

# Multiple chaotic central pattern generators for locomotion generation and leg damage compensation in a hexapod robot

Guanjiao Ren, Weihai Chen, Christoph Kolodziejcki,  
Florentin Wörgötter, Sakyasingha Dasgupta, Poramate Manoonpong

**Abstract**—In chaos control, an originally chaotic system is modified so that periodic dynamics arise. One application of this is to use the periodic dynamics of a single chaotic system as walking patterns in legged robots. In our previous work we applied such a controlled chaotic system as a central pattern generator (CPG) to generate different gait patterns of our hexapod robot AMOSII. However, if one or more legs break, its control fails. Specifically, in the scenario presented here, its movement permanently deviates from a desired trajectory. This is in contrast to the movement of real insects as they can compensate for body damages, for instance, by adjusting the remaining legs' frequency. To achieve this for our hexapod robot, we extend the system from one chaotic system serving as a single CPG to multiple chaotic systems, performing as multiple CPGs. Without damage, the chaotic systems synchronize and their dynamics is identical (similar to a single CPG). With damage, they can lose synchronization leading to independent dynamics. In both simulations and real experiments, we can tune the oscillation frequency of every CPG manually so that the controller can indeed compensate for leg damage. In comparison to the trajectory of the robot controlled by only a single CPG, the trajectory produced by multiple chaotic CPG controllers resembles the original trajectory by far better. Thus, multiple chaotic systems that synchronize for normal behavior but can stay desynchronized in other circumstances are an effective way to control complex behaviors where, for instance, different body parts have to do independent movements like after leg damage.

## I. INTRODUCTION

Legged locomotion has properties such as movement agility and adaptability to uneven terrains that are difficult to achieve with wheeled and tracked locomotion. Most of the species in the terrestrial world employ legged locomotion. However, bio-inspired legged robots still can up to now not mimic all the advantages of real walking animals.

In recent years, research on the neural basis of walking and its application to robot control has become more and more popular. It was found that leg movements are dominated by a series of central oscillation originating from the spinal

cord (vertebrate) or ganglions (invertebrate), so called central pattern generators (CPGs) [1]. This mechanism has been applied to different types of legged robots, such as the bipedal robot designed by Taga et al. [2], the quadruped robot *Tekken* by Kimura et al. [3], the hexapod robots in our previous works [4][5]. Bio-inspired amphibious robots proposed by Ijspeert et al. [6] also employ this kind of control strategy. More details on CPG-based locomotion control can be seen in [7].

Although the previous CPG-based algorithms generate sophisticated gait patterns and can even deal with the irregularities of terrain to some extent, the problem of leg damage was to our knowledge not fully addressed in CPG-based control. The difficulty arises as the main controller usually contains CPGs which always control all legs with the same frequency [8]. In contrast, insects can adjust the frequency of each leg individually [9]. If one leg is broken, insects can still achieve locomotion through changing the oscillation frequency of legs independently.

Traditional methods to compensate for leg damage are complicated [10]. The robot has to detect which leg is broken, then replan the gait pattern and choose another proper foot contact point. With this, the trajectory is calculated once again using inverse kinematics. For different legs, different trajectories have to be calculated. Hence, all situations have to be considered and the procedure uses many computational resources.

In contrast, we develop a neural-based control strategy to not only generate multiple gaits but also to deal with leg damage. For this, we use multiple CPG modules inspired by the 3-CPGs model proposed by Daun-Gruhn et al. [11][12]. We extend our previously proposed chaotic CPG controller to three CPGs, which include one master-CPG and two client-CPGs. The three CPGs are related to the three pairs of legs respectively and can be regarded as the pro-, meso- and meta-thoracic ganglion. The structure of the master-CPG follows our previous model [4] which can generate different gaits such as slow wave gait, fast wave gait, tetrapod gait, transition gait and tripod gait. It can also generate chaotic leg motion in order to overcome irregularities in terrain and to get out of some traps. The two client-CPGs are slightly different from the master CPG since a synchronization and desynchronization mechanism is added. If the two client-CPGs synchronize to the master CPG, the neural outputs are the same for all CPGs. If they are desynchronized, they can oscillate with different frequencies. Thus, supposing some legs are broken, other legs can independently change their

This research was supported by the Emmy Noether Program (DFG, MA4464/3-1) and the Federal Ministry of Education and Research (BMBF) by a grant to the Bernstein Center for Computational Neuroscience II Göttingen (01GQ1005A, project D1). This work was also supported by National Natural Science Foundation of China under the project 61175108, National 863 Program of China under the project 2011AA040902 and Innovation Foundation of BUAA for PhD Graduates

G. J. Ren and W. H. Chen are with School of Automation Science and Electrical Engineering, Beijing University of Aeronautics and Astronautics, Beijing, 100191, China [sqrgj@163.com](mailto:sqrgj@163.com)

C. Kolodziejcki, F. Wörgötter, S. Dasgupta and P. Manoonpong are with Bernstein Center for Computational Neuroscience, III Physikalisches Institut - Biophysik, Georg-August-Universität Göttingen, 37077 Göttingen, Germany [Poramate@Manoonpong.com](mailto:Poramate@Manoonpong.com)

oscillation frequency and can compensate for the leg damage.

This paper is organized as follows. First, we introduce the walking machine platform - AMOSII, which is inspired by real insects. Second, we present the control algorithm where the chaotic CPG is briefly addressed since it has been discussed in detail in our previous works. After that, we show how the central oscillation is extended to three CPGs and also state how the three CPGs can be synchronized and desynchronized. Third, we demonstrate the performance of the model on simulations and real robot experiments. Finally, a conclusion is given.

## II. THE WALKING MACHINE PLATFORM AMOSII

In order to test our algorithm in a physical system, the six-legged walking machine AMOSII is employed (see Fig. 1). It has identical leg structure with three linkages (coxa, femur, and tibia, see Fig. 2). 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 tibia. Comparing to a real insect [1], the tarsus is ignored in the current design. Nevertheless, a spring is installed in the leg to substitute part of the function of the tarsus; i.e., absorbing the impact force during touchdown on the ground. In addition, passive coupling is installed at each joint (see Fig. 2) in order to yield passive compliance and to protect the motor shaft. The body consists of two parts: two front legs belong to the front part and the middle and hind legs belong to the hind part. The two body parts are connected by an active backbone joint which enables the rotation around the lateral or transverse axis. This backbone joint is mainly used for climbing which is not the main focus here (but see [13]). All leg joints as well as its backbone joint are driven by digital servo motors.

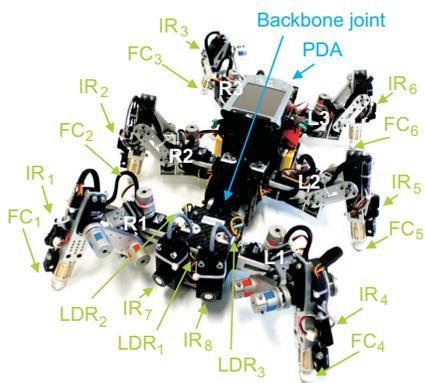


Fig. 1. Biologically inspired walking machine platform AMOSII.

The robot has six infrared sensors ( $IR_{1,\dots,6}$ ) at its legs, six force sensors ( $FC_{1,\dots,6}$ ) in its tibiae, three light dependent resistor sensors ( $LDR_{1,2,3}$ ) arranged in a triangle shape on the front body part, and two ultrasonic sensors ( $US_{1,2}$ ) at the front body part (see Fig. 1). The force sensors are for recording and analyzing the walking patterns. The infrared

sensors are used for detecting obstacles near the legs while the ultrasonic sensors are used for detecting obstacles in front. The light dependent resistor sensors serve to generate positive tropism like phototaxis. We use a Multi-Servo IO-Board (MBoard) installed inside the body to digitize all sensory input signals and generate a pulse-width-modulated signal to control servomotor position. The MBoard can be connected to a personal digital assistant (PDA) or a personal computer (PC) via an RS232 interface. For the robot walking experiments, here, the MBoard is connected to a PC on which the neural controller is implemented.

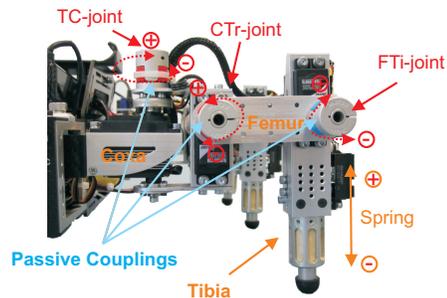


Fig. 2. Leg structure of AMOSII inspired by a cockroach leg.

## III. MULTIPLE CHAOTIC CENTRAL PATTERN GENERATORS AND THEIR SYNCHRONIZATION

In this section, we introduce the main controller of the hexapod robot, which is a CPG-based locomotion controller. First, we briefly introduce a single chaotic CPG presented in our previous work [4]. Second, we show how the single CPG is extended into multiple CPGs, e.g., three CPGs. The three CPGs generate either different periodic patterns (i.e., different frequencies) independently, or they become synchronized and generate the same pattern. Here, they will be synchronized for basic locomotion generation and desynchronized for leg damage compensation shown in the Experiments and Results section.

### A. Single Chaotic CPG

The chaos control CPG unit is shown in Fig. 3. Originally, the two neurons have self-connections as well as their mutual connections ( $w_{11}, w_{12}, w_{21}, w_{22}$ ) [5]. They also oscillate spontaneously and generate a series of waves. In order to enable complex behavior of the hexapod robot (e.g., chaotic leg motion for self untrapping [4]), we modify the CPG structure by removing the self-connection of the second neuron). Using appropriate parameters ( $w_{11} = -22.0, w_{12} = 5.9, w_{21} = -6.6, w_{22} = 0.0, \theta_1 = -3.4, \theta_2 = 3.8$ ), the CPG now exhibits chaotic dynamics. To achieve different walking patterns, we simultaneously add inputs to the two neurons, i.e., the control signals  $c_1$  and  $c_2$ , which act as extra biases that depend only on the period  $P$  of the walking cycle. With an increase of  $P$ , the robot walks slower. The output of the neurons is detected every  $P$  steps and the chaos is controlled to  $P$ -period orbit through adjusting the control

input. The discrete time dynamics of the activity (output) states  $x_i(t) \in [0, 1]$  of the circuit satisfies

$$x_i(t+1) = \sigma(\theta_i + \sum_{j=1}^2 w_{ij}x_j(t) + c_i^{(p)}(t)) \text{ for } i \in \{1, 2\} \quad (1)$$

where  $\sigma(x) = (1 + \exp(-x))^{-1}$  is a sigmoid activation function with biases  $\theta_i$ .  $w_{ij}$  is synaptic weight from neuron  $j$  to  $i$ .

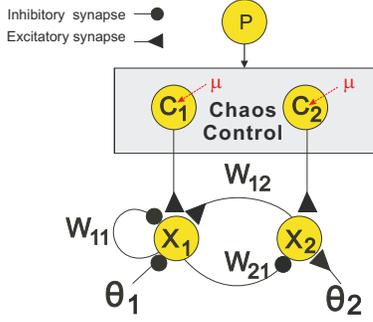


Fig. 3. Single CPG with the chaos controller.  $x_1$  and  $x_2$  indicate the neurons that generate the oscillation, while  $c_1$  and  $c_2$  are the control inputs depending only on the period  $P$  with a control strength  $\mu$ .  $w_{11}, w_{12}, w_{21}$  represent the synaptic weights and  $\theta_1$  and  $\theta_2$  mean the biases.

For a given period  $P$ , the control input

$$c_i^{(p)}(t) = \mu^{(p)}(t) \sum_{j=1}^2 w_{ij} \Delta_j(t) \quad (2)$$

is calculated every  $P$  time steps while the other steps are set to 0. In Eq. 2,  $\Delta_j(t)$  indicates the activity difference between the current step and  $P$ -steps before:

$$\Delta_j(t) = x_j(t) - x_j(t-p) \quad (3)$$

and  $\mu^{(p)}(t)$  means the control strength, which changes its value adaptively according to

$$\mu^{(p)}(t+1) = \mu^{(p)}(t) + \lambda \frac{\Delta_1^2(t) + \Delta_2^2(t)}{p} \quad (4)$$

with an adaption rate  $\lambda$ , e.g. 0.05.

Thus, using this single chaotic CPG different gait patterns can be easily obtained just by changing the  $P$  value. Periods 9, 8, 6, 5, 4 correspond to slow wave gait, fast wave gait, transition gait, tetrapod gait and tripod gait, respectively (see Fig. 4). A blue area means that this leg is in a support phase, i.e., touches the ground, while a white area indicates the swing phase. The red columns in panel (1) show the different walking speeds when the robot moves with different gaits.

Another useful function of this algorithm is the chaotic output. If we set the control signal  $c_i^{(p)}(t) \equiv 0$ , the neural CPG circuit shows chaotic dynamics, which can be applied for self-untrapping, e.g., when a leg falls into a hole. Usually, the output signals of the CPG are transferred to the leg joints after passing through additional neural circuits (i.e., a CPG post processing network, velocity regulating networks, and a phase switching network, not shown here but see [4][5] for more details). This neural control can also enable the robot to perform omnidirectional walking.

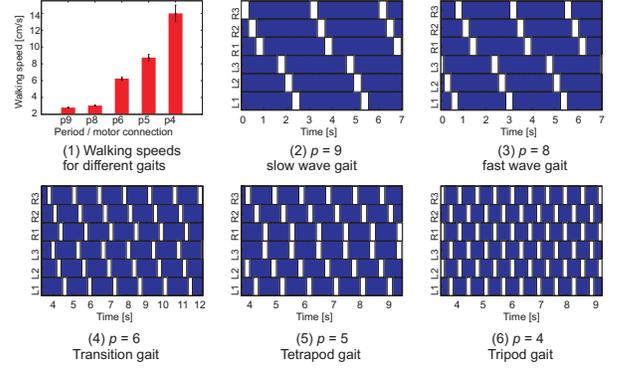


Fig. 4. Different gaits for changing  $P$  and the corresponding velocity (reproduced from [4]).

### B. Multiple Chaotic CPGs and Synchronization Mechanism

Supposing one or more legs are broken, the robot cannot use the same gait to stay on the original trajectory. In contrast, real insects can control their locomotion to continue with their trajectory even though some legs are broken. Legs then show different frequencies to maintain the body balance and compensate for leg damage.

Thus, we extend the single CPG network to three CPGs. The control structure is shown in Fig. 5. Each pair of legs is controlled by one CPG (yellow). In blue we depict the motor neurons, whose output can be directly applied to the joints. Green lines represent the spreading direction of the signals.

The first CPG is called master and the other two CPGs are called client. The two client-CPGs can synchronize to the master to keep pace with the oscillation frequency. When synchronized, the controller generates the same outputs as if there is only one chaos control CPG. When the legs are for example broken, the three CPGs can lose synchronization and oscillate with different frequencies.

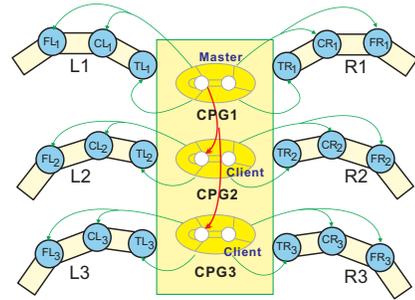


Fig. 5. Multiple chaotic CPGs of AMOSII. Three yellow ellipses represent the three CPGs. Blue circles are motor neurons. Red lines indicate the synchronization mechanism. Green lines indicate the directions of control signals spreading. For simplicity, the CPG post processing networks, the velocity regulating networks, and the phase switching networks are not depicted here but see [4][5].

The inner structure of the master CPG is like a normal chaos control CPG as shown in Fig. 3. However, the inner structure of the client CPGs is different. We add a synchronization mechanism to the CPG circuit as shown in Fig. 6.

When the client CPG needs to synchronize to the master CPG, the M-neuron becomes active (i.e., 1.0) shunting the synaptic weight from the inputs ( $c_1, c_2$ ). Thus, the outputs of the network are uncontrolled. Then, the output from the master CPG ( $X_{1master}$ , see Fig. 6) that was inhibited before is passed to the client due to disinhibition. So the output of the client-CPG can oscillate at the same frequency as the master CPG. When the client CPG needs to oscillate at its own frequency, the M-neuron becomes inactive (i.e., 0.0) switching off the inhibition and cutting down the connection from the master CPG. As a consequence, the two client CPGs are controlled by their control inputs ( $c_1$  and  $c_2$ ).

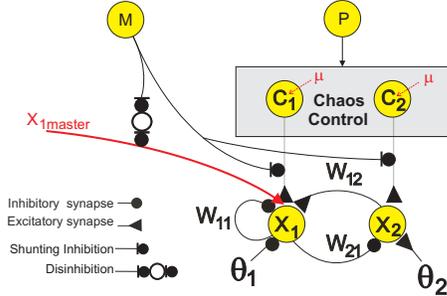


Fig. 6. Details of the client CPG. It is similar to the master CPG except for the synchronization mechanism, i.e. the M-neuron which enables the signal from the master CPG.

The following formulas describe the details of the client CPG. The outputs of the two neurons satisfy

$$\begin{aligned} x_1(t+1) &= \sigma(a_1(t)) + \alpha(x_{1master} - \sigma(a_1(t))) \\ x_2(t+1) &= \sigma(a_2(t)) \end{aligned} \quad (5)$$

where, the activity satisfies

$$a_i(t) = \theta_i + \sum_{j=1}^2 w_{ij}x_j(t) + c_i^{(p)}(t) \quad (6)$$

and  $\alpha$  is the synchronization parameter. It is set to 1.0 if the M-neuron is active and 0.0 if it is inactive. The output of  $x_1$  for synchrony and asynchrony is shown in Fig.7.

Although, in this paper, we implement three CPGs. This framework is general such that it can be easily extended to six CPGs (each leg one CPG) or 18 CPGs (each joint one CPG) for future research.

## IV. EXPERIMENTS AND RESULTS

### A. Simulation

We used LPZROBOTS<sup>1</sup> as simulation software. To simulate the legs' damage, we fixed the movement of the CTr-joints of the Left-middle (LM), Right-front (RF) and Right-hind (RH) legs, i.e., the CR1, CR3 and CL2 joints, as shown in Fig. 8(a) and in Fig. 9. By doing so, these three legs could now only support the body but they could not lift up. The period of the master CPG was set to 8 and the two client

<sup>1</sup>It is based on the Open Dynamics Engine (ODE). For more details of the LPZROBOTS simulator, see <http://robot.informatik.uni-leipzig.de/software/>.

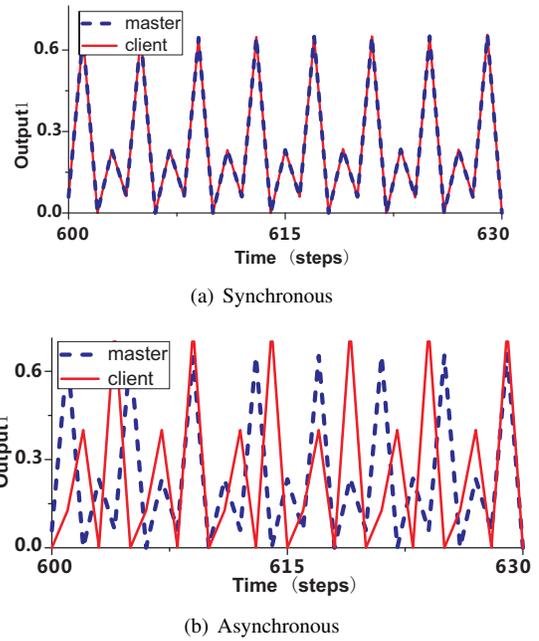


Fig. 7. The outputs of the CPG network for synchrony (a) and asynchrony (b). The master CPG has period 4; the client CPG has period 5.

CPGs were set to synchronize to the master. In this situation, the robot could not follow the former trajectory, a straight line, but turned right (see Fig. 10(d)).

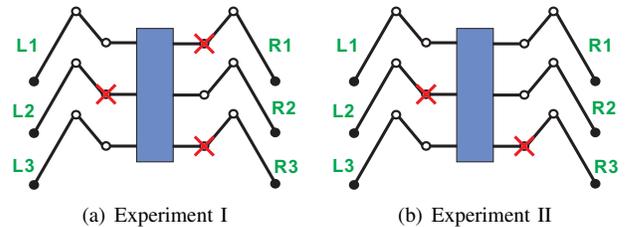


Fig. 8. This figure shows the broken legs in our two experiments. Red crosses indicate the broken joints. In the simulation and in experiment I, we fixed the LM, RF and RH legs. In experiment II, we fixed the LM and RH legs.

The reason is that the body does not any more obtain the same propelling force from each side: two legs on the left side but only one on the right side. Thus, we increased the driving frequency of CPG2 to period 5 (see Fig. 9(b)) while CPG3 remained synchronized to the master (see Fig. 9(a) and 9(c)). As a consequence, the RM leg can supply more propelling force. Normally every leg should perform like Fig. 9(c). However, the CR1, CR3 and CL2 joints were fixed, such that they performed like shown in Fig. 9(a). The oscillation frequency of the RM leg was increased, which is plotted in Fig. 9(b).

We tested four different periods in the simulation environment and the trajectory of the hexapod's movement is shown in Fig. 10. When the CPG2 was set to period 4, the robot performed a left turning curve. Period 5 was suitable for maintaining the straight-line walking of the robot. When

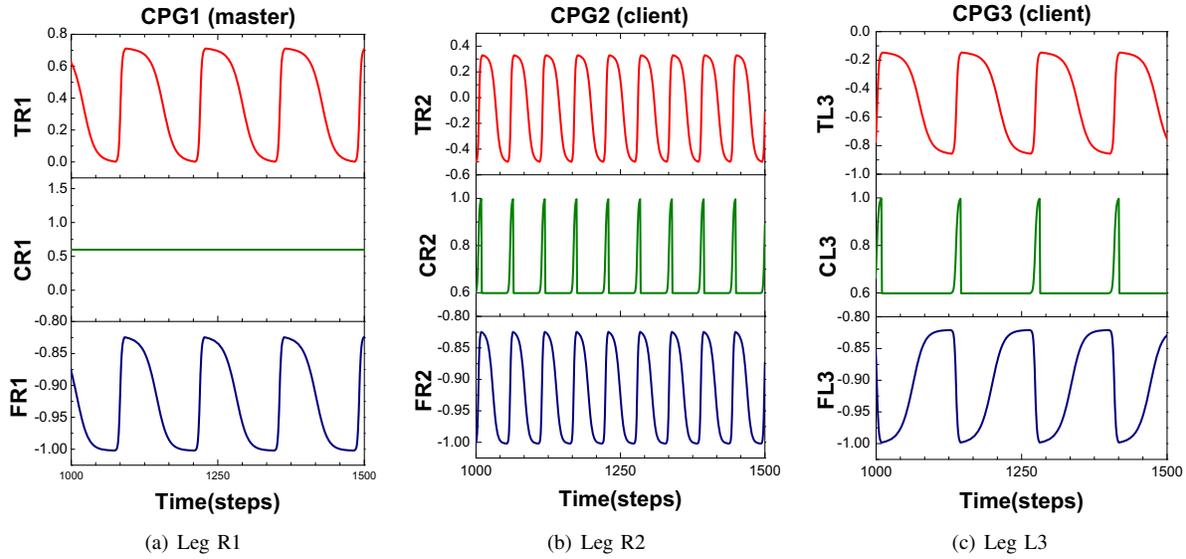


Fig. 9. Output of the motor neuron. (a) Output of leg R1 where the CTr- joint (CR1) was fixed. output of leg R3 had also a similar pattern. (b) Output of leg R2 where its oscillation was increased. (c) Output of leg L3 which oscillated with original frequency. Output of leg L1 had also a similar pattern. The outputs of the FTi- joints of hind legs (e.g. Leg L3) were inverted in order to push the leg outwards to stabilize the body.

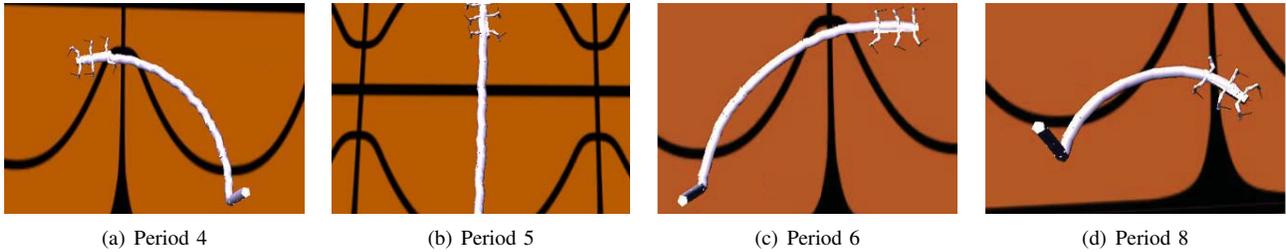


Fig. 10. Robot walking trajectories when leg R2 was set to different periods. Leg R2 was set to period 4, 5, 6 and 8, respectively.

CPG2 was set to period 6, the robot still turned right. When it was set to synchronize with the master CPG (i.e., period 8), the right turning curve was even sharper.

### B. Real Robot Experiment I

We also tested our leg damage scenario with our walking platform AMOSII (see Fig. 11). The setup was the same as for the simulation, i.e., the RF, RH, LM legs were fixed, as shown in Fig. 8(a). We used a straight white tape on the ground to compare the trajectory with a straight line (cf. Fig 10). The walking distance was set to 200 cm and we measured deviation in lateral direction after the distance.

The recorded data is plotted in Fig. 12. First, all CPGs had period 8 (S1, red) and because RF, RH and LM legs were broken, the robot did not arrive at the destination (cf. Fig. 11(a)). Next, CPG2 was set to period 5 (S2, blue), so that the RM leg moved with a higher frequency. Note that, CPG3 was still synchronized to the master. As a result, the robot walked straighter and reached the destination, however, with a deviation (cf. Fig. 11(b)). The blue bar (S2) depicts the average deviation for 5 trials. For comparison, the green bar (S3) shows the average deviation for 5 trials when all legs were functional. Because the mechanical structure

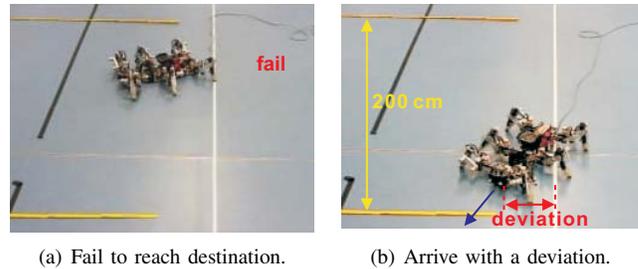


Fig. 11. Screenshot of the experiment . (a) The robot failed to reach the endpoint. (b) The robot arrived at the destination with a lateral deviation.

is not perfectly symmetrical, there is also some deviation in the lateral direction. The experimental video can be seen at [http://www.manoonpong.com/IROS2012/supple\\_video.wmv](http://www.manoonpong.com/IROS2012/supple_video.wmv)

The reason that there are differences between simulation and the real experiment is the friction condition and the mechanical asymmetry. Even with no legs broken, the trajectory was not a straight line. Note that we are currently working on modelling the friction condition and mechanical asymmetry

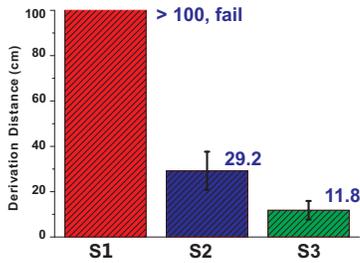


Fig. 12. Experiment I results: deviation in lateral direction. S1: We fixed RF, RH and LM legs and in three test trials, it always failed. S2: RM leg moved with period 5. The deviations in five trials were 20, 40, 24, 36 and 26 (cm) respectively. S3: The experiment with all legs functional and the deviations were 12, 13, 16, 13 and 5 (cm). The number above the bar is the mean value of the deviations for five trials.

in the simulation to reduce the differences.

### C. Real Robot Experiment II

We conducted another experiment to show the generality of this algorithm. In this experiment, all of the three CPGs were set to period 4 in order to generate a tripod gait. Then, we fixed the RH and LM legs (see Fig. 8(b)). As a consequence, the walking trajectory showed a left turning curve. This is because the middle legs serve a different function compared with the hind legs. To let the robot walk straight again, we made the three CPGs desynchronized and changed the period of them to 5, 4, 6, i.e., front legs in period 5, middle legs in period 4 and hind legs in period 6. The results are shown in Fig. 13. As can be seen, although the trajectory was still deviating from its original straight line (S2, blue), it was much better than if all the legs were in period 4 (S1, red). We also show the deviation when all legs were functional (S3, green). The experimental videos can be seen at [http://www.manoonpong.com/IROS2012/supple\\_video.wmv](http://www.manoonpong.com/IROS2012/supple_video.wmv)

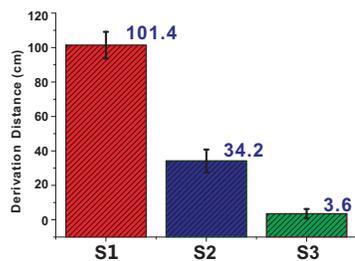


Fig. 13. Experiment II results: deviation in lateral direction. S1: We fixed RH and LM leg and all legs used period 4. Deviations in five tests were 100, 110, 100, 107 and 90 (cm). S2: The three CPGs had period 5, 4 and 6, respectively. Deviations in five tests were 40, 36, 40, 25 and 30 (cm). S3: Tripod gait with no leg broken. Deviations: 2, 5, 7, 0 and 4 (cm). The number above the bar is the mean value of the deviations for 5 trials.

## V. CONCLUSION

In this paper, we focused on the neural locomotion control of a hexapod, which is a continuation of our previous investigation. The central oscillator is extended from one

CPG to three CPGs, each of which controls a pair of opposite legs. Our experimental results show that one CPG might not be enough to maintain the body balance and compensate for leg damage. Here, we show that using more than one CPG provides a simple mechanism to cope with this problem.

However, the current 3 CPGs controller can only deal with some specific situations as shown here and the period parameter was manually tuned. Thus, our future plan is to further investigate by extending the controller to 6 or 18 chaos control CPGs with leg loading mechanisms [14][15] as the natural counterparts [9]. This will allow us to obtain more flexible adaptation, ensure statical stability, and attenuate the curve walking behavior in case of leg damage. In addition, we will employ a learning mechanism such that the robot could find an appropriate period automatically.

## VI. ACKNOWLEDGMENTS

We thank Dr. Frank Hesse and Xiaofeng Xiong for technical advice on simulator and real robot implementations.

## REFERENCES

- [1] F. Delcomyn, "Walking robots and the central and peripheral control of locomotion in insects", *Autonomous Robots*, vol. 7, No. 3, 1999, pp. 259-270.
- [2] G. Taga, Y. Yamaguchi and H. Shimizu, "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment", *Biological Cybernetics*, vol. 65, 1991, pp. 147-159.
- [3] H. Kimura, Y. Fukuoka and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts", *International Journal of Robotics Research*, vol. 26, No. 5, 2007, pp. 475-490.
- [4] S. Steingrube, M. Timme, F. Wörgötter and P. Manoonpong, "Self-organized adaptation of a simple neural circuit enables complex robot behaviour", *Nature Physics*, vol. 6, 2010, pp. 224-230.
- [5] P. Manoonpong, F. Pasemann and F. Wörgötter, "Sensor-driven neural control for omnidirectional locomotion and versatile reactive behaviors of walking machines", *Robotics and Autonomous Systems*, vol. 56, No. 3, 2008, pp. 265-288.
- [6] A. J. Ijspeert, A. Crespi, D. Ryczko and J. M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model", *Science*, vol. 315, No. 5817, 2007, pp. 1416-1420.
- [7] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review", *Neural Networks*, vol. 21, No. 4, 2008, pp. 642-653.
- [8] D. Spenneberg, K. McCullough and F. Kirchner, "Stability of walking in a multilegged robot suffering leg loss", In: *Proceedings of ICRA2004*, New Orleans, USA, 2004, vol. 3, pp. 2159-2164.
- [9] U. Bässler and A. Büschges, "Pattern generation for stick insect walking movements - multisensory control of a locomotor program", *Brain Research Reviews*, vol. 27, 1998, pp. 65-88.
- [10] J. M. Yang and J. H. Kim, "Fault-tolerant locomotion of the hexapod robot", *IEEE Transaction on System, Man and Cybernetics - Part B: Cybernetics*, vol. 28, No. 1, 1998, pp. 109-116.
- [11] S. Daun-Gruhn and T. I. Tóth, "An inter-segmental network model and its use in elucidating gait-switches in the stick insect", *Journal of Computational Neuroscience*, vol. 31, No. 1, 2010, pp. 43-60.
- [12] S. Daun-Gruhn and A. Büschges, "From neuron to behavior: dynamic equation-based prediction of biological processes in motor control", *Biological Cybernetics*, vol. 105, No. 1, 2011, pp. 71-88.
- [13] D. Goldschmidt, F. Hesse, F. Wörgötter, P. Manoonpong, "Biologically Inspired Reactive Climbing Behavior of Hexapod Robots", In: *Proceedings of the IROS2012*, Portugal, 2012.
- [14] M. Schilling, H. Cruse, P. Arena, "Hexapod Walking: an expansion to Walknet dealing with leg amputations and force oscillations", *Biological Cybernetics*, vol. 96, No. 3, 2007, pp. 323-340.
- [15] M. Görner, G. Hirzinger, "Analysis and evaluation of the stability of a biologically inspired, leg loss tolerant gait for Six- and eight-legged walking robots", In: *Proceedings of the ICRA2010*, Alaska, USA, 2010, pp. 4728-4735.