Modular Neural Control for Object Transportation of a Bio-inspired Hexapod Robot

Chris Tryk Lund Sørensen and Poramate Manoonpong(✉)

Embodied AI and Neurorobotics Lab, Centre for BioRobotics,
Mærsk Mc-Kinney Møller Institute, University of Southern Denmark,
Odense M, Denmark
chsoe14@student.sdu.dk, poma@mmmi.sdu.dk

Abstract. Insects, like dung beetles, can perform versatile motor behaviors including walking, climbing an object (i.e., dung ball), as well as manipulating and transporting it. To achieve such complex behaviors for artificial legged systems, we present here modular neural control of a bio-inspired hexapod robot. The controller utilizes discrete-time neurodynamics and consists of seven modules based on three generic neural networks. One is a neural oscillator network serving as a central pattern generator (CPG) which generates basic rhythmic patterns. The other two networks are so-called velocity regulating and phase switching networks. They are used for regulating the rhythmic patterns and changing their phase. As a result, the modular neural control enables the hexapod robot to walk and climb a large cylinder object with a diameter of 18 cm (i.e., ≈ 2.8 times the robot's body height). Additionally, it can also generate different hind leg movements for different object manipulation modes, like soft and hard pushing. Combining these pushing modes, the robot can quickly transport the object across an obstacle with a height up to 10 cm (i.e., ≈ 1.5 times the robot's body height). The controller was developed and evaluated using a physical simulation environment.

Keywords: Object manipulation · Locomotion · Modular neural network · Central pattern generator · Walking machines · Autonomous robots

1 Introduction

Over the last few decades, a number of animal-like walking robots have been developed. Most of them can perform only locomotion, like walking [1], climbing [2], and swimming [3]. Typically, if object manipulation or transportation tasks are required, additional manipulators/grippers need to be installed [4–6] instead of using existing legs. This becomes energy inefficient due to added load and the requirement of additional energy to power the manipulator or gripper system. Only a few works have shown walking robots which can locomote and transport an object using existing legs [7–9]. However, these robots require precise kinematic and force control; thereby they can only move or hold an object with the

E. Tuci et al. (Eds.): SAB 2016, LNAI 9825, pp. 67–78, 2016.
DOI: 10.1007/978-3-319-43488-9_7

stop-and-go motion. In other words, they cannot perform continuous movements for transporting an object, especially a large one.

In contrast, dung beetles with little neural computing can use their legs to continuously walk and at the same time move large objects - dung balls that can be larger than their body size [10]. In order to do so, the beetle walks backwards, climbs onto it, and uses its hind legs sometimes together with its middle legs to push the ball while its front legs are for walking. Inspired by the strategy of the beetle, we present here a modular neural control approach which allows a bio-inspired hexapod robot to walk backwards with a tripod gait, autonomously climb a large cylinder object, and use its hind legs to manipulate (i.e., push) the object while its front and middle legs are for walking. This results in continuous locomotion as well as object manipulation and transportation. With this technique, the robot can even perform different object manipulation modes including soft push, hard push, and boxing-like motion. A combination of soft and hard pushing strategies enables the robot to effectively transport a large cylinder object (larger than its body height) across an obstacle. We believe that the study in this direction will expand the usability of robots towards domains, like transportation and agriculture, in which (autonomous) mobile robots with multi functions are in high demand.

However, the rationale behind this study is not only to demonstrate the hexapod robot with multi functions (i.e., locomotion with object manipulation and transportation) but also to show that such complex functions can be achieved by a combination of neural modules. This pure neural network control has a layered, modular architecture which is inspired by the biological neural systems of insects [11]. Such a structure is also considered as a major advantage [12], compared to many other controllers [1], since it is able to deal with transferring and scaling issues; i.e., applying to different robots [13–15]. Thus, this modular neural control approach can be a powerful technique to solve sensorimotor coordination problems of many degrees-of-freedom systems (like walking robots) and to effectively provide complex multi functions to the systems.

2 Modular Neural Control for Object Transportation

To control the locomotion and object manipulation of a bio-inspired hexapod robot for continuous transporting an object, we employ neural mechanisms as the key ingredient of our controller. Although different methods [1] can be employed for the task, this neural control with a layered, modular architecture is selected in order to provide a basic control structure to the hexapod robot system. This way, neural learning mechanisms with synaptic plasticity for control parameter adaptation [16] could be later applied to obtain adaptive behavior.

The modular neural control is manually designed in a hierarchical way with seven neural modules (CPG, PSN1-4, and VRN1-2, Fig. 1(a)). There are four inputs $I_{1,2,3,4}$ (Fig. 1(a)) which are used to activate different motor patterns for forward/backward walking and different object manipulation modes. The complete structure of this modular neural control and the location of the

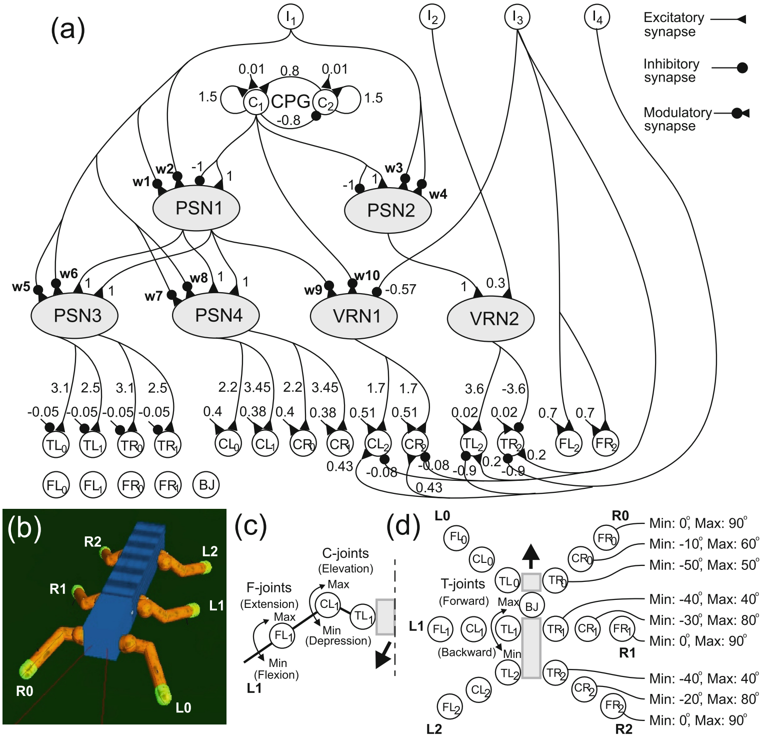

Fig. 1. (a) Modular neural control for locomotion and object manipulation. It is manually designed where its connection weights and inputs $I_{1,2,3,4}$ are tuned to obtain desired behavior (e.g., locomotion, object manipulation, etc.). Switching from one behavior to the other is achieved by manually setting the input values. By activating locomotion using the front and middle legs and object manipulation using the hind legs of a bio-inspired hexapod robot, the robot can perform object transportation (i.e., continuously transporting a cylinder object). Abbreviations are: BJ = a backbone joint, TL(R) = thoraco-coxal joints of left (right) legs, CL(R) = coxa-trochanteral joints of left (right) legs, FL(R) = femur-tibia joints of left (right) legs. (b) The simulated bio-inspired hexapod robot using the LPZRobots simulation environment (see http://robot.informatik.uni-leipzig.de/software). The robot consists of 19 joints: three joints for each leg and one backbone joint. The robot model is qualitatively consistent with our real hexapod robot AMOSII [16] in the aspect of size, mass distribution, motor torque/speed, and sensors. Its joint orientations follow the ones of the dung beetle *Geotrupes stercorarius*; i.e., the front legs are oriented slightly to the front while the middle and hind legs are oriented to the back. (c) The movements of the C- and F-joints. (d) The location of the motor neurons on the simulated robot and the movements of the T-joints. Minimum and maximum angles can be seen for all joints of the right legs where the same values are also set to the left ones.

corresponding motor neurons on the hexapod robot are shown in Fig. 1. The structural design of the control is based on our previous developed neural locomotion control [15,16].

The seven neural modules of the controller are derived from three generic neural networks[1]: A neural oscillator network (abbreviated CPG), a velocity regulating network (VRN), and a phase switching network (PSN). The neural oscillator network serves as a central pattern generator (CPG) module. It generates basic rhythmic signals. Here, the output signal C_1 of the CPG module (see Fig. 1(a)) is used to drive the joints of the robot for locomotion and object manipulation. To obtain proper motor patterns for locomotion and object manipulation, the CPG output signal is post-processed at the PSN and VRN modules. These modules act as premotor neuron networks. Here, the PSN1 and PSN2 modules receive the CPG output signal through excitatory and inhibitory synapses; i.e., they obtain the original CPG signal and its inversion. The outputs of these PSN modules are projected to the thoraco-coxal (T-) and coxa-trochanteral (C-) joints through the other PSN modules (PSN3 and PSN4) and the VRN modules (VRN1 and VRN2). These PSN modules are basically used to switch the phase of the T- and C-joint signals of the front and middle legs for forward/backward walking while the VRN modules are to regulate the amplitude of the hind legs to obtain different object manipulation modes (e.g., soft and hard pushing and boxing-like motion) as well as to maintain stability during object transportation. Note that the femur-tibia (F-) joints of the front and middle legs are kept fixed to a certain position while the F-joints of the hind legs are controlled by I_3 for object manipulation.

All these CPG, PSN, and VRN networks are described in details in the following sections. Their neurons are modelled as discrete-time non-spiking neurons with an update frequency of approx. 10 Hz. The activity of each neuron develops according to $a_i(t+1) = \sum_{j=1}^{n} w_{ij}\, o_j(t) + b_i;\ i = 1, \ldots, n$ where n denotes the number of units, b_i represents a fixed internal bias term of neuron i, w_{ij} the synaptic strength of the connection from neuron j to neuron i. The neuron output o_i is given by a hyperbolic tangent (tanh) transfer function. Input neurons ($I_{1,2,3,4}$) are here configured as linear buffers ($a_i = o_i$). All connection strengths together with bias terms are indicated by the small numbers (Fig. 1(a)) except w_{1-10} which are modulatory synapses (see section below for details). These fixed bias and synaptic connection values are here empirically set to obtain the desired locomotion and object manipulation patterns. However, they can be changed depending on robot configuration, e.g., the position of actuators.

2.1 Neural Oscillator Network (CPG)

The concept of central pattern generators (CPGs) for legged locomotion [11] has been studied and used in several robotic systems in particular walking robots.

[1] These networks have been successfully applied for locomotion control of various robot systems [14–16]. They are, for the first time here, employed for locomotion and object manipulation and transportation of a bio-inspired hexapod robot.

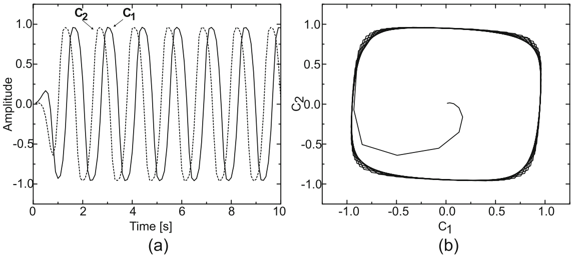

Fig. 2. (a) The periodic output signals from the CPG with the defined parameters shown in Fig. 1(a). Here we use the output signal C_1 for generating locomotion and object manipulation. (b) The phase space with the quasi-periodic attractor of the oscillator network [17].

Here, the model of a CPG is realized by using the discrete-time dynamics of a simple 2-neuron oscillator network with full connectivity (see Fig. 1(a)). Such a CPG model has been successfully used for locomotion control [15]. We empirically adjust the synaptic weights of this network to achieve a proper frequency of leg movements for stable locomotion and object manipulation. Figure 2 shows the outputs from the CPG network.

2.2 Phase Switching Network (PSN)

To obtain different modes (i.e., forward/backward locomotion and object manipulation), one possibility is to reverse the phase of the periodic signals driving the T- and C-joints (Fig. 1). That is, these periodic signals can be switched to lead or lag behind each other depending on the given input I_1. To do so, we use four phase switching network (PSN) modules (PSN1-4). The PSN was developed in our previous study [15]. It is a hand-designed feedforward network consisting of four hierarchical layers with 14 neurons P_{1-14} (Fig. 3). The synaptic weights and bias terms of the network were determined in a way that they do not change the periodic form of its input signals and keep the amplitude of the signals as high as possible (i.e., between -0.5 and $+0.5$). The detail of the network development is referred to [15]. For our implementation here (Fig. 1(a)), $P_{1,2}$ of the PSN1 and PSN2 modules receive the CPG signal C_1 through an excitatory synapse $(+1)$ and its inversion through an inhibitory synapse (-1) while their $P_{3,4}$ receive the input I_1 through the modulatory synapses $w_{1,2}$ for the PSN1 module and $w_{3,4}$ for the PSN2 module (Fig. 1(a)). $P_{1,2}$ of the PSN3 and PSN4 modules in a lower layer receive the outputs $P_{13,14}$ of the PSN1 module through an excitatory synapse $(+1)$ while their $P_{3,4}$ receive the input I_1 through the modulatory synapses $w_{5,6}$ for the PSN3 module and $w_{7,8}$ for the PSN4 module (Fig. 1(a)). The final outputs $P_{13,14}$ of the PSN3 and PSN4 modules are directly connected to the motor neurons of the T- and C- joints of the front and middle legs. The modulatory synapses of all PSN modules (Fig. 1(a)) are modelled as $w_{1,4,6,7} = I_1$ and $w_{2,3,5,8} = -I_1$. In this study, the bias terms $b_{1,2}$ (Fig. 3(a)) of the PNS1

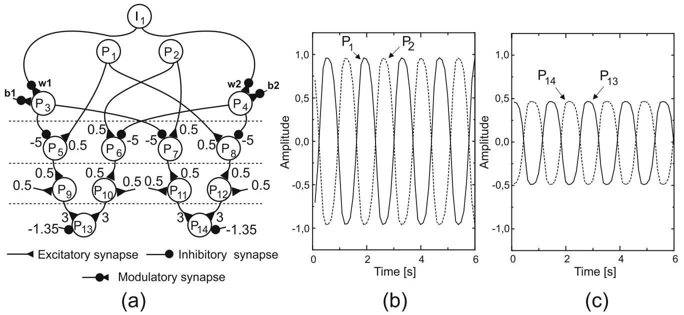

Fig. 3. (a) The phase switching network of the PSN1 module. The other PSN modules have the same structure. (b) The inputs of the PSN1 module which come from the CPG module. (c) The outputs of the PSN1 module where $I_{1,2}$ are set to (1,1) or (0,1). The outputs are inverted if $I_{1,2}$ are set to (1,−1), (−1,−1), or (1,0). Note that the inputs and outputs of the other PSN modules behave in a similar way.

and PSN4 modules are modelled as input-driven functions and described as $b_1 = \frac{-(I_1^2 I_2(I_2+1))}{2}$, $b_2 = -b_1$, while the ones of the PNS2 and PSN3 modules are set to $b_1 = -1$ and $b_2 = 0$. Note that the input-driven functions used here will basically activate or deactivate the neurons $P_{3,4}$ with respect to the inputs $I_{1,2}$.

2.3 Velocity Regulating Network (VRN)

To obtain different object manipulation modes (e.g., soft and hard pushing and boxing-like motion) and to maintain stability during object transportation, we need to regulate the signals controlling the T- and C-joints (TL_2, TR_2,CL_2, CR_2, see Fig. 1(a)) of the hind legs. According to this, we use two velocity regulating network (VRN) modules (VRN1,2) where one is for controlling the T-joints (TL_2, TR_2) and the other is for the C-joints (CL_2,CR_2). The VRN taken from [15] is a simple feed-forward neural network with two input $V_{1,2}$, four hidden V_{3-6}, and one output V_7 neurons (Fig. 4). It was trained by using the backpropagation algorithm to act as a multiplication operator on two input values on the neurons $V_{1,2} \in [-1, +1]$ (see [15] for details). For our purpose here, the neuron V_1 of the VRN1 module receives the input I_3 through an inhibitory synapse (e.g., −0.57, Fig. 1(a)) while the one of the VRN2 module receives the input I_2 through an excitatory synapse (e.g., 0.3, Fig. 1(a)). The bias term of the neuron V_1 of the VRN1 module is set to 1 while the one of the VRN2 module is set to 0.7 (Fig. 4(a)). The neuron V_2 of the VRN1 module receives two inputs (x, y) from the CPG output C_1 and the output P_{13} of the PSN1 module, respectively, through the modulatory synapses $w_{9,10}$ while the one of the VRN2 module receives only one input (x) from the output P_{13} of the PSN2 module through an excitatory synapse (+1, Fig. 1(a)). Additionally, the neuron V_2 of the VRN1 module has the bias term b_3 which is modelled as an input-driven function and described as $b_3 = 0.02((I_1^2 - I_2^2)^2 + \frac{I_1 I_2(I_2+I_1)}{2})^2)$ while there is no bias term for

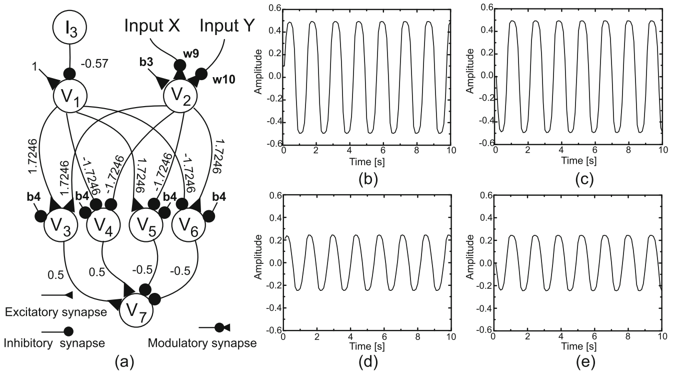

Fig. 4. (a) The velocity regulating network of the VRN1 module. The bias b_4 is equal to -2.48285. The VRN2 module has the same structure. (b–e) Different output signals of the VRN1 module for forward/backward walking (b), hard pushing (c), soft pushing (d), boxing-like motion (e). The VRN2 output behaves in a similar way.

the neuron V_2 of the VRN2 module (Fig. 4(a)). According to this input-driven function, b_3 will be 0.02 for all cases except soft pushing where it will be zero. Here, the synaptic weights $w_{9,10}$ are driven by the inputs $I_{1,2}$ and described as $w_9 = 2((I_1^2 - I_2^2)^2 + \frac{I_1 I_2 (I_2+I_1)}{2})^2)$ and $w_{10} = 1 - \frac{w_9}{2}$. According to these equations, w_9 will be equal to 2 for all actions except the soft pushing action for which it will be zero and the weight w_{10} will be zero for all actions except the soft pushing action for which it will be one. Finally, the outputs V_7 of the VRN1 and VRN2 modules are set to control the C-joints (CL_2,CR_2) and the T-joints (TL_2, TR_2), respectively. Note that all these functions of $b_3, w_{9,10}$ are used to scale the input signals (x, y) into proper ranges for different behavioral modes.

2.4 Neural Control Parameters for Different Behavior Modes

The integration of the different functional neural modules described above gives the complete modular neural controller. It can generate different behavioral modes[2] (locomotion, object manipulation, and their combination (i.e., object transportation)) through the four input parameters $I_{1,2,3,4}$. Appropriate input parameter sets for the different modes are presented in Table 1. $I_{1,2}$ are basically for generating different motor modes through the PSN and VRN modules while $I_{3,4}$, which can vary between -1.0 and 1.0, are for shifting the offsets of the leg joints upward/downward for object manipulation. Additionally, I_3 is used to scale the CPG and PSN signals through the VRN1 module to obtain proper movements for soft pushing and boxing-like motion. Note that the input values shown in Table 1 can be changed with respect to, e.g., robot configuration.

[2] See http://manoonpong.com/SAB2016/V1.mp4.

Table 1. Input parameters for different behavior modes.

Actions	I_1	I_2	I_3	I_4
Locomotion: Forward walking	1	1	0	0
Locomotion: Backward walking	0	1	0	0
Object manipulation: Soft pushing	1	−1	1	1
Object manipulation: Boxing-like motion	−1	−1	1	0
Object manipulation: Hard pushing	1	0	0	−0.3

3 Experiments and Results

To evaluate the performance of the developed controller, we used the simulated bio-inspired hexapod robot (see Fig. 1(b)) with a body height of 6.5 cm and a weight of ≈ 5 kg and a cylinder object (see Fig. 5(b)) with a length of 60 cm, a diameter of 18 cm (i.e., ≈ 2.8 times the robot's body height), and a weight of 2 kg. The friction coefficient of robot feet was set based on a rubber material used for the feet of the real robot while the friction coefficients of the object and ground were empirically set to obtain high friction and to avoid slipping during locomotion and object transportation. With the controller, the robot can walk forward with a tripod gait and can walk backward by changing the phase of the T-joints through the PSN2 and PSN3 modules. Note that the C-joint signals are clipped to ensure that the legs touch the ground during the stance phase; resulting in a stable walking behavior. Here, the F-joints stay in a certain position.

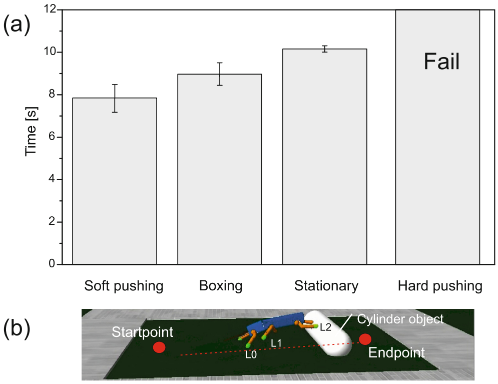

Fig. 5. Result of the speed test for object transportation using different object manipulation modes. (a) The bars show the average object transportation time with the standard deviation. The time was measured from the starting point to the end point where the distance is 1.46 m. We performed in total ten tests for each mode. (b) The startpoint and endpoint locations from which the robot has to transport the object. Note that in this experiment the robot was activated to walk backward without any additional steering command.

To let the robot transport the object, we drive the robot to walk backward. While walking backward and approaching an object, the robot will automatically climb the object since we set the backbone joint (BJ) in a slightly bending position. With this BJ setup, the body of the robot bends slightly upwards; thereby allowing the robot to swing its hind legs slightly more upwards during a swing phase and place its leg tips above the center line of the object during a stance phase. This way, the robot can climb the object. Once the robot has been stayed on the object partly, which is detected by a body inclination sensor, specific hind leg movements for different object manipulation modes will be activated while the front and middle legs remain unchanged. For soft pushing, the hind legs will slowly roll a cylinder object while the robot walks backward. For the boxing-like motion, as the word describes, the robot uses the hind legs to hit or punch the object and in this way move it. For hard pushing, the robot uses the hind legs to dig under the object in order to make it across an obstacle.

Two main experiments were carried out for our evaluation. The first experiment evaluates an object transportation speed of the robot without an obstacle when different object manipulation modes were used. The soft pushing, hard pushing, and boxing modes, where the hind legs actively move in specific patterns (see Footnote 2), were tested. Additionally, we also compare them with a situation where the hind legs were kept fixed in a certain position (not moving) and stayed on top of the object to avoid it run away (i.e., stationary mode[3]). Figure 5 shows the result of this experiment. It can be seen that the robot can transport or move the object with the fastest speed (i.e., less time) in a straight backward direction when the soft pushing mode was used while other modes required more time to reach the target location. The robot failed to do the task when the hard pushing mode was used because with this mode it pushed the object away in an arbitrary direction (see Footnote 2).

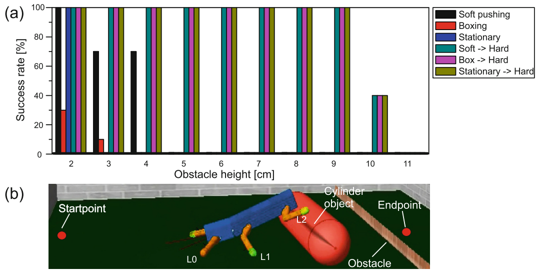

Fig. 6. Result of the obstacle test for the different object manipulation techniques. (a) Success rate in a total of ten experiments for each strategy. (b) The startpoint and endpoint locations from which the robot has to transport the object. In this experiment the robot was activated to walk backward without any additional steering command.

[3] See http://manoonpong.com/SAB2016/V2.mp4.

The second experiment evaluates the performance of the robot with different manipulation strategies to transport the object across an obstacle at different heights. The obstacle width was set to 1 mm while the obstacle height was varied from 2 cm to 11 cm. In total, we tested six strategies including soft pushing, boxing, stationary, and their combination with hard pushing. For the combination modes (i.e., soft pushing (Mode1) → hard pushing (Mode2), boxing (Mode1) → hard pushing (Mode2), and soft pushing (Mode1) → hard pushing (Mode2)), we switch from one pushing mode (Mode1) to another pushing mode (Mode2) when the object has reached or hit the obstacle. This is detected by the joint angle sensors of the F-joints of the front legs. If the angle sensors decrease below a threshold, then the switching occurs. Figure 6 presents the success rate of object transportation; i.e., the percentage of success from ten experiments each. A success is considered if the object gets across the obstacle within one minute. It can be seen that the combination modes outperform individual modes and allow the robot transport the object across the obstacle at the maximum height of 10 cm. However, when we take the transportation time into account the combination of soft pushing (Mode1) → hard pushing (Mode2) is the best since, with this mode, the robot uses first the soft pushing mode to roll the object leading to

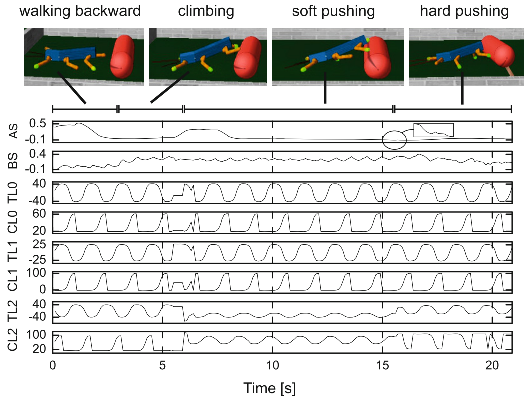

Fig. 7. Example of sensor and motor signals of the hexapod robot for object transportation. The robot first walked backward and then autonomously climbed the object due to the interaction between the leg movements and the object. Afterwards it performed soft pushing and finally hard pushing to move the object across the obstacle with a height of 5 cm. The soft pushing behavior was activated by the body inclination sensor signal (BS). It will be activated if the sensor value is higher than a threshold after a certain time step. The hard pushing behavior was activated by the joint angle sensor signals of the F-joints of the front legs. We used the average value of the angle signals (AS) for this activation. Basically, the hard pushing behavior will be activated if the value is smaller than a threshold. $TL_{0,1,2}$ are the thoraco-coxal (T-) joints of the left front, middle, and hind legs. $CL_{0,1,2}$ are the coxa-trochanteral (C-) joints of the left front, middle, and hind legs (see Fig. 1). The F-joints are not shown since they have constant values. The joint angle signals of the right legs are shown in degree. The left angle signals are similar to the right ones.

fast transportation speed compared to the others (see Fig. 5) and then the hard pushing mode to strongly push the object across the obstacle. Figure 7 shows the sensors and motor signals of the robot during object transportation using the combination of the soft pushing and hard pushing modes[4].

4 Conclusion

We present the modular neural controller of a bio-inspired hexapod robot. The controller is derived from three neural networks (CPG, PSN, and VRN). Each network has its functional origin in biological neural systems (see [14] for details). The controller can generate various motor patterns for locomotion, object manipulation, and their combination (resulting in object transportation). Different object manipulation strategies can be obtained from the controller. Among them, the strategy that combines soft pushing and hard pushing allows the robot to quickly roll a large cylinder object (i.e., ≈ 2.8 times the robot's body height) and to strongly push it across an obstacle with a height up to ≈ 1.5 times the robot's body height. Although the resulting object transportation behavior is inspired by the strategy of a dung beetle, the object used in this study is still smaller than and different from the one that the beetle can transport (i.e., dung ball). Furthermore, the beetle can also transport the ball on rough terrain using its middle and hind legs while walking with its front legs. Thus, in the future work, we will investigate another object transportation mode using the middle and hind legs to transport a large ball on rough terrain. We will also apply this approach to a real hexapod robot and test it in a real environment.

Acknowledgments. We would like to thank Georg Martius for technical advise about the LpzRobots simulation software.

References

1. Cully, A., Clune, J., Tarapore, D., Mouret, J.B.: Robots that can adapt like animals. Nature **521**, 503–507 (2015)
2. Inoue, K., Fujii, S., Takubo, T., Mae, Y., Arai, T.: Ladder climbing method for the limb mechanism robot asterisk. Adv. Robot. **24**, 1557–1576 (2010)
3. Crespi, A., Karakasiliotis, K., Guignard, A., Ijspeert, A.J.: Salamandra robotica II: an amphibious robot to study salamander-like swimming and walking gaits. IEEE Trans. Robot. **29**, 308–320 (2013)
4. Bartsch, S., Planthaber, S.: Scarabaeus: a walking robot applicable to sample return missions. In: Gottscheber, A., Enderle, S., Obdrzalek, D. (eds.) EUROBOT 2008. CCIS, vol. 33, pp. 128–133. Springer, Heidelberg (2009)
5. Rehman, B.U., Focchi, M., Frigerio, M., Goldsmith, J., Caldwell, D.G., Semini, C.: Design of a hydraulically actuated arm for a quadruped robot. In: Proceedings of the International Conference on Climbing and Walking Robots, pp. 283–290 (2015)

[4] see http://manoonpong.com/SAB2016/V3.mp4.

6. Heppner, G., Buettner, T., Roennau, A., Dillmann, R.: Versatile - high power gripper for a six legged walking robot. In: Proceedings of the International Conference on Climbing and Walking Robots, pp. 461–468 (2014)
7. Koyachi, N., Adachi, H., Arai, T., Izumi, M., Hirose, T., Senjo, N., Murata, R.: Walk and manipulation by a hexapod with integrated limb mechanism of leg and arm. J. Robot. Soc. Jpn. **22**, 411–421 (2004)
8. Inoue, K., Ooe, K., Lee, S.: Pushing methods for working six-legged robots capable of locomotion and manipulation in three modes. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4742–4748 (2010)
9. Takeo, G., Takubo, T., Ohara, K., Mae, Y., Arai, T.: Internal force control for rolling operation of polygonal prism. In: Proceedings of the IEEE International Conference on Robotics and Biomimetics, pp. 586–591 (2009)
10. Philips, T.K., Pretorius, E., Scholtz, C.H.: A phylogenetic analysis of dung beetles (Scarabaeinae): unrolling an evolutionary history. Invertebr. Syst. **18**, 53–88 (2004)
11. Bässler, U., Büschges, A.: Pattern generation for stick insect walking movements-multisensory control of a locomotor program. Brain Res. Rev. **27**, 65–88 (1998)
12. Valsalam, V., Miikkulainen, R.: Modular neuroevolution for multilegged locomotion. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 265–272 (2008)
13. Hornby, G., Takamura, S., Yamamoto, T., Fujita, M.: Autonomous evolution of dynamic gaits with two quadruped robots. IEEE Trans. Robot. Autom. **21**, 402–410 (2005)
14. Manoonpong, P., Wörgötter, F., Laksanacharoen, P.: Biologically inspired modular neural control for a leg-wheel hybrid robot. Adv. Robot. Res. **1**, 101–126 (2014)
15. Manoonpong, P., Pasemann, F., Wörgötter, F.: Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines. Robot. Auton. Syst. **56**, 265–288 (2008)
16. Grinke, E., Tetzlaff, C., Wörgötter, F., Manoonpong, P.: Synaptic plasticity in a recurrent neural network for versatile and adaptive behaviors of a walking robot. Front. Neurorobot. **9**, 1–15 (2015). doi:10.3389/fnbot.2015.00011
17. Pasemann, F., Hild, M., Zahedi, K.: So(2)-networks as neural oscillators. In: Proceedings of 7th International Work-Conference on Artificial and Natural Neural Networks (IWANN 2003), pp. 1042–1042 (2003)