Advances in Robotics Research, *Vol. 1, No. 1 (2014) 101-126*
DOI: http://dx.doi.org/10.12989/arr.2014.1.1.101

Biologically inspired modular neural control for a leg-wheel hybrid robot

Poramate Manoonpong[*1], Florentin Wörgötter[1] and Pudit Laksanacharoen[2]

[1]*Bernstein Center for Computational Neuroscience (BCCN), the Third Institute of Physics, Georg-August-Universität Göttingen, D-37077 Göttingen, Germany*
[2]*Mechanical and Aerospace Engineering Department, Faculty of Engineering, King Mongkut' s University of Technology North Bangkok, Bangkok 10800, Thailand*

(Received February 22, 2013, Revised June 10, 2013, Accepted July 27, 2013)

Abstract. In this article we present modular neural control for a leg-wheel hybrid robot consisting of three legs with omnidirectional wheels. This neural control has four main modules having their functional origin in biological neural systems. A minimal recurrent control (MRC) module is for sensory signal processing and state memorization. Its outputs drive two front wheels while the rear wheel is controlled through a velocity regulating network (VRN) module. In parallel, a neural oscillator network module serves as a central pattern generator (CPG) controls leg movements for sidestepping. Stepping directions are achieved by a phase switching network (PSN) module. The combination of these modules generates various locomotion patterns and a reactive obstacle avoidance behavior. The behavior is driven by sensor inputs, to which additional neural preprocessing networks are applied. The complete neural circuitry is developed and tested using a physics simulation environment. This study verifies that the neural modules can serve a general purpose regardless of the robot's specific embodiment. We also believe that our neural modules can be important components for locomotion generation in other complex robotic systems or they can serve as useful modules for other module-based neural control applications.

Keywords: neural networks; mobile robot control; autonomous robots; obstacle avoidance; reactive behavior

1. Introduction

Recently, roboticists have become more interested in multi-modal locomotion to enhance mobility and ensure integrity of robotic systems for locomotion in different terrains/surfaces as well as for autonomous exploration missions, e.g., planetary exploration and scouting in hazardous areas including transportation and rescue tasks. Thus, new types of robots like leg-wheel hybrid robots with different configurations have been increasingly developed to solve these tasks (Besseron *et al.* 2005, Nakajima and Nakano 2008, Tanaka and Hirose 2008, Klavins *et al.* 2000, Allen *et al.* 2003, Eich *et al.* 2008, Halme *et al.* 2001, Mahmoud *et al.* 2008). For example, the robot HyLoS (Besseron *et al.* 2005) consisting of four legs and four standard wheels was

*Corresponding author, Ph.D., E-mail: poramate@physik3.gwdg.de

http://www.techno-press.org/?journal=arr&subpage=7 ISSN: 2287-4976 (Print), 2287-4984 (Online)

developed to traverse different slopes. The robot Chariot III (Nakajima and Nakano 2008) was constructed with four legs and two wheels beside its body. It can locomote on rough terrain including climbing up a stair. One impressive hybrid robot is the leg-wheel hybrid jumping robot AirHopper II (Tanaka and Hirose 2008). It combines leg, jumping, and wheel mechanisms. As a consequence, the robot can move on flat surfaces using wheels and jump over large obstacles as well as land using its legs. In contrast to HyLoS, Chariot III, and AirHopper II where each of their legs has more than one degree of freedom (DOF), robots like RHex (Klavins *et al.* 2000), Whegs (Allen *et al.* 2003), and ASGUARD (Eich *et al.* 2008) have only one active joint (i.e., one DOF) at each "wheel-like" leg for propulsion to overcome various obstacles including rugged and sandy grounds, slopes and even stairs.

To tackle this challenging problem (i.e., multi-locomotion modes for autonomous exploration missions), during the last years we have developed a leg-wheel hybrid robot consisting of three legs with omnidirectional wheels which is able to transform into a spherical shape (Laksanacharoen and Jearanaisilawong 2009, Chadil *et al.* 2011). It combines the idea of using legs, wheels, and a rolling sphere for multi-modal locomotion. The conceptual design of this robot was that it will be initially packed and deployed in a spherical configuration. Due to its closed spherical shape, the robot allows for easy transportation and deployment; for instance, a number of these robots can be packed and deployed together from an aircraft. Cushioning materials can be added on the shell for soft landing. After landing, the robot will passively roll for some distance (Chadil *et al.* 2011). Afterwards it will transform into two inter-connected hemispheres and extend its three legs to further locomote using wheels or legs for autonomous exploration. To the best of our knowledge, this type of robot, which combines legs, wheels, and a rolling sphere for multi-modal locomotion, so far has not been developed by other researches. As described above, there are several leg-wheel hybrid robots but without rolling sphere while there are spherical rolling robots but without legs and wheels (Kim *et al.* 2010, Armour and Vincent 2006).

Continuing the development of our robot system, this article presents neural control of our leg-wheel hybrid robot for the generation of active locomotion using wheels and legs as well as controlling a reactive obstacle avoidance behavior in cluttered environments. This pure neural network control has a modular structure which is inspired by biological neural locomotion systems of insects (Bassler and Büschges 1998, Delcomyn 1999). Such a structure is also considered as a major advantage (Valsalam and Miikkulainen 2009, Valsalam and Miikkulainen 2008), compared to many other controllers (see the Discussion section for details). The entire control is composed of two main components: the neural sensory preprocessing and the modular neural locomotion control. The neural sensory preprocessing is for processing sensory inputs used to drive corresponding behaviors while the modular neural locomotion control consisting of four modules is for locomotion generation. Each module has its functional origin in the biological neural locomotion systems (i.e., neural and biological justifiable, see the Discussion section for details). This study also verifies that these neural modules can serve as a general purpose largely regardless of the robot's specific embodiment; i.e., it can be applied to not only walking robots and simple wheel robots studied in the past but also to this special designed leg-wheel hybrid robot (i.e., transferable and generic, see the Discussion section for details). The complete neural circuitry is also robust to changes of structures; i.e., modules can be completely removed leading to graceful degradation of the agent's functionality while as a whole the system can still function partially (see the Discussion section for details). All the essential features required from our modular neural control circuit distinguish it from others. Here, the controller will be developed and evaluated

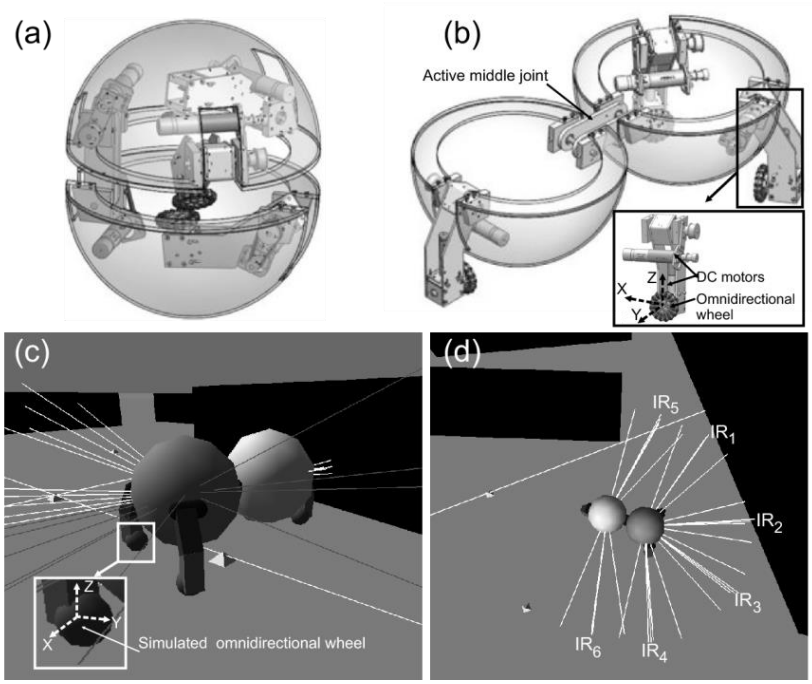

Fig. 1 (a), (b) The physical robot in dormant and transformed modes, respectively. (c), (d) Simulated three-legged robot with omnidirectional wheels in its virtual environment. Beams around the robot are infrared sensors $IR_{1,2,3,4,5,6}$. To model the omnidirectional wheels, we set friction coefficients for two orthogonal directions (x- and y-axes in (c)) of each wheel independently. As a consequence, the wheel rolls forward like a normal wheel (i.e., rolling with full force around y-axis) but it will slide sideways with almost no friction (i.e., freely rotating around x-axis)

using a physics simulation environment.

2. Simulated leg-wheel hybrid robot

We use the physical simulation environment called "Yet Another Robot Simulator" (YARS) (Zahedi *et al.* 2008) to simulate our leg-wheel hybrid robot (Fig. 1(a)-(b)). It provides a defined set of geometries, joints, motors and sensors which are adequate to create the robot in a virtual environment. The robot model (Fig. 1(c)-(d)) is qualitatively consistent with the real one (Chadil *et al.* 2011) in the aspect of geometry, mass distribution, motor torque/speed, and sensors. The robot is generally designed based on the concept of a spherical form where its three identical legs, each attached with an omnidirectional wheel at its end, are kept inside its shells (body) in order to perform passive rolling motion (Chadil *et al.* 2011). This form (called dormant mode, Fig. 1(a)) provides the compact shape of the robot. For active locomotion, it will transform into two hemispherical shells where the wheeled legs are projected out of the shells (called transformed mode, Fig. 1(b)). In this study, we consider the robot locomotion only in the transformed mode. Describing a controller for locomotion in the spherical mode (Shu *et al.* 2009) will go beyond the scope of this work.

All in all the robot has seven degrees of freedom driven by DC motors: one active middle joint for the transformation process and two active joints of each leg where each of them moves the leg up and down and drives the wheel. In addition to the robot mechanical feature, we also simulate six infrared proximity sensors ($IR_{1,2,3,4,5,6}$, Fig. 1(d)) to generate obstacle avoidance behavior.

Here, we add a Gaussian-distributed noise with a standard deviation of, e.g., 5% to each sensor value. The simulated robot with its virtual environment is shown in Figs. 1(c)-(d). This simulation environment is embedded in the Integrated Structure Evolution Environment (ISEE) (Hülse *et al.* 2004). The ISEE is a software platform consisting of the evolution program EvoSun, the execution program Hinton and the simulators, e.g., YARS (see (Manoonpong 2007) for the scheme of the ISEE). Hinton allows constructing and analyzing neural control while EvoSun can optimize it using a special evolutionary algorithm, the ENS^3 (evolution of neural systems by stochastic synthesis, see (Hülse *et al.* 2004) for more details). In this study, we implement neural control on Hinton. Since the evolution program is part of ISEE, it is straightforward to optimize the (recurrent) neural parameters, if required.

3. Biologically inspired modular neural control

To control the active locomotion of the leg-wheel hybrid robot, we employ modular neural control. While different methods can be applied for locomotion generation, this modular neural control is selected in order to provide a basic control structure to the system where an online neural learning mechanism (Manoonpong *et al.* 2013) or an evolutionary algorithm (Hülse *et al.* 2004) for parameter adaptation could be later applied to obtain adaptive and robust behaviors. Moreover, a modular approach is able to deal with transferring and scaling issues; i.e., applying to different robots (Hornby *et al.* 2005) or when more degrees of freedom are added (Valsalam and Miikkulainen 2009, Valsalam and Miikkulainen 2008). We also discuss a major advantage of using a modular approach in more details in the Discussion section.

Here it is used to generate various locomotion patterns (e.g., omnidirectional motion including sidestepping) as well as a reactive obstacle avoidance behavior. The modular neural control consists of four main modules or networks: a minimal recurrent control (MRC) network, a velocity regulating network (VRN), a neural oscillator network (abbreviated CPG, see below), and a phase switching network (PSN). The MRC network is for sensory signal processing and directly drives two front wheels ($M_{left.\ wheel}$, $M_{right.\ wheel}$, Fig. 2) while a rear wheel ($M_{rear.\ wheel}$, Fig. 2) is indirectly controlled through the VRN module. In parallel, the neural oscillator network serves as a central pattern generator (CPG) (Ijspeert 2008). It produces the basic rhythmic signals for driving leg movements ($M_{left.\ lea}$, $M_{riath.\ lea}$, Fig. 2) to obtain, e.g., sidestepping while the stepping directions are controlled through the PSN module. All modules are described in detail in the following sections. The complete structure of this modular controller and the location of the corresponding motor neurons are shown in Fig. 2.

All neurons of the network are modelled as discrete-time non-spiking neurons. The state and output of each neuron are governed by Eqs. (1)-(2), respectively

$$a_i(t+1) = \sum_{j=1}^{n} w_{ij}\, o_j(t) + b_i \quad i = 1, \ldots, n, \tag{1}$$

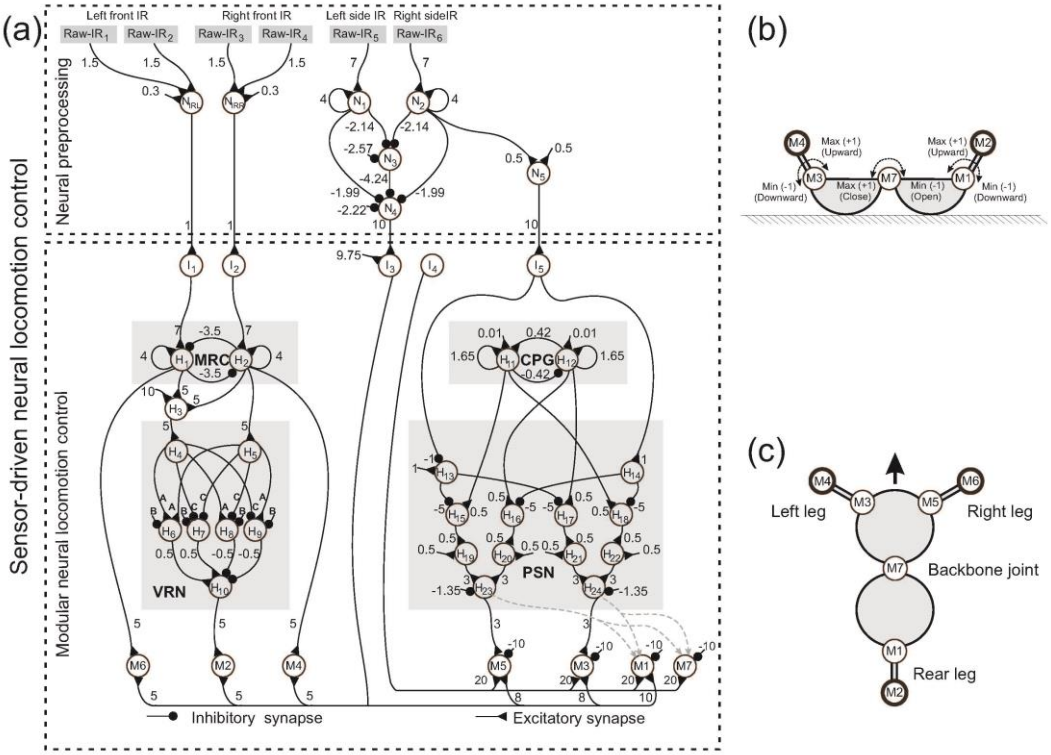

Fig. 2 (a) The sensor-driven neural locomotion control of the leg-wheel hybrid robot consists of two components: Neural sensory preprocessing and modular neural locomotion control. Each of them has their own input neurons (i.e., $IR_1, \cdots, IR_6$ and $I_1, \cdots, I_5$) modelled as linear buffers while other remaining neurons are modelled with respect to Eqs. (1)-(2). The modular neural control has three different neuron groups: input, hidden, and output. Input neurons *I* receive sensory signals. Hidden neurons *H* are divided into four modules (MRC, VRN, CPG, and PSN) having different functionalities (see text for details). Output neurons are described as motor neurons ($M_{1,\ldots,7}$). All connection strengths together with bias terms are indicated by the small numbers except some parameters of the VRN given by $A = 1.7246$, $B = -2.48285$, $C = -1.7246$. Dashed arrows indicate additional synapses which can be added to obtain more locomotion behaviors. (b) The movements of the leg joints ($M_{1,3}$) and the body joint (M_7). The right leg joint (M_5) having the same movement as the left one (M_3) is omitted. (c) The location of the motor neurons on the robot. For clarity and trackable in text, indexing of motor neurons is used as: $M_1 = M_{rear,leg}$, $M_2 = M_{rear,wheel}$, $M_3 = M_{left,\ leg}$, $M_4 = M_{left,\ wheel}$, $M_5 = M_{right,\ leg}$, $M_6 = M_{right,\ wheel}$, $M_7 = M_{body}$

$$o_i = \tanh(a_i) = \frac{2}{1 + e^{-2a_i}} - 1, \tag{2}$$

where n denotes the number of units, b_i represents a fixed internal bias term together with a stationary input to neuron i, a_i their activity, w_{ij} the synaptic strength of the connection from neuron j to neuron i, and o_i the neuron output. Input neurons are here configured as linear buffers ($a_i = o_i$). The entire network is constructed, experimented, and analyzed through the ISEE.

3.1 Wheeled locomotion control

In this section, we describe the first network group used to control three wheels ($M_{rear,wheel}$, $M_{left,wheel}$, $M_{right,wheel}$, Fig. 2. It consists of the minimal recurrent control (MRC) network and the velocity regulating network (VRN).

3.1.1 Minimal recurrent control (MRC)

The MRC has been originally evolved through the evolutionary algorithm ENS^3, integrated into the ISEE (see (Hülse *et al.* 2004) for more details), for generating obstacle avoidance behavior of a miniature Khepera robot (Hülse *et al.* 2004), which is a two wheeled platform. The fitness function was simply given as: For a given time go straight ahead as long and as fast as possible (see (Pasemann *et al.* 2003b) for more details). The result of which shows that the network consisting of two mutually inhibiting neurons with self-connection is sufficient for solving the task (Fig. 2(a)). The MRC has been formulated as the dynamical neural Schmitt trigger by Hülse and Pasemann (Hülse and Pasemann 2002) such that one can manually modify the connection parameters to obtain appropriate obstacle avoidance behavior for specific properties of different robot platforms and environments.

Here, we apply it to directly drive the two front wheels ($M_{left,wheel}$, $M_{right,wheel}$, Fig. 2) of our robot. It basically serves for driving robot motion and controlling the turning directions of the robot to avoid obstacles and to escape from a corner and even a deadlock situation. We here empirically adjusted the connection weights of the network for our robot (see Supplementary Information for more details of weight adjustment). The resulting weights are shown in (Fig. 2(a)). Using these weights, the network exhibits hysteresis effects (Supplementary Fig. 1) which guarantee 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.

We use four infrared (IR) sensor signals (two for each side, $IR_{1,2}$ and $IR_{3,4}$ (Fig. 1(d))) for obstacle detection at its front. They are transmitted to the inputs $I_{1,2}$ of the network (Fig. 2). The sensor signals are mapped onto the interval $[-1, +1]$, with -1 representing no obstacles, and $+1$ representing near obstacles. I_1 corresponds to an approximate mean value of the two left IR sensor signals and I_2 to that of the two right ones.

Applying the output signals of H_1 and H_2 directly to their target motor neurons $M_{left,wheel}$, $M_{right,wheel}$ and indirectly to the motor neuron $M_{rear,wheel}$ (Fig. 2) via the VRN (described below), the robot motion can be (autonomously) switched; for instance, switching from moving forward to turning left when there are obstacles on the right, and vice versa. The network outputs also determine in which direction the robot should turn in deadlock situations depending on which sensor side has been previously active. In a special situation, like moving toward a wall or a corner, I_1 and I_2 (Supplementary Fig. 1(a)) would have a value around 1.0 at the same time. As a consequence, H_1 and H_2 would then have a value of around 1.0; thereby the robot will move backward. During moving backward, the activation of the sensory signal of one side might be still active while the other might be inactive. Correspondingly, the robot will turn into the opposite direction of the active signal and it can finally leave from the wall or the corner.

3.1.2 Velocity regulating network (VRN)

In general, as one locomotion mode, one could use only the two front wheels ($M_{left,wheel}$,

$M_{right,wheel}$, Fig. 2) while the rear wheel ($M_{rear,wheel}$, Fig. 2) is inhibited and lifted above ground by the rear leg ($M_{rear,leg}$, Fig. 2(b)) during turning. However, to obtain more effective turning behavior we control the rear wheel with respect to turning motion driven by $M_{left,wheel}$, $M_{right,wheel}$. The rear wheel $M_{rear,wheel}$ is indirectly driven through the VRN which changes its rotational direction. The VRN is a simple feed-forward neural network with two input, four hidden, and one output neurons (Fig. 2(a)). It was trained by using the backpropagation algorithm. It works as a multiplication operator (Supplementary Fig. 2). Since it was developed and already presented in our previous work (Manoonpong *et al.* 2007a), we do not explain its detail here. The detail of the network development is referred to (Manoonpong *et al.* 2007a) and also Supplementary Information.

Here, we directly apply the VRN for our task. It is integrated into the modular neural locomotion control (Fig. 2(a)) where the neuron H_3 is added to combine both outputs of the MRC and project to the input neuron H_4 of the VRN. Another input neuron H_5 of the VRN receives its input from the neuron H_2 of the MRC. As a result, the VRN drives the rear wheel with low ≈ -1 activation to rotate clockwise leading to a right turn when there are obstacles on the left (i.e., the output of H_1 is $\approx +1$ while the output of H_2 is ≈ -1) and vice versa. In case no obstacle is detected (i.e., the output of $H_{1,2}$ is ≈ -1), the rear wheel will be inactive (zero activation) leading to forward motion of the robot. Forward motion will be achieved due to the properties of the omnidirectional wheel. The summary of the wheeled locomotion driven by the inputs $I_{1,2}$ is shown in Table 1. The motor neurons of the wheels are driven by binary values since we consider here only rotational directions of the wheels. However, one could add additional sensors and the VRNs for speed regulation of the wheels.

3.2 Legged locomotion control

In this section, we describe the second network group used to allow the robot to perform legged locomotion. The network consists of the neural oscillator network serving as a central pattern generator (CPG) to generate periodic movements and the phase switching network (PSN) to control the sidestepping directions (i.e., left and right). According to the robot configuration, if the front legs ($M_{left,leg}$, $M_{right,leg}$, Fig. 2) periodically move with phase difference $\pi/2$ to each other, we automatically obtain sidestepping (Fig. 3). We do not use any planning to lift a leg during sidestepping here. The left and right legs basically follow the CPG signals having phase difference $\pi/2$ to each other (see below). There is also no balancing mechanism used here. The robot basically uses its rear body part as a supporter (i.e., this body part always touches the ground) during leg movement. It is important to note that forward and backward stepping motions are not possible with the current robot setup. To achieve forward and backward stepping motions, we need to add other active shoulder joints that can move the legs forward and backward (Manoonpong 2007).

3.2.1 Neural oscillator network

The concept of CPGs for legged locomotion has been studied and used in several robotic systems particular in walking robots (Ijspeert 2008). Various CPG models have been proposed, such as nonlinear oscillators (Ijspeert 2008) and continuous-time neural oscillators (e.g., Matsuoka model (Matsuoka 1985), Terman-Wang model (Terman and Wang 1995), Wilson-Cowan model (Wilson and Cowan 1972)).

Here the model of a CPG is realized by using the discrete-time dynamics of a simple 2-neuron

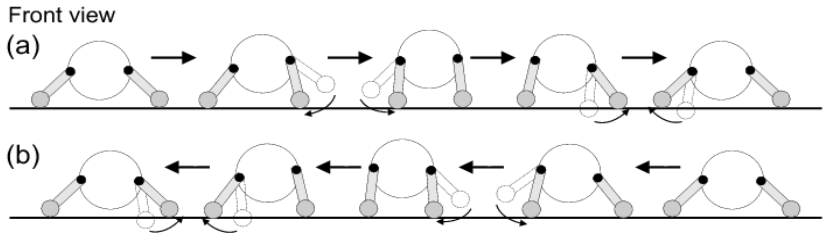

Fig. 3 Simple animation showing a sequence of leg movements leading to the stepping motions to the left (a) and the right (b) with respect to the robot view. Note that during stepping the rear leg of the robot is lifted such that its rear hemispherical shell is on the ground while its front part is kept above the ground. By doing so, its rear leg will not resist sideways motion

Table 1 Control parameters for the different motions [a]

Motions	H_1	H_2	$M_{rear,wheel}$	$M_{left,wheel}$	$M_{right,wheel}$
Forward	≈ -1.0	≈ -1.0	0.0	≈ -1.0	≈ -1.0
Turn left	≈ -1.0	$\approx +1.0$	$\approx +1.0$	$\approx +1.0$	≈ -1.0
Turn right	$\approx +1.0$	≈ -1.0	≈ -1.0	≈ -1.0	$\approx +1.0$
Turn left [*]	$\approx +1.0$	$\approx +1.0$	$\approx +1.0$	$\approx +1.0$	$\approx +1.0$

[a] The robot will move *forward* when there are no obstacles on the left and right sides; such that H_1 and H_2 show low ≈ -1 activation. It will *turn left* when there is an obstacle on its right; such that H_1 shows low ≈ -1 activation while H_2 shows high $\approx +1$ activation. It will *turn right* when there is an obstacle on its left; such that H_1 shows high $\approx +1$ activation while H_2 shows low ≈ -1 activation. In a special case, if there are obstacles on left and right sides making H_1 and H_2 show high $\approx +1$ activation, it will then *turn left*[*] due to the added hidden neuron H_3 of the network. This will allow the robot to effectively avoid obstacles and escape from corner and deadlock situations. We intuitively set to turn left in this special situation. However, one could also modify the network such that the rear wheel turns right in this situation. In general, without rotating the rear wheel the robot can move to left or right with respect to the rotation of the front wheels. However, the rear wheel would produce a small resistance resulting in slightly difficult to turn. In addition, rotating this rear wheel will allow the robot to perform better turning and can simply avoid obstacles without the resistance of the rear wheel when it does not rotate or rotates passively.

network presented by Pasemann *et al.* (2003a). The network consists of two neurons with full connectivity (Fig. 2(a)) and additional biases. Its weight matrix W is an element in the special orthogonal group which is associated with a rotation in the plane and represented by functions of the rotation angle φ. The weight matrix is given by Eq. (3) (compare Fig. 2(a))

$$\mathbf{W} = \begin{pmatrix} w_{H_{11}H_{11}} & w_{H_{11}H_{12}} \\ w_{H_{12}H_{11}} & w_{H_{12}H_{12}} \end{pmatrix} = \alpha \cdot \begin{pmatrix} cos(\varphi) & sin(\varphi) \\ -sin(\varphi) & cos(\varphi) \end{pmatrix}. \tag{3}$$

The parameters (φ, α) need to be selected in accordance with the dynamics of the system staying near the Neimark-Sacker bifurcation set where quasi-periodic attractors occur (Pasemann *et al.* 2003a). As a consequence, the network can generate almost sine-shaped waveforms with α

$\approx +1$. Increasing α will increase amplitude but slightly distort the waveforms. The frequency of the oscillations depends crucially on $\varphi \in [-\pi, \pi]$. We first set α to a high value, e.g., 1.7, to obtain a high amplitude and an appropriate waveform while we empirically adjust φ using the simulation to achieve an appropriate frequency for generating sidestepping of the robot. As a result, it is set to 0.25 which is appropriate for our purposes here (see Supplementary Information for more detail of the network).

3.2.2 Phase switching network (PSN)

To steer the sidestepping directions (i.e., lateral motions to the left and right), one possibility is to reverse the phase of the periodic signals driving the motors ($M_{left,leg}$, $M_{right,leg}$, Fig. 2). That is, these periodic signals can be switched to lead or lag behind each other by $\pi/2$ in phase depending on the given input I_5.

To do so, we apply the PSN developed in our previous study (Manoonpong *et al.* 2008). The PSN is a hand-designed feedforward network consisting of four hierarchical layers with 12 neurons. 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. The detail of the network development is referred to (Manoonpong *et al.* 2008) and Supplementary Information.

Here, the PSN receives a binary input from the neuron I_5 (i.e., binary neuron) which can be set manually or driven by and infrared sensor signal. Simultaneously, it also receives continuous periodic inputs from the neurons $H_{11,12}$ (i.e., continuous neurons governed by Eqs. (1)-(2)) of the neural oscillator. And it finally provides continuous periodic outputs at its output neurons $H_{23,24}$. All other neurons of the PSN are also continuous neurons. In fact, the network switches the phase of the two sinusoidal signals originally coming from the neural oscillator network when I_5 is changed from 0 to 1 and vice versa (see section below). By applying this network property, the movements of the left and right legs will be reversed corresponding to the modification of I_5. Consequently, the robot will change its sidestepping directions from the right to the left and vice versa. The summary of the legged locomotion driven by the input I_5 is shown in Table 2.

In order to control the sidestepping via sensory signals, i.e., here using the infrared sensors $IR_{5,6}$ (Fig. 1(d)), for, e.g., obstacle avoidance, we add another network designed as an XOR gate (Fig. 4(a)). This neural network has two input neurons, one hidden neuron, and one output neuron. All neurons are modelled as a standard additive neuron with the sigmoidal transfer function according

Table 2 Control parameters for sidestepping directions[b]

Actions	I_3	I_4	I_5
Sideways left	1.0	0.0	1.0
Sideways right	1.0	0.0	0.0

[b] Note that I_3 is set to 1.0 for legged locomotion in order to stop rotating motion of the wheels while the robot performs sidestepping. Otherwise I_3 is set to 0.0 for wheeled locomotion. In other words, I_3 is used to select between wheeled and legged locomotion. I_4 is used to control the robot to be the dormant mode or the transformed mode. The robot will be in the dormant mode (Fig. 1(a)) when I_4 is set to 1.0 while it will be in the transformed mode (Fig. 1(b)) when I_4 is set to 0.0.

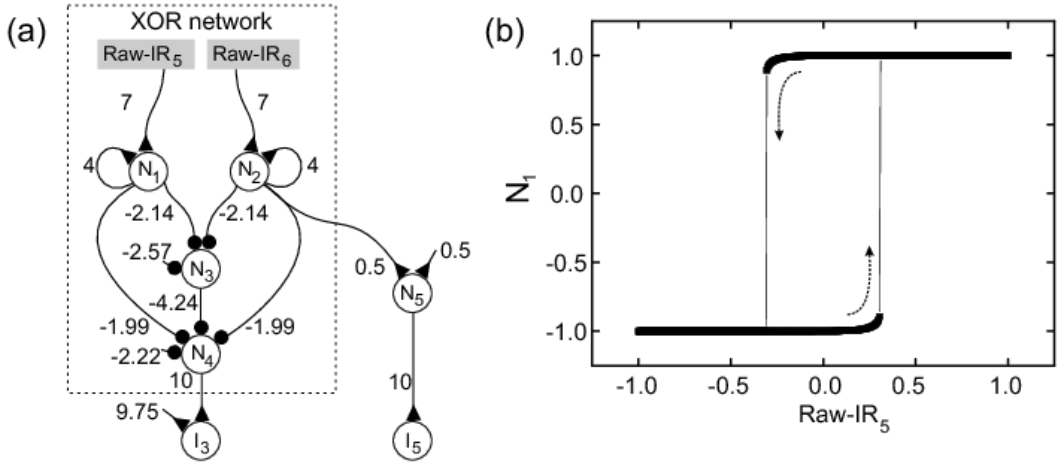

Fig. 4 (a) Neural preprocessing network of the IR signals $I_{5,6}$ (dashed frame). It is created as an XOR network with a self-connection at its input neurons. As a result, the input neurons function as hysteresis elements. The output of the network N_4 is fed to I_3 and the output of one hysteresis element N_2 is indirectly connected to I_5 via the additional hidden neuron N_5. (b) The hysteresis effect between the input and output of N_1. The input IR_5 varies between ≈ -1.0 and $\approx +1.0$ while its output shows high $\approx +1.0$ activation when the input increases to values above ≈ 0.3. On the other hand, it will show low ≈ -1.0 activation when the input decreases below ≈ -0.3

to Eq. (2). The network was trained by using the backpropagation algorithm (Rumelhart *et al.* 1980). Here we use the XOR network instead of an XNOR module which was developed for sideward walking of six- and eight-legged robots (Manoonpong *et al.* 2008) since the XOR network provides proper function for this task, i.e., activating sidestepping of the leg-wheel hybrid robot. Note that the widely used version of an XOR network with the standard sigmoidal transfer function having the output range of $[0,\cdots,1]$ is not used here since we want to keep all hidden and motor neurons with the same transfer function (i.e., tanh) for simplicity. The look up table of an XOR gate is also not considered here for consistence reasons and we want to keep the complete controller as one neural circuit.

The IR signals ($IR_{5,6}$) are linearly mapped onto the interval $[-1, +1]$, with -1 representing no obstacles, and $+1$ representing near obstacles. They are provided to the input neurons of the network. However, these sensory signals have to be first filtered before feeding them to the XOR network. Therefore we again apply the hysteresis effect of the recurrent neural network to eliminate sensory noise. Thus, the input neurons of the XOR network are configured as the hysteresis elements. Each of them has input and recurrent weights similar to the MRC network. The hysteresis effect of these single recurrent neurons is shown in Fig. 4(b) and the complete XOR network for sensory preprocessing together with its weights is shown in Fig. 4(a). The output N_4 of the network corresponding to the given inputs (i.e., the filtered IR signals $N_{1,2}$) is presented in Table 3.

We directly feed the output of N_4 to its target neuron I_3 in the neural locomotion control. Note that we add here a bias term at I_3 in order to scale the input signal to the range between 0.0 and 1.0. As a consequence, the sidestepping will be activated when N_4 gets high activation $\approx +1.0$

Table 3 The input-output characteristic for the XOR network

N_1	N_2	N_4
$\approx +1.0$	$\approx +1.0$	≈ -1.0
$\approx +1.0$	≈ -1.0	$\approx +1.0$
≈ -1.0	$\approx +1.0$	$\approx +1.0$
≈ -1.0	≈ -1.0	≈ -1.0

meaning that one of the N_1 and N_2 driven by the IR signals ($IR_{5,6}$) shows a high output signal $\approx +1.0$ (compare Table 3). Furthermore, the output signal of the hysteresis element N_2 serves to control the lateral direction through I_5 of the neural locomotion controller. The neuron N_5 is added to also scale the signal to a range between 0.0 and 1.0.

Once the sidestepping pattern is activated, the robot will step laterally to the right as long as the N_2 signal shows low ≈ -1 activation; otherwise it will step laterally to the left (compare Tables 2-3). Hence, the robot will perform sidestepping to the right if it detects an obstacle at its left side via the IR_5 sensor and vice versa. In special conditions, e.g., detecting obstacles on both lateral sides during moving forward, the IR_5 and IR_6 sensors will give high output activations at the same time resulting in the inhibition of sideways motions. The robot then continues to move forward using its wheels.

4. Experiments and results

In this section, five experiments demonstrating the robot behavior under the neural control (Fig. 2) are described. It is implemented on the ISEE. These experiments present locomotion behaviors of the robot using legs and wheels including its reactive obstacle avoidance behavior in the physics simulator (YARS). Here we report data acquired during various (reactive) locomotion behaviors. Video clips of these can be found at http://www.manoonpong.com/HybridRobot/. It is important to note that in all experiments the robot was set to the transformed mode. That is the input I_4 (Fig. 2) was set to 0 resulting in the motor of the body joint M_{body} being driven by low ≈ -1 activation (cf. Fig. 2(b)).

4.1 Wheeled and legged locomotion

The first and second experiments have been done to show wheeled and legged locomotion behaviors. In these experiments, all input neurons ($I_{1,2,3,4,5}$, Fig. 1) were set manually[1]. While the inputs $I_{1,2}$ were set to -1 and the input $I_4$was set to 0 (i.e., the transformed mode with forward motion), the inputs $I_{3,5}$ were regulated. In the first experiment, we let the robot move over flat terrain and continuously changed inputs $I_{3,5}$ to investigate its basic locomotion. As a consequence, by simply controlling the input I_3 (cf. Fig. 2) the robot can quickly change its locomotion from using wheels to legs and vice versa. During legged locomotion, changing the input parameter I_5

[1]Note that we do not apply infrared sensor signals to the input neurons ($I_{1,2,3,4,5}$, Fig. 1) in order to clearly see the robot locomotion behaviors.

from zero to one leads to sidestepping to its left and changing I_5 back to zero leads to sidestepping to its right. These behaviors have been carried out sequentially and with continuous transitions. The robot moves using its wheels with a speed of ≈ 10 cm/s while it performs sidestepping using its legs with a speed of ≈ 6 cm/s. The input parameters and motor signals during the experiment are shown in Fig. 5. The video clips of this experiment showing forward motion using wheels and sidestepping using legs can be found at

http://www.manoonpong.com/HybridRobot/Forward.mpg,
http://www.manoonpong.com/HybridRobot/SidewaysRight.mpg, and
http://www.manoonpong.com/HybridRobot/SidewaysLeft.mpg, respectively.

4.2 Escape behavior using wheeled and legged locomotion

In the second experiment, the robot was steered through flat terrain with obstacles having a height of $\approx 70\%$ of a wheel radius (≈ 2.0 cm). This is the highest climbable obstacle. Here, at the

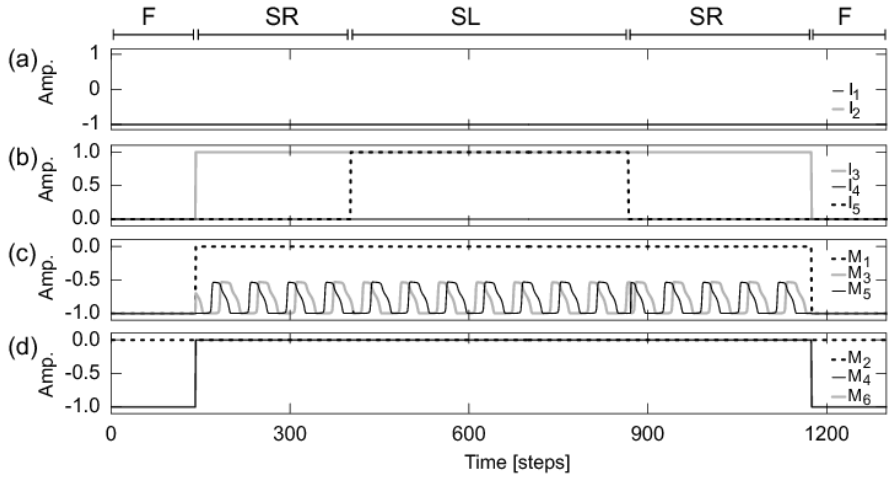

Fig. 5 Input parameters and motor signals during the first experiment. (a) Input parameters $I_{1,2}$ (Fig. 2) were set to -1 at all times in order to inhibit turning motions thus resulting in only forward motion. (b) Input parameters $I_{3,4,5}$ (Fig. 2). I_3 is used to switch between wheeled (I_3 = 0) and legged (I_3 = 1) locomotion. I_4 was here set to 0 in order to keep the robot in the transformed mode (Fig. 1(b)). Setting I_4 to 1 leads to the dormant mode (Fig. 1(a)). I_5 is used to steer the sidestepping directions. Setting I_5 to 0 leads to the sidestepping to the right SR and setting it to 1 leads to the left SL. (c) Motor signals at the leg joints ($M_{rear,leg}$, $M_{left,leg}$, $M_{right,leg}$, (Fig. 2)). The motors $M_{left,leg}$, $M_{right,leg}$ show the periodic signals when the legged locomotion mode is activated. One can observe that when the robot steps sideways to its right the periodic signal of $M_{right,leg}$ leads the one of $M_{left,leg}$ by $\pi/2$ in phase and vice versa when the robot steps sideways to its left. (d) Motor signals at the wheels ($M_{rear,wheel}$, $M_{left,wheel}$, $M_{right,wheel}$, (Fig. 2)). Low ≈ -1 activation drives the wheels in a way that the robot moves forward F while zero activation means the wheels have no motion (i.e., they will roll freely in the direction of their axis (cf. Fig. 1(b)). Indexing of motor neurons is used as: $M_1 = M_{rear,leg}$, $M_2 = M_{rear,wheel}$, $M_3 = M_{left,leg}$, $M_4 = M_{left,wheel}$, $M_5 = M_{right,leg}$, $M_6 = M_{right,wheel}$

beginning the robot moved forward using wheeled locomotion. As soon as it got stuck; we set the input I_3 to high $\approx +1$ activation in order to enable legged locomotion. As a result, the legged locomotion allows the robot to climb over obstacles obstructing its path and thereby enhances its mobility. A series of photos of this experiment is shown in Fig. 6. The video clip of this experiment can be found at
http://www.manoonpong.com/HybridRobot/ClimbingOverObstacle.mpg. Note that the limitation of climbing over higher obstacles (> 2.0 cm) is because of the physical constraints of the robot which are: 1) motor torque and 2) one degree of freedom legs. Due to these constraints, the stepping behavior of the robot cannot generate strong drag force to propel the robot body over higher obstacles. In addition, since no balance control was integrated in this current controller, its rear body part always touches the ground during climbing; thereby producing additional resistance.

4.3 Obstacle avoidance behavior using wheeled locomotion

The third and fourth experiments have been performed to assess the ability of the neural

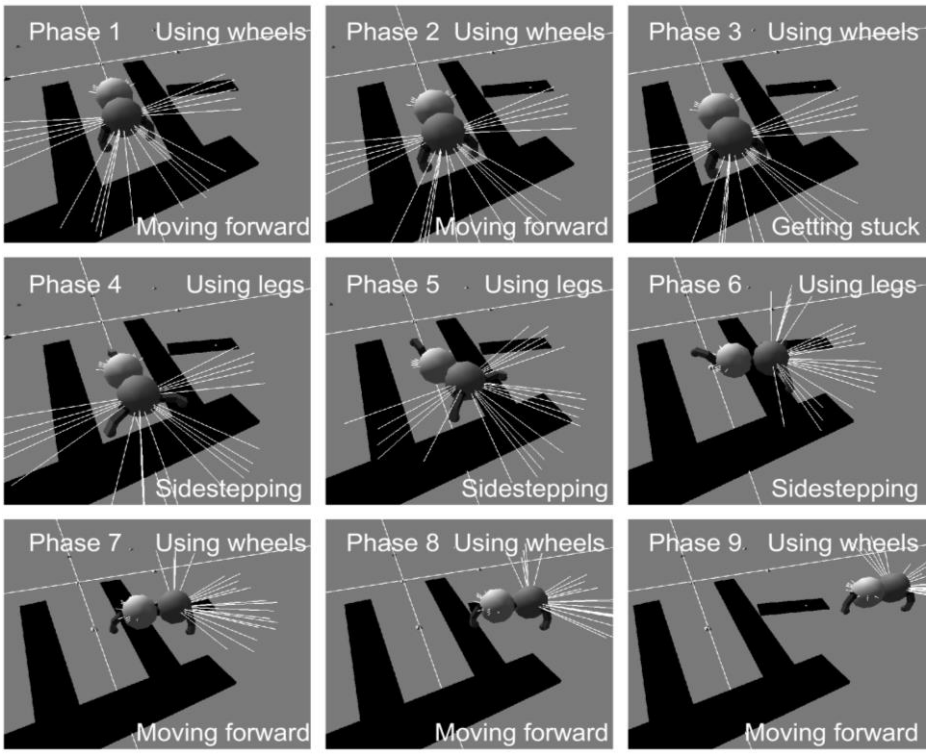

Fig. 6 Escape behavior of the robot using the wheeled and legged locomotion in the second experiment. Input parameters and motor signals are comparable to Fig. 5. Phases 1-2: the robot was set to the wheeled locomotion mode ($I_3 = 0$) for moving forward. Phase 3: it got stuck. Phases 4-6: it was manually set to the legged locomotion mode ($I_3 = 1$) for sidestepping to its left. It then climbed over obstacles. Phases 7-9: it was manually set to return to the wheeled locomotion mode ($I_3 = 0$) for again moving forward. As a result, it can escape from an obstacle area

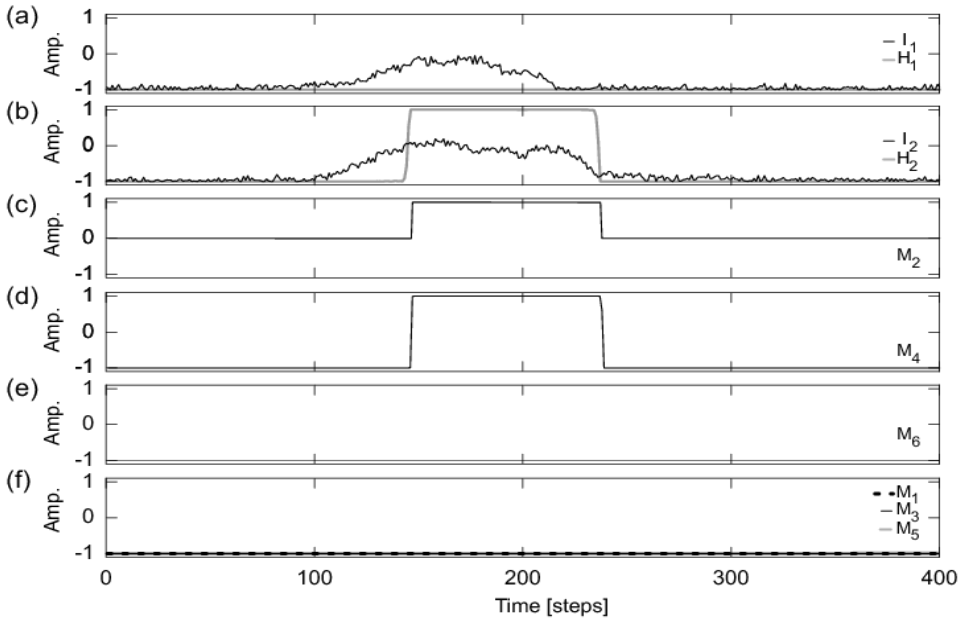

Fig. 7 Sensor and motor signals during the third experiment. (a), (b) Approximate mean values of the two left I_1 and right I_2 IR sensor signals before processing by the MRC network and the output signals $H_{1,2}$ after processing (cf. Fig. 2). (c)-(f) The motor neuron signals (cf. Fig. 2(c)). Indexing of motor neurons is used as: $M_1 = M_{rear,leg}$, $M_2 = M_{rear,wheel}$, $M_3 = M_{left,leg}$, $M_4 = M_{left,wheel}$, $M_5 = M_{right,leg}$, $M_6 = M_{right,wheel}$

controller (Fig. 2) generating obstacle avoidance behavior. Here wheeled locomotion was activated while the legged locomotion was inhibited and the robot moved through flat terrain with very high obstacles. Fig. 7 shows the sensor and motor signals during avoiding obstacles and escaping from a sharp corner and Fig. 8 displays a series of photos according to these signals.

It can be seen that the robot moved forward at the beginning. During moving forward, the motors of two front wheels $M_{left,wheel}$, $M_{right,wheel}$, were driven by low ≈ -1 activation while the motor of a rear wheel $M_{rear,wheel}$ was inactive (i.e., having zero activation). In this case the motors of leg joints $M_{rear,leg}$, $M_{left,leg}$, $M_{right,leg}$ were inhibited with low ≈ -1 activation to stay in the downward position (cf. Fig. 2(b)). After around 105 time steps, the robot encountered the corner and I_2 gradually activated to a high level. At around 10 time steps later, I_1 activated showing a pattern similar to I_2. As I_2 strongly activated, first H_2 became activated such that it then inhibited H_1. As a consequence, $M_{rear,wheel}$ became activated (i.e., showing high $\approx +1$ activation) and $M_{left,wheel}$ changed its activation from low ≈ -1 to high $\approx +1$ (i.e., changing its rolling direction) making the robot turn left. The robot kept on turning left until around 235 time steps and then returned to normal forward motion. Fig. 9 shows the sensor and motor signals of another obstacle avoidance behavior while Fig. 10 displays a series of photos according to these signals.

In this situation the robot moved toward a wall and at around 170 time steps[2] it was very close to the wall thereby activating I_1 and I_2. However, I_1 got higher activation than I_2 which activated H_1

[2]The update frequency of the system is approximately 25 Hz. Thus 170 time steps are about 6.8 seconds.

and then inhibited H_2. As a result, $M_{rear,wheel}$ switched to the low activation and $M_{right,wheel}$ changed its activation from low to high (i.e., changing its rolling direction) leading to a right turn. After around 250 time steps, it returned to normal forward motion. At around 450 time steps, it again detected another wall where the signals developed in similar patterns to those of the previous wall encounter resulting in another right turn. Eventually, the robot was able to avoid the obstacles and continued to move forward. We encourage readers to also watch a video clip showing another obstacle avoidance behavior at

http://www.manoonpong.com/HybridRobot/ObstacleAvoidanceI.mpg.

4.4 Obstacle avoidance behavior using wheeled and legged locomotion

The last experiment uses the complete controller (Fig. 2) with all sensors to demonstrate the use of wheels and legs for a reactive obstacle avoidance behavior. In this case we use the infrared sensors $IR_{5.6}$ to allow the robot to detect obstacles at its lateral sides. The sensors will then drive sidestepping motions making the robot step away from the obstacles. Simultaneously the front IR sensors are still used to let it turn away from the obstacles using its wheels. Fig. 11 shows the sensor and motor signals of this experiment and Fig. 12 presents a series of photos according to these signals.

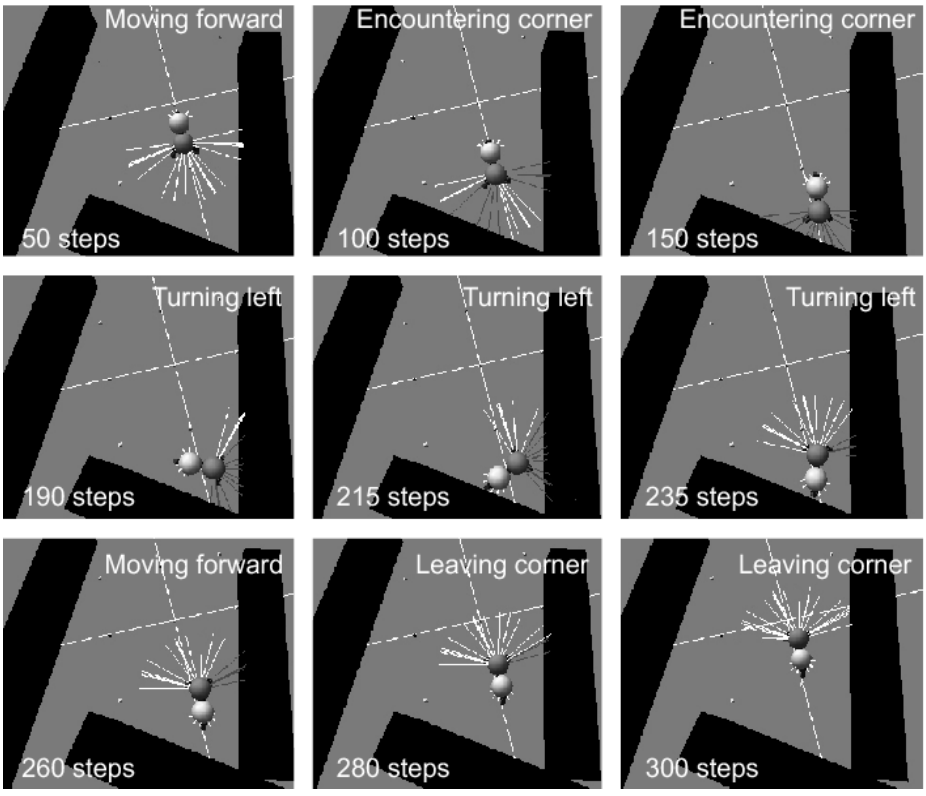

Fig. 8 The obstacle avoidance behavior of the robot with respect to the signals shown in Fig. 7

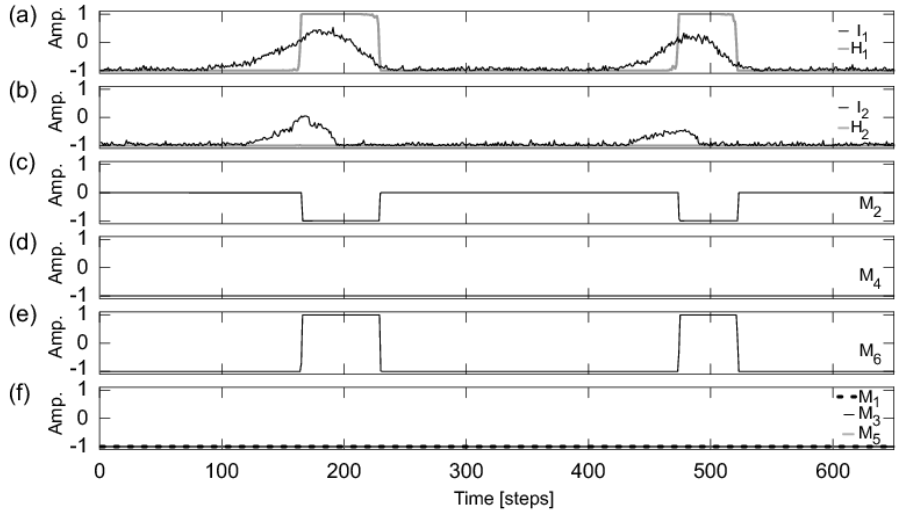

Fig. 9 Sensor and motor signals during the fourth experiment. (a), (b) Approximate mean values of the two left I_1 and right I_2 IR sensor signals before processing by the MRC network and the output signals $H_{1,2}$ after processing (cf. Fig. 2). (c)-(f) The motor neuron signals (cf. Fig. 2(c)). Indexing of motor neurons is used as: $M_1 = M_{rear,leg}$, $M_2 = M_{rear,wheel}$, $M_3 = M_{left,leg}$, $M_4 = M_{left,wheel}$, $M_5 = M_{right,leg}$, $M_6 = M_{right,wheel}$

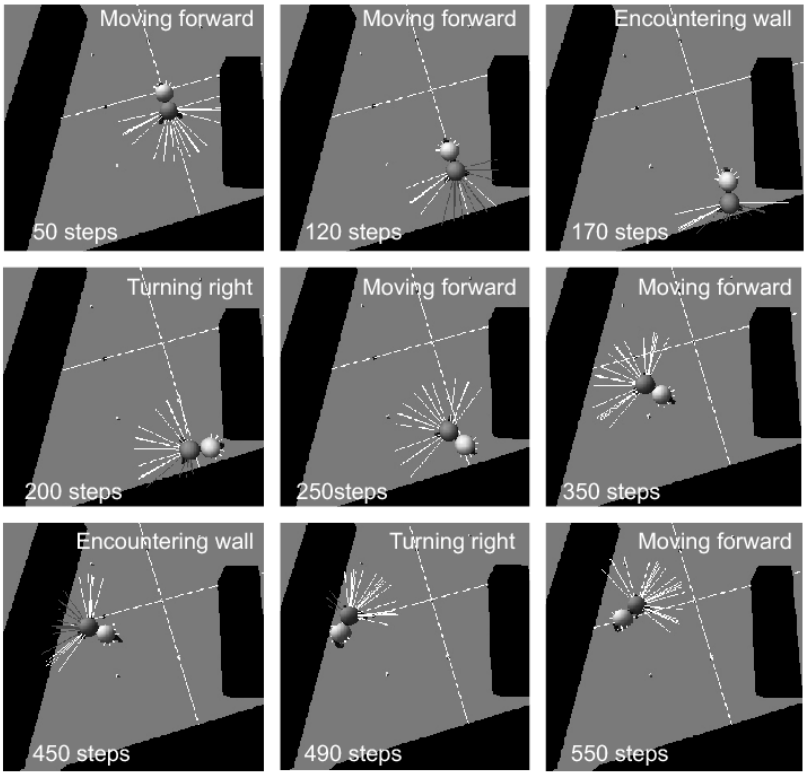

Fig. 10 The obstacle avoidance behavior of the robot with respect to the signals shown in Fig. 9

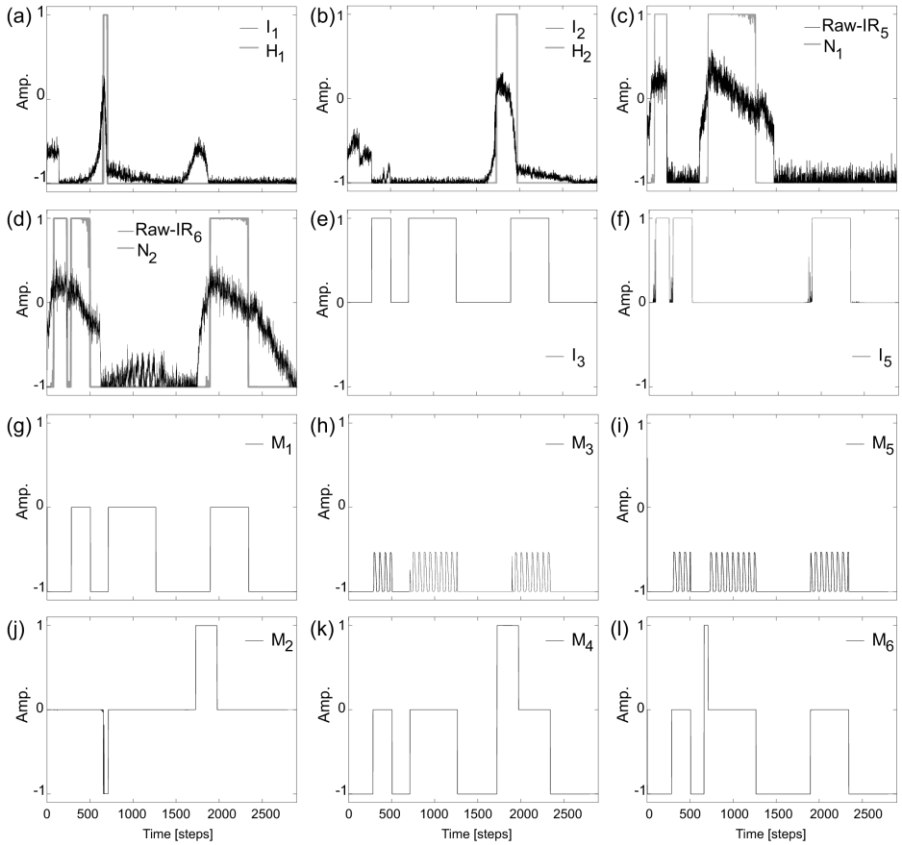

Fig. 11 Sensor and motor signals during the fifth experiment. (a), (b) Approximate mean values of the two front left I_1 and right I_2 IR sensor signals before processing by the MRC network and the output signals $H_{1,2}$ after processing (cf. Fig. 2). (c), (d) $IR_{5,6}$ signals before preprocessing and the output signals $N_{1,2}$ after preprocessing (compare Fig. 4). (e) The signal of the input parameter I_3 switching between wheeled (I_3 = 0) and legged (I_3 = 1) locomotion. It is driven by the output of the XOR network N_4. (f) The signal of the input parameter I_5 controlling the sidestepping directions to the right (I_5= 0) or the left (I_5 = 1). It is driven by the IR_6 indirectly through hidden neurons $N_{2,5}$ (cf. Fig. 4). (g)-(l) The motor neuron signals (cf. Fig. 2(c)). Indexing of motor neurons is used as: $M_1 = M_{rear,leg}$, $M_2 = M_{rear,wheel}$, $M_3 = M_{left,leg}$, $M_4 = M_{left,wheel}$, $M_5 = M_{right,leg}$, $M_6 = M_{right,wheel}$

At the beginning (around 100 time steps), since the robot detected obstacles on both lateral sides, it continued to move forward using its wheels due to the functionality of the XOR network described above. During moving forward, the motors of the two front wheels $M_{left,wheel}$, $M_{right,wheel}$ performed rolling motions while the rear wheel $M_{rear,wheel}$ was inactive. From around 300 to 500 time steps, the robot stepped to the left in order to avoid a lateral obstacle on its right. In this situation, the front leg joints $M_{left,leg}$, $M_{right,leg}$ performed periodic movements while the rear leg joint $M_{rear,leg}$ lifted the leg above ground. At around 620 time steps it was far enough from the obstacle; therefore, it returned to move forward using its wheels. At around 700 time steps, it detected an

obstacle on its left. Hence, it then turned right. During turning its left side IR sensor IR_5 increased ($\approx +1.0$) leading to sidestepping to the right at around 1000 time steps. Afterwards, it returned to normal forward motion. At around 1700 time steps, it approached another obstacle leading to turn left and then step sideways to its left. Finally, after avoiding the obstacle it returned to the forward motion at around 2400 time steps.

As demonstrated, the sensor-driven neural controller (Fig. 2) enables the robot to successfully solve the obstacle avoidance task and this controller also shows an example of how both locomotion modes could be driven by sensory signals. Additionally, the controller can even protect the robot from getting stuck in corners as shown in Figs. 8-10. Thus, due to this functionality, the robot can autonomously perform exploration. We encourage readers to see another demonstration at http://www.manoonpong.com/HybridRobot/ObstacleAvoidanceII.mpg.

5. Discussion

We simulated a leg-wheel hybrid robot using the physics simulation environment called YARS. The simulated robot is intended to be used to develop and test neural controllers before

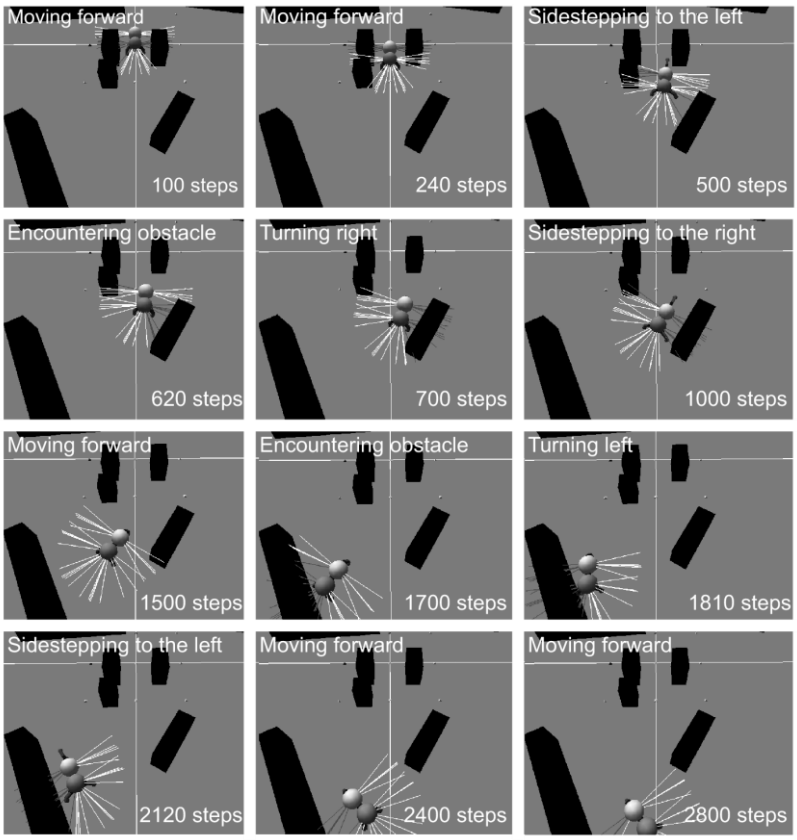

Fig. 12 The obstacle avoidance behavior of the robot with respect to the signals shown in Fig. 11

implementation on the real one (Chadil *et al.* 2011). The complete sensor-driven neural controller of the robot developed here enables it to avoid obstacles and escape from corners as well as deadlock situations. Thus, due to this functionality, the robot can autonomously perform exploration. This controller can also automatically select locomotion modes according to sensory signals. In general, the robot uses its wheels for basic locomotion and its legs to step away from obstacles driven by IR sensor signals. It can also use its legs to climb over small obstacles where in this study we manually control the robot for testing this function. However, later this climbing behavior will be driven by proprioceptive sensory information. In addition to the use of hybrid locomotion for climbing over obstacles (cf. Fig. 6) and obstacle avoidance (cf. Fig. 12), it is important to note that the robot could use its legs for movement when its wheels are broken; i.e., changing its locomotion from forward motion using wheels to sidestepping using legs (cf. Fig. 5).

This sensor-driven controller can be considered as a model-based control technique which was constructed with an artificial neural network using discrete-time dynamics. Part of it was developed by realizing dynamical properties of recurrent neural networks; i.e., hysteresis effects and quasi-periodic attractors. The controller is composed of two main components: the modular neural locomotion control and the neural sensory preprocessing networks. The modular neural locomotion control has four subnetworks (modules): a minimal recurrent control (MRC) network, a velocity regulating network (VRN), a neural oscillator network (CPG), and a phase switching network (PSN). It generates a wide range of locomotion patterns like omnidirectional motion including sidestepping using wheels and legs. The neural preprocessing networks serve to filter sensory noise as well as preprocess the sensory data where the network outputs appropriately change locomotion direction and/or speed through our premotor neuron modules (i.e., VRN and PSN). This results in a reactive behavior; i.e., obstacle avoidance. Nevertheless, controlling locomotion direction and speed can be also achieved by modifying neural parameters or control inputs projecting onto motor neurons (von Twickel *et al.* 2011).

In fact, our modular neural locomotion control exhibits some essential features: Neural and biological justifiable, modular, transferable, generic, and robust. These features distinguish it from conventional control methods (e.g., classical control (Besseron *et al.* 2005, Nakajima and Nakano 2008), AI control algorithms (Halme *et al.* 2001, Mahmoud *et al.* 2008), and biologically inspired control (Klavins *et al.* 2000, Eich *et al.* 2008, Allen *et al.* 2003)) which have been applied to other leg-wheel hybrid robots (as described in the Introduction section).

Neural and biological justifiable:

MRC: In the context of neural systems, hysteresis phenomena have already been discussed as models for short-term memory (Harth *et al.* 1970). These effects allow an agent to keep on doing a task till the task is completed even if the stimulus has decayed or is removed. Without such memory, the agent might switch between tasks reactively without completing any of them and, thus fail to complete tasks. In addition, hysteresis effects require relatively large changes of neural activity in order to switch between quasi-stable states; i.e., they are robust to small changes (e.g., noise) such that they can be also described as low-pass filters (Manoonpong 2007). From this point of view, the hysteresis effects allow robots to memorize their state and exhibit behavior robust against noise. These hysteresis effects are directly reproduced by our MRC module. In other words, the MRC module with the defined network parameters can be considered as simple short term memory and low-pass filters. Note that one can further optimize these network parameters for specific tasks, for instance, by using the evolutionary algorithm ENS^3 (Hülse *et al.* 2004). The

experiments here show that our MRC module with the resulting weights allows the robot to memorize its state such that it performs smooth motion and successfully avoids obstacles (see, e.g., Figs. 8-9 as well as http://www.manoonpong.com/HybridRobot/ObstacleAvoidanceI.mpg). In contrast, using simple finite state control (Chadil *et al.* 2011) or classical Braitenberg control (Braitenberg 1984) without state memory the robot needs to turn several times in order to avoid obstacles or it sometimes gets stuck (see http://www.manoonpong.com/ HybridRobot/ObstacleAvo idanceI-BC.mpg).

CPG: The basic locomotion and rhythm of stepping in walking animals mostly rely on central pattern generators (CPGs) (Ijspeert 2008, Büschges 2005). CPGs generate basic rhythmic outputs in the complete absence of any sensory feedback, and appear to underlie all types of rhythmic behavior. Although sensory feedback is not required for generating the basic rhythms, it has an important role in modulating and shaping the rhythmic patterns including switching their phase (Pearson and Iles 1973) for the production of appropriate motor behavior during locomotion like changing walking patterns and directions in order to escape from a predator or to avoid obstacles. Besides, evidences show that neural elements of CPGs are part of sensorimotor loop and functionally sensory signals might contribute to generation rhythmic activity depending on context (Daun *et al.* 2009). In this study, we implement the functionality of CPGs in part by a neural oscillator module (i.e., CPG module). The CPG module can generate basic rhythmic pattern without any sensory feedback. Its outputs allow the robot to perform stepping while the phase of the rhythmic pattern is controlled by a sensory input through the PSN module. It is important to note that here we exclude a sensorimotor loop for the module and use only one module for controlling all leg joints, rather than each module for each leg joint (see, e.g., (von Twickel *et al.* 2011) for simulations and (von Twickel *et al.* 2012) for robotic studies) as found in animals (Marder and Bucher 2001, Grillner 2006). However, this CPG module shows a minimal approach for robot control and if required, one could also optimize the parameters of the CPG network using, for instance, the evolutionary algorithm ENS^3 (see (Hülse *et al.* 2004) for more details) integrated into the ISEE. The simplest way to define the fitness function for this optimization process is to take the Euclidean distance from the start to the end point of the robot's trajectory.

PSN: There is strong evidence for a phase shifting property since around 1973 from the study of Pearson and Iles (1973), who have in the cockroach recorded from inter-segmental neurons in the connective elements. Phase relationships between these neurons can change as would be required for emulating the functionality of our PSN module.

VRN: Recent studies by Akay *et al.* (2007) show that in stick insect locomotion motorneuron pools are able to not only drive protractor (swing) and retractor (stance) muscle activities but also "reverse" their activities leading to the change of locomotion directions (e.g., from walking forward to backward and vice versa). This reversion is also influenced by sensory feedback like load signals from the leg. The functionality of these motorneuron pools is directly reproduced by our VRN module which controls and reserves motor signals. In principle, the VRN performs as a multiplication operator which can inverse motor signals as well as regulate their magnitude (Manoonpong 2007). Hence, our study predicts such a multiplicative function at the premotor interneurons of stick insects (Akay *et al.* 2007, Gabriel and Büschges 2007).

Modular:

From neuroethological studies in walking animals, it is known that the neural network of nonspiking interneurons for the locomotion system contributes to different functional "modules"

(Bässler and Büschges 1998, Delcomyn 1999). They govern the different leg joints resulting in walking or stepping behavior. According to Delcomyn (1999), insects exhibit a modular organization (i.e., modular structure) of locomotion control elements. Inspired by this finding, our neural locomotion control uses a modular structure where its modules (MRC, CPG, VRN, and PSN) also have a functional origin in biological neural systems (described above).

A modular structure, relevant to biological systems, is considered as a major advantage, compared to many other approaches due to the following aspects: 1) It is flexible, allowing to simply rearrange, add, and/or remove modules for controlling different types of robots (see Transferable). 2) Each module can be decoupled where its functioning still remains (see Generic). 3) It is robust and has fault tolerance capabilities. Damage to a part of the system can result in a loss of some of the abilities of the system, but, the whole system can still function partially (see Robust). 4) It is able to deal with scaling issue; i.e., when more degrees of freedom are added (Valsalam and Miikkulainen 2009, Valsalam and Miikkulainen 2008). 5) It is a way of embedding a priori knowledge in a neural network or providing basic functions (as shown in this study), which can integrate different neural functions (Manoonpong *et al.* 2007b), different neural structures (Manoonpong 2007) or different kinds of learning mechanisms (Steingrube *et al.* 2010, Manoonpong *et al.* 2013), depending on the task at hand. 6) The modules generally have a simpler structure as compared to the network as a whole. Thus, their functions and dynamics are analyzable by observing the input/output relationship of an individual module (see, e.g., hysteresis effects of the MRC module). In contrast, approaches using evolutionary algorithms (Yosinski *et al.* 2011, Parker and Lee 2003) or reservoir computing (Krause *et al.* 2010, Salmen and Ploeger 2005) with non modularity might end up with large networks which are difficult to understand or analyze their dynamics in particular if they use a massive recurrent connectivity structure. Removing some connections or neurons of the networks might result in instability or drastically reduce some functions of the system. Furthermore, for most of these networks it is difficult to transfer them successfully onto different robots without re-evolving or retraining.

Transferable:

The entire locomotion network consists of four main modules or subnetworks: 1) the MRC module, 2) the CPG module, 3) the PSN module, and 4) the VRN module. They have been so far successfully implemented on four-, six- and eight-legged robots as well as two wheeled robots (Manoonpong 2007, Manoonpong *et al.* 2008, Manoonpong and Roth 2008). Thus they are transferable. Applying to the different systems, the structures and internal parameters of the PSN and VRN modules normally remain unchanged. However, only the parameters of the CPG and MRC modules (i.e., synaptic weights) might be necessary to be adjusted in order to obtain suitable walking frequency and obstacle avoidance behavior, respectively.

Generic:

As shown in this paper, only very few components (MRC, CPG, PSN, VRN) are required to achieve a very rich, functionality (i.e., a wide range of locomotion patterns as well as a reactive obstacle avoidance behavior). As suggested by their names, the modules each serve a general purpose largely regardless of the robot's specific embodiment (see Transferable) and behavioral repertoire. For example, the PSN module can switch the phase of not only periodic signals shown here but also different forms like sawtooth signals which are generated by a chaotic CPG module as shown in (Steingrube *et al.* 2010). We believe that our neural modules can serve as useful building blocks (i.e., transferable and generalization) for other module-based neural control.

Robust:

The neural circuitry is not sensitive to changes of parameters and can be adjusted within large intervals making fine tuning unnecessary, e.g., synaptic weights between modules and synaptic weights projecting to motor neurons. Furthermore, synaptic connections can be completely cut or a module can be completely removed leading to graceful degradation of the agent's functionality while as a whole the system can still function partially. For instance, removing the CPG module the robot will not be able to perform sidestepping but it will still be able to locomote using its wheels and vice versa when the MRC module is removed while the CPG module is kept. Such situations can be considered from the experiment shown in Fig. 5. For example, during the first period of the experiment, the robot moved forward (F) using its wheels. In this case, it can be assumed that the CPG and PSN modules are completely removed. This way, the robot cannot perform legged locomotion where the motor neurons (M_1, M_3, M_5) controlling the leg joints have their activation of -1.0 due to their bias term. During the second period of the experiment, the robot stepped to the right (SR) using its legs. In this case, it can be assumed that the MRC and VRN modules are completely removed. This way, the robot cannot perform wheeled locomotion where the motor neurons (M_2, M_4, M_6) controlling the wheels have their default activation of 0.0. Besides this, due to the hysteresis loops in the MRC module, the system is insensitive to moderate changes of sensory signals, e.g., noise.

It is important to note that, although our controller is proper for leg-wheel hybrid robot behavior generation and exhibits essential features (described above), it has not so far 1) exploited sensorimotor loops on its low level motor control and 2) considered real biological data for locomotion generation as shown in (von Twickel *et al.* 2012). These two components will allow for robust locomotion under varying environmental conditions.

6. Conclusions

This article has presented various locomotion behaviors and a reactive obstacle avoidance behavior of a leg-wheel hybrid robot in the YARS physics simulation environment. The robot has a special mechanical design which consists of two (hemi) spherical body shells and three legs with omnidirectional wheels. It combines the idea of using legs, wheels, and rolling sphere for the multi-modal locomotion (one of locomotion modes will be activated at a time) which so far has not been shown by other researches.

In this study, active locomotion behaviors using wheels or legs are generated by biologically inspired modular neural control. It consists of four different functional modules having their origin in biological neural systems: a minimal recurrent control (MRC) network, a velocity regulating network (VRN), a neural oscillator network (CPG), and a phase switching network (PSN). The MRC module is for sensory signal processing and state memorization. Its outputs directly drive the motions of two front wheels while the rear wheel is indirectly controlled through a velocity regulating network (VRN) module. In parallel, a simple neural oscillator network module serves as a central pattern generator (CPG) producing basic rhythmic signals for driving leg movements to obtain, e.g., sidestepping or climbing over small obstacles. Controlling sidestepping directions is achieved by a phase switching network (PSN) module. As a result, this modular neural locomotion control serves as a basic control structure and can produce omnidirectional locomotion including sidestepping for climbing over obstacles by using four inputs ($I_{1,2,3,5}$, Fig. 2). Note that the input I_4

is used to transform the robot into a spherical mode for passive rolling motion[3]. We want to emphasize that the controller not only works for the leg-wheel hybrid robot presented here but it has been also applied equally efficiency to other types of robots (Manoonpong 2007, Manoonpong *et al.* 2008, Hülse *et al.* 2004).

Integrating the neural preprocessing networks of sensor signals provides effective sensor-driven behavior control based completely on neural network techniques. The preprocessing obtained by simple additive neurons, single recurrent neurons and the XOR network, which combine the sensory data and are robust against sensory noise by utilizing hysteresis phenomena of the recurrent neurons. As a consequence, the robot can autonomously perform the desired behaviors, like obstacle avoidance and exploration, with respect to corresponding sensory inputs. Because the design comprises independent modules one can simply replace the neural preprocessing module of the IR signals with other types of signal processing units to acquire different reactive behaviors, e.g., phototaxis (Manoonpong *et al.* 2007b) and sound tropism (Manoonpong 2007).

Recently, we are working on testing the purposed neural controller on the real robot (see (Chadil *et al.* 2011, Laksanacharoen and Jearanaisilawong 2009) for more details of the robot mechanics). Preliminary results for legged and wheeled locomotion can be seen at http://www.manoonpong.com/HybridRobot/RealbotSidewaysRight.mpg and http://www.manoonpong.com/HybridRobot/RealbotObstacleAvoidance.mpg, respectively. Due to mechanical problems of this first prototype robot (i.e., backlash and slip of the leg driving mechanisms using gears and belts, respectively), its legs cannot follow the motor commands all the times as expected. As a result, sidestepping using its legs cannot be effectively performed. This problem will be addressed in the next prototype but apart from this hardware-based behavior is the same as that of the simulation.

Hence, to overcome the remaining mechanical problems, our next step will be the improvement of the leg driving mechanisms. We will also enhance our simulation to achieve a detailed match with hardware by following an effective approach called iterative testing presented in (von Twickel *et al.* 2012). The approach uses single joint pendulum test setups to investigated nonlinear joint properties, backlash and activation to torque and velocity characteristics where the resulting data is integrated into simulation. In addition to this, we will use proprioceptive sensors (i.e., rotational sensors of wheels and joint angle sensors of leg joints) for damage detection and apply neural learning mechanisms based on correlation and/or reward information (Steingrube *et al.* 2010, Manoonpong *et al.* 2013) to find behaviorally useful motor responses after damage. For instance, the robot will learn to use its legs for movement when its wheels are broken (i.e., changing its locomotion from forward motion using wheels to sidestepping using legs); it will learn to find an appropriate combination of using wheels and/or legs if one wheel or leg or both of them damage; or it will learn to find an appropriate frequency of the CPG of leg movement when one leg is damaged. In principle, the robot will learn in an unsupervised manner (i.e., learning through correlation between its sensory signals) or learn on the basis of a reinforcement learning concept (i.e., learning to maximize a given reward). Besides, we will also investigate on autonomous transformation from rolling locomotion to obstacle climbing.

[3] Transformation and passive rolling are not the focus of this study but see http://www.manoonpong.com/HybridRobot/RollingAndTransforming.mpg for demonstration and (Chadil *et al.* 2011) for description.

Supporting Information

Supplementary information accompanies this paper on
http://www.manoonpong.com/HybridRobot/

Acknowledgements

This research was supported by Emmy Noether grant MA4464/3-1 of the Deutsche Forschungsgemeinschaft (DFG), Bernstein Center for Computational Neuroscience II Göttingen (BCCN grant 01GQ1005A, project D1), and the Higher Education Commission of Thailand. We thank Martin Biehl and Frank Hesse for correction of the text and Natthaphon Bun-athuek for his help in real robot experiments.

References

Akay, T., Ludwar, B., Goritz, M., Schmitz, J. and Büschges, A. (2007), "Segment specificity of load signal processing depends on walking direction in the stick insect leg muscle control system", *J. Neurosci.*, **27**, 3285-3294.

Allen, T., Quinn, R., Bachmann, R. and Ritzmann, R. (2003), "Abstracted biological principles applied with reduced actuation improve mobility of legged vehicles", *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1370–1375, Las Vegas, Nevada, USA, October.

Armour, R. and Vincent, J. (2006), "Rolling in nature and robotics: a review", *J. Bionic Eng.*, **3**(4),195-208.

Bässler, U. and Büschges, A. (1998), "Pattern generation for stick insect walking movements-Multisensory control of a locomotor program", *Brain Res. Rev.*, **27**, 65-88.

Besseron, G., Grand, C., Ben Amar, F., Plumet, F. and Bidaud, P. (2005), "Locomotion modes of an hybrid wheel-legged robot", *Proceedings of the 7th International Conference on Climbing and Walking Robots*, pages 825–833, London, UK, September.

Braitenberg, V. (1984), *Vehicles: Experiments in Synthetic Psychology*, Cambridge, MA: MIT Press.

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

Chadil, N., Phadoognsidhi, M., Suwannasit, K., Manoonpong, P. and Laksanacharoen, P. (2011), "A reconfigurable spherical robot", *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2380-2385, Shanghai, China, May.

Daun, S., Rubin, J. and Rybak, I. (2009), "Control of oscillation periods and phase durations in half-center central pattern generators: a comparative mechanistic analysis", *J. Comput. Neurosci.*, **27**(1), 3-36.

Delcomyn, F. (1999), "Walking robots and the central and peripheral control of locomotion in insects", *Auton. Robot.*, **7**, 259-270.

Eich, M., Grimminger, F., Bosse, S., Spenneberg, D. and Kirchner, F. (2008), "ASGUARD: a hybrid legged wheel security and SAR-robot using bio-inspired locomotion for rough terrain", *Proceedings of the IARP/EURON Workshop on Robotics for Risky Interventions and Enviromental Surveillance*, Benicassim, Spain, January.

Gabriel, J. and Büschges, A. (2007), "Control of stepping velocity in a single insect leg during walking", *Philos. T. Roy. Soc. A*, **365**, 251-271.

Grillner, S. (2006), "Biological pattern generation: the cellular and computational logic of networks in motion", *Neuron*, **52**(5), 751-766.

Halme, A., Leppaenen, I., Montonen, M. and Yloenen, S. (2001), "Robot motion by simultaneously wheel and leg propulsion", *Proceedings of the 4th International Conference on Climbing and Walking Robots*.

Karlsruhe, Germany, September.
Harth, E., Csermely, T., Beek, B. and Lindsay, R. (1970), "Brain functions and neural dynamics", *J. Theor. Biol.*, **26**, 93-120.
Hornby, G., Takamura, S., Yamamoto, T. and Fujita, M. (2005), "Autonomous evolution of dynamic gaits with two quadruped robots", *IEEE T. Robotic. Autom.*, **21**, 402-410.
Hülse, M. and Pasemann, F. (2002), "Dynamical neural schmitt trigger for robot control", *Proceedings of the International Conference on Artificial Neural Networks*, volume 2415, pages 783-788, Madrid, Spain, August.
Hülse, M., Wischmann, S. and Pasemann, F. (2004), "Structure and function of evolved neuro-controllers for autonomous robots", *Connect. Sci.*, **16**(4), 249-266.
Ijspeert, A.J. (2008), "Central pattern generators for locomotion control in animals and robots: a review", *Neural Networks*, **21**(4), 642-653.
Kim, Y., Ahn, S. and Lee, Y. (2010), "Kisbot: new spherical robot with arms", *Proceedings of the 10th WSEAS International Conference on Robotics, Control and Manufacturing Technology*, pages 63-67, Hangzhou, China, April.
Klavins, E., Komsuoglu, H., Full, R.J. and Koditschek, D.E. (2000), *Neurotechnology for Biomimetic Robots*, chapter The Role of Reflexes Versus Central Pattern Generators in Dynamical Legged Locomotion, MIT Press, Boston.
Krause, A., Dürr, V., Blässing, B. and Schack, T. (2010), "Evolutionary optimization of echo state networks: multiple motor pattern learning", *Proceedings of the 6th International Workshop on Artificial Neural Networks and Intelligent Information Processing*, volume 2, pages 63–71, Punchal, Madeira, Portugal, June.
Laksanacharoen, S. and Jearanaisilawong, P. (2009), "Design of a three-legged reconfigurable spherical shape robot", *Proceedings of the 2009 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1730-1733, Singapore, July.
Mahmoud, A., Okada, T. and Shimizu, T. (2008), "Circular path estimation of a rotating four-legged robot using a hybrid genetic algorithm LSM", *Proceedings of the JSME Conference on Robotics and Mechatronics*, pages 2P1–C10, Fukuoka, Japan, May.
Manoonpong, P. (2007), *Neural Preprocessing and Control of Reactive Walking Machines: Towards Versatile Artificial Perception-Action Systems*, Cognitive Technologies, Springer.
Manoonpong, P., Kolodziejski, C., Wörgötter, F. and J., M. (2013), "Combining correlation-based and reward-based learning in neural control for policy improvement", *Advs. Complex Syst.*, DOI: 10.1142/S021952591350015X.
Manoonpong, P., Pasemann, F. and Roth, H. (2007a), "Modular reactive neurocontrol for biologically-inspired walking machines", *Int. J. Robot. Res.*, **26**(3), 301-331.
Manoonpong, P., Pasemann, F. and Wörgötter, F. (2007b), "Reactive neural control for phototaxis and obstacle avoidance behavior of walking machines", *Int. J. Mech. Syst. Sci. Eng.*, **1**(3), 172-177.
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.
Manoonpong, P. and Roth, H. (2008), "Reactive neural control for autonomous robots: From simple wheeled robots to complex walking machines", *Proceedings of the Fifth International Conference on Neural Networks and Artificial Intelligence (ICNNAI '2008)*, Minsk, Belarus, May.
Marder, E. and Bucher, D. (2001), "Central pattern generators and the control of rhythmic movements", *Curr. Biol.*, **11**(23), R986-R996.
Matsuoka, K. (1985), "Sustained oscillations generated by mutually inhibiting neurons with adaptation", *Biol. Cybern.*, **52**(6), 367-376.
Nakajima, S. and Nakano, E. (2008), "Adaptive gait for a leg-wheel robot traversing rough terrain (second report: Step-up gait)", *J. Robot. Mechatronics*, **20**(6), 912-919.
Parker, G. and Lee, Z. (2003), "Evolving neural networks for hexapod leg controllers", *Proceedings of the 2003 IEEE/RSJInternational Conference on Intelligent Robots and Systems*, volume 2, pages 1376-1381, Las Vegas, Nevada, USA, October.

Pasemann, F., Hild, M. and Zahedi, K. (2003a), "SO(2)-networks as neural oscillators", *Computational Methods in Neural Modeling: Proceedings of the 7th International Work-Conference on Artificial and Natural Networks*, volume 2686, pages 144-151, Maó, Menorca, Spain, June.

Pasemann, F., Hülse, M. and Zahedi, K. (2003b), "Evolved neurodynamics for robot control", *Proceedings of European Symposium on Artificial Neural Networks 2003*, pages 439–444, Bruges, Belgium, April.

Pearson, K. and Iles, J. (1973), "Nervous mechanisms underlying intersegmental coordination of leg movements during walking in the cockroach", *J. Exp. Biol.*, **58**, 725-744.

Rumelhart, D., Hinton, G. and Williams, R. (1980), "Learning internal representations by error propagation", *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1, 318-362.

Salmen, M. and Ploeger, P. (2005), "Echo state networks used for motor control", *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1953–1958, Barcelona, Spain, April.

Shu, G., Zhan, Q. and Cai, Y. (2009), "Motion control of spherical robot based on conservation of angular momentum", *Proceedings of International Conference on Mechatronics and Automation*, 599-604, Changchun, Jilin, China, August.

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.

Tanaka, T. and Hirose, S. (2008), "Development of leg-wheel hybrid quadruped "AirHopper": lightweight leg-wheel design", *J. Robot. Mechatronics*, **20**(4), 526-532.

Terman, D. and Wang, D.L. (1995), "Global competition and local cooperation in a network of neural oscillators", *Physica D*, **81**, 148-176.

Valsalam, V. and Miikkulainen, R. (2008), "Modular neuroevolution for multilegged locomotion", *Proceedings of the Genetic and Evolutionary Computation Conference*, 265-272, Atlanta, Georgia, USA, July.

Valsalam, V. and Miikkulainen, R. (2009), "Evolving symmetric and modular neural networks for distributed control", *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 731-738, Montreal, Canada, July.

von Twickel, A., Büschges, A. and Pasemann, F. (2011), "Deriving neural network controllers from neuro-biological data—Implementation of a single-leg stick insect controller", *Biol. Cybern.*, **104**, 95-119.

von Twickel, A., Hild, M., Siedel, T., Patel, V. and Pasemann, F. (2012), "Neural control of a modular multi-legged walking machine: Simulation and hardware", *Robot. Auton. Syst.*, **60**(2), 227-241.

Wilson, H. and Cowan, J. (1972), "Excitatory and inhibitory interactions in localized populations of model neurons", *Biophys. J.*, **12**, 1-24.

Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Cristobal-Zagal, J. and Lipson, H. (2011), "Evolving robot gaits in hardware: The hyperneat generative encoding vs. parameter optimization", *Proceedings of the 20th European Conference on Artificial Life*, 890-897, Paris, France, August.

Zahedi, K., von Twickel, A. and Pasemann, F. (2008), "Yars: a physical 3D simulator for evolving controllers for real robots", *Simulation, Modeling and Programming for Autonomous Robots (SIMPAR 2008)*, volume 5325 of LNAI, pages 75-86, Venice, Italy, November.

CC