Experimental study of limit cycle and chaotic controllers for the locomotion of centipede robots

Loïc Matthey, Ludovic Righetti and Auke Jan Ijspeert

Abstract— **In this contribution we present a CPG (central pattern generator) controller based on coupled Rössler systems. It is able to generate both limit cycle and chaotic behaviors through bifurcation. We develop an experimental test bench to measure quantitatively the performance of different controllers on unknown terrains of increasing difficulty. First, we show that for flat terrains, open loop limit cycle systems are the most efficient (in terms of speed of locomotion) but that they are quite sensitive to environmental changes. Second, we show that sensory feedback is a crucial addition for unknown terrains. Third, we show that the chaotic controller with sensory feedback outperforms the other controllers in very difficult terrains and actually promotes the emergence of short synchronized movement patterns. All that is done using an unified framework for the generation of limit cycle and chaotic behaviors, where a simple parameter change can switch from one behavior to the other through bifurcation. Such flexibility would allow the automatic adaptation of the robot locomotion strategy to the terrain uncertainty.**

I. Introduction

The control of legged locomotion on unknown uneven terrains is not yet a solved problem. In this case, model based controllers are difficult to use since it is extremely hard to maintain an accurate model of the environment. Recently researchers have proposed to use Central Pattern Generators (CPGs), taking direct inspiration from biology, to generate control policies for locomotion of legged robots. In robotics these CPGs are often modeled as coupled dynamical systems, mostly oscillators. The advantages of CPGs are their stability properties (limit cycle behavior) and synchronization abilities (e.g. coordination between the legs and/or coupling to the body and the environment). Moreover since these methods are model free and since limit cycles with sensory feedback help to deal with perturbations, they are well adapted to locomotion in unknown environments. CPG based controllers for salamander robots [1], quadruped robots [2], [3] and humanoid robots [4], [5] are a few examples of successful applications.

As an alternative to CPGs, Kuniyoshi et al. [6] have proposed the use of chaotic controllers coupled to the robot and the environment via sensory feedback to generate control policies. They showed that it was possible to have emergent coordinated patterns that would change and reorganize according to environmental changes. This allows for a fast exploration of possible patterns of coordination, with sensory feedback stabilizing the useful patterns. The use of chaos to design controllers is spreading slowly [7].

This work was supported by the European Commission's Cognition Unit, project no. IST-2004-004370: RobotCub (L.R.) and by a grant from the Swiss National Science Foundation (L.R. and A.I.).

School of Communication and Computer Science, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland loic.matthey@epfl.ch, ludovic.righetti@a3. epfl.ch, auke.ijspeert@epfl.ch

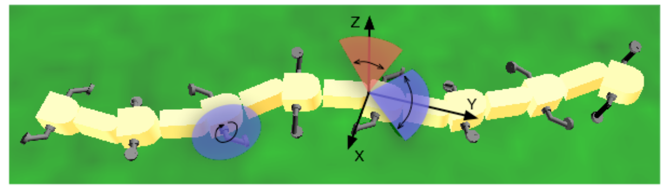

Fig. 1. Model of the Centipede robot in the physics simulator Webots. Consists of 8 body segments, each equipped with 2 rotary legs. Z axis goes up.

However, several questions arise from these experiments. First, how can the structural properties of the chaotic system be used to enhance exploration? Specifically, to what extent is the use of chaotic controllers to explore new coordination patterns more efficient than just using controllers that generate random signals and are directed by some sensory feedback? Second, does the use of chaotic systems provide more flexibility to the controller, as compared to limit cycle systems with adequate sensory feedback loops? Third, is the choice of appropriate sensory feedback pathways more important than the underlying feedforward controller (e.g. chaotic or limit cycle controller)?

In this work, we try to partially answer these questions by experimentally exploring the advantages of using chaotic controllers to locomote in uneven terrains compared to limit cycle systems and purely random movement generators. To do so we design a CPG controller based on coupled Rössler systems. The system is able to generate stable limit cycles for efficient locomotion on flat terrains. Then by a simple parameter change the system can bifurcate to a chaotic regime. Our goal is to propose a controller that uses limit cycle behavior for steady-state locomotion, and chaos for dealing with complex terrain and getting out of difficult situations. We use a simulated centipede robot (with 16 legs and a total of 32 degrees of freedom) locomoting on terrains of increasing difficulty. For each setup we also explore the importance of sensory feedback in the control loop.

We start by presenting our robot model, the CPG controller based on Rössler systems and all controllers compared in the experimental study. We then introduce the methodology used in the test bench. Finally, we comment on the results obtained for different environments' difficulty and the implications on the movement patterns created.

II. Model

We start by presenting the simulated robot and the different controllers we use. First, we describe the CPG based on

Rössler systems that generates a limit cycle and show how to switch it to chaotic behavior. Then, we present the random movement controllers used in the test bench. Finally, we describe how to integrate sensory feedback into our controllers.

A. Centipede robot

For this experimental study, we use a simulation of a centipede robot, see Fig. 1. We choose this type of robot because of the large variety of gaits its many degrees of freedom allowed. Moreover, it does not require complex controllers to maintain equilibrium, being very close to the ground (as opposed to humanoid or quadruped robots). The robot consists of 8 similar body segments. Each segment has the following active DOFs: two independent rotating legs and a joint connected to the next segment. This joint actively rotates in the XY-plane for movement and passively rotates in the YZ-plane to allow compliance on uneven ground. All active DOFs are implemented as position servomotors (i.e. are controlled by PD control loops that receive desired positions from the locomotion controllers). There are pressure sensors on the feet of the robot.

Each segment is based on an existing module that was used for a salamander robot previously introduced in [1] and [8], except for the pressure sensors that we added to the feet in this work. We perform our study using Webots [9], which is a simulator based on ODE [10], an open source physics engine for simulating 3D rigid body dynamics. The model of a module has the same DOF's, mass distribution and inertia matrix as a real one. A real implementation is possible in future works.

B. Limit cycle controller

We want to reproduce with a CPG the movements used by real centipedes, like *scolopendra heros*. They move by propagating traveling waves along their legs while undulating their body lightly [11]. We chose to build a CPG that could switch between limit cycle or chaotic behavior with a parameter-controlled bifurcation. We use Rössler systems as basic oscillators, as they offer this bifurcation feature. They also are well studied systems, with known parameter ranges, behaviors and synchronization possibilities. Moreover, they create simple limit cycles, easy to transform into motor commands for our robot.

We use the classical system definition of Rössler systems, with the addition of a scaling term ω to control the frequency of the oscillators. The equations are as follows:

$$\left\{ \begin{array}{lcl} \dot{x_i} & = & \omega\big(-(y_i+z_i)+\sum_j C(\vec{x}_i,\vec{x}_j)+\Gamma_i\big) \\ \dot{y_i} & = & \omega\big(x_i+ay_i\big) \\ \dot{z_i} & = & \omega\big(b+z_i(x_i-c)\big) \end{array} \right. \quad (1)$$

Where $i \in \{1,...,24\}$ is the oscillating element index and a, b and c are free parameters (set to $a = 0.2$, $b = 3.0$ and $c = 5.7$ for a limit cycle behavior). By changing the b parameter, the system can undergo a bifurcation that drives the system into chaotic regime. Γ_i is the sensory feedback for element i. The frequency of oscillations is defined by $F = \frac{\omega}{2\pi}$ Hz.

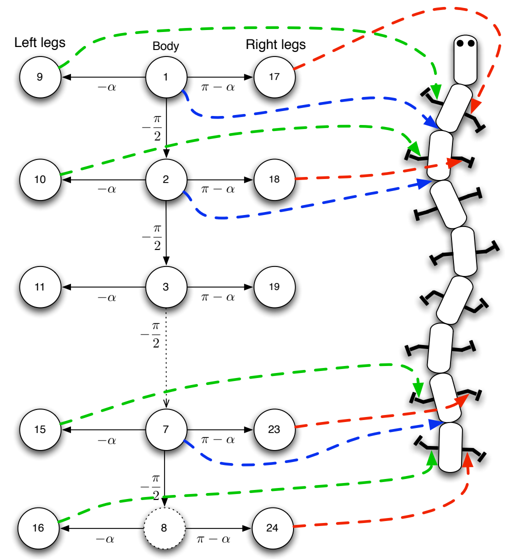

Fig. 2. Rössler CPG for centipede robot locomotion. Create a nature-like movement pattern which is then mapped to the centipede robot. Arrows show enforced phase differences. α is an open parameter controlling the phase lag between the body and the limbs.

$C(\vec{x}_i,\vec{x}_j)$ defines the coupling between two oscillators. This coupling term allows us to couple two Rössler elements with any phase difference desired by taking advantage of the geometric properties of the phase plane. The idea is to rotate one phase space with respect to the other, and then to use diffusive coupling. For the Rössler oscillator, we can approximate the phase plan by projecting the state variables on the (x, y) plane (i.e. use (x_i, y_i) directly). This is generalizable to every dynamical system if we can define a phase plane where the rotation has to take place. The coupling can then be written as:

$$C(\vec{x}_i,\vec{x}_j) = K \cdot \big((cos\phi \cdot x_j - sin\phi \cdot y_j) - x_i\big) \quad (2)$$

where $\vec{x}_j$ is rotated by an angle ϕ on the (x, y) plane and K is a gain parameter ($= 0.5$ in our case). Note that we do not use the coupling term on the y variables as it introduced instabilities (due to competition between oscillators), without improving the convergence time.

We use Eqs. (1) with a Rössler oscillator element for every motor and specific phase differences to create a movement pattern for our robot (Fig. 2). Body segments have a phase difference of $\frac{\pi}{2}$, producing a traveling wave along the body. Legs are set to be in anti-phase with a open bias α by design choice and in accordance to nature. α is a parameter controlling the phase lag between the body and the limbs (see Fig. 2).

The body joints receive the following position command: $\chi_i = A\tilde{x}_i$ (sine-like signal). We can control the movement by modifying the amplitude of body oscillations A (scaling the output of the CPG), the frequency of the oscillators and tuning the α parameter. $\tilde{x}_i$ is a dynamic normalization of the signal x_i in a moving window of 50 samples. We need to normalize x_i because of amplitudes variations and because our motors take input values in a fixed range. The legs are in permanent rotation following a phase signal calculated using x_i and y_i, $\Theta_i = atan(\frac{y_i}{x_i})$ (this gives an angle of rotation).

We implemented three types of limit cycle controllers. The *limit cycle* controller uses arbitrary chosen A and α parameters, without sensory feedback ($\Gamma_i = 0$). *Limit cycle with sensory feedback* uses the same parameters but with sensory feedback (explained in Section II-E). For the *optimized limit cycle* controller, we performed a systematic exploration of the amplitude A and the α parameter, for fixed frequencies of oscillations, to find the gait with highest speed on flat terrains. The obtained optimized movement pattern closely resembles the one obtained for *scolopendra heros*[11], except its body amplitude is bigger. We think this is due to the smaller mobility of our robot's legs, which is compensated by bigger body oscillations increasing the accessible space for each foot.

C. Chaotic controller

We make the Rössler CPG bifurcate to the chaotic regime by setting $b = 0.2$. We use the same number of oscillators and relations to the robot motors. As we want to know if chaos could make efficient locomotion patterns emerge, we want our controller to be as unconstrained as possible. Therefore, we remove coupling between the elements.

The only interactions between the chaotic elements will be through sensory feedback integration. The oscillators are thus indirectly coupled via their interaction with the outside world. This idea of indirect interaction is consistent with the previous work on chaotic controllers by Y.Kuniyoshi [6] [12].

D. Random controller

1) Pure random controller: The first random controller is a simple phase oscillator with randomly varying frequency:

$$\dot{\Theta}_i = \omega \cdot (2 \cdot U(0,1) - 1) \tag{3}$$

where $U(0,1)$ is a random number following a uniform distribution between 0 and 1 and ω controls the desired frequency with $F = \frac{w}{4\pi}$. Θ_i is updated every 32ms. We use an oscillator for each motor. The legs receive directly the angle Θ_i. The body elements receive $A\sin(\Theta_i)$ as angular position. They do not receive sensory feedback. This controller can make the legs go forward and backward.

2) Forward random controller: Since for locomotion, it is likely that the legs always rotate forward, we introduce another random oscillator:

$$\dot{\Theta}_i = \omega \cdot U(0,1) \tag{4}$$

The controls are sent to the robot as before.

E. Sensory feedback integration

We consider the feedback given by pressure sensors placed on every foot of the robot, and study its effect on different controllers. It is the easiest feedback that gives actual insight on the effect of movement patterns and terrain features. A rhythmic pressure sensory feedback indicates a constant movement pattern on an even terrain. More complicated terrains will introduce different patterns in the pressure sensory feedback. The signal coming from the pressure sensor is preprocessed and confined to a $[0;15]\frac{N}{m^2}$ range.

TABLE I
PARAMETER SETS OF THE CONTROLLERS FOR $F = 2Hz$.

	A	ω	α	K_f
Optimized limit cycle	1.6	6π	2.93	0
Limit cycle	0.8	6π	$-\frac{\pi}{2}$	0
Limit cycle with sensory feedback	0.8	6π	$-\frac{\pi}{2}$	0.5
Random forward	0.4	8π	–	–
Random pure	0.4	8π	–	–
Chaos without sensory feedback	0.3	6π	–	0
Chaos with sensory feedback	0.3	6π	–	-0.5

The sensor information from a given leg is sent to the Rössler system controlling this leg and the one controlling the opposite leg of the same segment. This creates a local feedback for legs. The body segments do not receive any feedback, as no good feedback scheme was found to create interesting patterns. We define the feedback as:

$$\Gamma_i = \begin{cases} K_f \cdot (T_i - T_k) & \text{If } i \text{ is a leg} \\ 0 & \text{If } i \text{ is not a leg} \end{cases} \tag{5}$$

T_i is the pressure sensor signal coming from the leg corresponding to oscillator i, T_k is the signal coming from the opposite leg on the same body segment. K_f is a signed open gain parameter.

This sensory integration is limited and can only periodically affect the oscillators. It can modify the swing time of one segment based on timing discrepancies of the sensory signals. It cannot be used to predict coming difficulties, but only reacts to instantaneous changes in the outside world.

III. EXPERIMENTAL SETUP

We want to measure quantitatively how these different controllers behave when they face unknown terrain. Obviously, when the terrain is known and simple, a limit cycle behavior will produce the best results. But when the environment becomes unknown, we do not have such insight anymore. Chaos could be a way to explore interesting patterns of coordination when facing uneven terrain or when stuck in a hole, for instance.

We developed a complete and precise test bench to assert this fact, with the following components:

1) An experiment area (see Fig. 3) consisting of a starting zone for the robot, a randomly generated uneven terrain as test area and extra flat terrain after that. The robot has to walk through the test area and beyond.
2) A defined difficulty level between 0.1 and 1.0 for the uneven terrain. The test area is generated automatically, by creating a mesh of triangles with randomly chosen heights. The derivative along successive heights is used to define the difficulty level. We use 10 difficulty levels, 0.1 being the simplest terrain and 1.0 the hardest.
3) 8 terrains are generated for each difficulty level. Runs are distributed equally between them.
4) 120 runs are done for each combination of difficulty level and controller. Controllers are initialized randomly.
5) The random controllers are used as references to see if the chaotic regime is more efficient than a random exploration.

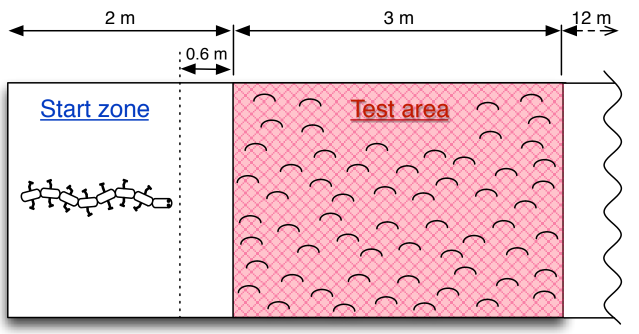

Fig. 3. Sketch of the Experiment area created for the study. The Test area consists of a triangle mesh with randomly generated heights. The derivative along successive heights is used as a difficulty level parameter.

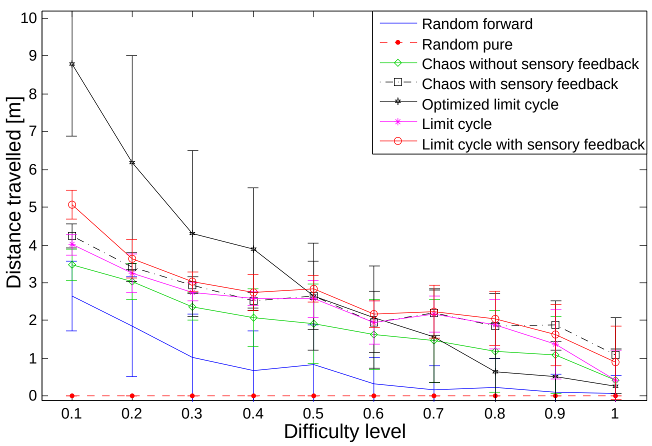

Fig. 4. Benchmark results for all controllers. We show the average distance traveled as a function of the terrain difficulty together with its standard deviation.

We remove as much as possible random side effects of the simulations by doing a large number of experiments. The controllers use fixed parameters defined in Table I. ω are chosen to get a common frequency $F = 2Hz$ for every controller. All the parameters α, A and K_f in Table I (except for *limit cycle* and *limit cycle with sensory feedback*, as explained before) have been found by performing an extensive parameter search. We systematically vary them for each controller in order to optimize the forward speed on flat terrain (over 20 runs).

IV. Results

A. Performance on uneven terrains

We measure the performance of the controllers as the distance traveled in a fixed period (20 seconds) on the different terrains[1]. The traveled distances for all controllers on the different worlds are shown in Fig. 4.

We see that:

- *Optimized limit cycle* performs best on easy terrain, but its performance drops dramatically while the difficulty level increases. It nearly does not move when the difficulty is higher than 0.7.
- *Limit cycle* behaves less well on easy terrain, but manages to keep acceptable performances when the difficulty level increases. We think this is due to the smaller body amplitude, which, while being sub-optimal on easy terrain, tends to be more efficient on uneven terrain.
- When adding sensory feedback to the *limit cycle*, we see an increase in performance on every world. This is an interesting result, as it shows that sensory information is used to increase the speed independently of the terrain configuration. It even manages to allow movement on the hardest terrain, where the other limit cycle controllers produce no real movement (see Fig. 5).
- The *random pure* controller cannot make the robot move on any terrain. The *random forward* controller has minimal performances on all difficulty levels, which confirms initial assumptions.
- The chaotic controllers have good performances, even when the difficulty increases. Moreover, when sensory feedback is present, the performance attains the one from a limit cycle controller, which is a surprising fact, since there is no explicit coordination between the elements of the controller. Compared to the network architecture of the *limit cycle*, with its carefully designed phase differences, this is quite an achievement. Another interesting feature is its high performance on very difficult worlds. Looking at a box-plot view of the performances on difficulty 1.0 (Fig. 5), we see that *chaos with sensory feedback* is the only controller with a positive median, indicating a good robustness on this difficulty.

[1] A video showing the obtained movement patterns and the experimental area is available on http://birg.epfl.ch/page67190.html

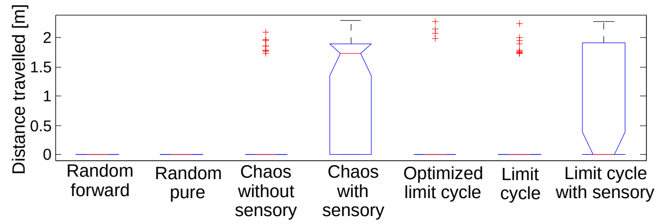

Fig. 5. Boxplot of the performances of controllers on terrain of difficulty 1.0. Horizontal bars are the median values, boxes shows the interquartile distance and crosses are outlier values.

The best performing controllers on uneven terrain are thus the ones with sensory feedback information, which indicates its importance on unknown terrains. Moreover, the chaotic controller does better on average than the limit cycle systems on worlds of difficulty higher than 0.6 (Fig. 4 and 5). It is interesting to note that the best performance on flat terrain is the limit cycle controller without feedback, which can be explained by the fact that in control theory feedforward controllers achieve the best performance in perfectly known environments but their performance degrades rapidly under unexpected disturbances.

B. Patterns of coordination for the different controllers

The movement pattern of the limit cycle system on flat terrain is easy to observe, as we have explicitly designed it. On the contrary, observing the movement pattern of the chaos controller with sensory feedback is more complicated, as no stable movement emerges, and because the state values are intrinsically hard to study. We cannot study directly the behavior of the oscillator in state space, as the sensory feedback

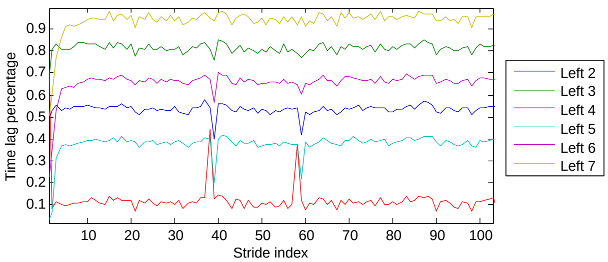

Fig. 6. Time lag for the left legs, with respect to the first left leg, for *optimized limit cycle* on flat terrain. Stride index is the number of the current stride. Distribution of time lags, indicating a traveling wave along the body.

adds unknown components to it. A simple visual observation did not reveal any stable structure in state space over time. But while observing the movement of the chaos controller, we detected short bursts of synchronization between legs of the same segment, so synchronization or patterns should exist (data not shown).

To find out about those patterns, we studied the temporal distribution of the touchdown of the different legs of our robot. We take inspiration from an approach for studying the coordination patterns during unsteady locomotion of mammals, proposed by Abourachid et al.[13]. We observe the times at which different legs touch the ground with respect to a reference leg, to define a dynamical way of expressing gaits. We take the first left leg as a reference. The intervals between two touchdowns of this leg define the dynamical strides. If we then express the moment of touch of the other legs in term of time lag percentage with respect to this interval, we can detect movement patterns. The time lag is just the difference between the time of touch of some leg with respect to the start of the current reference interval of the first left leg. By dividing by the duration of the current reference interval, we have the time lag percentage. A time lag percentage of 50% indicates "anti-phase" synchronization, and values around 0% or 100% indicate "in-phase" synchronization. A time lag percentage can be thought of as a dynamical gait definition.

Fig. 6 shows the time lag percentages for the left legs, with respect to the first left leg for the *optimized limit cycle* controller. The distribution of the percentages are in accordance with the traveling wave along the body. The time lags percentages are fairly constant, which indicate a stable movement pattern.

When applying the same methodology to the *chaos with sensory feedback* controller, we obtain quite different results. Fig. 7 shows in the upper part the time lag percentage of the first right leg with respect to the first left leg. This figure could show the possible synchronization in a same segment. It only reveals a big change of the time lag percentage performance over time, which indicates a strong irregularity of the movement patterns used by the chaos controller. But as can be seen in the lower part of the figure, the speed stays quite constant on average.

Another way to interpret the average presence of movement patterns is to show the distribution of time lag percentages using histograms. Typical results for the first right leg and second right leg, using the *optimized limit cycle*, the

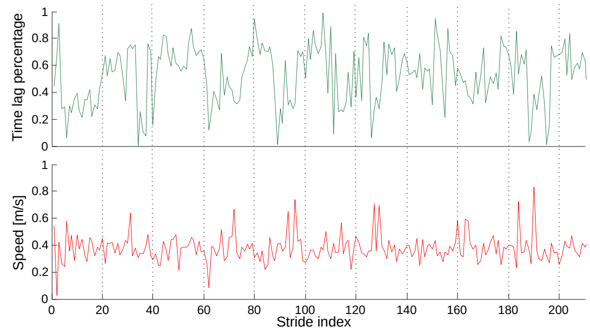

Fig. 7. Top: Time lag of first right leg for *chaos with sensory feedback* on flat terrain. Bottom: instantaneous speed of the robot.

limit cycle with sensory feedback and the *chaos with sensory feedback* controllers are shown in Fig. 8. From these figures, we see that:

- *Optimized limit cycle*: Uses nearly always the same movement pattern, with a 50% time lag for the Right leg, corresponding to anti-phase synchronization, and 75% time lag for the second right leg, which corresponds to the designed relation between segments. On difficult terrain, the robot gets stuck, so aberrant values appear (near 80%) and fewer strides are done in total, which decreases the relevance of those values.
- *Limit cycle with sensory feedback*: We find the previous time lags again, but now with greater variance around them. The number of successful strides is much higher than for the *optimized limit cycle*, which shows the effect of the sensory feedback on uneven terrain. For a difficult terrain, the controller manages to keep the desired patterns while exploring a little around them, which is sufficient if the terrain is not too complicated.
- *Chaos with sensory feedback*: We see that some movement patterns indeed exists for the this controller. For the first right leg on easy terrain, there is a Gaussian-like distribution of time lags with a mean of about 50%. Anti-phase synchronization between legs of the same segment is thus promoted, even if these synchronized patterns are quickly changing. This is still the case in difficult terrain. The results for the second right leg are harder to interpret; it seems that no real synchronization occurs between segments. This could be explained by the fact that no sensory feedback is exchanged between segments. This is more visible for the difficult terrain, where we have a quasi uniform distribution of time lag percentages. This distribution shows that the system explores a lot of possible coordination patterns without stabilizing to a specific one.

We thus showed that motion patterns are promoted by *chaos with sensory feedback*. Moreover, sensory feedback adds flexibility to the possible movement patterns since it allows exploration of these patterns, which is useful in uneven terrain.

V. Conclusion

In this work, we have developed a CPG controller for a centipede robot based on coupled Rössler systems. The same controller can be used to generate both limit cycle and

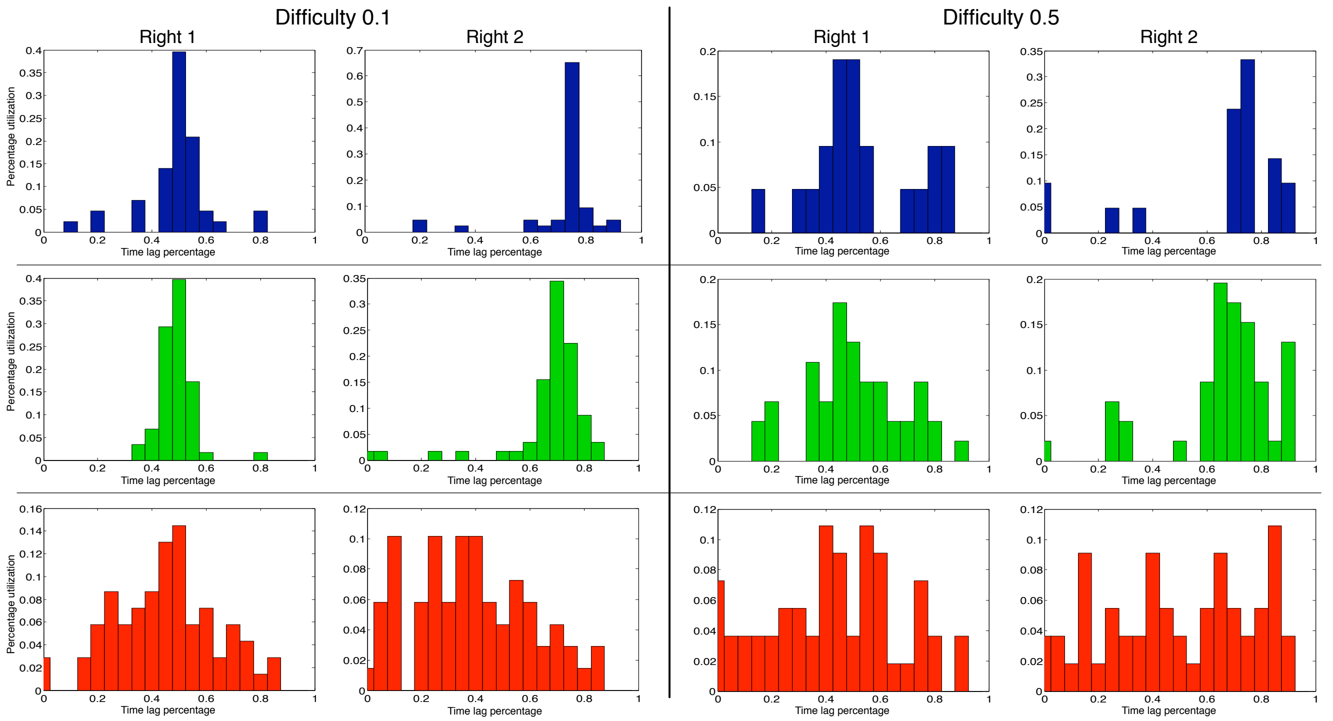

Fig. 8. Distribution of the time lag percentages used by the different controllers over a run of 30 sec. These time lags are calculated with respect to the first left leg, and are percentages of the dynamical strides of this leg. First row is the *optimized limit cycle*. Second row is the *limit cycle with sensory feedback*. Third row is *chaos with sensory feedback*.

chaotic behaviors by a simple parameter change, using the bifurcation characteristics of the Rössler elements. We added sensory feedback to the system.

First, comparison with random controllers show that our chaotic controller is able to explore new coordination patterns in difficult terrains in an efficient way, certainly exploiting some structural properties that are absent in random movement generators. Second, the chaotic system provides more flexibility in the exploration of patterns when compared to the limit cycle system and thus performs better in very difficult terrains. However, the limit cycle systems are still better on simple terrains. Third, appropriate sensory feedback loops are extremely important, independently of the type of controller, since on the very difficult terrains it was the only way to have a moving robot. Finally, since we use a unified framework for the different controllers, it is possible to choose the appropriate controller behavior according to the environment.

In future work it would be interesting to explore the relation between the structural properties of the chaotic controller and the sensory feedback loops, in order to be able to provide general design methodologies for chaotic controllers for locomotion. Extension of the test bench to other types of terrains would also be needed. Finally, we could make our robot switch automatically between limit cycle and chaos regime when stuck, to exploit the exploration capabilities of the chaotic regime to resume movement.

References

[1] 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, pp. 1416–20, Feb 2007.

[2] Y. Fukuoka, H. Kimura, and A. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, Jan 2003.

[3] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 819 – 824, Apr 2008.

[4] S. Aoi and K. Tsuchiya, "Locomotion control of a biped robot using nonlinear oscillators," *Autonomous Robots*, vol. 19, no. 3, pp. 219–232, Jan 2005.

[5] G. Endo, J. Nakanishi, J. Morimoto, and G. Cheng, "Experimental studies of a neural oscillator for biped locomotion with qrio," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 596 – 602, Mar 2005.

[6] Y. Kuniyoshi and S. Suzuki, "Dynamic emergence and adaptation of behavior through embodiment as coupled chaotic field," *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 2, pp. 2042 – 2049 vol.2, Jan 2004.

[7] S. Dreissigacker, T. Geisel, and F. Wörgötter, "Chaos control in neuromodules and application to gait generation," *Bernstein Center of University of Göttingen*, Jun 2007, master's Thesis.

[8] A. Ijspeert, "A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander," *Biological Cybernetics*, vol. 84, no. 5, pp. 331–348, Dec 2001.

[9] O. Michel, "Webots: Professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[10] Open dynamic engine documentation. [Online]. Available: http://www.ode.org

[11] B. Anderson, J. Shultz, and B. Jayne, "Axial kinematics and muscle activity during terrestrial locomotion of the centipede scolopendra heros," *J Exp Biol*, vol. 198, no. Pt 5, pp. 1185–95, Dec 1995.

[12] Y. Kuniyoshi and S. Sangawa, "Early motor development from partially ordered neural-body dynamics: experiments with a cortico-spinal-musculo-skeletal model," *Biological cybernetics*, vol. 95, no. 6, pp. 589–605, Dec 2006.

[13] A. Abourachid, M. Herbin, R. Hackert, and L. Maes, "Experimental study of coordination patterns during unsteady locomotion in mammals," *Journal of Experimental Biology*, vol. 210, pp. 366–372, Jan 2007.