Proceedings of the Post-Graduate Conference on Robotics and Development of Cognition
10-12 September 2012, Lausanne, Switzerland

Morphology Dependent Distributed Controller for Locomotion in Modular Robots

Avinash Ranganath [1], Juan Gonzalez-Gomez [2], Luis Moreno Lorente [1]

[1] Robotics Lab, Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Madrid, Spain
[2] Robotics and Cybernetics Group, Departamento de Automática, Ingeniería Electrónica e Informática Industrial, Universidad Politécnica de Madrid, Madrid, Spain
arangana@uc3m.es, juan@iearobotics.com, moreno@ing.uc3m.es

Abstract

Stigmergy is defined as a mechanism of coordination through indirect communication among agents, which can be commonly observed in social insects such as ants. In this work we investigate the emergence of coordination for locomotion in modular robots through indirect communication among modules. We demonstrate how intra-configuration forces that exist between physically connected modules can be used for self-organization in modular robots, and how the emerging global behavior is a result of the morphology of the robotic configuration.
Index Terms: modular robot, locomotion, distributed controller, self-organization, embodiment

1. Introduction

Modular robots are systems composed of several individual unit modules, which with self-reconfigurable capability can autonomously change their morphology. Modular robots can be broadly classified into lattice-type and chain-type systems. Lattice-type systems achieve locomotion through continuous self-reconfiguration, where each module has the ability to move independently in the configuration, giving the notion of modules flowing on the ground and around obstacles. Locomotion in a chain-type system is achieved by controlling the actuator of individual modules in a fixed configuration.

One of the earliest demonstrations of locomotion in chain-type reconfigurable modular robots was provided by Mark Yim in [1], which included several locomotion modes such as walking, crawling, rolling, climbing etc. Distributed controllers for locomotion in chain-type modular robots have been researched in [2], [3], [4], [5] and [6]. Shen et al. have used a biologically inspired method called Digital Hormone Method (DHM) [2], [3], [4] for adaptive communication of state information between modules, based on which a module can decide an action from the gait table, resulting in the emergence of locomotion. Gonzalez-Gomez et al. have demonstrated in [5] how simple sinusoidal oscillators can be used on minimal configuration modular robots with two and three modules to generate locomotion in one and two dimensions respectively. In [6] Ijspreet et al. at the Biorobotics Laboratory, EPFL, have used Central Pattern Generators (CPG) [7] for producing locomotion oscillations on their modular robotic platform called YaMoR. CPGs are specialized neurons found in the spinal cord of vertebrate animals, which have the capability of producing rhythmic output without rhythmic sensory or central input. The mathematical model of CPGs used for controlling locomotion in modular robots are usually one or two CPG neurons per module, which are coupled in different ways with CPGs of other modules based on the configuration.

Though DHM and CPGs are distributed control methodologies, they rely on explicit inter-modular communication. The simple oscillators for locomotion in modular robots demonstrated by Gonzalez-Gomez et al. in [5] is a distributed controller as well, but the phase relation between modules are predetermined, making the controller heterogeneous. We have, in this work, attempted to develop a locomotion controller for chain-type modular robots that is distributed, homogeneous and which does not rely on explicit communication between modules.

2. Simulation and robotic platform

In this work, we test our locomotion controller on modular robotic configurations built using the simulated model of the *Y1* modular robot modules, developed by Juan Gonzalez-Gomez. OpenRAVE is the simulation environment used for experiments in this work. OpenRAVE is physics based, open-source, robotics simulator that has Open Dynamic Engine as its core. The *Y1* modules are an open source, low cost, flexible, and easy to build modular robotic platform, which have been used as a research platform in several research projects. The*Y1*s, as could be seen in Fig.1, are open-ended cube shaped modules, which have a single degree of freedom, with a rotation range of 180º. The dimensions of these modules are 72x52x52 mm. The simulated modules are kept consistent with the real modules, both structure wise, and with respect to actuator features.

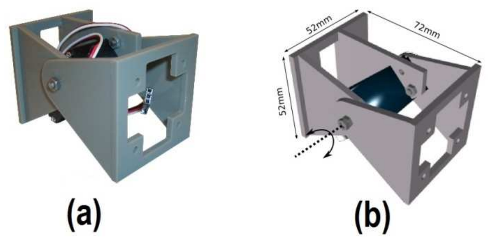

Figure 1: Y1 module (a) Real and (b) Simulated versions.

2.1. Modular robot configurations

We have tested our locomotion controller on three different modular robotic configurations, as could be seen in Fig.2. Each configuration is explained in the following subsections.

2.1.1. Minimal configuration

The Minimal configuration is a two module, one-dimensional configuration, and according to [5], this is the smallest possible configuration for producing locomotion in one-dimension. When both the modules are actuated with simple sinusoidal oscillators with predefined phase difference, they produce a caterpillar gait, which resembles a travelling sine wave, with the phase value determining the direction of locomotion.

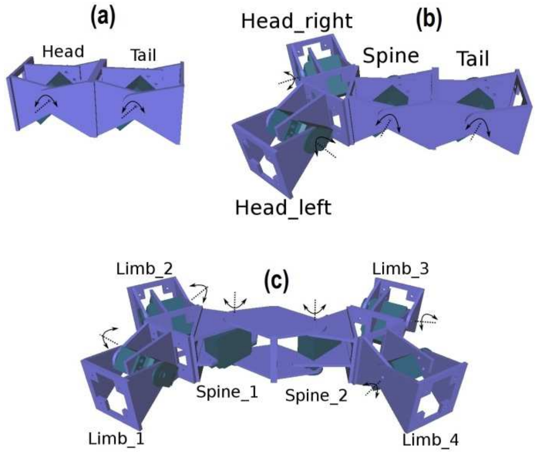

Figure 2: (a) Minimal configuration (b) Y-bot and (c) Lizard.

2.1.2. *Y-bot*

Y-bot is a four module configuration that can be seen as an extension of the Minimal configuration when two more modules (*Head_left* and *Head_right*) are connected to the *Spine* module at an angle of +/-60º. Locomotion in two-dimension is possible with this configuration, although we focus only on one-dimensional locomotion gait in this work. Again, with simple phase-differed sinusoidal oscillators, this configuration produces a caterpillar like gait, when modules *Head_left* and *Head_right* remain in phase.

2.1.3. *Lizard*

Lizard is a six module configuration that has four *Limb* modules, and two *Spine* modules. The *Spine* modules are rotated by +/-90º along the pitch axis, in relation with the rest of the configuration. When modules in this configuration are actuated with phase-controlled sinusoidal oscillators, as shown in Table 1 (derived empirically), the result is a quadruped walking gait, resembling that of a reptile.

Table 1. Phase relation between modules in a Lizard configuration with respect to the module 'Limb_1'.

Module	Phase Angle
Limb_1	0º
Limb_2	160º
Spine_1	80º
Spine_2	-80º
Limb_3	160º
Limb_4	0º

3. Controller

Locomotion in general, whether a gallop of a horse, flapping of a bird, or walking of a human, can be seen as repetitive and coordinated movement of limbs, through which the locomotion gait emerges. Looking at locomotion as a collection of oscillators, the phase relation between these oscillators determines the generated gait. This phase relation can be brought about by sharing actuation information among modules through explicit inter-module communication in a modular robotic system. But since a modular robot is an embodied system comprising of physically connected robot modules, our controller relies on the intra-configuration forces that exist among modules for coordination.

3.1. Intra-configuration forces

In a simulated Minimal configuration, when one module is actuated with a sinusoidal oscillator, with amplitude of 60º, and the other module is made to remain at a constant 0º, the oscillating module is seen to affect the other module. As could be seen in Fig.3, the unactuated module oscillates as well with low amplitude and an offset, due to the force exerted on it by the oscillating module. This is because a robot is an embodied system, where physically connected modules exert force on each other when actuated, which can be seen as an implicit communication among modules. Since the simulation tool used here is based on physics, similar (if not exactly the same) results can be expected in the real system.

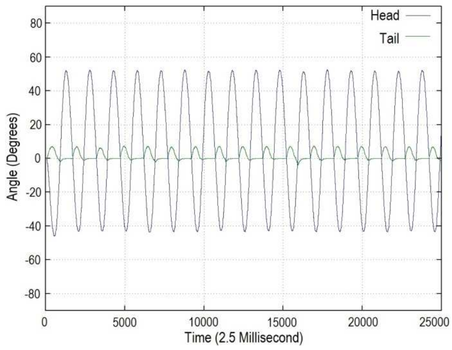

Figure 3: Plot of actuator values in a Minimal configuration, demonstrating the effects of the oscillating module over an unactuated module.

3.2. Simple controller

Since oscillation is fundamental to all locomotion gaits, we made the modules oscillate independently with fixed amplitude and an offset as defined in (1). Conditions (2) and (3) are used to determine if the module's actuator has reached the desired oscillation angle, and if either of the two conditions satisfies, then the direction of rotation of the module's actuator is switched by obtaining the next oscillatory angle from (1). Fig.4(a) depicts the control strategy. Condition (2) checks if the actuator is within a range of $+\alpha$ and $-\alpha$ of the desired position determined by (1). Condition (3) checks if the rate of actuation is above a certain limit specified by β. The value of the parameters A, o, α and β are determined empirically.

$$y_i := (-1)^i A + o, \forall i \in \mathrm{N} \quad (1)$$

$$|y - \theta_t| \leq \alpha \quad (2)$$

$$\Delta\theta_t \leq \beta \quad (3)$$

Where y_i is the i^{th} input to the module's actuator, A is the amplitude, o is the offset, θ_t is the positional feedback from the module's actuator at time instance t. Parameters α and β are constants.

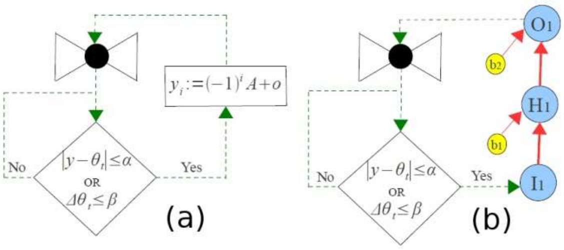

Figure 4: Control flow of (a) Simple controller (b) Neural controller.

3.3. Neural controller

Extending the previous model to include adaptive oscillation rather than a fixed-amplitude-offset oscillator, (1) is replaced with a fully connected feed-forward multilayer perceptron Artificial Neural Network [ANN], as shown in Fig.4(b). The ANN has one input neuron, one hidden layer with a single hidden neuron, and one output neuron. The input to the neural network is the positional feedback from the module's actuator, and the output is the control signal for the same. The lone hidden neuron and the output neuron have one bias node each. *Flood*, an open source ANN library, is used for implementing the ANN. The parameters of this controller are optimized using Genetic Algorithm [GA].

4. Experiment and results

4.1. Evolution

The parameter β and the synaptic weights of the ANN in the neural controller are optimized using GA, individually for each of the three configurations. A robotic configuration is set up in the simulation environment, with each module controlled independently with the neural controller, starting with random initial parameters. The evaluation criteria for evolving optimal parameters, is the distance travelled at the end of the simulation cycle. Each individual in the population is evaluated for 50 seconds in the simulation environment. A fairly standard GA approach is followed, with Roulette Wheel selection method and Intermediate Recombination method for reproducing new offspring. Table 2 contains the GA parameters employed.

Table 2. GA Parameter values used for evolution.

Parameters	Value
Population Size	50
Evolution length	50 generations
Crossover percentage	50.0%
Elite population percentage	12.5%
Mutation rate	1/Size of genome

4.2. Evaluation

The resulting neural controller was evaluated by controlling modules in a given configuration with the most optimal control parameters evolved for that configuration. When actuated, the modules in the Minimal configuration started oscillating in phase, but quickly develop and maintain a steady phase difference, and resulted in a caterpillar locomotion gait. The frequency of oscillation is not predefined in the controller, but intrinsic to the system, and it is inversely-proportional to the amplitude. The amount and stability of phase relation between modules is a result of the morphology. A plot of the oscillation, frequency and phase values of the emerged locomotion gait in this configuration is as shown in Fig.5, Fig.6 and Fig.7 respectively.

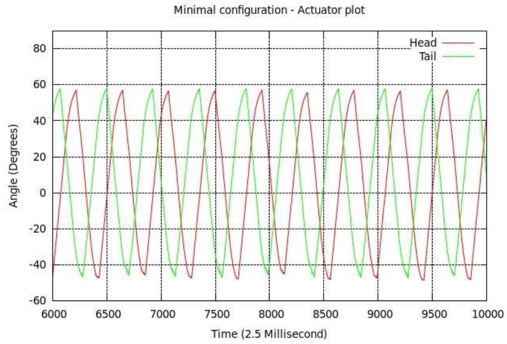

Figure 5: Plot of actuator values in the Minimal configuration actuated with the neural controller.

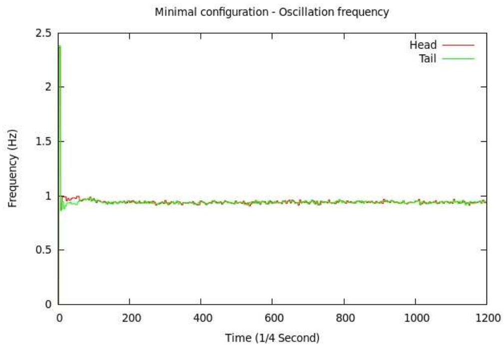

Figure 6: Oscillation frequency graph of modules in the Minimal configuration when actuated with the neural controller.

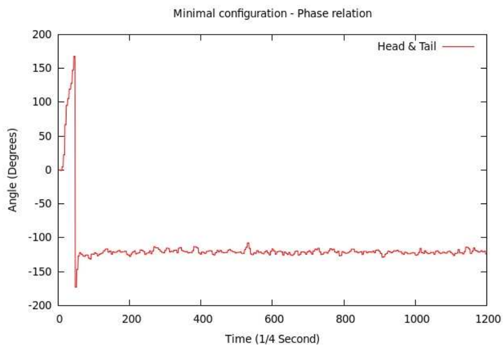

Figure 7: Graph containing phase relation between modules in the Minimal configuration when actuated with the neural controller.

When modules in the Y-bot configuration were actuated with the best evolved neural controller, a similar caterpillar gait emerged and the phase relation graph is as shown in Fig.8. In the Lizard configuration, the neural controller produced a quadruped walking gait, similar to that of a reptile. Each configuration with its respective neural controller was evaluated for a period of 300 seconds. Table 3 contains the speed of locomotion, averaged over 10 evaluations. Fig.9 and Fig.10 contains the phase relation graph of the emerged locomotion gait in the Lizard configuration. The graphs in Fig.9 and Fig.10 are from a single evaluation, but presented separately as two different conventions are used with respect to the Y-axis range for better visualization.

Table 3. Speed of locomotion averaged over 10 evaluations.

Configuration	Speed (Cms/Sec)
Minimal configuration	3.35
Y-bot	4.18
Lizard	2.09

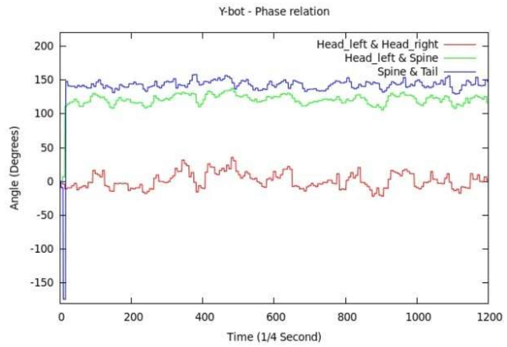

Figure 8: Graph containing phase relation between modules in the Y-bot configuration when actuated with the neural controller.

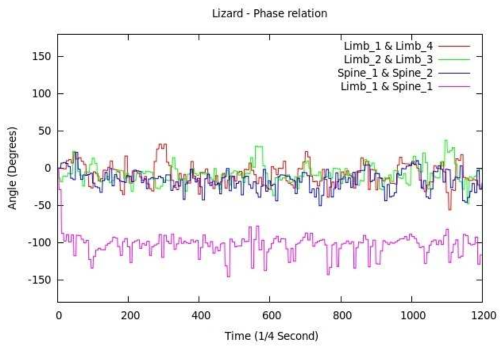

Figure 9: Graph containing phase relation between some pairs of modules in the Lizard configuration when actuated with the neural controller. The phase angle is represented as a value between -180º and +180º for better visualization.

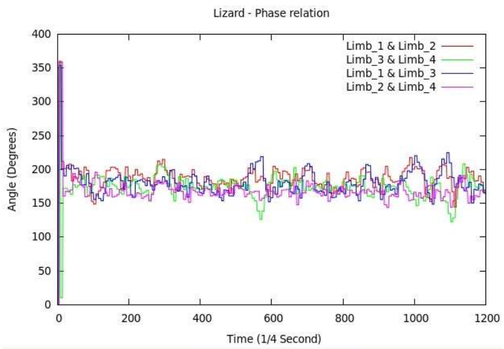

Figure 10: Graph containing phase relation between a few other pairs of modules in the Lizard configuration when actuated with the neural controller. The phase angle is represented as a value between 0º and 359º for better visualization.

4.3. Cross-evaluation

Considering both, the difference in morphology and the dynamics of the emerged locomotion gait in the Y-bot and the Lizard configurations, the required coordination among modules of both the configurations must be very different. To test how a controller evolved for a particular configuration would fair when applied on a different configuration, we cross-evaluated the neural controller evolved for the Y-bot configuration on the Lizard configuration, and vice versa. The emerged locomotion gait when cross-evaluated was virtually similar to the configuration's original locomotion gait in both the cases, implying that the controller is able to adapt its behavior based on the change in morphology.

5. Conclusions

In a multi-robot system like modular robots, coordination among modules is required to produce a stable locomotion gait, and with our controller we have been able to demonstrate how such coordination among modules can emerge based only on indirect local interaction among connected modules, without the need for any direct communication between them. Furthermore, by cross-evaluating the controller, we have been able to demonstrate the dependency of the emerged gait on the morphology of the robot, supporting the notion of embodiment in a robot.

Moving forward, we would like to first evaluate the proposed controller on configurations with real *Y1* modules. In the current model, although the parameter β which determines the actuation rate threshold is optimized using GA, it is a constant during the control phase. We would like to extend our model in such a way that the activation rate threshold value is adaptive during the control phase.

6. References

[1] Yim, M. 1994. Locomotion with a unit-modular reconfigurable robot. Ph.D dissertation, in Department of Mechanical Engineering, Stanford University.
[2] Shen, W.-M., Salemi, B., Will, P. 2000. Hormones for Self-Reconfigurable Robots. in Intelligent Autonomous Systems. Venice, Italy. pp. 918–925.
[3] Salemi, B., Shen, W.-M., Will, P. 2001. Hormone-Controlled Metamorphic Robots. in International Conference on Robotics and Automation. Seoul, Korea.
[4] Hou, F., Shen, W.-M. March 2006. Hormone-inspired Adaptive Distributed Synchronization of Reconfigurable Robots. In The 9th Intl. Conf. Intelligent and Autonomous Systems (IAS-9), Tokyo, Japan.
[5] Gonzalez-Gomez, J., Boemo, E. September 2005. Motion of Minimal Configurations of a Modular Robot: Sinusoidal, Lateral Rolling and Lateral Shift. Proc. of the 8th International Conference on Climbing and Walking Robots, CLAWAR, London. pp. 667-674.
[6] Sproewitz, A., Moeckel, R., Maye, J., Ijspeert, A.J. 2008. Learning to move in modular robots using central pattern generators and online optimization, Int. J. Rob. Res., vol. 27, num. 3-4, p. 423--443.
[7] Ijspeert, A.J. 2008. "Central pattern generators for locomotion control in animals and robots: A review," Neural Networks, vol. 21, no. 4, pp. 642 – 653.