Motion development as direct CPG adjustment by touching

Fabio Dalla Libera*, Takashi Minato**, Hiroshi Ishiguro** ***, Emanuele Menegatti*

*Department of Information Engineering (DEI), University of Padua, Italy
**ERATO, Japan Science and Technology Agency, Osaka University, Japan
***Department of Adaptive Machine Systems, Graduate School of Engineering, Osaka University, Japan

Central Pattern Generators (CPGs) are a powerful bio-inspired method to generate motions. Usually the numerous parameters of the CPGs are set by learning algorithms like genetic algorithms or policy gradient. This gives the user little control over the resultant motions, which might look awkward. We propose a methodology to set the CPG parameters by user interaction, in detail by touch interaction. We describe the elements of the developed system and verify its feasibility by the realization from scratch of a crawling motion.

Key Words Touch interaction, motion development, CPG

1. Introduction

Many animal and human rhythmic movements, like walking or chewing, are controlled by Central Pattern Generators that make the system automatically entrain to the environment and therefore adapt and recover from disturbances [1, 2].

In order to have a similar robustness on robotic systems, this structure is mimicked and several CPG models are applied to control different kind of robots [3, 4, 5, 6, 7, 1]. In particular, in many works each actuator is controlled (with position or torque control) by an oscillator composed of a couple of neurons, representing the extensor and flexor neurons. In order to allow synchronization between the joints the oscillators are interconnected, usually by weak coupling. The sensory feedback is then opportunely introduced to have entrainment between the environment and the robot.

The movement generated by the CPG depends on a huge number of parameters, consisting of the neuron model parameters and the weight interconnections. These are usually set by automatic search algorithms such as genetic algorithms [8], policy gradient [9] or reinforcement learning [10].

One of the drawbacks in using these techniques is that the user has little control over the resulting motions. In fact the programmer can express the quality of the motion only in terms of a fitness/score/evaluation function. Imagine that the user wants to develop a crawling motion for a humanoid robot and that the optimization criterion used by a genetic algorithm to evaluate the motion quality is the velocity. The derived motion would probably contain movements that would look awkward and the overall movement would be similar to canter, while often the purpose is to get a baby-like movement that makes the robot proceed at a reasonable speed. In other words, in many cases we cannot easily replace human evaluation with a mathematical formulation.

Manually setting each CPG parameter would anyway be both time consuming and not intuitive for unexperienced users that cannot predict how changing a parameter value will modify the motion.

In this paper we propose to have the user create the motion by setting the CPG parameters through interaction with the robot. Among the possible communication means (verbal communication, learning by watching [11], etc.) we decided to utilize tactile interaction, a very powerful and natural way of communication in human-human and human-robot interaction [12]. In particular, with our approach the user can touch the robot while this is performing the motion to tune the CPG parameters and therefore change the resultant motion.

This implies, of course, that the relationships between parameter changes and movement changes should be predictable. Section 2 reports the details on the easily predictable oscillator network we employed. Mapping touch patterns to motion modifications and expressly to CPG parameter changes is not straightforward, since several features like the amplitude of the movement or the velocity of each joint motion should be changed by touch. Therefore, in section 3 we illustrate the touch protocol defined for our experiment. We will report results of an experiment conducted with a simulated humanoid robot in section 4 and conclude the paper in section 5.

2. CPG Network

While often the purpose of a CPG network is predefined (for example, make the robot walk) we would like to derive a quite general structure for the network. More precisely in the network structure design we must focus in having a network suited for teaching by touching, instead of aiming at a certain task and optimize the connections consequently. Users interacting with the robot must be able to predict the consequences of their commands. In other terms, when a touch pattern is perceived and the intended motion modification is identified, it must be clear how to change the parameters to obtain the desired movement alteration. This implies that one of the main requirements in the choice of the oscillator model is to have a predictable system.

In existing research numerous types of oscillators were presented (Rayleigh, Van Der Pol [5], Matsuoka's [13], etc.). Hopf oscillator was chosen for its simplicity and predictability. The predictability of the system is also strongly influenced by the interconnections between the oscillators. Oscillator networks present in literature present essentially five structures:

1. chain [2], used mainly for snake robots

2. star [14], that is a "peacemaker"/ "clock" oscillator provides a synchronizing signal to all the others
3. tree [15], where essentially the oscillators are connected as a tree, from the proximal to the distal joints
4. connection between homologous joints [5], i.e. joints with a similar function
5. full connection between the oscillators [16]

We opted for a star structure, which allows all the oscillators to be synchronized without unpredictable interactions between subgroups of oscillators. More precisely each of the n degrees of freedom of the robot is controlled by one oscillator, and a further "clock" oscillator provides a reference signals for all these oscillators. Let us identify by C_0 the reference oscillator and by C_j, $1 \leq j \leq n$ the oscillators controlling the robot joints,

Using the complex number representation for the Hopf oscillator [17] we have for the j-th oscillator, $0 \leq j \leq n$

$$\begin{gathered} \dot{z_j} = \gamma \left(\mu_j - |z_j|^2\right) z_j + i\omega_j z_j + F_j(t) \\ m_j = \Re\{z_j\} + o_j \end{gathered} \tag{1}$$

Where:

- $z_j \in \mathbb{C}$ is the state of the oscillator
- $m_j \in \mathbb{R}$ is the control signal for the actuator
- γ is a coefficient for the speed of recovery after perturbation [18]
- $\mu_j \in \mathbb{R}, \mu_j \geq 0$ controls the amplitude of the oscillation
- $\omega_j \in \mathbb{R}, \omega_j \geq 0$ controls the oscillation frequency
- $F_j(t)$ is an external perturbation signal
- o_j is an offset value used to set the position around which the joint oscillates

We decided to restrict to periodic motions, and to ensure rational ratios between the frequencies of oscillation of each pair of joints we set

$$\omega_j = p_j \omega_0 \tag{2}$$

$1 \leq j \leq n$, $p_j \in \mathbb{N}$. In the current implementation we do not use any feedback signal, so $F_0(t)$ is zero (the main clock is not influenced by anything, but in further implementation we plan to introduce a signal from the gyroscope) while for $1 \leq j \leq n$

$$F_j(t) = we^{i\phi_j} z_0^{p_j} \tag{3}$$

that is $F_j(t)$ consists essentially in the perturbation from the clock oscillator that permits synchronization of the system. The reference signal z_0 is elevated to the power p_j so the frequencies of the oscillator and of the perturbation are close. The similarity of the frequencies leads to an easier synchronization and a predictable phase between the j-th oscillator and the reference one [19]. The coefficient w determines the coupling strength between C_0 and the other oscillators while the term $e^{i\phi_j}$ allows to change the phase difference between the clock oscillators and the other ones. In the current implementation $\mu_0 = 1$,$w = 0.1$ and $\gamma = 1000$.

3. Touch protocol

As previously stated a protocol that maps touches on the robot sensors into parameter changes must be defined. In detail we can imagine for simplicity the situation where each robot's link is a parallelepiped and that each face constitutes a touch sensor. We need to convert a series of touches on the sensors to CPG parameter changes. Since this work aims exclusively at verifying the feasibility of the approach, we decided to simplify the system as much as possible and employed a reasonable static mapping between the user actions and the parameter changes. We also assume the touch sensors to be binary and measure just their pressure time. Research on the identification of the features of intuitive touch protocols is of course of fundamental importance, and preliminary works in this direction had been presented in [20]. From Equations 1 and 2 it is possible to see that for each joint we can control

- the amplitude of the oscillation of each joint, by μ_j
- the frequency of the movement of each joint, by p_j
- the phase of the movement, with respect to the main oscillator, through the parameter ϕ_j
- the zero position (offset) around which the joint moves, by o_j

and changing ω_0 we can change the global speed of the movement. We adopted, mostly arbitrarily, the following protocol. When a sensor s is pushed we determine the most distal joint, along the kinematic chain, that causes a movement of the sensor center in the direction perpendicular to the sensor surface. More formally we assume the robot's main body (in our case the torso of a humanoid robot) is fixed in the space. We denote by n_s the vector perpendicular to the pushed sensor surface and by d_j the vector representing the derivative of the position of the center of the pushed sensor when the $j-th$ joint is rotated. Let $j_1, j_2, ...j_d$ be the indices of the joints placed between the robot's main body and the link where the sensor is located, in order from the most proximal to the most distal; we identify the joint j_s such that $\rho_s =< n_s, d_{j_s} > \neq 0$ and $< n_s, d_{j_k} >= 0$ for $s < k \leq d$. If this joint doesn't exist we simply ignore the sensor pressure (unless it is on the main body, as will be specified later). Once j_s is determined the phase of joint j_s ($\angle z_{j_s}$) is used as a time reference. Expressly the pushing time τ_{j_s} is measured in terms of phase difference between the release time and the pushing time, counting for the phase resettings (i.e. the difference is positive, monotonically increasing and can be larger than 2π). We distinguish the following cases:

- If $\tau_{j_s} > \Theta_O$ (the user pushes for a very long time) the offset parameter is changed according to the applied force, i.e. $o_{j_s,new} = o_{j_s,old} + sgn(\rho_{j_s})\Delta_O$, where sgn is the sign function.
- If $\Theta_A < \tau_{j_s} \leq \Theta_O$ the amplitude parameter is updated by the value $sgn((\rho_{j_s} * m_{j_s})\Delta_A$ where m_{j_s} is the value of the output at the pushing time.
- If the user pushes for a time $\tau_{j_s} \leq \Theta_A$, releases the sensor and doesn't push it for a time Θ_P then the phase parameter ϕ_{j_s} is updated such that in the following cycles the closest maximum of oscillation occurs at the pushing time, i.e. the quantity $-\angle(m_{j_s} * z_{j_s})$ is added to ϕ_{j_s}, where z_{j_s} and m_{j_s} are considered at the pushing time.

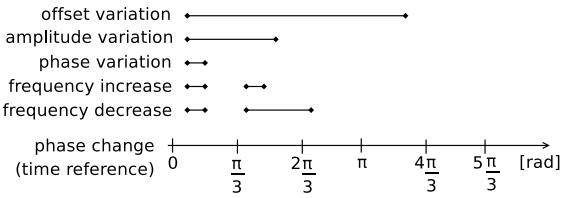

Figure 1: Schematization of the touch patterns recognized by the system. The presence of a line indicates the pressure of a sensor. The phase change axis(time reference) is also reported.

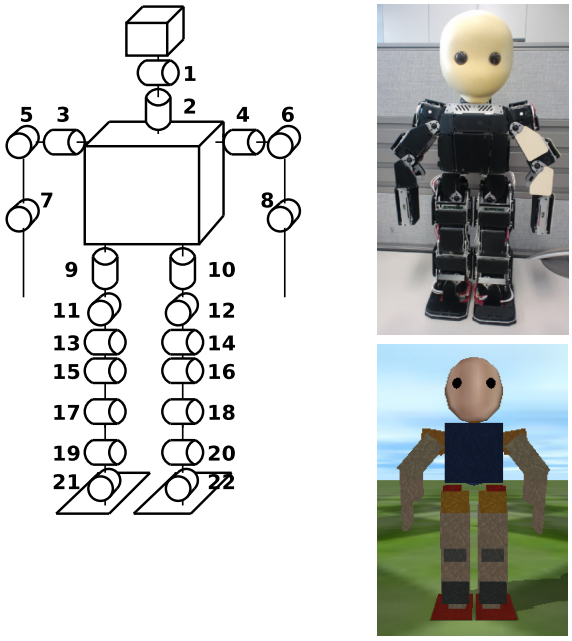

Figure 2: Structure of the robot, picture of the real robot and its simulated counterpart.

- If the user pushes for a time $\tau_{j_s} \leq \Theta_A$, releases the sensor and before a phase change of Θ_P pushes the sensor again then p_{j_s} is incremented or decremented respectively if this second pushing time $\tau_{j_s,2}$ is greater or lower than Θ_A.
- Similarly, if the user pushes the robot main body for a time $\Delta\phi_0 \leq \Theta_A$ releases the sensor and before a phase change of Θ_P pushes the sensor again then ω_0 is increased or decreased by the quantity Δ_{ω_0} respectively if the second pushing time is greater or lower than Θ_A.

In our implementation all the Δ and Θ values are constants, expressly $\Theta_O = \pi$, $\Theta_A = \frac{\pi}{6}$, $\Theta_P = \frac{2\pi}{3}$, $\Delta_O = \Delta_A = \frac{\pi}{12}$, $\Delta_{\omega_0} = 1$.

4. Experiments

In order to test the feasibility of the approach we developed, from scratch, a crawling motion for a humanoid robot. Specifically we used a simulator based on ODE that models VStone's Vision4G[1]. The touch is simulated by mouse clicks, whose time

[1] See `http://www.vstone.co.jp` and `http://www.ode.org` for Vision4G and ODE, respectively.

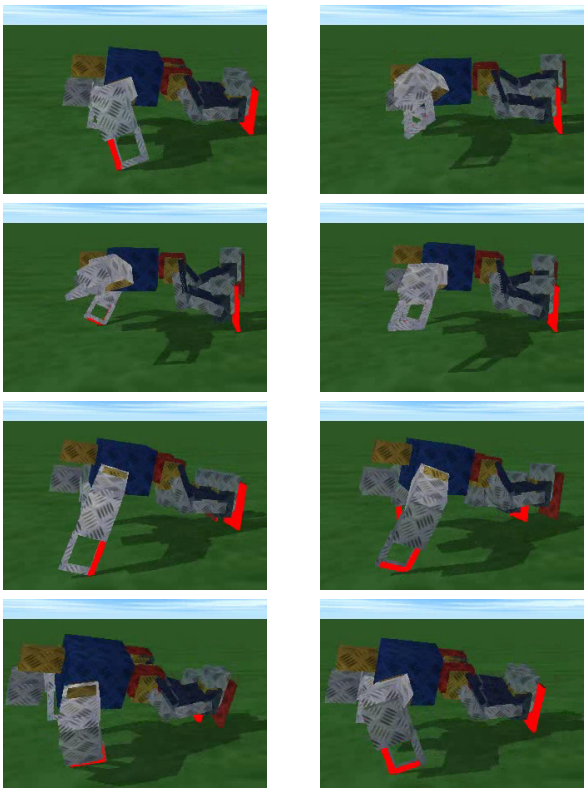

Figure 3: Screenshots of the developed crawling motion.

length is measured and used to modify the CPG parameters.

Employing a simulator is a strong simplification because we can easily discriminate the user touch from self touches and contact with the ground. Apart this aspect, however, our approach is applicable to real robots, and in fact future experiments will be conducted using Vision4G. One detail to be considered when employing the real hardware is that self collisions, generated by some CPG parameter settings, could cause servomotor breakdown. The self collisions must be predicted online therefore the computation should be very fast. The collision detection can in any case be inaccurate. More precisely, while we should assure that collision positions are discarded, we are not required to recognize each collision free posture as such, i.e., we can reduce the set of allowed postures to increase the computation speed. We thus chose to model each link by a (bigger) parallelepiped, and limit the computation to the collision of those parallelepipeds. Note that increasing the size of the parallelepiped we can simply obtain a "safety margin". The realized collision checker, employed even in this preliminary experiment conducted on a simulator, is able to perform around 17000 collisions per seconds on the robot's CPU.

Starting with $\mu_j = o_j = \phi_j = 0$ and $p_j = 1$, $1 \leq j \leq n$ the crawling motion was developed by a a single user in 56 minutes. The user employed 57 amplitude changes, 39 phase changes, 22 offset changes and 2 frequency changes to obtain a satisfactory movement according to his subjective evaluation (screenshots are presented in Fig. [**?**]). Table 1 reports the final values obtained for the CPG parameters that were altered from their initial conditions. The final value for ω_0 is 3.256.

Table 1: Final values obtained by user interaction.

Joint \ Parameter	μ_j	p_j	ϕ_j	o_j
3	0.5236	1	0.6599	0.7854
4	0.2618	1	1.6120	-0.5236
5	0	1	0.7172	0.5236
6	0.5236	1	2.3118	0
7	0.2618	1	2.2020	0.7854
8	0.7853	1	5.6918	-0.5236
11	0	1	5.6814	0
15	0.2618	1	1.2783	-1.0472
16	0.2618	1	0	1.0472
17	0.2618	1	0	0
18	0.2618	1	0	0

5. Conclusions

This paper proposes to employ human-robot interaction as a way to set the parameters of a CPG. This allows controlling the resultant motion (and therefore generate the expected movement, in terms, for instance, of similarity to human motion) by tuning all the parameters in a natural way. We described a Hopf oscillator network that ensures predictability, a basic requirement if we want the user to be able to intuitively change the parameters. To maximize the usability of the system we employed tactile interaction, a very direct way of communicating the desired motion modification. We defined a touch protocol that allows to change amplitude, frequency, phase and offset of the movement of the robot's joints and present how to change the CPG parameters in response to user touches.

As a validation we developed a crawling motion using exclusively the proposed technique. We stress that purpose of the experiment is not to achieve the fastest crawling speed which is possible but to achieve a motion satisfactory for the user (in terms, for instance, of similarity to human movements). Obviously the motions obtained with this approach could be used as initial solutions to be optimized with classical automatic methods. Further tuning, for instance to assure perfect movement symmetry or to maximize the crawling speed could be also performed but this is out to the scope of this paper.

Future works will need to measure quantitatively the advantages of the presented approach. More precisely we should evaluate how often automatically optimized motions appear satisfactory to the user (i.e. compare our approach to motions derived from genetic algorithms or policy gradient) and study the advantages over direct input of the parameters values w.r.t. simplicity of motion development, for instance comparing the motion development time. As previously stated we will also need to study the features of a good touch protocol, which was fixed nearly arbitrarily in this preliminary work. Finally future research will need to consider techniques to discriminate between user touches, self touches and contact with the floor, a problem here avoided employing a simulator but that must be tackled when working with a real robot.

References

[1] Taga, G. and Yamaguchi, Y. Shimizu, H. "Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment". *Biological Cybernetics*, vol. 65, 147–159, 1991.

[2] Ijspeert, A.J. and Crespi, A. "Online trajectory generation in an amphibious snake robot using a lamprey-like central pattern generator model". *2007 IEEE International Conference on Robotics and Automation (ICRA 2007)*, pp. 262–268. Roma, Italy, 2007.

[3] Morimoto, J., Endo, G., Nakanishi, J., Hyon, S.H., Cheng, G., Bentivegna, D.C. and Atkeson, C.G. "Modulation of simple sinusoidal patterns by a coupled oscillator model for biped walking". *2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*, pp. 1579–1584. Orlando, USA, 2006.

[4] de Pina Filho, A.C., Dutra, M.S. and Raptopoulos, L.S.C. "Modeling of a bipedal robot using mutually coupled rayleigh oscillators". *Biological Cybernetics*, vol. 92(1), 1–7, 2005.

[5] Roy, P. and Demiris, Y. "Analysis of biped gait patterns generated by van der pol and rayleigh oscillators under feedback". *Proc. 3rd International Symposium on Adaptive Motion in Animals and Machines (AMAM2005)*. Ilmenau, Germany, 2005.

[6] Collins, J.J. and Richmon, S.A. "Hard-wired central pattern generators for quadrupedal locomotion". *Biological Cybernetics*, vol. 71(5), 1994.

[7] Mathayomchan, B. and Beer, R.D. "Center-crossing recurrent neural networks for the evolution of rhythmic behavior". *Neural Computation*, vol. 14, 2043–2051, 2002.

[8] Inada, H. and Ishii, K. "Behavior generation of bipedal robot using central pattern generator(cpg) (1st report: Cpg parameters searching method by genetic algorithm)". *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 2003)*, vol. 3, pp. 2179–2184. Las Vegas, USA, 2003.

[9] Endo, G., Morimoto, J., Matsubara, T., Nakanishi, J. and Cheng, G. "Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot". *International Journal of Robotics Research*, vol. 27(2), 213–228, 2008.

[10] Nakamura, Y., Mori, T., aki Sato, M. and Ishii, S. "Reinforcement learning for a biped robot based on a cpg-actor-critic method". *Neural Networks*, vol. 20(6), 2007.

[11] Kuniyoshi, Y., Inaba, M. and Inoue, H. "Learning by watching: extracting reusable task knowledge from visual observation of human performance". *IEEE Transactions on Robotics and Automation*, vol. 10(6), 799–822, 1994.

[12] Voyles, R. and Khosla, P. "Tactile gestures for human/robot interaction". *1995 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS 1995)*, pp. 7–13. Pittsburg, USA, 1995.

[13] Matsuoka, K. "Sustained oscillations generated by mutually inhibiting neurons with adaptation". *Biological Cybernetics*, vol. 52, 367âĂŞ376, 1985.

[14] Schaal, S., Kotosaka, S. and Sternad, D. "Nonlinear dynamical systems as movement primitives". *IEEE-RAS 1st International Conference on Humanoid Robots (Humanoids 2000)*, pp. 117–124. Cambridge, USA, 2000.

[15] Endo, K., Yamasaki, F., Maeno, T. and Kitano, H. "A method for co-evolving morphology and walking pattern of biped humanoid robot". *2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, pp. 2775–2780. Washington, USA, 2002.

[16] K.Tsujita, K.Tsuchiya and A.Onat. "Decentralized autonomous control of a quadruped locomotion robot". *Artificial life and robotics*, vol. 5, 152–158, 2003.

[17] Kern, A. and Stoop, R. "Nonlinear dynamics of cochlear information processing". *Proc Int Workshop of Nonlinear Dynamics of Electronic Systems (NDES-04), Evora, Portugal*, pp. 190–193, 2004.

[18] Righetti, L. and Ijspeert, A. "Programmable central pattern generators: an application to biped locomotion control". *2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*, pp. 1585–1590. Orlando, USA, 2006.

[19] Pikovsky A., Rosenblum M., K.J. *Synchronization: A Universal Concept in Nonlinear Science*. Cambridge University Press, Cambridge, 2001.

[20] Dalla Libera, F., Minato, T., Fasel, I., Ishiguro, H., Menegatti, E. and Pagello, E. "Teaching by touching: an intuitive method for development of humanoid robot motions". *IEEE-RAS 7th International Conference on Humanoid Robots (Humanoids 2007)*. Pittsburg, USA, 2007.