Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation

Benjamin Ellenberger

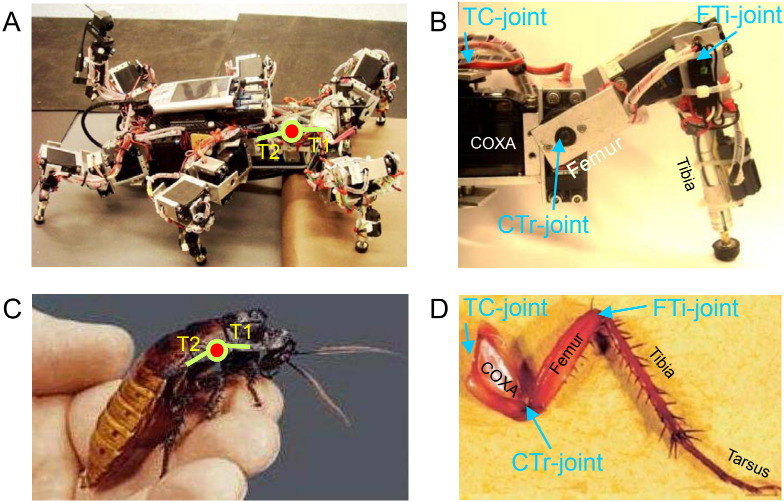
Abstract

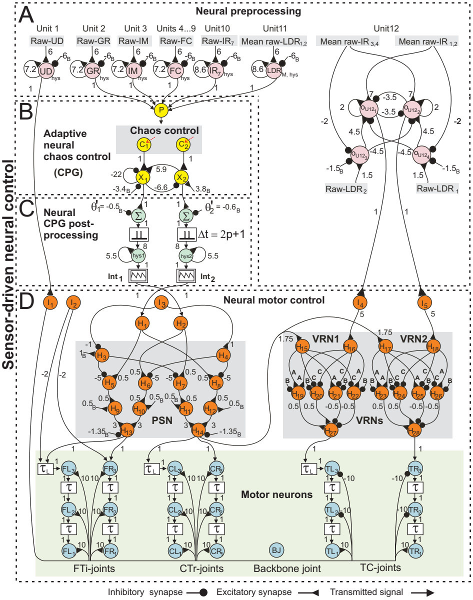
A B S T R A C T

An originally chaotic system can be controlled into various periodic dynamics. When it is implemented into a legged robot's locomotion control as a central pattern generator (CPG), sophisticated gait patterns arise so that the robot can perform various walking behaviors. However, such a single chaotic CPG controller has difficulties dealing with leg malfunction. Specifically, in the scenarios presented here, its movement permanently deviates from the desired trajectory. To address this problem, we extend the single chaotic CPG to multiple CPGs with learning. The learning mechanism is based on a simulated annealing algorithm. In a normal situation, the CPGs synchronize and their dynamics are identical. With leg malfunction or disability, the CPGs lose synchronization leading to independent dynamics. In this case, the learning mechanism is applied to automatically adjust the remaining legs' oscillation frequencies so that the robot adapts its locomotion to deal with the malfunction. As a consequence, the trajectory produced by the multiple chaotic CPGs resembles the original trajectory far better than the one produced by only a single CPG. The performance of the system is evaluated first in a physical simulation of a quadruped as well as a hexapod robot and finally in a real six-legged walking machine called AMOSII. The experimental results presented here reveal that using multiple CPGs with learning is an effective approach for adaptive locomotion generation where, for instance, different body parts have to perform independent movements for malfunction compensation.

Problem

Meet AMOS II


A
T2
T1
B
TC-joint
FTi-joint
COXA
Femur
Tibia
CTr-joint
C
T2
T1
D
TC-joint
FTi-joint
COXA
Femur
Tibia
CTr-joint
Tarsus



- A Preprocessing
  - Denoising
  - Shaping
  - Orientation signal preparation
- B Adaptive neural chaos control (CPG)
  - Discrete time dynamics
  - Sigmoid activation function
  - Additional control signal *c* based on period p appplied every p+1 time step
- C CPG post-processing
  - time window
  - hysteresis
  - signal integration
- D Neural motor control
  - Phase switching network (PSN)
  - Velocity regulation network (VRN)
- E Delay lines

Chaotic CPG overview

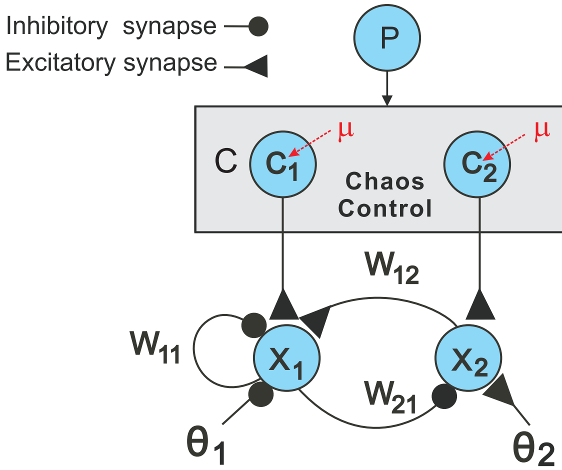


Fig. 1. Single CPG with the chaos controller.

CPG activity (discrete time dynamics)

Neuron state

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

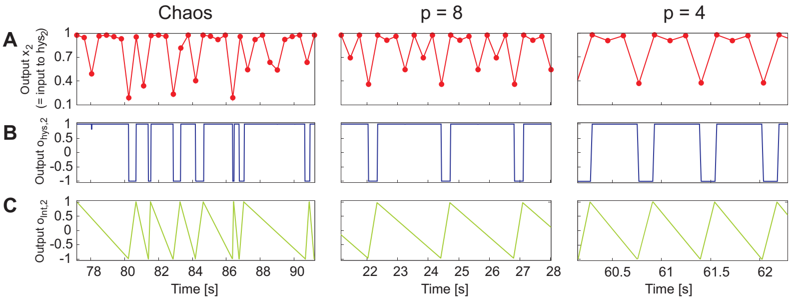
Control signal

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

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

$$\mu_{(p)} = \mu_{(p)}(t) + \lambda \frac{\Delta_1^2(t) + \Delta_2^2(t)}{p} \textit{ with } \lambda = 0.05$$

Postprocessing motor neuron signals



Supplementary Figure 3: **Postprocessing signals for different periods.** **(A)** Output signals of the time window function unit Δt. **(B)** Output signals of the hysteresis unit $hys2$. **(C)** Output signals of the signal integrator unit $Int2$.

Different hexapod gaits

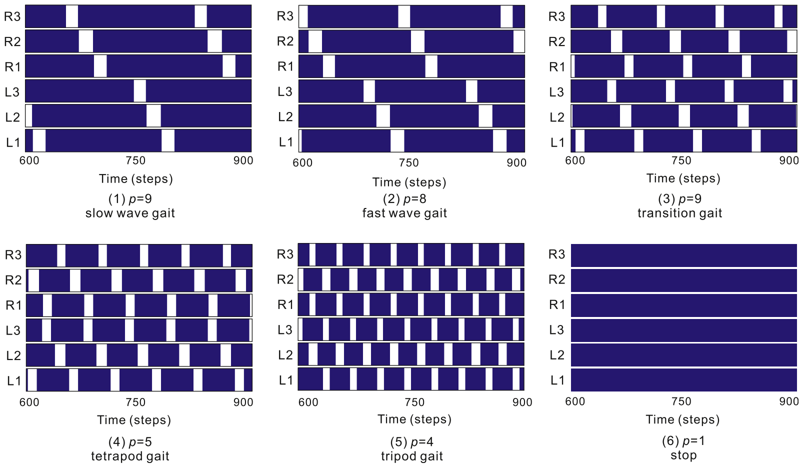
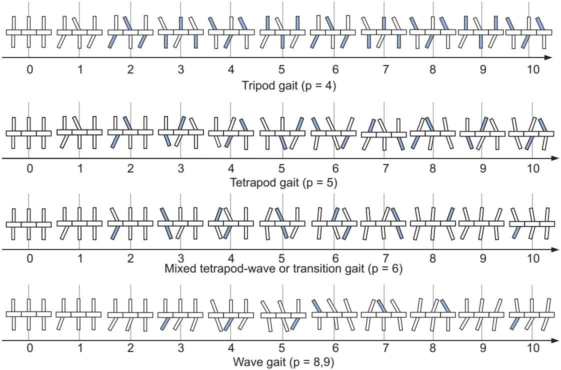


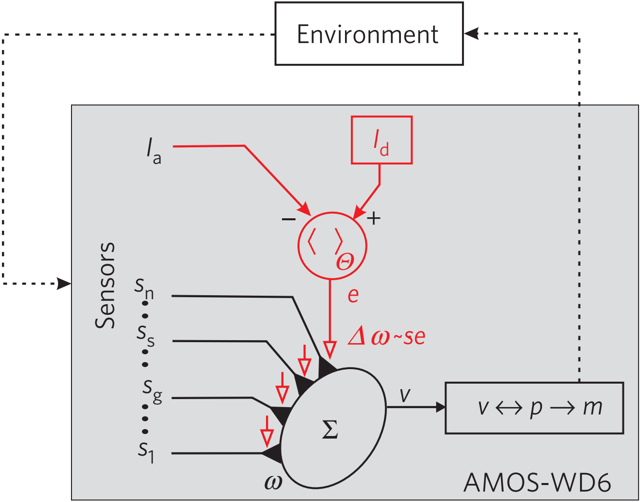
Fig. 2. Different hexapod gaits for changing p and the stop status ($p = 1$).

Sketched walking patterns

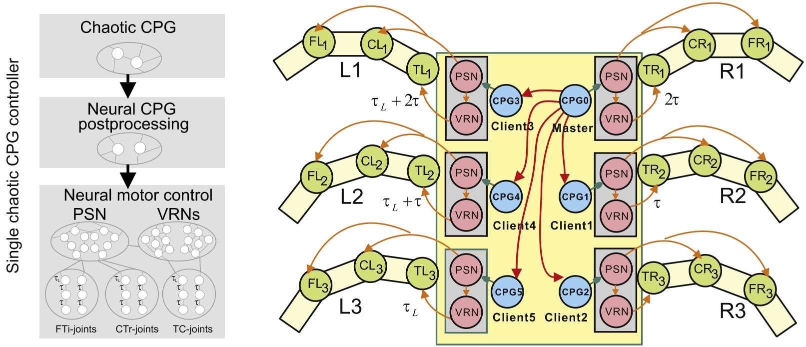


Supplementary Figure 4: **Sketch of different walking patterns according to target periods p.** A blue colored leg is in the air, a white leg is on the ground. To keep the images concise, only very coarse schemes are shown that do not take into account the different timings of swing and stance phases.

Motor pattern learning neuron



Multi CPG extension



(a) Single CPG controller (b) Multiple CPGs controller

Fig. 3. Single chaotic CPG (a) and multiple chaotic CPGs (b) for a multi-legged robot.

Client CPG

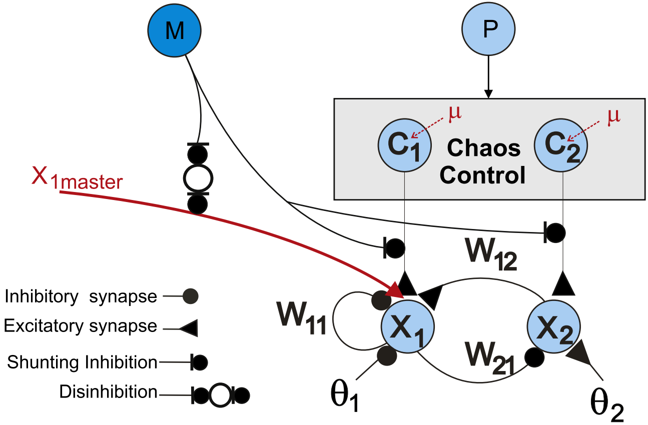


Fig. 4. The inner structure of the client CPG.

Synchronous and asynchronous CPGs

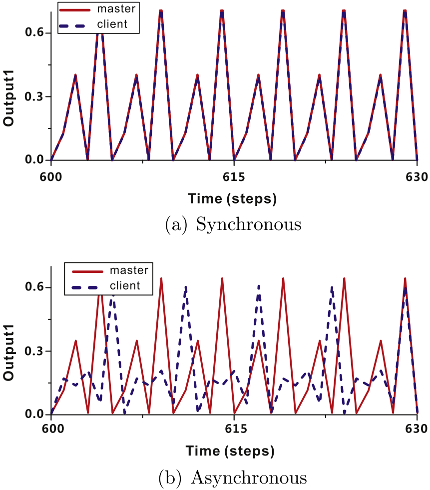
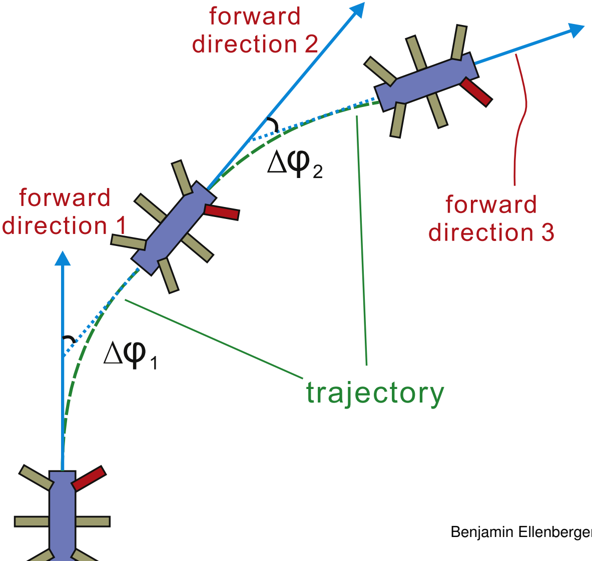


Fig. 5. The outputs of the CPG network for (a) synchrony and (b) asynchrony.

Trajectory deviation measure



Simulated annealing for leg period learning

initialize $C(1) = [1/41/24; 4; 4; 4; 4; 4]$; $\Delta\phi = 0.0$; $E_1 = 0.0$
repeat:
    At repetition n
    do
        randomly pick a leg l, $l \in [R1; R2; R3; L1; L2; L3]$
        change the period of leg l to a random value, $P(l) \in [4; 5; 6; 8; 9]$
        compare this combination of leg periods, $C'(n)$, to the leg periods of $C(n-1)$
    until $C'(n)$ is a new combination of leg periods

    run the robot
    // calculate the evaluation function and its variation
    $E_n = \Delta\phi$
    $\Delta E = E_n - E_{n-1}$

    // choose the combination of leg periods
    if $\Delta E < 0$ ***then***
        $C(n) = C'(n)$
    else
        if $X \geq e^{-\beta\Delta E}$ ***then*** // $X \in [0; 1]$, *learning rate* β
            $C(n) = C'(n)$
        else
            $C(n) = C(n-1)$
        end if
    end if
until: The evaluation function E_n is less than a required value E_{req}

β Tuning (Changing the acceptance rate)

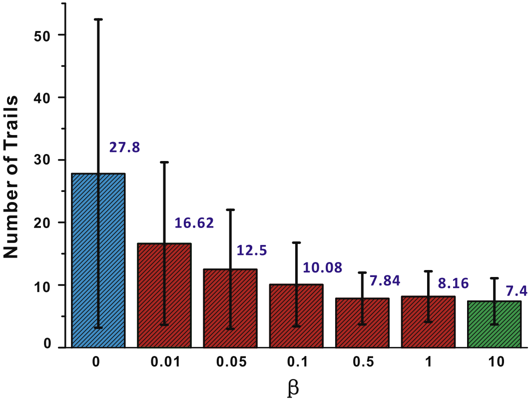


Fig. 14. The average number of trials with different β (see text for details).

Learning process

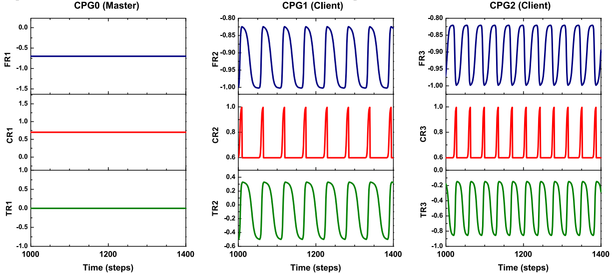
NO.	R1	R2	R3	L1	L2	L3	Deviation Angle (degree)	Decision
0	4	4	4	4	4	4	37.8604	Start
1	4	4	4	5	4	4	21.4578	Keep
2	4	4	9	5	4	4	64.1167	Return to No.1
3	4	4	4	5	9	4	14.0168	Keep
4	4	4	4	5	6	4	9.5276	Keep
5	4	5	4	5	6	4	9.6890	Keep
6	4	5	4	5	6	8	21.8642	Return to No.5
7	4	5	4	5	6	5	0.2451	End

Fig. 8. The learning process for one scenario (see text for details).

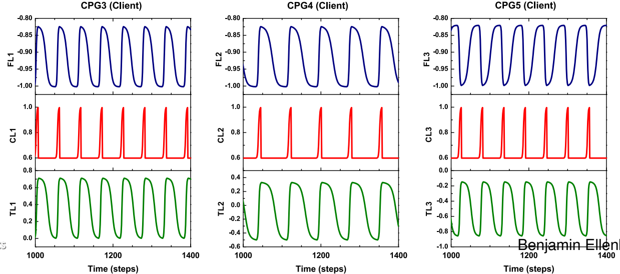
Different leg disabilities

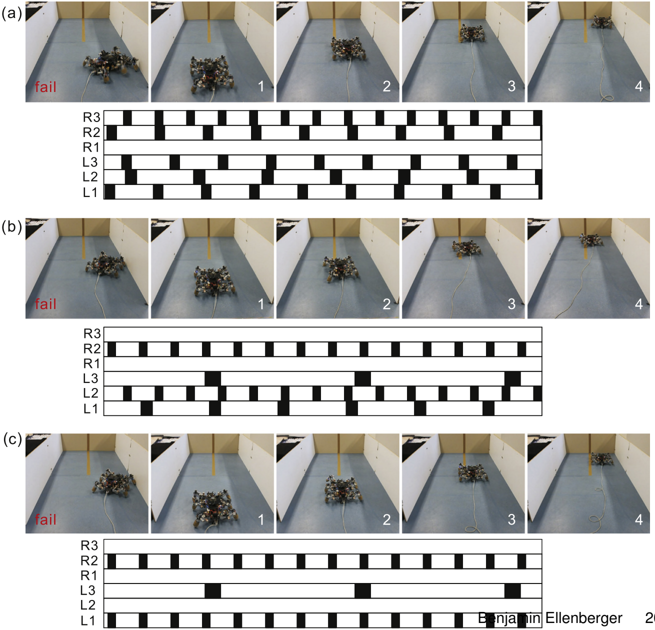
Leg status	Disabled leg(s)	Example of periods after learning	Average deviation angle (deg) + SD	Trials + SD
	R1	4, 5, 4, 5, 6, 5	3.61 ± 2.10	2.8 ± 1.99
	R2	4, 4, 4, 6, 4, 4	3.52 ± 2.00	1.9 ± 1.51
	R3	4, 4, 4, 9, 4, 4	3.45 ± 2.45	7.9 ± 9.05
	L1	4, 6, 4, 4, 4, 4	4.57 ± 2.30	2.4 ± 1.69
	L2	4, 4, 6, 4, 4, 4	4.88 ± 1.98	2.6 ± 2.11
	L3	4, 6, 4, 4, 4, 4	4.47 ± 1.66	4.6 ± 4.72
	R2, R3	4, 4, 4, 5, 8, 9	4.24 ± 2.69	8.0 ± 5.31
	R1, R3	4, 4, 4, 6, 4, 9	2.64 ± 2.05	6.1 ± 3.94
	R1, R2	4, 4, 4, 8, 9, 6	4.35 ± 1.59	7.6 ± 2.20
	R1, L3	4, 5, 4, 4, 5, 4	3.29 ± 2.16	10.9 ± 5.84
	R2, L3	8, 4, 4, 4, 8, 4	4.17 ± 2.29	10.7 ± 6.77
	R1, L2	4, 4, 4, 4, 4, 4	0.83 ± 0.04	0.0 ± 0.0
	R1,R3,L1	4, 4, 4, 4, 5, 5	4.39 ± 2.09	7.7 ± 6.51
	R1,R3,L3	4, 4, 4, 4, 9, 4	5.08 ± 1.29	3.6 ± 3.83
	R1,R3,L2	4, 4, 4, 4, 4, 9	2.01 ± 1.61	5.7 ± 5.92
	R1,R2,L1	4, 4, 5, 4, 8, 8	4.67 ± 2.50	4.9 ± 3.08
	R1,R2,L3	4, 4, 4, 5, 9, 4	3.81 ± 2.31	5.8 ± 4.04
	R1,R2,L2	4, 4, 4, 4, 4, 9	2.98 ± 2.21	3.5 ± 2.46
	R2,R3,L1	4, 4, 4, 4, 8, 5	2.49 ± 2.07	3.8 ± 2.52
	R2,R3,L3	4, 4, 4, 6, 5, 4	4.94 ± 2.37	3.7 ± 2.10
	R2,R3,LM	4, 4, 4, 4, 4, 8	5.03 ± 1.93	3.3 ± 1.55

Leg periods after R1 compensation



(a) Leg R1: disabled (b) Leg R2: p5 (c) Leg R3: p4




(a)
fail
1
2
3
4
R3
R2
R1
L3
L2
L1
(b)
fail
1
2
3
4
R3
R2
R1
L3
L2
L1
(c)
fail
1
2
3
4
R3
R2
R1
L3
L2
L1

Real performance video

Comparison to other approaches

	Ren & Chen	Cully & Clune
Model	Chaotic CPG	Gaussian Process
Additional Prior	None	Search space premodelling & prior performance values
Optimization	Simulated Annealing (GD)	Intelligent Trial & Error (Bayesian Optimization)
Trials	Few (1-10)	Few (1-10)

How can the performances be so similar?

Keypoints

- A chaotic system can be controlled into showing periodic dynamics, so as to be implemented as a CPG to accomplish the locomotion control of a bio-inspired hexapod robot
- Multiple coupled chaotic CPGs with learning can be used for legged locomotion and malfunction compensation
- A simulated annealing optimization is sufficient if the basic locomotion mechanism aids the optimization
- Chaotic ground state can improve the overall performance of a robot and make it easier to search for matching periodicities

Flaws

- The system does not use any additional sensors to detect malfunction.
- The PSN/VRN networks and the delay lines make it a beautiful walking machine, but unfortunately make it unable to do anything else.

Discussion!

- Any questions?

References

- G. Ren, W. Chen. et al., "Multiple chaotic central pattern generators with learning for legged locomotion and malfunction compensation", Information Sciences 294 (2015)
- S.Steingrube, M. Timme et al., "Self-organized adaptation of a simple neural circuit enables complex robot behaviour", Nature Physics 6 (2010)
- A. Cully, J. Clune et al. "Robots that can adapt like animals", Nature 521 (2015)