2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Evolving a Sensory-Motor Interconnection for Dynamic Quadruped Robot Locomotion Behavior

Azhar Aulia Saputra, Wei Hong Chin, János Botzheim, and Naoyuki Kubota

Abstract— **In this paper, we present a novel biologically inspired evolving neural oscillator for quadruped robot locomotion to minimize constraints during the locomotion process. The proposed sensory-motor coordination model is formed by the interconnection between motor and sensory neurons. The model utilizes Bacterial Programming to reconstruct the number of joints and neurons in each joint based on environmental conditions. Bacterial Programming is inspired by the evolutionary process of bacteria that includes bacterial mutation and gene transfer process. In this system, either the number of joints, the number of neurons, or the interconnection structure are changing dynamically depending on the sensory information from sensors equipped on the robot. The proposed model is simulated in computer for realizing the optimization process and the optimized structure is then applied to a real quadruped robot for locomotion process. The optimizing process is based on tree structure optimization to simplify the sensory-motor interconnection structure. The proposed model was validated by series of real robot experiments in different environmental conditions.**

I. INTRODUCTION

In recent years, biologically inspired approach in locomotion has been widely researched for realizing the dynamical system in robot for moving in an unpredictable area or other environmental conditions such as rocky surface, boggy soil, and slippery ground. Quadruped robot can be applied as a multi-purposed robot in these environmental conditions for performing rescue mission, space exploration and environment mapping. Conventional locomotion systems have been applied in sloppy and slippery environments. However, there are some drawbacks in biologically inspired approach that impede its performance such as environmental constraint, neuron constraint, and joint constraint.

Smith et al. [1] proposed a biologically inspired approach for recognizing the appropriate number of legs of a multi-legged robot based on different environmental conditions. In [2], a muscle-based skeletal model has been applied for controlling the biped robot locomotion. Conventional research works applied two neurons for representing one joint that generated mutual inhibition between certain neurons termed as central pattern generator (CPG). Nassour et al. [3] proposed a neuro-locomotion model based on multi-layered neuron structure for robot locomotion which achieved an adequate and stable walking pattern.

In recent muscle-based locomotion research, most of them are tested and verified in simulation experiments [4], [2], [5]. Lee et al. [4] realized the muscle-based locomotion with more than one hundred Hill-type muscles for humanoid models actuated. However, the muscle model has limitation for real robot implementation. The main drawback of muscle model and adaptive locomotion is the optimization process. Some researchers applied genetic algorithm for evolving the configuration of the coupled neural oscillator and optimizing the locomotion gaits [6]. However, the proposed method heavily relies on parameters value settings, a small change of its parameter value can cause a big impact on the generated locomotion pattern, which may fail to achieve the sensory-motor coordination.

Therefore, in this paper a joint angle based model is proposed as the targeted actuator generated by motor neurons. Locomotion models based on a biological approach were proposed by several researchers [7], [8], [9], [3], [10]. This approach is inspired by the spinal cord system in the vertebrates. Nassour et al. used CPG in for biped robot locomotion. He proposed multi-layered neuron structure for improving the performance of stability [3].

In our previous work [11], [10], [12], we have developed a fix neural structure and utilized multi-objective evolutionary algorithm for the synaptic weight optimization for biped robot locomotion. However, our previous works have constraints for real robot implementation. One of the constraint is that we have to conduct preliminary tests beforehand for designing the appropriate neural structure. In [11], the interconnection structure can be optimized, but the number of neurons were fixed and unable to be optimized.

In this paper, we propose an evolving neural oscillator in robot locomotion that can optimize not only the interconnection structure of the neurons but also reconstruct the number of joints and neurons in each joint based on environmental conditions. The contribution of this work is that either the number of joints, or the number of neurons, or the interconnection structure are dynamically changed depending on the condition acquired from the sensor equipped on the robot. In addition, we propose a joint angle-based model as the targeted actuator generated by motor neurons and represent the solution of the neural interconnection structure by tree structure for simplifying the optimization strategy. The proposed method comprises Bacterial Programming (BP) [13] for optimizing the unpredictable neural structure. The optimization approach was inspired by the evolutionary

Azhar Aulia Saputra and Wei Hong Chin are with Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo, 191-0065, Japan (e-mail: azhar-aulia-saputra@ed.tmu.ac.jp; azhar.aulia.s@gmail.com; weihong@ed.tmu.ac.jp).

János Botzheim is with Budapest University of Technology and Economics, 4-6 Bertalan Lajos Street, Budapest, 1111, Hungary (email: dr.janos.botzheim@ieee.org).

Naoyuki Kubota is with Tokyo Metropolitan University, 6-6 Asahigaoka, Hino, Tokyo, 191-0065, Japan (Phone and Fax: +81-42-585-8441; e-mail: kubota@tmu.ac.jp).

978-1-5386-8094-0/18/$31.00 ©2018 IEEE

process of bacteria and has been implemented in different environmental conditions.

The paper is organized as follows. In Section II, we discuss the neural model that represents the sensory-motor coordination in quadruped robot locomotion. Section III presents the tree structured optimization strategy using bacterial programming. Simulation and real robot experiments are presented in Section IV. Finally, Section V concludes the paper.

II. Sensory-Motor Interconnection Model

We utilize the central pattern generator as locomotion generator in which the motor neurons serve as the output signal and sensory neurons serve as the feedback signal. The internal sensory information collected from tilt sensor (pitch and roll) and ground touch sensor in all legs are used as feedback signals. The number of activated joints in the robot's legs is depending on the environmental condition. Unused joints will be deactivated and therefore decreasing the computational cost and energy consumption.

The proposed locomotion model comprises 3 main elements: gait generator system, optimization system, and adaptive intelligent control system. The role of the gait generator is to generate the walking pattern signal in joint angle level based on neural oscillator. Next, the optimization system optimizes the best neuron structure of gait generator in different environmental conditions. The adaptive intelligent control system manipulates the neural structure according to the change of environmental conditions.

The proposed neural model of joints is illustrated in Fig. 1, its mathematical model is shown in Eqs. (1), (2), (3) and mathematical notation definition as tabulated in Table I.

$$\tau \dot{u}_i = \left(u_0 - u_i - \sum_{j=1}^{n} w_{ij} y_j J_j^k + \sum_{l=1}^{n} m_{il} s_l - b v_i \right) \tau_f \quad (1)$$

$$\tau' \dot{v}_i = (y_i - v_i)\tau_f \quad (2)$$

$$y_i = \max(u_i, 0) \quad (3)$$

$$\Theta_i^{(l,r)} = \sum_{j=1}^{N_i^{neuron}} J_j^{l,r} y_j \quad (4)$$

TABLE I

The Proposed Method's Mathematical Notations

Notation	Definition
w_{ij}	motor neurons interconnection
m_{il}	sensory motor neurons interconnection
u_i	inner state
y_i	output value
v_i	variable of self-inhibition effect of neurons
J_j^k	operator representing +1, -1, or 0
s_l	output of the lth sensory neuron
b	rate of the adaptation value
u_0	external input for coupled neurons
τ and τ'	time constant of the inner state and the adaptation effect

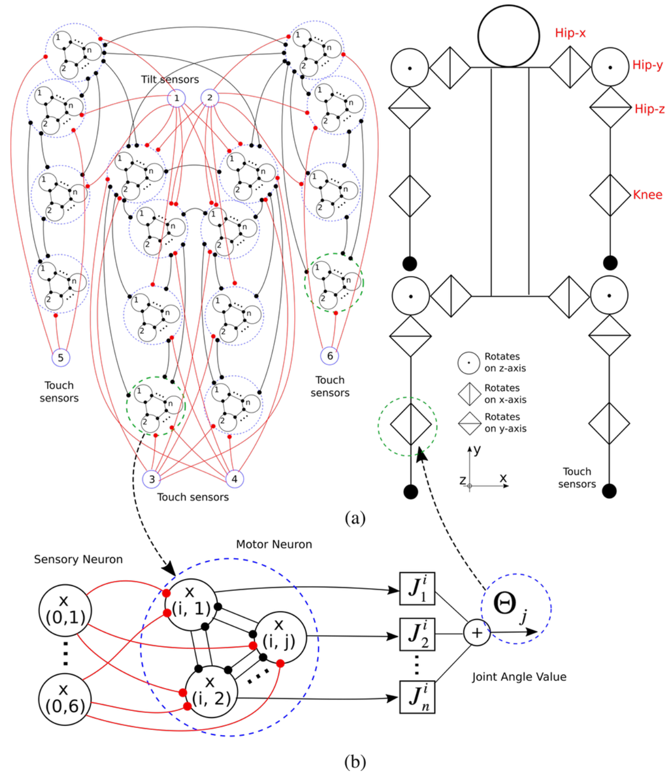


Fig. 1. Representation of the evolving neural model. The black line represents the motor–motor interconnection and the red line represents the sensory–motor interconnection. (a) Neural structure and mechanical structure of the robot. (b) Neuron structure in joint angle level.

In Eq. (1), w_{ij} represents the strength of the inhibitory effect between the motor neurons that is optimized offline; m_{il} represents the strength of the sensory signal effect to the motor neurons; $\sum_{j=1}^{n} w_{ij} y_j$ represents the total of the signal input from the motor neuron; $\sum_{l=1}^{n} m_{il} s_l$ represents the total of the feedback signal from the sensory neurons. In Eqs. (1) and (2), τ_f is used for controlling the frequency of oscillation. In Eq. (4), J_j^i is the neuron effect of the jth neuron in the ith joint represented by joint node in the tree structure of bacterial programming (see Section III) and N_i^{neuron} is the number of neurons in the ith joint. Each joint angle is represented by different number of neurons depending on the environmental conditions.

Regarding the walking pattern, knee (θ_2^d) and hip joints are the main joints (θ_0^d & θ_1^d) for generating the walking pattern. Ankle joints (θ_3^d & θ_4^d) adapt the condition of the main joint and support the stabilization and landing system in locomotion. Ankle joints are required for generating the walking pattern in different conditions. Eqs. (5), (6), (7) and (8) represent the angle value in every joint; where θ_k^d represents the signal to the robot's joint at k joint ID on d side, (l) and (r) represent left and right side; Θ_k represents the signal output from the neuron in joint angle level.

$$\theta_0 = \begin{cases} \theta_1^{(l)} - \theta_1^{(r)} & \text{if } N^{joint} < 3 \\ \Theta_3 & \text{otherwise} \end{cases} \quad (5)$$

$$\theta_1^{(l,r)} = \Theta_0 \tag{6}$$

$$\theta_2^{(l,r)} = \Theta_1 \tag{7}$$

$$\theta_3 = \begin{cases} \theta_2^{(l,r)} - \theta_1^{(l,r)} & \text{if } N^{joint} < 4 \\ \Theta_4 & \text{otherwise} \end{cases} \tag{8}$$

When the environmental condition requires the robot needs only two joints in each leg ($N^{joint} = 2$), the system prefers to choose the hip joint in x-axis position and knee joint to be activated. When $N^{joint} = 3$, the system choose the hip joint x-axis, y-axis position, and knee joint to be activated. When $N^{joint} = 4$, the system activates ankle joint as the additional joint. This condition represents the high complexity of the environment when the robot needs to activate the ankle joint. The neural representation of the robot structure and the neural model in joint angle level are depicted in Fig. 1.

The proposed system is applied in a quadruped robot which has 4 legs and 4 joints in each leg for real world implementation. However, only the important joints are generated by motor neurons.

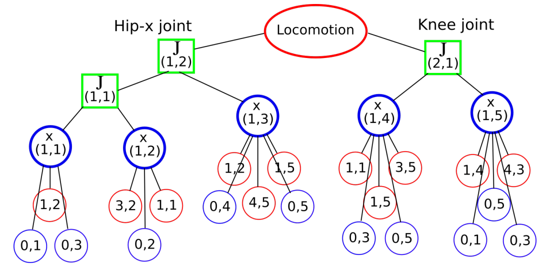
III. Bacterial Programming

Bacterial Programming [13] is an evolutionary computation technique which is based on the combination of Bacterial Evolutionary Algorithm [14] and Genetic Programming [15]. The algorithm is based on the bacterial operations, however, it uses tree structures similar to the ones in the GP. The novelty of our proposed model is inspired by the evolutionary process of bacteria applied for robot locomotion. Bacteria can transfer genes to other bacteria. The bacterial mutation performs local optimization whilst the gene transfer allows the bacteria to directly transfer information to another individuals in the population. Based on these bacterial operations, we optimize the neural structure of locomotion by using the tree structures in certain ground condition and certain slope terrain. In BP, the population is composed of several individuals that are represented by a tree model. The evolutionary process of BP is explained in [13].

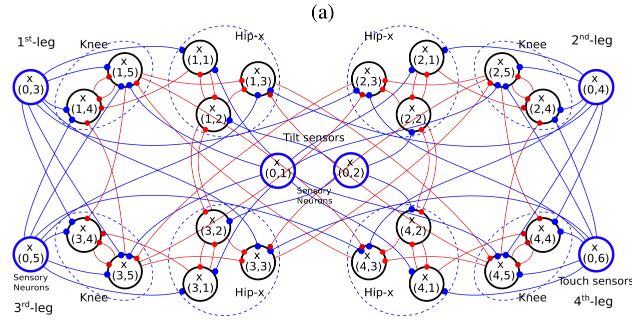
A. Encoding Model

In order to simplify the complexity of the neural oscillator model for robot locomotion, one bacterium is represented by a tree structure, composed of joint and neuron nodes as shown in Fig. 2a. A joint node can be composed of operators "+", "-", or "0" which represents the addition operation, reduction operation, or no operation respectively. These operators are randomly generated and will affect the motor neuron signal that is calculated in Eq. (1). A neuron node is composed of the synaptic weight values from other neurons.

This tree structure, as well as the characteristics of the nodes, evolves from generation to generation. The tree model structure is converted to neural oscillator structure which refers to a symmetric structure as shown in Fig. 2 in order to reduce the size of the search space. The neural oscillator



(a)



(b)

Fig. 2. a) The tree structure represents the neuro-based locomotion; b) Symmetric model of the neuron structure as the result of the tree structure conversion.

structure of a leg is mirrored to another leg. Therefore, in the robot, the right leg mirrors to the left leg of the robot, and the front leg mirrors to the backward leg of the robot.

We obtain a tree structure (i.e. a bacterium) from the evolutionary process, as shown in Fig. 2a, where the red circle (k,l) represents the synaptic weight value from the lth neuron of the kth leg to the neuron (m,n), the blue bold circle with the nth neuron of the mth leg. The parameters will be generated between p^{min} and p^{max}. This information will be encoded to the synaptic weight parameter w_{ij}, where $i = (k-1) \cdot N_k^{neuron} + l$ and $j = (m-1) \cdot N_m^{neuron} + n$, where N_k^{neuron} and N_m^{neuron} represent the number of neurons in the kth leg and mth leg, respectively. The blue bold circle represents the inner state of neuron calculated in Eqs. (1), (2), (3). The green square represents the signal effect in joint angle level. The red and blue connections in Fig. 2 represent motor neurons interconnection and sensory-motor interconnection, respectively. In the optimization, number of joints will be generated from N_{min}^{joint} until N_{max}^{joint}, while number of neurons from N_{min}^{neuron} until N_{max}^{neuron}.

The tree structure in Fig. 2a has 2 joints which are hip-x joint and knee joint and it is converted to neural structure that can be seen in Fig. 2b.

The BP process starts with the initial population generation in which the algorithm randomly generates N_{ind} individuals (i.e. bacteria) one by one. After that the two main operators of the algorithm are repeated in N_{gen} generations, the bacterial mutation and the gene transfer operations.

B. Bacterial Mutation

The bacterial mutation is applied to each bacterium one by one. First, N_{clones} copies (clones) of the bacterium are generated. A certain node of the bacterium is then randomly

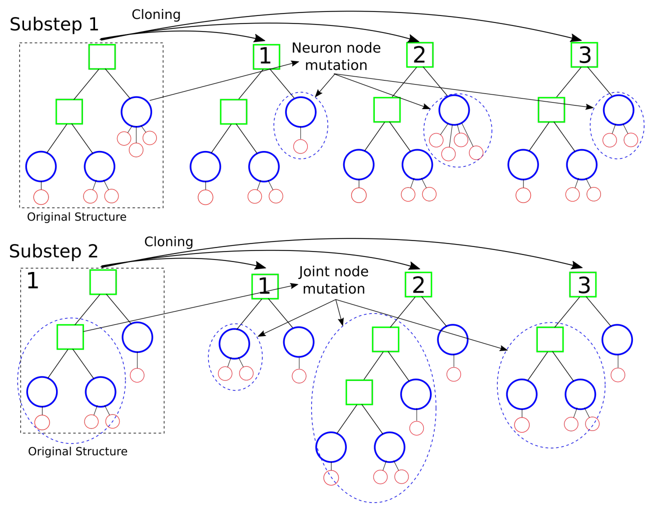


Fig. 3. Illustration of bacterial mutation

selected and the subtree is defined by this node that is randomly changing in each clone (mutation). In our proposed method, because coding is given by an expression tree for neural oscillator based robot locomotion, there are two types of mutation nodes: joint node mutation and neuron node mutation. Then, all the clones and the original bacterium are evaluated by a performance criterion (see Section III-D). The best individual among the clones and the original bacterium transfers the mutated part into the other individuals. This cycle is repeated for the remaining nodes until all of the subtrees of the bacterium have been mutated and evaluated. At the end, the best bacterium is kept and the remaining N_{clones} are discharged. The bacterial mutation operation is illustrated in Fig. 3 and it shows the example of two substeps of the bacterial mutation process, where three clones are applied.

C. Gene Transfer

The gene transfer operation represents the exchange of genetic information between two bacteria. In this process, the population of bacteria is ordered according to the performance criterion (see Section III-D). Then, a source bacterium is randomly selected from the superior half of the population and a destination bacterium is randomly selected from the inferior half. The source bacterium transfers one of its subtree to the destination bacterium which overwrites one of its subtree by the transferred subtree. The above process (ordering the population, selecting the source and destination bacteria, and transferring the subtree) is repeated N_{inf} times. The gene transfer operation is illustrated in Fig. 4.

D. Evaluation

Two evaluation criteria are calculated in the evaluation process which are the desired walking length and the body tilt oscillation in pitch and roll direction $\bar{\dot{\beta}}$. The desired walking length, $\bar{\upsilon}$ is represented by the remaining distance to the target.

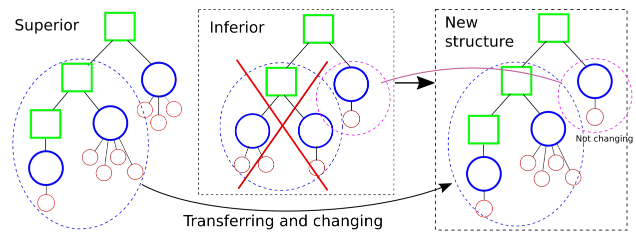


Fig. 4. Illustration of gene transfer

The value of tilt body oscillation represents the stability of movement. If the robot locomotion has low oscillation, it implies good stabilization. The remaining distance represents the speed of the robot walking. If the robot has a high value in the remaining distance, the robot has a low speed in walking. Another motivation of the proposed method is to realize a locomotion pattern with maximum possible speed. Therefore, our objective is to acquire locomotion that has a good stabilization and a high walking speed.

In Eq. (9), $\dot{\beta}_{pitch}$ and $\dot{\beta}_{roll}$ are tilt oscillation in pitch and roll direction that has absolute value. In Eq. (10), $\ell(t, w_{ij})$ is the resultant value of $\mathrm{x}(t)$ and $\mathrm{y}(t)$ in each time sampling. The $\ell(t, w_{ij})$, $\mathrm{x}(t)$, $\mathrm{y}(t)$ notations were defined as real numbers. Parameter w_{ij} is synaptic weight which has been explained in Section II. Parameters α_1 and α_2 represent the weight factor of fitness. The goal of the optimization problem is to minimize the fitness described in Eq. (11).

$$\bar{\dot{\beta}} = \frac{1}{T} \sum_{t=0}^{T} \left(\beta_{pitch}(t) + \beta_{roll}(t) \right) \tag{9}$$

$$\bar{\upsilon}_{(w_{ij})} = \Upsilon - \frac{1}{T} \sum_{t=0}^{T} \frac{\delta}{\delta t} \ell \left(t, w_{ij}, \mathrm{x}(t), \mathrm{y}(t) \right) \tag{10}$$

$$f = \bar{\dot{\beta}} \alpha_1 + \bar{\upsilon} \alpha_2 \tag{11}$$

This evaluation is conducted in computer simulation by using the Open Dynamics Engine (ODE) [16]. The evaluation time requires 10 seconds (1000 time samplings), since a time sampling requires 0.01 second. Thus, the timing process in the evaluation is the same as the real timing that is applied in the real robot.

IV. Experimental Results

We validated the proposed method through several computer simulation and real robot implementation. Two experiments were conducted to validate the locomotion optimization and its application in the middle size quadruped robot.

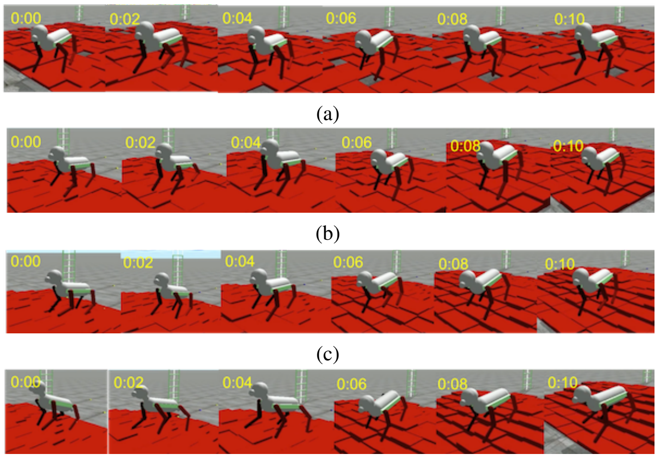
A. Locomotion Optimization

The experiment was conducted on a rough terrain with different slope degrees by using the proposed neural evolving algorithm, whose parameter values have been tabulated in Table II. We then evaluate the walking performance based on the two aspects which are the speed and the stability.

For simulation, we first set the friction to a certain value in Open Dynamics Engine [16]. The sample walking simulation can be seen in Fig. 5. The proposed algorithm succeeded

TABLE II
BACTERIAL PROGRAMMING PARAMETERS

Parameter	Value
N_{ind}, N_{gen}, N_{clones}, N_{inf}	100, 50, 10, 30
p^{min}; p^{max}	0; 4
N^{joint}_{min}; N^{joint}_{max}	1; 3
N^{neuron}_{min}; N^{neuron}_{max}	1; 4
α_1; α_2	0.75; 0.25



(a)

(b)

(c)

(d)

Fig. 5. Simulation of proposed locomotion on uneven terrain in different degree of slope (a) slope 0° (b) slope 5° (c) slope 14° (d) slope 20°

to form the neuron structure for locomotion on the uneven terrain.

The evolution of joints and neuron numbers in each joint are shown in Fig. 6. Those numbers are adaptively changed in some generations. In flat terrain experiment, after several generations, 3 joints in each leg were not stable enough for locomotion. Then, the number of joints was decreased and became 2 joints in each leg. Thus, in the final structure, there are 2 neurons in the first joint (hip-x joint) and 2 neuron in the second joint (knee joint). In this experiment, the locomotion model required 16 neurons for 4 legs, where their signal in joint angle level can be seen in Fig. 7.

The results showed the important joints in the robot locomotion. Like animal, these joints are required to produce rhythmic signal for walking; at least 2 neurons are required for representing extensor and flexor muscle. The hip-x joint is important for swinging the leg and controlling the walking phase.

In the rough terrain experiment, we can see the activity of sensory-motor coordination as shown in Fig. 9. The ground touch sensor input gave some effects to the motor neurons, therefore the signal is changing adaptively as shown in Fig. 9. The figure shows the signal between left part and right part of the robot legs.

The optimized tree structure and its neuron interconnection structure can be seen in Fig. 8. In the rough terrain, 2 joints in every leg was not stable enough. Therefore, after a few generations, the number of joints became 3 joints in every leg. In this experiments, there are 16 motor neurons

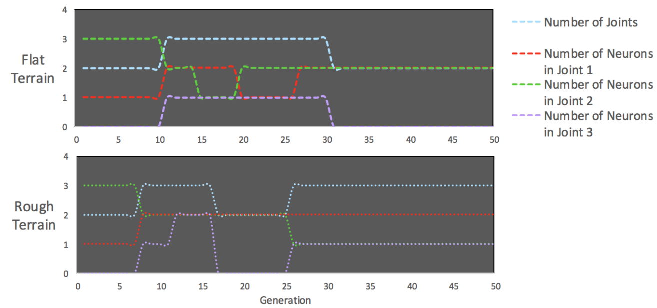


Fig. 6. The number of joints and the number of neurons in each joint in every generation

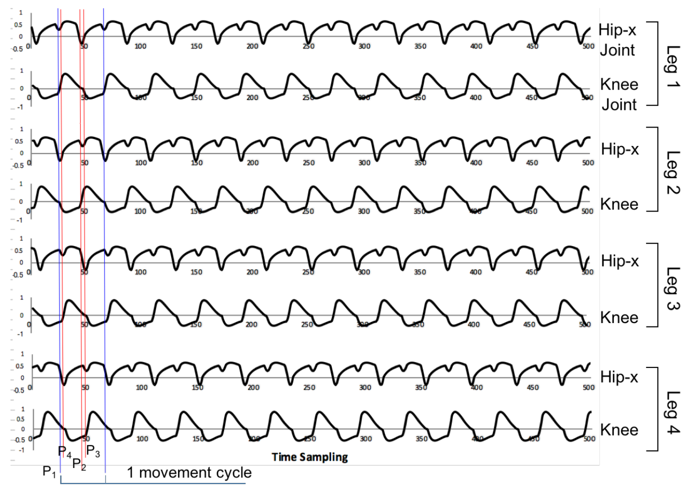


Fig. 7. Sample of signal output in joint angle level ($\Theta_i^{(l,r)} = \sum_{j=1}^{N_i^{neuron}} J_j^{l,r} y_j$) on the flat terrain. The signal movement pattern tends to the walk-like movement. Signal in first (P_1) and third (P_3) leg has the same phase, and the second (P_2) and the fourth (P_4) has almost the same phase.

required for performing on the trained terrains, 1 motor neuron representing hip-x joint and hip-y joint, and 2 motor neurons representing knee joint.

In order to evaluate the stability level, we recorded the body tilt sensor information which can be seen in Fig. 10 and analyzed it using Poincare map and Cobweb map which are illustrated in Fig. 11.

B. Implementation in Real Robot

The proposed method was further validated by real robot implementation. We utilized the optimized neural structure and uploaded it to the robot. We built a quadruped robot with 55 cm of height and 6 kg of weight, where the mechanical structure of the robot is the same as the robot in simulation experiment. In this experiment we have built 2 middle size quadruped robots, where the first robot is shown in Fig. 12a and the second one is shown in Fig. 12b. The proposed model is successfully applied to the real robots, and both robots are able to walk on both flat terrain and small uneven obstacles. The sample figure of the implementation is depicted in Fig. 13.

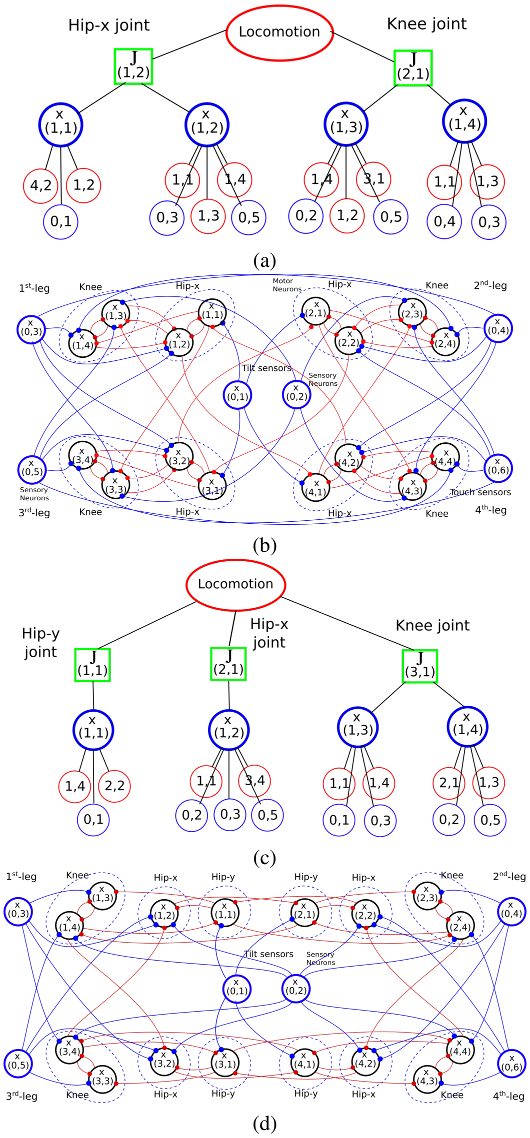


Fig. 8. (a) Optimized tree structure for flat terrain (b) Optimized neuron interconnection (c) Optimized tree structure for rough terrain (d) Optimized neuron interconnection

V. Conclusion

This paper proposed a novel biologically inspired neural based model for evolving the sensory-motor integration for robot locomotion. Since the sensory-motor neuron interconnection is getting more complex, a new tree structure based model of neuron interconnection structure is proposed. Bacterial programming is used as the optimization technique. The proposed tree structure can simplify the process and successfully generate stable walking for quadruped robots.

During the evolutionary process of neurons, the number of motor neurons in every joint were successfully optimized. The evolving model is able to reduce the number of neurons being involved, depending on the requirement of the robot's performance.

In order to represent the sensory-motor coordination, the proposed model can also generate the sensory-motor neuron interconnection. There are 6 sensory neurons representing 6

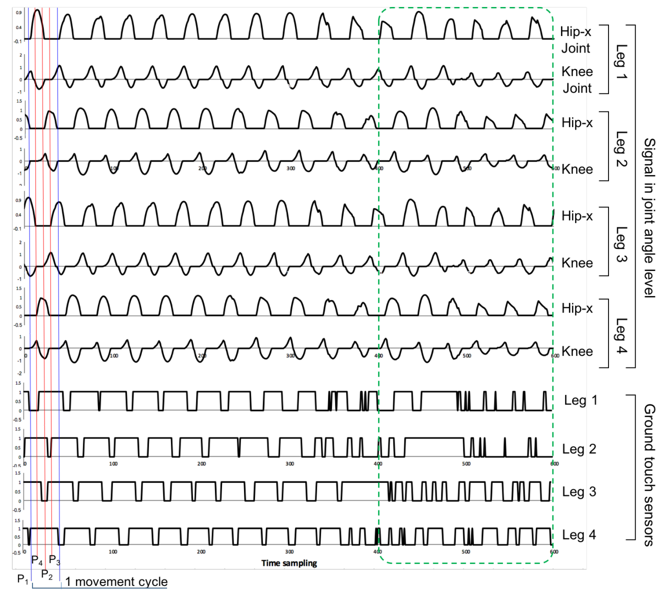


Fig. 9. The signal generated in the experiment on rough terrain. The effect of sensory input can be seen to the motor neuron signal. Green blocks show the difference of signal patterns because of the different sensory inputs. The signal phase in every leg ($P_1, \cdots, P_4$) has 0.25 phase difference that makes the movement slower than on the flat terrain.

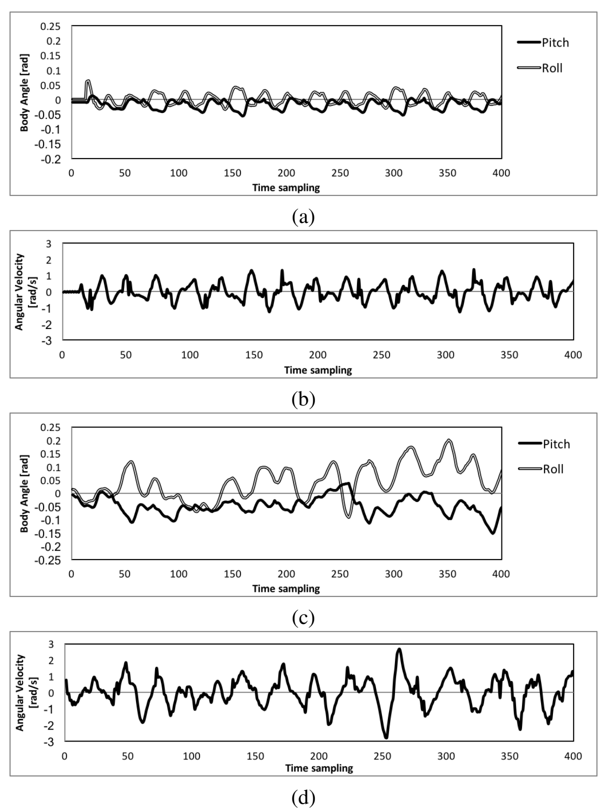
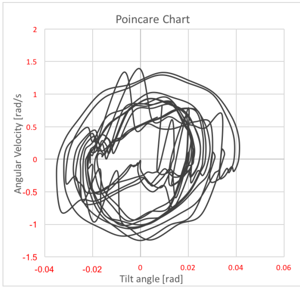
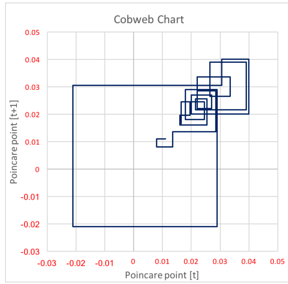


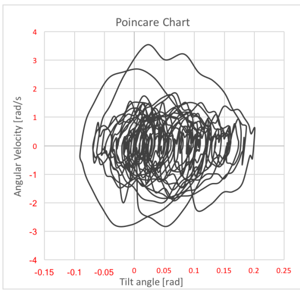
Fig. 10. The signal oscillation of angular velocity and tilt angle in pitch and roll direction measured from robot's body. (a) Body tilt signal on flat terrain. (b) Angular velocity signal on flat terrain. (c) Body tilt signal on rough terrain (outdoor grass). (d) Angular velocity signal on rough terrain. The signal on flat terrain is more stable than on rough terrain. Nevertheless, the signal oscillation is still acceptable for stability.



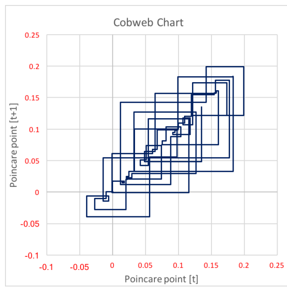
(a)



(b)



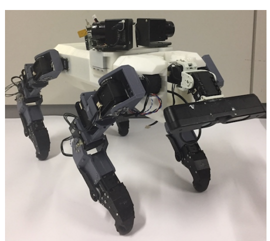
(c)



(d)

Fig. 11. Stability analysis diagram (a) Phase diagram of robot tilt angle and stability analysis on the flat terrain, based on Poincare map (b) Cobweb diagram representation of Fig. 11a (c) Poincare map during robot's performance on the rough terrain (d) Cobweb diagram representation of Fig. 11c

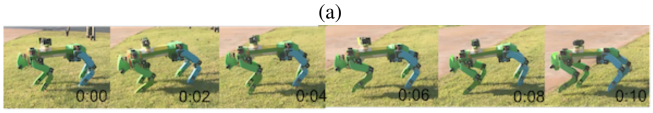
(a)



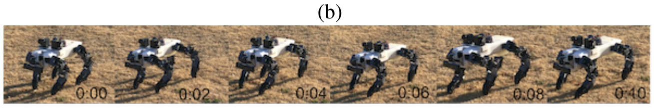
(b)

Fig. 12. Proposed middle size quadruped robot (a) first prototype (b) second prototype.

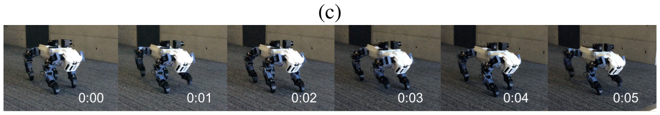
(a)



(b)



(c)



(d)

Fig. 13. Experimental result in a real robot. (a) First prototype of robot on the grass. (b) First prototype of robot on the grass with slope. (c) Second prototype of robot on the rough grass (d) Second prototype of robot on the flat terrain.

pieces of internal sensory information which are tilt sensor in pitch and roll, 4 ground sensors, one in each leg. According to the experimental results, signals generated by motor neurons are adaptively changing depending on the sensory information about the environment. The proposed model is able to reduce the computational cost to 67% compared to conventional models.

All in all, the proposed bacterial programming based tree structure model can be applied for simplifying the representation of sensory-motor interconnection in any multi-legged robots. For future works, we will conduct more experiments in different kind of walking surface for further validation.

Acknowledgement

János Botzheim was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] B. Smith, C. Saaj, and E. Allouis, "Evolving legged robots using biologically inspired optimization strategies," in *Proc. of Intl. Conf. on Robotics and Biomimetics (ROBIO)*, Dec 2010, pp. 1335–1340.

[2] A. Topchyan and T. Topchyan, "Muscle-based skeletal bipedal locomotion using neural evolution," in *Computer Science and Information Technologies (CSIT), 2013*, Sept 2013, pp. 1–6.

[3] J. Nassour, P. Henaff, F. B. Ouezdou, and G. Cheng, "Multi-layered multi-pattern CPG for adaptive locomotion of humanoid robots," *Biological Cybernetics*, pp. 291–303, 2014.

[4] Y. Lee, M. S. Park, T. Kwon, and J. Lee, "Locomotion control for many-muscle humanoids," *ACM Tran. on Graphics*, vol. 33, no. 6, p. 218, 2014.

[5] K. Glette and M. Hovin, "Evolution of artificial muscle-based robotic locomotion in physx," in *Proc. of Intl. Conf. on Intelligent Robots and Systems*, Oct 2010, pp. 1114–1119.

[6] D. Marbach and A. Ijspeert, "Online optimization of modular robot locomotion," in *Proc. of the 2005 IEEE International Conference on Mechatronics and Automation*, vol. 1, July 2005, pp. 248–253 Vol. 1.

[7] A. Ijspeert, "A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander," *Biological Cybernetics*, vol. 85, no. 5, pp. 331–348, 2001.

[8] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," 2008.

[9] G. Taga, Y. Yamaguchi, and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment," *Biological Cybernetics*, vol. 65, pp. 147–159, 1991.

[10] A. A. Saputra, N. N. W. Tay, Y. Toda, J. Botzheim, and N. Kubota, "Bézier curve model for efficient bio-inspired locomotion of low cost four legged robot," in *Intl. Conf. on Intelligent Robots and Systems*, 2016, pp. 4443–4448.

[11] A. A. Saputra, T. Takeda, J. Botzheim, and N. Kubota, "Multi-objective evolutionary algorithm for neural oscillator based robot locomotion," in *Proc. of the 41st Annual Conference of the IEEE Industrial Electronics Society*, Yokohama, Japan, November 2015, pp. 2655–2660.

[12] A. A. Saputra, J. Botzheim, I. A. Sulistijono, and N. Kubota, "Biologically inspired control system for 3-D locomotion of a humanoid biped robot," *IEEE Transaction on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 898–911, 2016.

[13] J. Botzheim, C. Cabrita, L. T. Koczy, and A. E. Ruano, "Genetic and bacterial programming for B-spline neural networks design," *Journal of Advanced Computational Intelligence Informatics*, vol. 11, no. 2, pp. 220–231, 2007.

[14] N. E. Nawa and T. Furuhashi, "Fuzzy system parameters discovery by bacterial evolutionary algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 7, no. 5, pp. 608–616, Oct. 1999.

[15] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.

[16] "Open Dynamics Engine," http://ode.org.