Reservoir-based Online Adaptive Forward Models with Neural Control for Complex Locomotion in a Hexapod Robot

Poramate Manoonpong*1,3 Sakyasingha Dasgupta*1 Dennis Goldschmidt1,2 Florentin Wörgötter1

***Abstract*—Walking animals show fascinating locomotor abilities and complex behaviors. Biological study has revealed that such complex behaviors is a result of a combination of biomechanics and neural mechanisms. While biomechanics allows for flexibility and a variety of movements, neural mechanisms generate locomotion, make predictions, and provide adaptation. Inspired by this finding, we present here an artificial bio-inspired walking system which combines biomechanics (in terms of its body and leg structures) and neural mechanisms. The neural mechanisms consist of 1) central pattern generator-based control for generating basic rhythmic patterns and coordinated movements, 2) reservoir-based adaptive forward models with efference copies for sensory prediction as well as state estimation, and 3) searching and elevation control for adapting the movement of an individual leg to deal with different environmental conditions. Simulation results show that this bio-inspired approach allows the walking robot to perform complex locomotor abilities including walking on undulated terrains, crossing a large gap, as well as climbing over a high obstacle and a fleet of stairs.**

I. INTRODUCTION

Walking animals show diverse locomotor skills to deal with a wide range of terrains and environments. They can effectively cross gaps [1], climb over obstacles [21], and walk on uneven terrain [15]. These capabilities are realized by a combination of biomechanics and neural mechanisms where the mechanisms include central pattern generators (CPGs), forward models, and reflex control. The CPGs generate basic rhythmic motor patterns for locomotion, while the reflex control employs direct sensory feedback [15] and/or expected feedback (predictive information) provided by the forward models [22] to shape the motor patterns for adaption. In principle, the forward models [22] are neural mechanisms that predict the future state of a system given the current state (feedback) and the control signals or motor commands (efference copies) [22]. While these three key neural mechanisms are important for locomotion control, only individual instances of them has been successfully realized on artificial systems [2], [11], [16], [20]; thereby achieving partial solutions.

A few studies have applied all these mechanisms to quadruped robots to achieve adaptive behavior [12], [17] but climbing over a high obstacle and crossing a large gap have not been included. Here, we present our neural mechanisms which capture the three bio-inspired neural mechanisms for complex locomotor generation including gap crossing, obstacle and stair climbing, and uneven terrain walking in a hexapod robot. This work also introduces the use of reservoir computing [13] with online learning as our adaptive forward models. By exploiting the dynamic reservoir embedded in the models, it allows for complex motor transformation at different gaits which cannot be achieved by our previous simple models [14]. We implement in total six reservoir-based adaptive forward models on the robot, each of which is for sensory prediction and state estimation of each leg. The outputs of the models are compared with foot contact sensory signals where differences between them are used for motor adaptation.

However, the main purpose of this article is not only to demonstrate the complex locomotor behaviors but also to show that the embodied modular neural closed-loop technique can be a powerful approach to solve sensorimotor coordination problems of many degrees-of-freedom systems as well as to achieve adaptive locomotion.

II. NEURAL MECHANISMS FOR COMPLEX LOCOMOTION

The neural mechanisms (Fig. 1a) are developed based on a modular structure. The mechanisms comprise i) central pattern generator (CPG)-based control, ii) reservoir-based adaptive forward models, and iii) searching and elevation control. The CPG-based control and the searching and elevation control have been presented in our previous work [14]. Thus here we discuss only their main functions while the reservoir-based adaptive forward models which are a main contribution of this paper will be presented in detail in the following section.

The CPG-based control basically generates a variety of rhythmic patterns and coordinates all leg joints of a hexapod robot (Fig. 1b), thereby leading to a multitude of different behavioral patterns and insect-like leg movements. The patterns include omnidirectional walking and insect-like gaits [14]. All these patterns can be set manually or autonomously driven by exteroceptive sensors, like a camera [23], a laser scanner [10], or range sensors [14]. Furthermore, synaptic long-term storage of behaviorally relevant motor patterns can be carried out using simple learning mechanisms as presented in our previous work [18]. While the CPG-based control provides versatile autonomous behaviors, the searching and elevation control using error signals provided by the reservoir-based adaptive forward models adapt the movement of an individual leg of the robot to deal with different environmental conditions. All neurons in the CPG-based control and the searching and elevation control are modeled as discrete-time non-spiking neurons with different activation functions (see [14] for details). They are updated with a frequency of ≈ 27 Hz.

* Equal contribution.

This research was supported by the Emmy Noether Program of the Deutsch Forschungsgemeinschaft (DFG, MA4464/3-1), the Federal Ministry of Education and Research (BMBF) by a grant to the Bernstein Center for Computational Neuroscience II Göttingen (01GQ1005A, project D1).

[1] Bernstein Center for Computational Neuroscience (BCCN), University of Göttingen, D-37077 Göttingen, Germany

[2] Institute of Neuroinformatics, University of Zurich, CH-8057 Zurich, Switzerland

[3] Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark.

Email: poma@mmmi.sdu.dk, sdasgup@gwdg.de

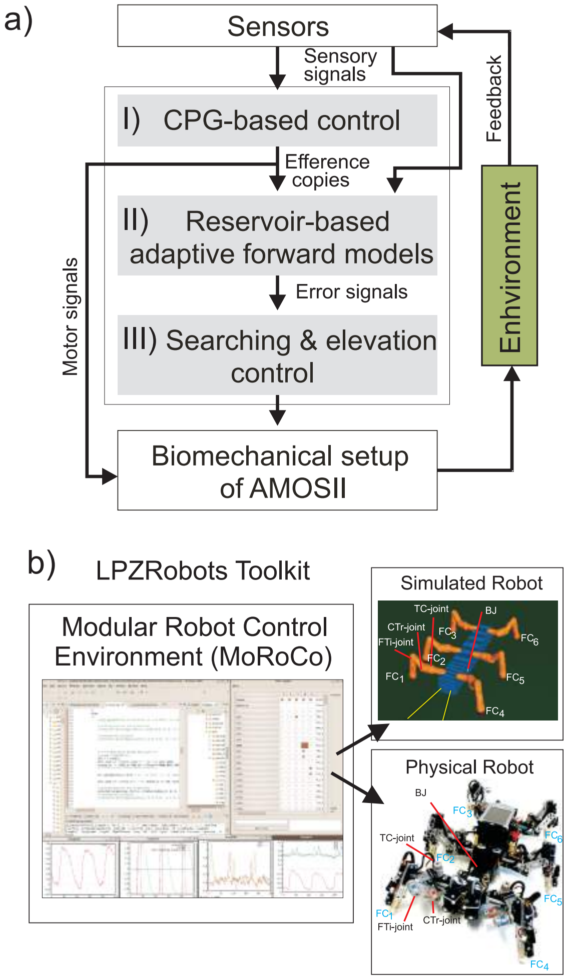

Fig. 1. (a) The diagram of an artificial bio-inspired walking system consisting of the biomechanical setup of the hexapod robot AMOS-II (i.e., six 3-jointed legs, a segmented body structure with one active backbone joint (BJ), actuators, and passive compliant components [14]), sensors (i.e., proprioceptive and exteroceptive sensors), and neural mechanisms (i,ii,iii). (b) Modular Robot Control Environment embedded in the LPZRobots toolkit [7]. It is used for developing a controller, testing it on the simulated hexapod robot, and transferring it to the physical one. FC_1, FC_2, FC_3, FC_4, FC_5, and FC_6 are foot contact sensors installed in the robot legs. Each leg has three joints: the thoraco-coxal (TC-) joint enables forward and backward movements, the coxa-trochanteral (CTr-) joint enables elevation and depression of the leg, and the femur-tibia (FTi-) joint enables extension and flexion of the tibia. The morphology of these multi-jointed legs based on a cockroach leg [24]. More details on BJ control for climbing are described in [5].

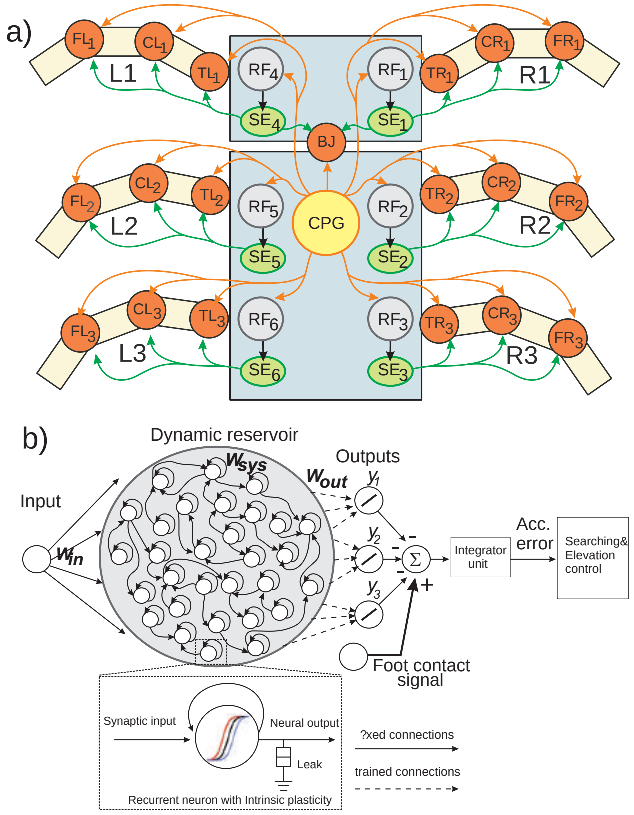

Fig. 2. (a) Neural mechanisms implemented on the bio-inspired hexapod robot AMOS-II. The yellow circle (CPG) represents the neural control mechanism which consists of a CPG mechanism with neuromodulation, neural CPG post processing, neural motor control, and motor neurons (see Fig. A2 in [14] and details therein). The grey circles $(RF_{1,2,3,\ldots,6})$ represent the reservoir-based adaptive forward models. The green circles $(SE_{1,2,3,\ldots,6})$ represent searching and elevation control modules. The orange circles represent leg joints where TR_i, CR_i, FR_i are TC-, CTr- and FTi-joints of the right front leg $(i = 1)$, right middle leg $(i = 2)$, right hind leg $(i = 3)$ and TL_i, CL_i, FL_i are left front leg $(i = 1)$, left middle leg $(i = 2)$, left hind leg $(i = 3)$, respectively. BJ is a backbone joint. The orange arrow lines indicate the motor signals which are converted to joint angles for controlling motor positions. The black arrow lines indicate error signals. The green arrow lines indicate signals for adapting joint movements to deal with different circumstances. b) An example of the reservoir-based adaptive forward model. The dashed frame shows a zoomed in view of a single reservoir neuron. In this setup, the input to each of the reservoir network comes from the CTr- joint of the respective leg. The reservoir learns to produce the expected foot contact signal for three different gaits (y_1, y_2, y_3). The signals of the output neurons are combined and compared to the actual foot contact sensory signal. The error from the comparison is transmitted to an integrator unit. The unit accumulates the error over time. The accumulated error is finally used to adapt joint movements through searching and elevation control.

III. Reservoir-based Adaptive Forward Models

A. Network description

Six identical adaptive[1] forward models $(RF_{1,2,3,\ldots,6})$ are used here, one for each leg (Fig. 2(a)). They serve for sensory prediction as well as state estimation. Specifically, each forward model transforms a motor signal (i.e., here the CTr-motor signal[2], efference copy) into an expected sensory signal to be able to compare it to the actual incoming one (i.e., here the foot contact signal d (Fig. 2b) of the leg). Each forward model is based on a recurrent neural network (RNN) of the reservoir computing (RC) [3], [9] type. Due to the dynamic reservoir, the network exhibits a wide repertoire of nonlinear activity which can be exploited for our motor signal transformation. Typically, the network has three layers: input, hidden (or internal), and output layers (Fig. 2b). The internal layer is constructed as a random RNN with N internal neurons and fixed randomly initialized synaptic connectivity. The recurrent neural activity within the dynamic reservoir varies as a function

[1] Adaptive, here, refers to the ability of each forward model to learn to multiple walking gaits as well as the ability of the recurrent neurons of the network to adjust their nonlinearity (firing rate).

[2] We use the CTr-motor signal instead of the TC- and FTi-motor signals since the signal shows clear swing (off the ground) and stance (on the ground) phases which can be simply matched to the foot contact signal.

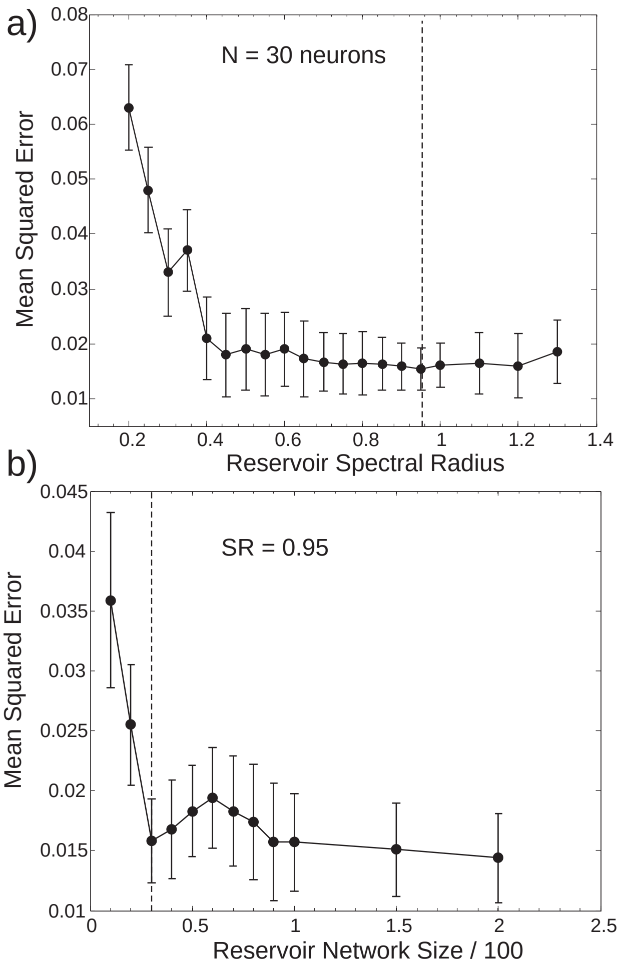

Fig. 3. (a) Plot of the change in the mean squared error for the forward model task for one of the front legs (R_1) of the walking robot with respect to the scaling of the reservoir weight matrix with different spectral radius (SR). As observed, very small values of (SR) have a negative impact on performance compared with values closer to one being better. Interestingly the performance does not change significantly for $SR > 1.0$. This is due to homeostasis introduced by intrinsic plasticity in the network. The optimal value of $SR = 0.95$ selected for our experiments is indicated with a dashed line. (b) Plot of the change in mean squared error with respect to different reservoir sizes (N). SR is fixed at the optimal value. Although increasing the reservoir size in general tends to increase performance, a smaller size of $N = 30$ gives same level of performance as $N = 100$. According to computational efficiency, we set our reservoir to 30 neurons. Results were obtained from 10 trials with different parameter settings on the forward model task for a single leg and a fixed walking gait.

of it's previous activity and the current driving input signal. The discrete time state dynamics of reservoir neurons is given by:

$$\mathbf{x}(t+1) = (1-\lambda)\mathbf{x}(t) + \lambda f_{sys}(\mathbf{W}_{in}u(t+1) + \mathbf{W}_{sys}\mathbf{x}(t) + b_o), \tag{1}$$

$$\mathbf{y}(t) = \mathbf{W}_{out}\mathbf{x}(t), \tag{2}$$

where $\mathbf{x}(t)$ is the N dimensional vector of reservoir state activations. N is the reservoir size. It is empirically selected and subsequently scaled to the optimal spectral radius (Figs. 3(a) and (b)). Here each reservoir contains $N = 30$ neurons with a spectral radius of 0.95. $u(t)$ is the input to the reservoir, which in this case, is a single CTr-motor signal. $\mathbf{y}(t)$ is the M dimensional vector of output neurons. In this study, M is set to three, since only three different gaits (i.e., wave (y_1), tetrapod (y_2), and caterpillar (y_3)) are used. The wave, tetrapod, and caterpillar gaits are used for climbing over an obstacle, walking on uneven terrain, and crossing a gap[3], respectively. Each output neuron, basically generates the expected foot contact signal of each gait. In principle, more gaits can be applied by adding further output neurons. As a result, once learned, this enables each reservoir forward model to do sensory/state prediction for a wide variety of walking patterns without the need of any re-training. The reservoir time scale is controlled by the parameter λ, where $0 < \lambda \leq 1$. Here it is set to 0.9. A constant bias $b_o = 0.001$ is applied to the reservoir neurons. $\mathbf{W}_{in}$ and $\mathbf{W}_{sys}$ are the input to reservoir weights and the internal reservoir recurrent connection weights, respectively. They are set randomly from an uniform distribution of [-0.1,0.1] and [-1,1] respectively. The reservoir neurons are updated with a frequency of ≈ 27 Hz using a tanh nonlinear activation function, $f_{sys}(.) = tanh(ax + b)$ where, x is the right hand side of Eq. 1 inside the function f_{sys}. Here the parameters a and b, allow adjustment of the shape and scale of the neuron transfer function (zoomed in view in Fig. 2 (b)) and are adapted through a generic intrinsic plasticity (IP) mechanism [19]. The IP mechanism is based on the Weibull distribution for unsupervised adaptation. In principle, adapting the parameters of the reservoir neuron nonlinearity allows the reservoir to homoeostatically maintain a stable firing rate and at the same time prevent unwanted chaotic neural activity (see [3] for more details).

The output neurons use a linear activation function. The output weights $\mathbf{W}_{out}$ are calculated online using the recursive least squares (RLS) algorithm at each time step, while the training input $u(t)$ is being fed into the reservoir. $\mathbf{W}_{out}$ are calculated such that the overall error is minimized; thereby, the network transforms the CTr-motor signal to the expected foot contact signal correctly. We implement the RLS algorithm using a fixed forgetting factor ($\lambda_{RLS} < 1$) as follows:

Here for each input signal $u(t)$, the reservoir state $\mathbf{x}(t)$ and network output $\mathbf{y}(t)$ are calculated using Eqs. 1 and 2. $e(t)$ is the online error calculated from the difference between the desired output (i.e., here foot contact signal) and the summation of all generated outputs. $\mathbf{K}(t)$ is the RLS gain vector and $\rho(t)$ the auto-correlation matrix updated at each time step. The reservoir to output weights $\mathbf{W}_{out}$ are initially set to zero. Exponential forgetting factor (λ_{RLS}) is set to a value less than one (here, we use 0.99). The auto-correlation matrix ρ is initialized as $\rho(0) = \mathbf{I}/\beta$, where $\mathbf{I}$ is unit matrix and β is a small constant (here, $\beta = 10^{-4}$). Details of all the fixed parameters and initial settings for the reservoir based forward model networks are summarized in Table I.

[3] These three gaits were empirically selected among 19 gaits. Previous studies show that wave and tetrapod gaits are the most effective gaits for climbing and walking on uneven terrains, respectively. While in this study we observe that the caterpillar gait is the best one for crossing a gap.

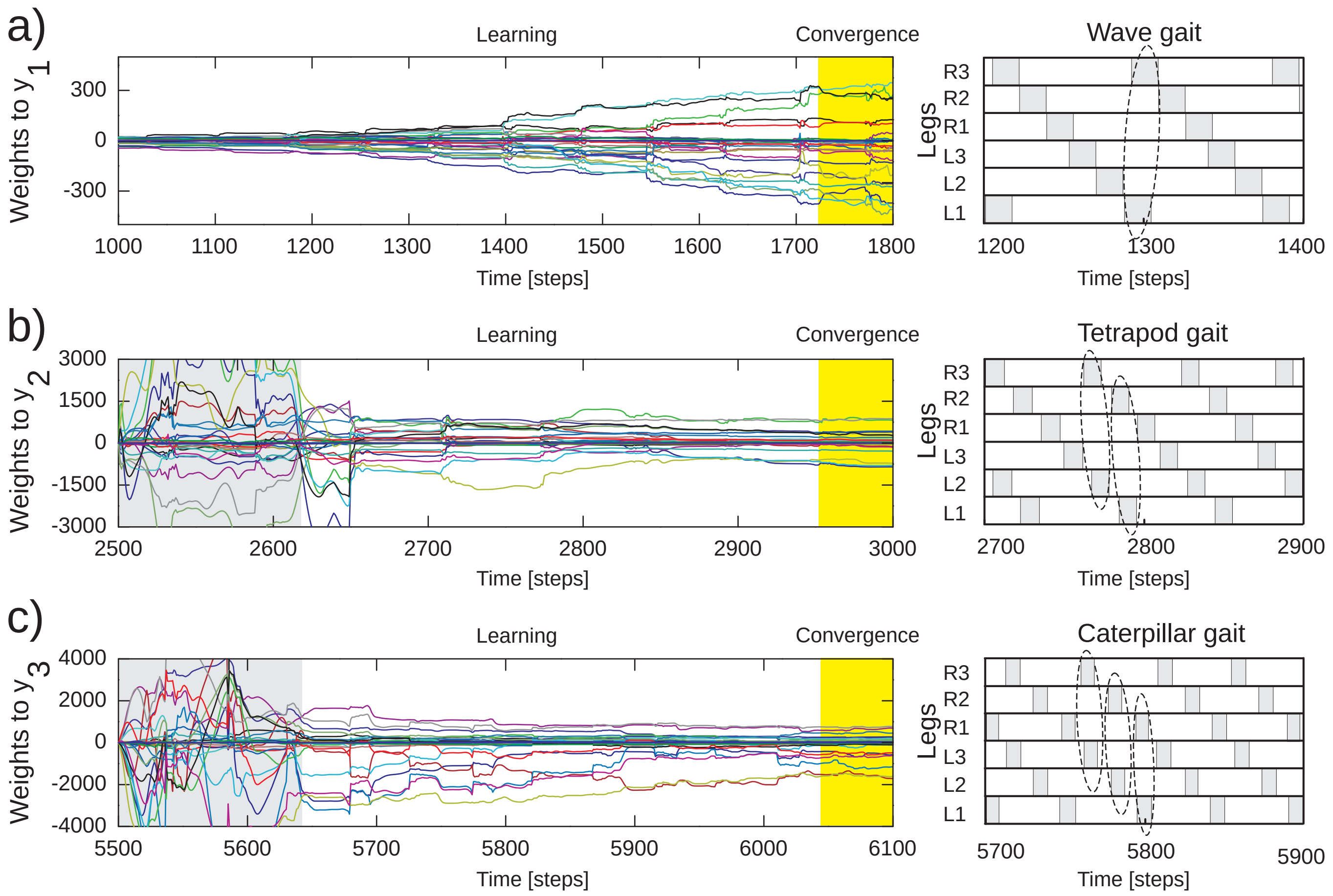

Fig. 4. Weight adaptation during online learning. (a) Changes of 30 weights projecting to the first output neuron (y_1) of the forward model of the right front leg (R_1) while walking with a wave gait. During this period, weights projecting to the second (y_2) and third (y_3) output neurons remain unchanged (i.e., they are zero). (b) Changes of the weights to y_2 while walking with a tetrapod gait. During this period, the weights to y_3 still remain unchanged and the weights to y_1 converge to around zero. (c) Changes of the weights to y_3 while walking with a caterpillar gait. During this period, the weights to y_1 and y_2 converge to around zero. At the end of each gait, all weights are stored such that they will be used for locomotion in different environments. The Grey areas represent transition phases from one gait to another gait and the yellow areas represent convergence. The gait diagrams are shown on the right. They are observed from the motor signals of the CTr-joints (Fig. 5). White areas indicate ground contact or stance phase and gray areas refer to no ground contact during swing phase. As frequency increases, some legs step in pairs (dashed enclosures). Note that one time step is ≈ 0.037 s.

TABLE I. THE LIST OF RESERVOIR NETWORK PARAMETER SETTINGS

Parameter	Value
Reservoir size (neurons)	30
Number of output neurons	3
Number of input neurons	1
Reservoir neuron bias (b_o)	0.001
Reservoir leak rate (λ)	0.9
RLS learning constant (β)	10^{-4}
Non-linearity shape initialization (a)	1.0
Non-linearity scale initialization (b)	0.0
RLS learning rate (λ_{RLS})	0.99
Reservoir network sparsity	50 %
Input to Reservoir sparsity	50 %
Reservoir spectral radius	0.95

B. Learning the Forward Models

In order to train the six forward models in an online manner, one for each leg, we let the simulated robot AMOS-II walk under normal condition (i.e., walking on a flat terrain with the three different gaits). Each gait is sequentially executed for a certain period of time. Specifically, we let the robot walk with a wave gait for 2500 time steps (Fig. 4a, right). Afterward, we change to a tetrapod gait (Fig. 4b, right) and finally to a caterpillar gait (Fig. 4c, right). In this way, synaptic weights projecting from reservoir neurons to the first output neuron (y_1) correspond to the motor signal transformation of the wave gait while synaptic weights projecting to the second (y_2) and third (y_3) output neurons correspond to the motor signal transformation of the tetrapod and caterpillar gaits, respectively. In this experimental setup, the weights for each gait converges within 2500 time steps. After each such learning period, all the reservoir forward model weights were stored in memory, such that they can be retrieved during testing the network for different behavioral scenarios. The weight adaptation of the different gaits during the online training process is shown in Fig. 4. It was sufficient to carry out the training process for a single run under normal walking condition. This was finally used as reference to compare with the foot contact signals generated for other walking conditions like crossing a gap, climbing, and walking on uneven terrain.

Figure 5 shows an example of the motor signal transformations at the different gaits. It can be seen that the CTr-motor

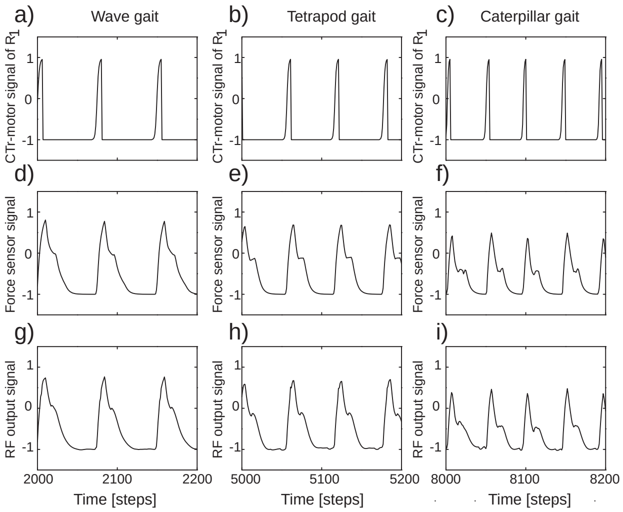

Fig. 5. (a-c) The CTr-motor signal of the right front leg (R_1) for wave, tetrapod, and caterpillar gaits, respectively. This motor signal is basically the input of the forward model. (d-f) The foot contact signal (force sensor signal) used as the target signal of the reservoir network. (g-i) The transformed motor signal or the final output of the forward model (RF output signal).

signal of the right front leg (R_1) (Fig. 5 a,b,c) is precisely transformed into the expected foot contact (FC) signal. The motor signals of the other legs are also precisely transformed into the corresponding foot contact signals (not shown). Note that the FC signals of the other legs show slightly different periodic patterns.

During testing, the output of each trained forward model (i.e., the expected FC signal, Fig. 5g,h,i) is used to compare it to the actual incoming FC signal of the leg (Fig. 5d,e,f). The difference (error signal) between them determines the walking state, where, a positive value indicates losing ground contact during the stance phase and a negative value indicates stepping on or hitting obstacles during the swing phase. These values are accumulated over time and then transmitted to the searching and elevation control for leg adaptation. The accumulated error is always reset to zero at the beginning of the swing phase (see [14] for more details of the searching and elevation control). For gap crossing, we use the accumulated error to control tilting of the backbone joint (BJ) and shifting of the TC- and FTi-joints such that the legs move extendedly forward (see the experiments and results section below, Fig. 6). For climbing and walking on uneven terrain (Fig. 7), we use them to control shifting of the CTr- and FTi-joints such that the respective leg searches for a foothold (i.e. extending the leg downwards till foot contact signal is active). This mechanism only occurs in the stance phase. In addition to such leg joint control, reactive backbone joint control is also applied to control the BJ for climbing (see [5] for details).

IV. Experiments and Results

In this section, we present the experiments carried out to assess the ability of the reservoir-based adaptive forward models in a neural closed-loop control system (see Fig. 1) at different behavioral conditions including crossing a gap, climbing over obstacles and up a fleet of stairs, and walking on uneven terrain. In all cases, we used the same learned forward models (Section 3). In the case of gap crossing, we let AMOS-II walk with a caterpillar gait (see Fig. 4c, right), such that each pair of legs moves simultaneously.

As shown in Fig. 6(1), at the beginning AMOS-II walked forward, straight towards the gap. During this period it performed regular movements as in the case of normal walking (walking on the flat surface of the first platform). Afterward, it encountered a 15 cm gap ($\approx 44\%$ of body length - the maximum crossable gap). In this situation, only its front legs lost ground contact during the stance phase. This results in the occurrence of the accumulated error signals of only the front legs. In order to activate the BJ and adapt the leg movements due to the error signals, the maximum accumulated error value of the previous step (Fig. 6, red line) was used to control the BJ and leg movements in the next step. This resulted in the BJ to lean upwards incrementally at around 1020-1170 time steps (Fig. 6(2)). Simultaneously, the TC- and FTi-joint movements of the left and right front legs were also adapted accordingly. Due to a predefined time-out period for tilting upwards, at around 1170 time steps (Fig. 6(3)), the backbone joint automatically moved downwards. As a result, the front legs touched the ground of the second platform at the middle of the stance phase; leading to a decrease in the accumulated error signals. In order to let the BJ to move back to the normal position ($-2\,\deg$) another time-out period of 1200 time steps was used to signal tilting downwards (Fig. 6(4)). Thereafter, the TC- and FTi-joints performed regular movements. At around 1300 time steps (Fig. 6(5)), the left and right hind legs lost the ground contact leading to body tilting. As a result the movements of the TC- and FTi-joints were slightly adapted. Finally, AMOS-II successfully crossed the gap (as observed from Fig. 6(6)).

Besides this experimental result, it is important to note that both adaptive locomotion and reactive backbone joint controllers have a distributed implementation, but they are indirectly coupled by sensory feedback and the mechanical structure of AMOS-II. In this way, the combined neural control network driven by the sensor signals synchronizes leg and backbone joint movements for stable walking and gap crossing.

Figure 7 shows that the neural mechanisms (see Fig. 1) not only enable AMOS-II to successfully generate gap crossing behavior (as shown above), but also allows it to climb over an obstacle and up a fleet of stairs as well as to walk on uneven terrain. In both cases, we directly used the accumulated errors for leg movement adaptation. For climbing, the reactive backbone joint control was also applied to the system (see [5] for more details) along with a slow wave gait (see Fig. 4a, right). Experimentally the wave gait was found to be the most effective for climbing, which allows AMOS-II to overcome the highest climbable obstacle (i.e., 15 cm height which equals $\approx 86\%$ of its leg length) and to surmount a fleet of stairs. For walking on uneven terrain, a tetrapod gait (see Fig. 4b, right) was used without the backbone joint control. This is the most effective gait for walking on uneven terrain (see also [14]). Recall that in all experiments the forward models basically generate the expected foot contact signals (i.e., sensory prediction) which are compared to the actual observed signal. Errors between the expected and actual signals during locomotion serve as state estimation and are used to adapt the joint movements accordingly. It is important to note

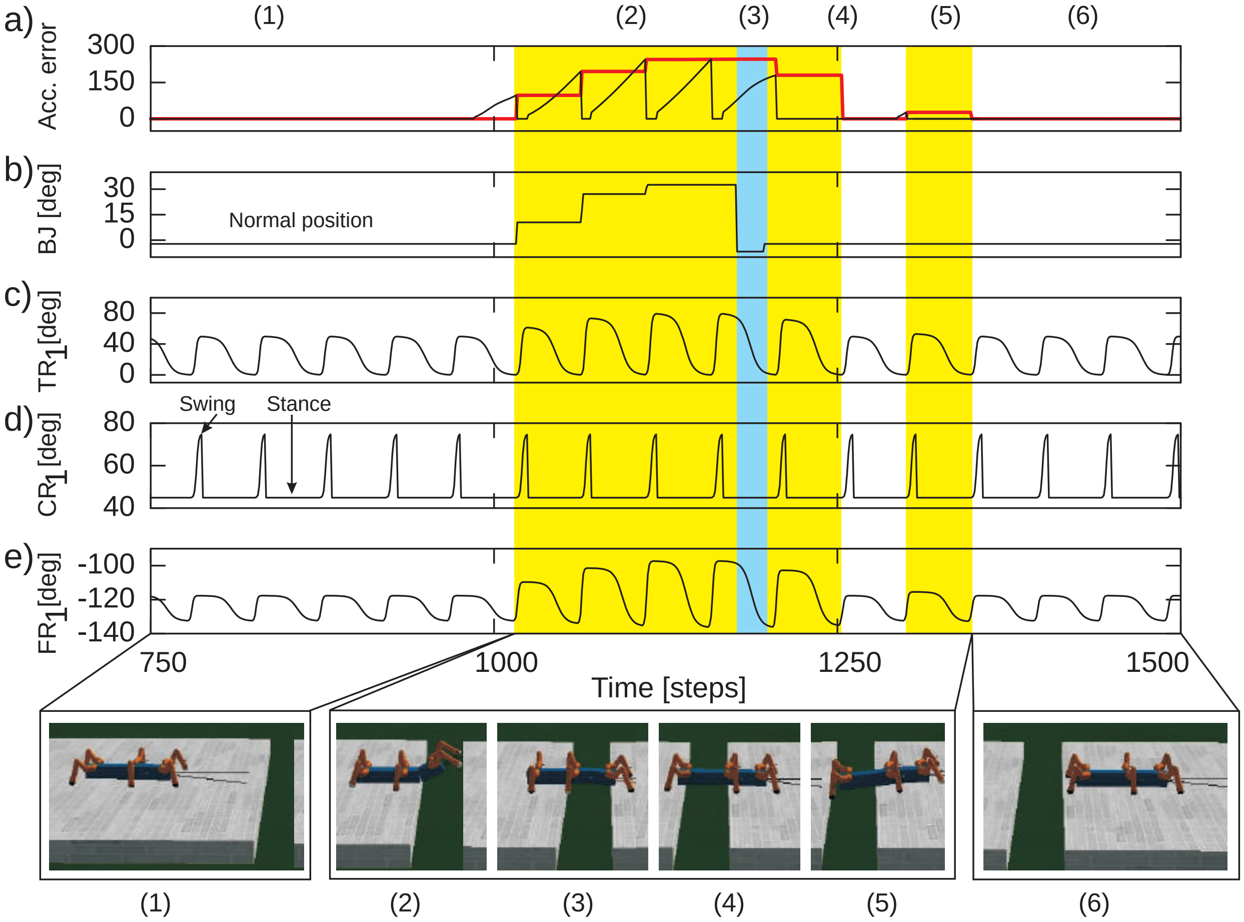

Fig. 6. Real-time data of walking and crossing a large gap. (a) The accumulated error (black line) and the maximum accumulated error value at the end of each stance phase (red line) of the right front leg (R_1). The accumulated error is reset to zero every swing phase. (b) The backbone joint (BJ) angle during walking and gap crossing. The BJ stayed at a normal position (-2 deg) during normal walking. It leans upwards and then bent downwards during gap crossing. (c-e) The TC-, CTr-, and FTi-joint angles of R_1 during walking and gap crossing. The joint adaptation was controlled by the maximum accumulated error value of a previous step (red line). Below pictures show snap shots of the locomotion of AMOS-II during the experiment. Note that one time step is ≈ 0.037 s. We encourage readers to also see the video of this experiment at http://manoonpong.com/ComplexLocomotion/S1.wmv.

that, the best gait for each specific scenario was experimentally determined (see also [14]) and fixed. However, this could be easily extended with learning mechanisms (see [18] [4]) to switch to the desired gait when the respective behavioral scenarios are encountered.

V. Conclusion

In this study, we presented adaptive forward models using reservoir computing for locomotion control. Each model is implemented on each leg of a simulated bio-inspired hexapod robot. It is trained online during walking on a flat terrain in order to transform an efference copy (motor signal) into an expected foot contact signal (i.e., sensory prediction). Afterward, the learned model of each leg is used to estimate walking states by comparing the expected foot contact signal with the actual incoming one. The difference between the expected and actual foot contact signals is used to adapt the robot's leg through elevation and searching control. Each leg is adapted independently. This enables the robot to successfully walk on uneven terrains. Moreover, using a backbone joint, the robot can also successfully cross a large gap and climb over a high obstacle as well as up a fleet of stairs. In this approach, basic walking patterns are generated by CPG-based control. It is also important to note that the usage of reservoir networks, as forward models in general, provides the added benefit of an inherent representation of temporal memory (due to the recurrent structure). The memory might be required to overcome scenarios with time lags between expected sensory signals and motor outputs or tasks with memory dependencies. Furthermore, online adaptation of only the output weights (readout) makes reservoir networks beneficial for simple and robust learning. However, reservoir parameters (e.g., sparsity, spectral radius, number of neurons) require proper tuning to achieve optimal performance for a given task. Although here we provided a proof of selecting some of these parameters, more rigorous analysis of all free parameters will be carried out in the future.

The concept of forward models with efference copies in conjunction with neural control has been suggested since the mid-20th century [8], [6] and increasingly employed for biological investigations [22]. This is because it can explain mechanisms which biological systems use to predict the conse-

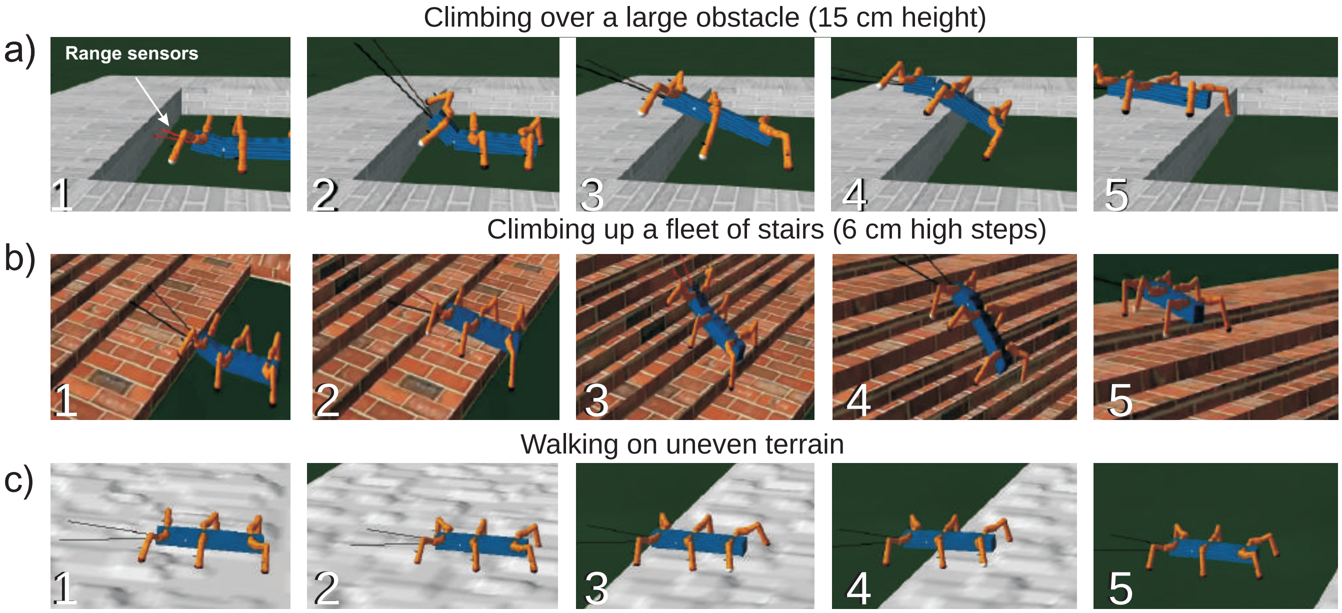
Climbing over a large obstacle (15 cm height)
a)
Range sensors
1
2
3
4
5
Climbing up a fleet of stairs (6 cm high steps)
b)
1
2
3
4
5
Walking on uneven terrain
c)
1
2
3
4
5

Fig. 7. Snap shots during climbing over a high obstacle, climbing up a fleet of stairs, and walking on uneven terrain. (a) AMOS-II walked with the wave gait and approached a 15 cm high obstacle (1). It detected the obstacle using its range sensors installed at its front part. The low-pass filtered range sensory signals control the BJ to tilt upwards (2) and then back to its normal position (3). Due to the missing foot contact of the front legs, the BJ moved downwards to ensure stability (4). During climbing, middle and hind legs lowered downwards due to the occurrence of the accumulated errors, showing leg extension, to support the body. Finally, it successfully surmounted the high obstacle (5). We encourage readers to also see the video of this experiment at http://manoonpong.com/ComplexLocomotion/S2.wmv. (b) AMOS-II climbed up a fleet of stairs (1-5) using the wave gait as well as the reactive backbone joint control. The climbing behavior is also similar to the one described in the case (a). We encourage readers to also see the video of this experiment at http://manoonpong.com/ComplexLocomotion/S3.wmv. (c) AMOS-II walked with the tetrapod gait. During traversing from the uneven terrain (1-4) to the even terrain (5), it adapted its legs individually to deal with a change of terrain. That is, it depressed its leg and extended its tibia to search for a foothold when losing a ground contact during the stance phase. Losing ground contact information is detected by a significant change of the accumulated errors. We encourage readers to also see the video of this experiment at http://manoonpong.com/ComplexLocomotion/S4.wmv.

quence of their action based on sensory information, resulting in adaptive and robust behaviors in a closed-loop scenario. This concept also forms a major motivation for robots inspired by biological systems. In this context, our work verifies that a combination of CPG-based neural control, adaptive forward models with efference copies, and searching and elevation control can be used for generating complex locomotion and adaptive behaviors in an artificial walking system. In the future, we will transfer the reservoir-based adaptive forward models with neural control to our real hexapod robot AMOS-II and test it in a real environment.

Acknowledgment

This research was supported by the Emmy Noether Program (DFG, MA4464/3-1), the Federal Ministry of Education and Research (BMBF) by a grant to the Bernstein Center for Computational Neuroscience II Göttingen (01GQ1005A, project D1) and the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273 (Xperience).

References

[1] Blaesing, B., Cruse, H.: Stick Insect Locomotion in a Complex Environment: Climbing Over Large Gaps. J. Exp. Biol. 207: 1273-1286 (2004).

[2] Blaesing B.: Adaptive Locomotion in a Complex Environment: Simulation of Stick Insect Gap Crossing Behaviour. In Proc. of the Eighth Int. Conf. on the Simulation of Adaptive Behavior: From Animals to Animats 9, MIT Press, pp. 173-182 (2004).

[3] Dasgupta, S., Wörgötter, F., Manoonpong, P.: Information Dynamics based Self-Adaptive Reservoir for Delay Temporal Memory Tasks. Evolving Systems, 4(4):235-249 (2013).

[4] Dasgupta, S., Wörgötter, F., Morimoto, J., Manoonpong, P.: Neural Combinatorial Learning of Goal-directed Behavior with Reservoir Critic and Reward Modulated Hebbian Plasticity. In Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics, pp. 993-1000 (2013).

[5] Goldschmidt, D., Wörgötter, F., Manoonpong, P.: Biologically Inspired Reactive Climbing Behavior Of Hexapod Robots. In Proc. IEEE Int. Conf. on Intelligent Robots and Systems, pp. 4632-4637 (2012).

[6] Held, R.: Exposure History as a Factor in Maintaining Stability of Perception and Coordination. Journal of Nervous and Mental Disease, 132: 26-32 (1961).

[7] Hesse, F., Martius, G., Manoonpong, P., Biehl, M., Wörgötter, F.: Modular Robot Control Environment Testing Neural Control on Simulated and Real Robots. Frontiers in Computational Neuroscience, Conference Abstract: Bernstein Conference 2012, doi: 10.3389/conf.fncom.2012.55.00179 (2012).

[8] Holst, E.v., Mittelstaedt, H.: Das Reafferenzprinzip. Naturwissenschaften, 37: 464-476 (1950).

[9] Jaeger, H., Haas, H.: Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science: 78-80 (2004).

[10] Kesper, P., Grinke, E., Hesse, F., Wörgötter, F., Manoonpong, P.: Obstacle/Gap Detection and Terrain Classification of Walking Robots based on a 2D Laser Range Finder. In Proc. 16th Int. Conf. on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR), pp. 419-426 (2013).

[11] Lewinger, W. A., Quinn, R. D.: Neurobiologically-based Control System for an Adaptively Walking Hexapod. Ind. Robot 38(3): 258-263 (2011).

[12] Lewis, M. A., Bekey, G. A.: Gait Adaptation in a Quadruped Robot.

Auton. Robot. 12(3): 301-312 (2002).

[13] Lukosevicius, M., Jaeger, H.: Reservoir Computing Approaches to Recurrent Neural Network Training. Computer Science Review 3(s): 127-149 (2009).

[14] Manoonpong, P., Parlitz, U., Wörgötter, F.: Neural Control and Adaptive Neural Forward Models for Insect-like, Energy-Efficient, and Adaptable Locomotion of Walking Machines. Front. Neural Circuits 7: 12, doi: 10.3389/fncir.2013.00012 (2013).

[15] Pearson, K., Franklin, R.: Characteristics of Leg Movements and Patterns of Coordination in Locusts Walking on Rough Terrain. Int. J. Robot. Res. 3(2): 101-112 (1984).

[16] Schilling, M., Paskarbeit, J., Schmitz, J., Schneider, A., Cruse H.: Grounding an Internal Body Model of a Hexapod Walker Control of Curve Walking in a Biological Inspired robot. In Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 27622768 (2012).

[17] Silva, P., Matos, V., Santos, C. P.: Adaptive Quadruped Locomotion: Learning to Detect and Avoid an Obstacle. In Proc. of Int. Conf. on the Simulation of Adaptive Behavior, pp. 361-370 (2012).

[18] Steingrube, S., Timme, M., Wörgötter, F., Manoonpong, P.: Self-organized adaptation of a simple neural circuit enables complex robot behaviour. Nat. Phys. 6(2): 224-230 (2010).

[19] Triesch, J.: Synergies between Intrinsic and Synaptic Plasticity Mechanisms. Neural Computation, 4: 885-909 (2007).

[20] von Twickel, A., Hild, M., Siedel, T., Patel, V., Pasemann, F.: Neural Control of a Modular Multi-Legged Walking Machine: Simulation and hardware. Robot. Auton. Syst. 60(2): 227-241 (2012).

[21] Watson, J. T., Ritzmann, R. E., Zill, S. N., Pollack, A. J.: Control of Obstacle Climbing in the Cockroach, Blaberus Discoidalis: I. locomotion. J. Comp. Physiol. 188: 39-53 (2002).

[22] Webb, B: Neural Mechanisms for Prediction: Do Insects Have Forward Models? Trends Neurosci. 27: 278-282 (2004).

[23] Zenker, S., Erdal Aksoy, E., Goldschmidt, D., Wörgötter, F., Manoonpong, P.: Visual Terrain Classification for Selecting Energy Efficient Gaits of a Hexapod Robot. In Proc. IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, pp. 577-584 (2013).

[24] Zill, S., Schmitz, J., Büschges, A.: Load Sensing and Control of Posture and Locomotion. Arthropod Struct. Dev., 33: 273-286 (2004).