “Control of hyper-redundant robot using QDSEGA.” *Proceedings of the 41st SICE Annual Conference* **3**: 1499-1504.

[9] Kamimura A., Kurokawa H., Yoshida E., Tomita K., Murata S., Kokaji S. (2003) “Automatic locomotion pattern generation for modular robots.” *Proceedings of 2003 IEEE International Conference on Robotics and Automation*: 714-720.

[10] Marbach D., Ijspeert A.J. (2004) “Co-evolution of configuration and control for homogenous modular robots.” *Proceedings of the eighth conference on Intelligent Autonomous Systems (IAS8)*: 712-719.

[11] Mataric M., Cliff D. (1996) “The challenge in evolving controllers for physical robot.” *Robotics and autonomous systems* **19** (**1**): 67-83.

[12] Smith R. (n.d.). “Open Dynamics engine.” Retrieved from http://ode.org

[13] Tanev I., Ray T., Buller A. (2005) “Automated evolutionary design, robustness and adaptation of sidewinding locomotion for simulated snake-like robot.” *IEEE Transactions on robotics* **21** (**4**): 632-645.

[14] Valsalam V. K., Miikkulainen R. (2008) “Modular neuroevolution for multilegged locomotion.” *Proceedings of GECCO*: 265-272.

[15] Vityaev E. (2006) “The logic of prediction.” *Proceedings of the 9th Asian Logic Conference*: 263-276.