Supplementary information accompanying the manuscript Biologically Inspired Modular Neural Control for a Leg-Wheel Hybrid Robot

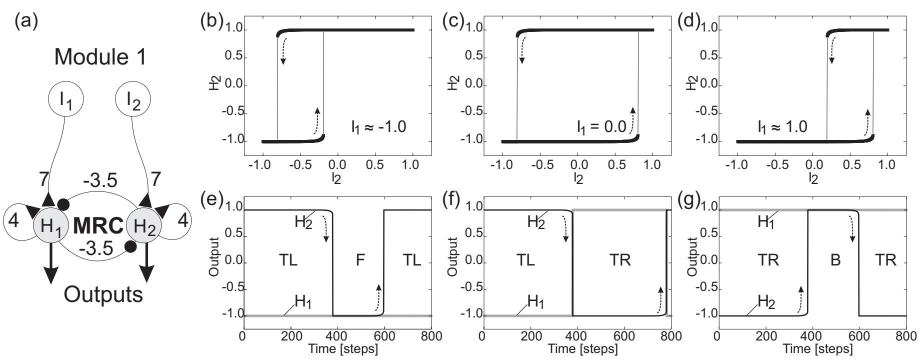
Poramate Manoonpong[a,*], Florentin Wörgötter[a], Pudit Laksanacharoen[b]

[a)]Bernstein Center for Computational Neuroscience (BCCN),
The Third Institute of Physics,
Georg-August-Universität Göttingen,
D-37077 Göttingen, Germany
[b)]Mechanical and Aerospace Engineering Department,
Faculty of Engineering,
King Mongkuts University of Technology North Bangkok,
Bangkok 10800, Thailand
* To whom correspondence should be addressed,
Email addresses: poramate@physik3.gwdg.de

This supplementary information includes:

1 Minimal Recurrent Control (MRC)

On the basis of the well understood functionalities and dynamics of the MRC (Hülse and Pasemann, 2002; Manoonpong, 2007), we here empirically adjusted the connection weights of the network for our robot as follows. First, the weights from the inputs ($I_{1,2}$, **Supplementary Figure 1(a)**) to the neurons ($H_{1,2}$, **Supplementary Figure 1(a)**) were set to a high value as amplification factors, i.e., 7.0. Then the self-connection weights of the neurons were adjusted to derive a reasonable hysteresis interval on the input space. In this case, the hysteresis effect determines the turning angle in front of the obstacles for avoiding them, i.e., the wider the hysteresis, the larger the turning angle. Both self-connections are set to 4.0 to obtain a suitable turning angle to avoid obstacles or sharp corners (less than 90 degrees, see the Experiments and Results section). It is important to note that this turning angle depends also on the application environment of the robot as well as the robot configuration. Finally, the recurrent connections between the neurons were symmetrized and adjusted to -3.5. Such inhibitory recurrent connections are formed as a so-called even loop (Pasemann, 1993), which also shows hysteresis (**Supplementary Figure 1**). In general conditions, only one neuron at a time is able to produce a positive output ($\approx +1$), while the other one has a negative output (≈ -1), and vice versa (**Supplementary Figure 1**). However, both neurons ($H_{1,2}$, **Supplementary Figure 1(a)**) can show high activation only if their inputs are very high, e.g., > 0.81 (**Supplementary Figure 1(c)**). This guarantees optimal functionality for avoiding obstacles and escaping from corner and deadlock situations (Hülse et al., 2004). Additionally, the setup parameters enable the network to eliminate the noise of the sensory signals. The complete network is shown in **Supplementary Figure 1(a)**. The hysteresis effects of the network and time evolution of its outputs are exemplified in **Supplementary Figures 1(b)–(g)**.



Supplementary Figure 1: (a) The MRC where its connection weights are empirically adjusted for controlling an obstacle avoidance behavior of the leg-wheel hybrid robot. (b), (c), (d) Hysteresis domain of the input neuron I_2 for the output neuron H_2 of the network with the input neuron I_1 fixed. (e), (f), (g) Time evolution of $H_{1,2}$ for varying I_2 (see text for details).

Supplementary Figure 1(b) shows that, setting I_1 to ≈ -1.0 (i.e., there is no obstacle on the left of the robot), the output neuron H_1 shows low $\approx$

-1 activation at all times while H_2 changes according to I_2 (**Supplementary Figure 1(e)**). In this case, the robot will move forward F as long as H_1 and H_2 give low activation but it will turn left TL as soon as I_2 increases to values above ≈ -0.2 leading to high activation of H_2; i.e., there is an obstacle on its right (**Supplementary Figure 1(e)**). However, it will return to move forward when I_2 decreases to values below ≈ -0.81 meaning that no obstacle is detected.

Supplementary Figure 1(c) shows that, setting I_1 to 0.0 (i.e., there is an obstacle on the left of the robot in a long distance), H_1 shows low ≈ -1 and high $\approx +1$ activation opposite to the activation of H_2 driven by I_2 (**Supplementary Figure 1(f)**). In this case, the robot will generally turn right TR but it will turn left TL (**Supplementary Figure 1(f)**) as soon as I_2 increases to values above ≈ 0.81. As a consequence, H_2 shows high activation which then inhibits H_1 (i.e., detecting a very close obstacle on its right). And the robot will turn right TR (**Supplementary Figure 1(f)**) when I_2 decreases to values below ≈ -0.81 such that H_2 becomes inactive (≈ -1) resulting that H_1 becomes automatically active ($\approx +1$).

Supplementary Figure 1(d) shows that, setting I_1 to ≈ 1.0 (i.e., there is a very close obstacle on the left of the robot), H_1 shows high $\approx +1$ activation at all times while H_2 changes according to I_2 (**Supplementary Figure 1(g)**). In this case, the robot will turn right TR as long as H_2 gives low activation but it will move backward B as soon as I_2 increases to values above ≈ 0.81 leading to high activation of H_2; i.e., there is also a very close obstacle on its right (**Supplementary Figure 1(g)**). However, it will return to turn right when I_2 decreases to values below ≈ 0.2 meaning that only the obstacle on its left is still detected. In reverse cases, if I_1 is varied while I_2 is fixed, it will derive the same hysteresis effect as I_2 does.

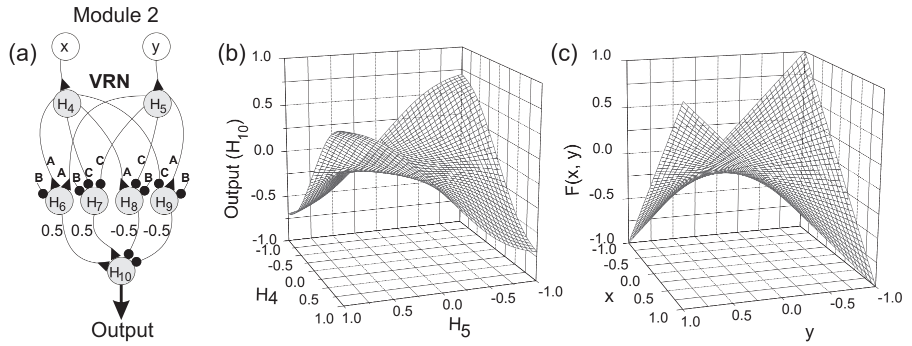
2 Velocity Regulating Network (VRN)

The VRN is derived from a multiplication of two values of the range $x, y \in [-1,1]$. It was constructed by four hidden neurons which are connected with an output neuron and was trained by using the backpropagation algorithm (Rumelhart et al., 1980). **Supplementary Figure 2(a)** presents the resulting network. It approximately works as a multiplication operator (**Supplementary Figures 2(b) and (c)**).

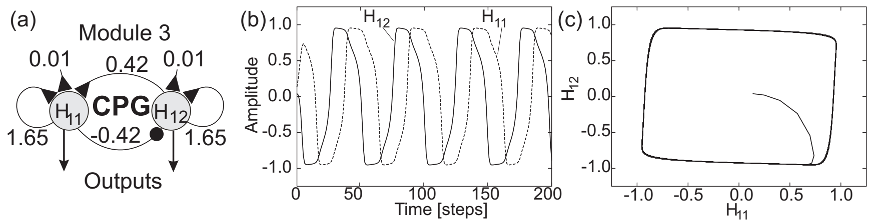
3 Neural Oscillator Network

The neural oscillator network is realized by using two neurons with full connectivity and additional biases (**Supplementary Figure 3(a)**). The network parameters was manually adjusted for our task here. The resulting weights and the outputs of the network according to these setup parameters are shown in **Supplementary Figure 3**. More investigation and analysis of the network can be found in (Manoonpong, 2007; Pasemann et al., 2003; Manoonpong et al., 2008).

It is important to note that this 2-neuron oscillator network is used here since: 1) it is inspired by neural structures found in insects (Büschges, 2005), 2) its output signals after post processing via the PSN can produce appropriate



Supplementary Figure 2: (a) The VRN where its parameters are given by $A = 1.7246$, $B = -2.48285$, $C = -1.7246$. (b) The approximation $H_{10}(H_4, H_5)$ of the VRN with average mean square error $(e^2) \approx 0.0046748$. The output H_{10} of the neuron is given by a sigmoidal transfer function *tanh*; therefore the suitable input values x, y projecting to H_4 and H_5 are in the range of $[-1,\cdot\cdot\cdot,1]$. (c) The multiplication function F(x, y) = x · y.



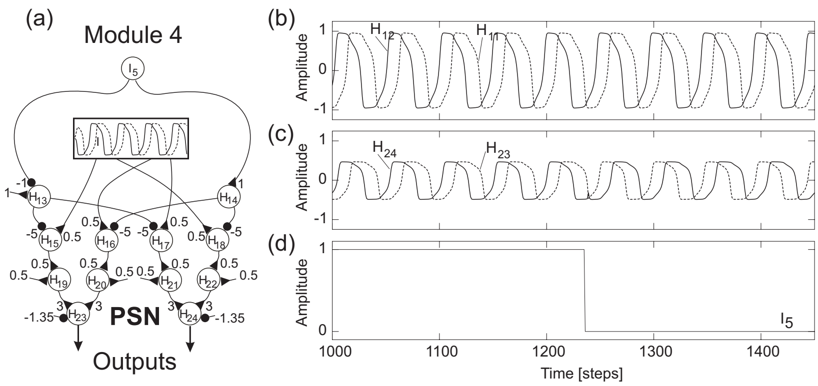
Supplementary Figure 3: (a) The 2-neuron oscillator network (CPG). (b) Output signals of neurons H_{11} (dashed line) and H_{12} (solid line) from the neural oscillator network. They differ in phase by $\pi/2$ and have a frequency of approximately 0.8 Hz. They are used to basically drive the legs (i.e., here, $M_{left,leg}$, $M_{right,leg}$) of the robot. (c) Phase space with quasi-periodic attractor of the oscillator network.

rhythmic patterns (i.e., asymmetry of ascending and descending slopes, see Experiments and Results section in the main manuscript) for sidestepping, and 3) the network can be later extended to achieve a so-called adaptive neural chaos oscillator which produces chaotic and a large variety of periodic patterns. Such patterns can be useful for specific behaviors necessary to appropriately respond to a changing environment like self-untrapping from a hole in the ground as shown in (Steingrube et al., 2010).

4 Phase Switching Network (PSN)

The PSN is a hand-designed feedforward network consisting of four hierarchical layers with 12 neurons. The development of this network is described as follows. First, the periodic signals of the neural oscillator network are provided to the PSN through two pairs of hidden neurons ($H_{15,16}$ and $H_{17,18}$, **Supplementary Figure 4(a)**). The synaptic weights projecting to them are determined such that they should not change the periodic form of their input signals and should keep the amplitude of the signals as high as possible. Thus, we set these synaptic weights to 0.5, which will convert the signals in the linear domain of the sigmoidal transfer function *tanh*. The activation of $H_{15,16,17,18}$ is controlled by higher layer neurons $H_{13,14}$ with large inhibitory connections (i.e., -5.0). H_{13} (or H_{14}) will inhibit its target neurons (**Supplementary Figure 4(a)**) if it is activated, where its activation will be controlled by the binary values of I_5. As a result, one neuron of each pair (H_{15} or H_{16} and H_{17} or H_{18}) will be activated while the other will be inhibited. For instance, if H_{15} and H_{17} are activated, they will give periodic outputs while H_{16} and H_{18} will give a constant value of -1.0 and vice versa. To preserve the periodic output of the activated neurons, e.g., $H_{15,17}$, we have to shift the signals of the inhibited neuron, e.g., $H_{16,18}$, from -1.0 to 0.0 before summing them. This is done by the hidden neurons $H_{19,20,21,22}$ of the lower layer.

The synaptic weights together with the bias terms connected to them are set in a way that the signals will be again converted in the linear domain and the output signals of the inhibited neurons will be shifted to minimally 0.0. That is, we again choose them as 0.5. Finally, we amplify the output signals of $H_{19,20}$ and $H_{21,22}$ with larger synaptic weights, i.e., 3.0, and combine them via the output neurons $H_{23,24}$. Additionally, we set the bias terms of $H_{23,24}$ to -1.35 to shift the offset of the resulting output signals down. **Supplementary Figure 4** shows the resulting network and the output signals of it with respect to the given input I_5.



Supplementary Figure 4: (a) The PSN. (b) Output signals of the neural oscillator network projecting to the PSN through hidden neurons $H_{15,16,17,18}$. (c) Output signals ($H_{23,24}$) of the PSN controlled by the input I_5 (d). From 1000 to around 1230 time steps I_5 is set to 1; such that H_{14} is activated while H_{13} is deactivated because of its bias term. Thus, H_{14} inhibits the activation of its targeting neurons $H_{16,18}$. As a result, $H_{23,24}$ of the network generate the periodic signals originally coming from $H_{11,12}$ of the CPG through $H_{15,19}$ and $H_{17,21}$. On the other hand, the periodic signals go through other neuron paths when I_5 is set to 0 after around 1230 time steps.

References

Büschges, A. (2005). Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion. *J. Neurophysiol.*, 93:1127–1135.

Hülse, M. and Pasemann, F. (2002). Dynamical neural schmitt trigger for robot control. In JR, D., editor, *Proceedings of the International Conference on Artificial Neural Networks*, volume 2415, pages 783–788. Springer Verlag.

Hülse, M., Wischmann, S., and Pasemann, F. (2004). Structure and function of evolved neuro-controllers for autonomous robots. *Connect. Sci.*, 16(4):249–266.

Manoonpong, P. (2007). *Neural Preprocessing and Control of Reactive Walking Machines: Towards Versatile Artificial Perception-Action Systems.* Cognitive Technologies. Springer.

Manoonpong, P., Pasemann, F., and Wörgötter, F. (2008). Sensor-Driven Neural Control for Omnidirectional Locomotion and Versatile Reactive Behaviors of Walking Machines. *Robot. Auton. Syst.*, 56(3):265–288.

Pasemann, F. (1993). Discrete dynamics of two neuron networks. *Open Syst. Inf. Dyn.*, 2:49–66.

Pasemann, F., Hild, M., and Zahedi, K. (2003). SO(2)-networks as neural oscillators. In Mira, J. and Alvarez, J., editors, *Computational Methods in Neural Modeling: Proceedings of the 7th International Work-Conference on Artificial and Natural Networks*, volume 2686, pages 144–151. Springer Berlin.

Rumelhart, D., Hinton, G., and Williams, R. (1980). Learning internal representations by error propagation. In *Parallel distributed processing: Explorations in the microstructure of cognition*, volume 1, pages 318–362.

Steingrube, S., Timme, M., Wörgötter, F., and Manoonpong, P. (2010). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nature Phys.*, 6:224–230.