ITAT 2016 Proceedings, CEUR Workshop Proceedings Vol. 1649, pp. 131–137
http://ceur-ws.org/Vol-1649, Series ISSN 1613-0073, © 2016 P. Milička, P. Čížek, J. Faigl

On Chaotic Oscillator-based Central Pattern Generator for Motion Control of Hexapod Walking Robot

Pavel Milička, Petr Čížek, and Jan Faigl

Czech Technical University in Prague, Technická 2, 166 27 Prague, Czech Republic
milicpav|cizekpe6|faiglj@fel.cvut.cz

Abstract: In this paper, we address a problem of motion control of a real hexapod walking robot along a trajectory of the prescribed curvature and desired motion gait. The proposed approach is based on a chaotic neural oscillator that is employed as the central pattern generator (CPG). The CPG allows to generate various motion gaits according to the specified period of the chaotic oscillator. The output signal of the oscillator is processed by the proposed trajectory generator that allows to specify a curvature of the trajectory the robot is requested to traverse. Such a signal is then considered as an input for the inverse kinematic task which provides particular trajectories of individual legs that are directly send to the robot actuators. Thus, the main benefit of the proposed approach is that only two natural parameters are necessary to control the gait type and the robot motion. The proposed approach has been verified in real experiments. The experimental results support feasibility of the proposed concept and the robot is able to crawl desired trajectories with the tripod, ripple, low gear, and wave motion gaits.

1 Introduction

Hexapod crawlers have a great potential in many applications such as rescue missions or exploration of unknown environments that are hostile or unreachable to human beings and where motion capabilities of legged robots can be utilized to walk over rough terrains. Regarding a particular mission, it is desirable a robot can deal with various locomotor skills to adapt and react to its environment. On the other hand, a solution of such a task has already been found in nature by the process of evolution. Hence, it is a source of motivation for the presented approach based on biologically inspired locomotion controller for the hexapod walking robot showed in Figure 1.

The considered approach builds on the idea that the insect locomotion is generated by the so called Central Pattern Generators (CPGs) located in central nervous system [1, 2, 3], which generates the rhytmic motor patterns carried out by muscles without the sensory feedback. The feedback is; however, essential in structuring the motor patterns when walking on uneven terrains [4]. In addition, the locomotion is also enhanced by muscle dynamics reacting immediately to terrain irregularities and thus creating reactions (preflexes) faster than any neural feedback [5].

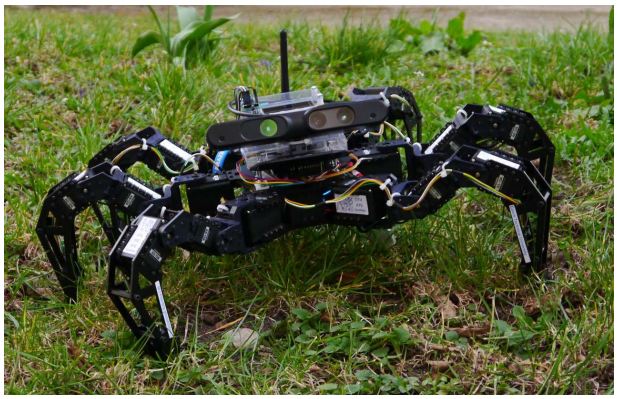


Figure 1: Hexapod robot used for experimental evaluation of the proposed chaotic oscillator-based motion control.

According to [6], individual limbs can produce motor patterns, which can be viewed as an evidence that each leg, or even each joint has its own pattern generator. The coordination of multiple legs is done by coupling the individual CPGs by mutual connections and the overall locomotion is composed of a synchronized motion of individual legs. Two phases can be recognized in a single motion step of each leg: the support phase and the swing phase. In the support phase, a leg is firmly touching the ground and moving backward relatively to the body, which results in the body moving forward relatively to the ground. On the other hand, in the swing phase, a leg swings forward in the air relative to the body to reach a position enabling another support phase.

Findings from biological observations of walking insects have been summarized by Wilson [7], who created a simple locomotion model consisting of five rules that are essential to the construction of motion patterns:

1. Swing phases go from rear to front and no leg swings until the one behind is in the support phase;
2. Contralateral legs of the same segment alternate in phase, i.e., they cannot both be in the swing phase simultaneously;
3. The swing phase duration is constant;
4. The support phase duration varies with the frequency of CPG oscillations, i.e., a lower frequency, which generates a slower gait, results in a longer support phase;
5. Intervals between steps of the hind leg and the middle leg and between the middle leg and the front leg are

constant, while the interval between the foreleg and the hind leg steps varies with the CPG frequency.

Based on these rules following locomotion patterns (gaits) can be identified. The most safe gait is called *wave* in which each leg swings separately. Another example is the *low gear* gait where an insect alternates between swinging two legs and one. Another types are *tetrapod* and *ripple* gaits which differ in phase lag between the left and right legs. Legs in the tetrapod gait are synchronized, whereas the ripple gait seems to be more fluent. Finally the fastest motion pattern is called the *tripod* gait. It is also worth mentioning that there are further variants of the tetrapod, ripple and low gear gaits depending on the phase lags between the left and right legs [8].

The main motivation of this work is to design and experimentally verify a biologically inspired locomotion controller for a hexapod walking robot to develop a suitable framework for further processing various sensory inputs and produce adequate control actions. Therefore, the design of the proposed controller is based on chaotic oscillator [9] for which we propose to post-process the oscillator output with a trajectory generator to generate the position of each leg's foot-tip and inverse kinematics module, which transforms the signal to the actual joints' positions. The proposed controller is comprehensible and it allows to control the hexapod locomotion with only two natural parameters: the turning rate *turn* of the robot and the period *p* influencing the gait type according to the aforementioned locomotion patterns.

The rest of the paper is organized as follows. Section 2 overviews the related work to give an insight in possible approaches and CPG implementation methods. Section 3 describes the design of the proposed locomotion controller and methods used for its development. Results of the performed experimental evaluation with the real robot are presented in Section 4. Concluding remarks and suggestions for further work are dedicated to Section 5.

2 Related Work

Biologically inspired strategies based on the CPGs have already been utilized in the control of legged locomotion. The existing methods mainly differs in the way how the CPG is implemented and how its output signal is processed. In strictly biological approaches, the signal is processed by neural networks with the motoneurons as the output layer. Alternatively, the signal can be post-processed and utilized in a trajectory generator to compute the desired foot-tips locations that are further transformed to the individual joints' positions using the inverse kinematics.

Regarding the CPG itself, there are many ways of achieving a patterned output [6, 10] that can be characterized as one of the three main types of implementation. First, the CPG can be implemented using neuron models mutually connected together, which produces (with a correct set of design parameters) a dynamical system capable of oscillations [11, 12]. The second type are CPG implementations based on the coupled non-linear oscillators (NLO), which are not strictly biologically based, but share many common characteristics with biophysical models. One of the most used models is Matsuoka oscillator [13] implementing the half centre principle: extensor and flexor neurons inhibiting each other with an adaptation mechanism. The Matsuoka NLO model was successfully simulated and implemented in hexapod [14, 15] and biped walking [16]. Finally, connectionist models are CPG implementations which tend to use simplified neuron models while focusing on the effect of inter-neuron connections. An example of this approach is a cellular neural network used in [17], where the network consists of identical cells arranged in a rectangular grid. Each cell is usually a first order dynamical system affected only by itself and its neighbours in the specified radius. Another example are spiking neurons used for hexapod locomotion in [18].

In this paper, we consider NLO-based approach for which we utilized chaotic neural oscillator proposed in [9] to produce rhythmic patterns of the desired CPG. The proposed solution is presented in the following section.

3 Proposed Solution

The proposed solution builds on the chaotic CPG [9] and the ideas of the adaptive locomotion controller for hexapod robots [14], where non-neural post-processing is suggested. However, this paper provides different method of the post-processing resulting in more emphasis put on the insect's locomotion rules. Moreover, the proposed approach utilizes the trajectory generator in the way that the hexapod is able to follow any given trajectory consisting of circular arcs.

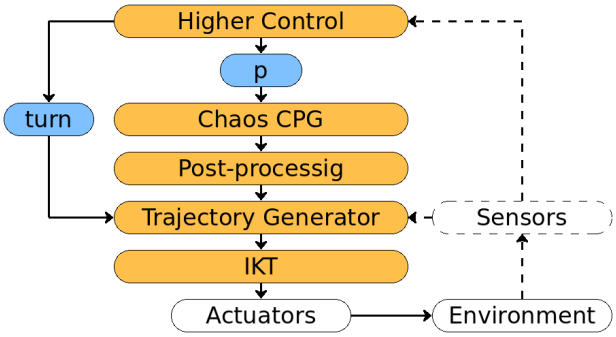


Figure 2: Structure of the proposed control system

The overall block scheme of the proposed modular controller is depicted in Figure 2. Note, the dashed lines denote the sensory feedback part of the system which has not yet been implemented but is considered future work. In general, the proposed controller works as follows. First of all, data from sensors are processed by the Higher Control module which sets the locomotion control parameters,

namely the turning radius *turn* and the period p.[1] The period p is used to adjust the CPG output which is then shaped and delayed by the post-processing module to generate a signal for each leg. Afterwards, this signal is fed into the trajectory generator which generates foot-tip positions using the second control parameter *turn*. Finally, the foot-tips' positions are transformed by the inverse kinematics module into the joint angles that are directly applied to the servo drives. The particular controller blocks are detailed in the following subsections.

3.1 Chaos CPG

The proposed CPG implementation is based on the chaotic neural oscillator [9]. The main feature of this oscillator is its ability to stabilize different periodic orbits by changing only a single parameter, the period p. A diagram of the used neural oscillator is shown in Figure 3.

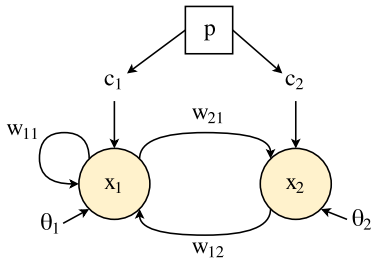


Figure 3: Chaotic neural oscillator

The oscillator is considered with the particular parameters $w_{11} = -22.0$, $w_{12} = 5.9$, $w_{21} = -6.6$, $w_{22} = 0$, $\theta_1 = -3.4$, $\theta_2 = 3.8$ for the network output computed as

$$x_i(t+1) = \sigma\left(\theta_i + \sum_{j=1}^{2} w_{ij}x_j(t) + c_i(t)\right), \qquad (1)$$

where w_{ij} is the synaptic weight from the neuron j to the neuron i, θ_i is the neuron bias and c_i is the control signal utilized for stabilizing periodic orbits. It is applied every $p+1$ step, otherwise it is set to zero. The signal is determined from

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

where Δ_j is the signal difference of the j-th neuron state separated by one period

$$\Delta_j(t) = x_j(t) - x_j(t-p), \qquad (3)$$

and $\mu(t)$ is the adaptive control strength given as

$$\mu(t+1) = \mu(t) + \frac{\lambda}{p}\left(\Delta_1(t)^2 + \Delta_2(t)^2\right). \qquad (4)$$

The period which is set to be stabilized is denoted as p. The adaptation speed λ has to be set carefully as too high values can prevent the stabilization of the given periodic orbit or the stabilization could take a long time. Besides, an usable learning rate gets lower with increasing period; so, we scale it by $1/p$. The adaptation speed λ has been empirically set to 0.05 in our implementation. The control strength μ is reset to -1 whenever the period p is changed.

Based on the real insect locomotion, periods p from the set $\{2, 3, 4, 6\}$ would be ideal for further processing. Unfortunately, not all periodic orbits can be stabilized or even exist in the proposed chaotic oscillator, such as the period $p = 3$. Therefore, p is selected from a set $\{4, 6, 8, 12\}$ which is then further utilized for the gait generation, where the swing phase is always constant.

Another drawback is that due to sensitivity of the oscillator to the initial conditions together with the control method used, the final stabilized period can differ from the desired one. For example, the control method cannot recognize that it actually stabilized period–2 orbit when period–6 orbit was set. This can be addressed by resetting the network states to zero every time the period is changed; so, the initial condition of the oscillator will not spoil the stabilization.

3.2 CPG Post-processing

The output produced by the chaos CPG has to be shaped and processed to get information about the current phase of each leg and to obtain an input for the trajectory generator, which can be used to position the legs' foot-tips. For this purpose, partially neural-based signal post-processing is proposed, which scheme is depicted in Figure 4.

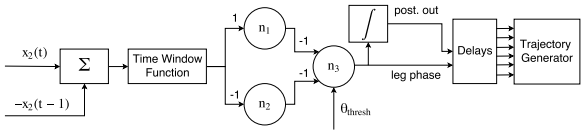


Figure 4: Post-processing module. n_1, n_2 and n_3 are neurons used for processing the output of the CPG.

The difference of the oscillator neuron output and its delay is used to obtain the signal for the phase generation

$$\Delta x_2(t) = x_2(t) - x_2(t-1). \qquad (5)$$

This signal goes through a time window function, which passes the signal every Δt step to the neural post-processing network. A suitable value Δt has been found experimentally as $\Delta t = 13$. The neural network compares the absolute difference of the signal with the threshold θ_{thresh} that is selected according to the actual period p and is listed in Table 1.

The output of the neural network corresponds to the current leg phase. If it is equal to -1, the leg is in the swing phase; otherwise it is in the support phase. Outputs of the

[1] Since in this work we focus on the CPG, trajectory generation and the experimental evaluation of the motion control of the hexapod walking robot, the control values are set manually.

Table 1: Post-processing constants

p	Gait	θ_{thresh}	τ_L
4	Tripod	0.20	26
6	Ripple	0.20	39
8	Low gear	0.50	26
12	Wave	0.78	78

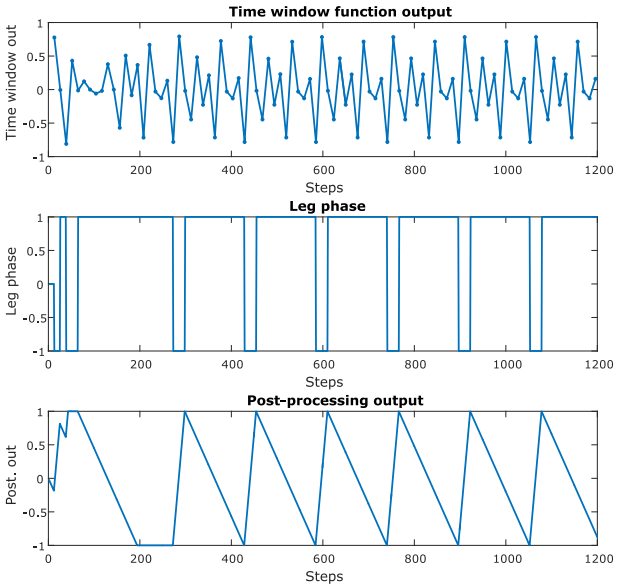


Figure 5: Post-processed signals, $p = 6$

individual neurons n_1, n_2, n_3 are computed as

$$n_1(t) = \max(0, \Delta x_2(t)), \quad (6)$$
$$n_2(t) = \max(0, -\Delta x_2(t)), \quad (7)$$
$$n_3(t) = f(-n_1(t) - n_2(t) + \theta_{thresh}), \quad (8)$$

where

$$f(x) = \begin{cases} -1 & \text{if } x \in (-\infty, 0), \\ 1 & \text{if } x \in [0, \infty). \end{cases} \quad (9)$$

Note, the neurons here serve only as processing units with no ability to learn or adapt.

According to the leg phase, a constant value is added to the previously post-processed output to achieve a suitable input for the trajectory generator *post*. This signal alternates from -1 to 1 in triangle waves after the period stabilization, where the up slope (swing phase) is a constant and the down slope (support phase) depends on the period p. Figure 5 shows the results of post processing of a CPG signal with the period $p = 6$.

The last part of the post-processing module is signal delaying. This is rather a simplification compared to biological systems as only a single oscillator is used to control all legs. By adjusting the delay τ_L (Table 1) and thus changing the phase lag between the right rear and the left rear leg, different walking patterns (i.e., *tripod*, *ripple*, *low gear*, and *wave* gaits) observed in nature can be achieved.

3.3 Trajectory Generator

The trajectory generator module decides the foot-tip coordinates $\hat{x}_i(t), \hat{y}_i(t), \hat{z}_i(t)$ of each leg i based on the post-processed output of the CPG – $post(t)$ at the time t, and a particular parameter determining the rate of the robot turning – *turn*. Foot-tips positions generation is generalized by considering the robot is always turning. Nevertheless, a turning radius set to a very high value results in a seemingly straight walking.

The $\hat{z}_i(t)$ coordinate of the foot tip is computed using

$$\hat{z}_i(t) = \begin{cases} \sin((post(t)+1)\frac{\pi}{2}) \cdot z_{max}, & \text{if swing} \wedge post(t) \leq 0, \\ \sin((-post(t)+1)\frac{\pi}{2}) \cdot z_{max}, & \text{if swing} \wedge post(t) > 0, \\ 0 \quad \text{if support}, \end{cases} \quad (10)$$

where z_{max} is the maximum step height. Unlike $\hat{z}_i(t)$, the coordinates $\hat{x}_i(t)$ and $\hat{y}_i(t)$ are influenced by the turning radius *turn*. In order to turn the hexapod, each leg has to move along an arc of a circle going through the foot default position and having center at the turn point, which is located on a line given by the default foot-tips positions of the middle legs. Therefore, only a single parameter, the turning radius *turn*, is sufficient to parametrize the robots' trajectory. Figure 6 shows how the *turn* parameter influences the foot-tip trajectories of the middle legs.

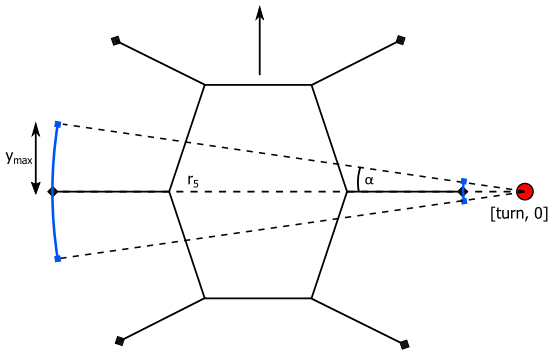


Figure 6: Trajectory generation – the turning point denoted as red circle is given and α is computed with the distance of the turning point to the furthest middle leg and with defined step length. α is then used to specify the trajectory of each foot-tip.

The next step of the trajectory generation is to find an angle determining the arc used for the foot-tips locations, because the step length is limited by the robot and leg construction. y_{max} is experimentally set to 35 mm, which means a leg can move about this distance in both forward and backward directions, i.e., the step length is at most 70 mm. When turning, each leg has a different step length and the longest step is always performed by the furthest leg j from the turning center

$$j = \text{argmax}_{j=\{2,5\}}(|P_{x_j} - turn|), \quad (11)$$

where P_{x_j} is the x-coordinate of the default foot-tip with respect to the robots' center. The angle α is then given as

$$\alpha = \text{asin}\left(\frac{y_{max}}{(P_{x_j} - turn)}\right). \tag{12}$$

Now, the arc radius r_i is computed for each leg foot-tip $i \in \{1,2,\ldots,6\}$ and the angle offset θ_{rad} given by the default leg position as

$$\theta_{rad_i} = \text{atan2}\left(P_{y_i}, P_{x_i} - turn\right), \tag{13}$$

$$r_i = \sqrt{(P_{x_i} - turn)^2 + (P_{y_i})^2}. \tag{14}$$

Then, the foot-tip location is computed

$$\hat{x}_i(t) = r_i \cos(\theta_{rad_i} + \alpha post(t)), \tag{15}$$
$$\hat{y}_i(t) = r_i \sin(\theta_{rad_i} + \alpha post(t)). \tag{16}$$

3.4 Inverse Kinematics

The last step of the trajectory generation is to transform the generated foot-tip positions into the joint angles which can be directly send to the actuators. The input of our inverse kinematic task (IKT) module are foot coordinates in the leg reference system. For brevity, the inputs $x_{foot_i}^{leg_i}$, $y_{foot_i}^{leg_i}$, $z_{foot_i}^{leg_i}$ are written as x_i, y_i, z_i in this section, where i stands for leg index. The coxa joint angles can be computed as

$$\theta_{1_i} = \text{atan2}\left(y_i, x_i\right) - \theta_{1_i}^{off}. \tag{17}$$

The other two joint angles are calculated using the following formulas. Note, the provided formulas do not hold in the whole operational space of the legs; however, they are correct for the foot-tip positions generated by our locomotion controller within the restricted operational space. The joint angles are determined as follows.

$$\theta_{2_i} = k_i(\beta + \omega_i) - \theta_{2_i}^{off}, \tag{18}$$
$$\theta_{3_i} = k_i(\gamma - \pi) - \theta_{3_i}^{off}, \tag{19}$$

where

$$\beta = \text{acos}\left(\frac{-l_3^2 + l_2^2 + d^2}{2 l_2 d}\right), \tag{20}$$

$$\gamma = \text{acos}\left(\frac{-d^2 + l_2^2 + l_3^2}{2 l_2 l_3}\right), \tag{21}$$

$$\omega_i = \text{atan2}\left(z_i, \sqrt{(B_{x_i} - x_i)^2 + (B_{y_i}) - y_i)^2}\right). \tag{22}$$

In which

$$d = \sqrt{(B_{x_i} - x_i)^2 + (B_{y_i} - y_i)^2 + (B_{z_i} - z_i)^2} \tag{23}$$

and B_i is the location of the i-th leg femur joint in the leg coordinates that can be computed as the first three values of

$$B_i = R_z(\theta_{1_i} + \theta_{1_i}^{off}) T_x(l_1) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} l_1 \cos\left(\theta_{1_i} + \theta_{1_i}^{off}\right) \\ l_1 \sin\left(\theta_{1_i} + \theta_{1_i}^{off}\right) \\ 0 \\ 1 \end{bmatrix}, \tag{24}$$

where $R_z(\cdot)$, $T_x(\cdot)$ stands for the rotation around z-axis and translation along x-axis, respectively. Joint offsets caused by the mechanical construction are listed in Table 2 among with k_i. The signs are influenced by the "asymmetric" behaviour of the left and right legs along with the fact that the x-axis of the leg coordinate system is heading right no matter whether it is on the hexapod right or left side. Values of θ_2^{off} and θ_3^{off} are obtained from the leg dimensions.

Table 2: IKT Constants

Leg	k_i	$\theta_{1_i}^{off}$	$\theta_{2_i}^{off}$	$\theta_{3_i}^{off}$
1 (RR)	1	$-\pi/4$	θ_2^{off}	θ_3^{off}
2 (RM)	1	0	θ_2^{off}	θ_3^{off}
3 (RF)	1	$\pi/4$	θ_2^{off}	θ_3^{off}
4 (LR)	-1	$-3\pi/4$	$-\theta_2^{off}$	$-\theta_3^{off}$
5 (LM)	-1	π	$-\theta_2^{off}$	$-\theta_3^{off}$
6 (LF)	-1	$3\pi/4$	$-\theta_2^{off}$	$-\theta_3^{off}$

4 Experimental Evaluation

The conducted experiments focus on the evaluation of the controller performance to follow trajectories consisting of straight line segments and arcs of circles with various radii and different gait types.

4.1 Walking Platform

The proposed locomotion controller is considered with an off-the-shelf hexapod walking robot *PhantomX Hexapod Mark II* (depicted in Figure 1). The robot consists of six legs each granting three Degrees-Of-Freedom (DOF), summing up to 18 controllable DOF for the whole robot. That gives us great maneuverability and agility that is useful for obstacle traversing and rough terrain walking. Each leg has three revolute joints motorized by the Dynamixel AX series intelligent servo motors. The servos are connected in daisy chain and communicate using a serial interface.

4.2 Results

The robot real trajectory is captured by a vision-based external localization system [19], which grants us an ability to measure the robot forward velocity for different gaits

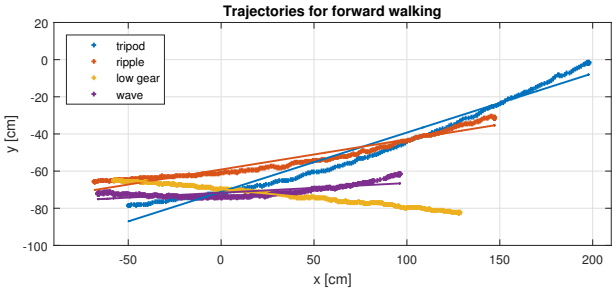


Figure 7: Trajectories of the forward motion

and quantify the relative radius error (RRE) given a particular value of the *turn* parameter. Thus, the herein presented empirical evaluation of the proposed stochastic-based CPG motion controller is focused on two control parameters: the period p and the turning radius *turn*.

The tracked body position is denoted as $X^{glob}(t) = [O_x^{glob}(t), O_y^{glob}(t)]$ with $t = 0, 1, \ldots N$, where N is the number of frames taken by the localization system during the experiment. Having the robot position in time, we can estimate the forward velocity for individual gaits defined by the parameter p as

$$v = \frac{\|X^{glob}(1) - X^{glob}(N)\|}{t_r} \quad [m \cdot s^{-1}], \qquad (25)$$

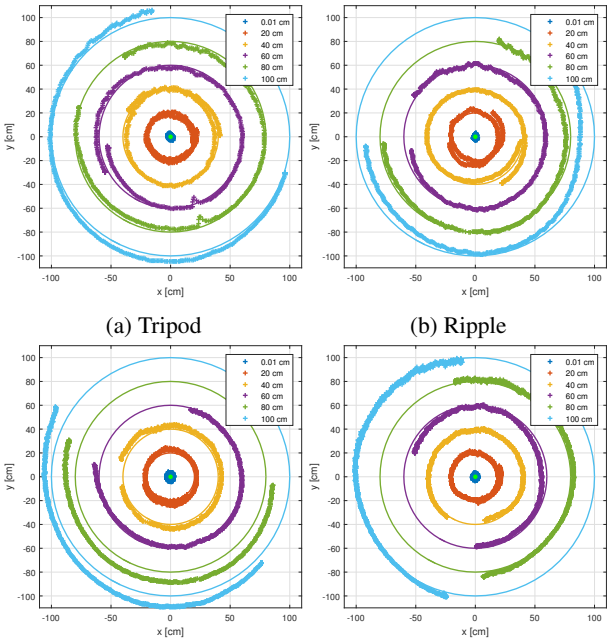
where t_r is the experiment runtime. The straight forward motion is achieved by setting the parameter *turn* to very high value, e.g., 10^9; however, too high values may cause numerical instability. The robot does not perfectly follow a straight line path, see Figure 7, due to the gaits and mechanical imprecision. Therefore, the forward velocity is established from a line fitted to the captured data. The established forward velocities for each individual gait are reported in Table 3.

Table 3: Forward Velocity

p	4	6	8	12
Gait	Tripod	Ripple	Low gear	Wave
Speed [m · s^{-1}]	0.172	0.097	0.062	0.036

The value of the *turn* parameter equals to the radius of the circle the hexapod body should follow and it is denoted as r_{ref}. Different values of *turn* allows to achieve various circular motion as rotation on the spot or crawling an arc trajectory of specified radius. Therefore, the robot has been requested to crawl circular trajectories with the radius 0.01 cm, i.e., rotating on the spot, and radii $r_{ref} \in \{20, 40, 60, 80, 100\}$ cm. A circle fitting method proposed by Pratt [20] is utilized to estimate the real circular trajectory with the radius r_{fit} and center $S = [S_x^{glob}, S_y^{glob}]$ in the global coordinate system of the localization system. The relative radius error (RRE) η_{turn} is used to quantify the robot performance in crawling circular trajectories as

$$\eta_{turn} = \frac{|r_{fit} - r_{ref}|}{r_{ref}} \cdot 100 \quad [\%]. \qquad (26)$$



(a) Tripod (b) Ripple

(c) Low gear (d) Wave

Figure 8: Trajectories for circular motion and different gaits. Crosses show the tracked hexapod body positions and the circles the reference trajectory. Trajectories and centres are aligned to have their center in the plot origin.

Particular trajectories for each gait type are depicted in Figure 8, where the detected hexapod positions are marked with crosses and reference orbits are drawn with the solid line. The resulting trajectories are collocated to share the same orbit point. The established η_{turn} are listed in Table 4 for $r_{ref} \geq 20$ cm, since for rotating on the spot η_{turn} would give too large number without carrying any information because of the division with small r_{ref}.

Table 4: Relative radius error η_{turn} for circular motion

turn [cm]/p	4	6	8	12
20	0.83 %	9.89 %	11.63 %	1.34 %
40	0.92 %	0.41 %	7.54 %	1.42 %
60	0.40 %	0.97 %	0.32 %	4.60 %
80	2.78 %	1.78 %	10.00 %	2.32 %
100	3.15 %	5.37 %	7.23 %	0.54 %

Discussion

The presented results indicate that the circular locomotion can be controlled with the proposed method, i.e., the robot is able to follow the orbit defined by the *turn* parameter with the error not exceeding more than 2.5 % for the majority of combinations of gait types and turn ratios. The outliers, e.g., for $p = 6$ and *turn* $= 20$ cm, are caused by a leg slippage on the floor and other factors like the experiment imperfections. The results obtained for wave gait

tend to have greater relative errors which can indicate that the "period stabilization" might have spoiled the resulting trajectory. The stabilization is a process in which the robot does not exactly have to be walking as desired and its legs move in a way not resulting in locomotion specified by the control parameters. It even may turn during forward locomotion as it can drag wrong legs while they should be swinging. The uncertain or even wrong foot coordination during the stabilization is caused by the time it takes the chaotic oscillator producing the patterned outputs to achieve oscillations with the specified period. This phenomenon is hardly noticeable for gaits other than wave as it takes less than a second, but in the case of the wave gait, it takes up to 4 seconds to achieve the desired motion, which is caused by the period stabilization.

5 Conclusion

In this paper, we have proposed a locomotion controller for a hexapod walking robot based on combination of the biologically inspired approach with a trajectory generator and inverse kinematics. The proposed controller is capable of controlling the hexapod locomotion by tuning only two parameters: the period p which modulates the chaotic CPG in order to stabilize different periodic orbits, hence different motion gaits, and the turning radius *turn* which allows the robot to perform different movements from turning on the spot to straight-forward walking. The designed control method has been evaluated using the real hexapod walking robot. The presented results of the experimental evaluation show that the proposed controller is plausible and can be used for walking on flat surfaces. The key concept in development has been to keep the controller modular to be easily usable with additional sensors. Therefore, our future work is to consider sensor feedback in the trajectory generator to allow the robot to negotiate rough terrains and autonomously set-up the control parameters. Also, chaotic state of the oscillator could be exploited for leg untrapping.

Acknowledgments – This work was supported by the Czech Science Foundation (GAČR) under research project No. 15-09600Y. The support of the Grant Agency of the CTU in Prague under grant No. SGS16/235/OHK3/3T/13 to Petr Čížek is also gratefully acknowledged.

References

[1] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: a review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[2] T. G. Brown, "On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system," *The Journal of Physiology*, vol. 48, no. 1, pp. 18–46, 1914.

[3] D. M. Wilson, "The central nervous control of flight in a locust," *J. exp. Biol*, vol. 38, no. 47, pp. l–490, 1961.

[4] M. H. Dickinson, C. T. Farley, R. J. Full, M. Koehl, R. Kram, and S. Lehman, "How animals move: an integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, 2000.

[5] J. Proctor and P. Holmes, "Reflexes and preflexes: on the role of sensory feedback on rhythmic patterns in insect locomotion," *Biological cybernetics*, vol. 102, no. 6, pp. 513–531, 2010.

[6] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current biology*, vol. 11, no. 23, pp. R986–R996, 2001.

[7] D. M. Wilson, "Insect walking," *Annual review of entomology*, vol. 11, no. 1, pp. 103–122, 1966.

[8] N. Porcino, "Hexapod gait control by a neural network," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1990, pp. 189–194.

[9] 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, no. 3, pp. 224–230, 2010.

[10] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on cpg-inspired control models and system implementation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 441–456, 2014.

[11] R. Haschke, "Bifurcations in discrete-time neural networks: controlling complex network behaviour with inputs," Ph.D. dissertation, Bielefeld University, 2003.

[12] F. Pasemann, "Complex dynamics and the structure of small neural networks," *Network: Computation in neural systems*, vol. 13, no. 2, pp. 195–216, 2002.

[13] K. Matsuoka, "Sustained oscillations generated by mutually inhibiting neurons with adaptation," *Biological cybernetics*, vol. 52, no. 6, pp. 367–376, 1985.

[14] W. Chen, G. Ren, J. Wang, and D. Liu, "An adaptive locomotion controller for a hexapod robot: CPG, kinematics and force feedback," *Science China Information Sciences*, vol. 57, no. 11, pp. 1–18, 2014.

[15] L. Xu, W. Liu, Z. Wang, and W. Xu, "Gait planning method of a hexapod robot based on the central pattern generators: Simulation and experiment," in *ROBIO*, 2013, pp. 698–703.

[16] G. L. Liu, M. K. Habib, K. Watanabe, and K. Izumi, "Central pattern generators based on matsuoka oscillators for the locomotion of biped robots," *Artificial Life and Robotics*, vol. 12, no. 1-2, pp. 264–269, 2008.

[17] M. Frasca, P. Arena, and L. Fortuna, *Bio-inspired emergent control of locomotion systems*, ser. Nonlinear Science A. World Scientific, 2004, vol. 48.

[18] H. Rostro-Gonzalez, P. Cerna-Garcia, G. Trejo-Caballero, C. Garcia-Capulin, M. Ibarra-Manzano, J. Avina-Cervantes, and C. Torres-Huitzil, "A CPG system based on spiking neurons for hexapod robot locomotion," *Neurocomputing*, vol. 170, pp. 47–54, 2015.

[19] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, and M. Mejail, "A practical multirobot localization system," *Journal of Intelligent & Robotic Systems*, vol. 76, no. 3-4, pp. 539–562, 2014.

[20] V. Pratt, "Direct least-squares fitting of algebraic surfaces," *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 145–152, 1987.