

Automatic generation of reduced CPG control networks for locomotion of arbitrary modular robot structures

Stéphane Bonardi, Massimo Vespignani, Rico Moeckel, Jesse van den Kieboom,
Soha Pouya, Alexander Sproewitz, and Auke Jan Ijspeert

Biorobotics Laboratory,

École Polytechnique Fédérale de Lausanne (EPFL)

Lausanne, Switzerland

Email: {stephane.bonardi, massimo.vespignani, auke.ijspeert}@epfl.ch

Abstract—The design of efficient locomotion controllers for arbitrary structures of reconfigurable modular robots is challenging because the morphology of the structure can change dynamically during the completion of a task. In this paper, we propose a new method to automatically generate reduced Central Pattern Generator (CPG) networks for locomotion control based on the detection of bio-inspired sub-structures, like body and limbs, and articulation joints inside the robotic structure. We demonstrate how that information, coupled with the potential symmetries in the structure, can be used to speed up the optimization of the gaits and investigate its impact on the solution quality (i.e. the velocity of the robotic structure and the potential internal collisions between robotic modules). We tested our approach on three simulated structures and observed that the reduced network topologies in the first iterations of the optimization process performed significantly better than the fully open ones.

I. INTRODUCTION

Self-Reconfigurable Modular Robots (SRMR) are robotic units able to assemble and disassemble to form various structures depending on the task to be performed [23]. SRMR are potentially well suited for unknown environments and exploration tasks since they can adapt their structure autonomously to changes in their surroundings. But this adaptation capability brings additional challenges in terms of locomotion control of the resulting structure, since the morphology of such a structure might not be known beforehand. One successful approach to control the locomotion of modular structures are bio-inspired Central Pattern Generators (CPGs) [8], a network of coupled oscillators that allows to generate complex locomotion behaviors with a reduced set of control parameters. The main difficulty when using CPGs is the design of the best suited network for a given morphology. This step is most of the time based on trial and error and can quickly become time-consuming for large irregular structures. In order to find the most suited set of control parameters for the CPG network, optimization methods, such as Powell’s Method [17] or Particle Swarm Optimization [10, 15], can be used [16]. In these methods, the time required to optimize the gait of a structure is highly correlated to the number of parameters to optimize.

In this paper, we propose an automated method to generate reduced control networks for the locomotion of arbitrary structures made of modular robots, instead of considering a fully connected network with many parameters. In this work we consider structures that are neither fully linear (i.e. being composed of modules connected only in a open chain) nor fully cyclic (i.e. being composed of modules connected only in closed chain). Our approach is based on the decomposition of the robotic structure into morphologically relevant sub-structures, like body and limbs, and on the automated identification of bio-inspired articulation joints inside the structure. The number of optimization parameters is further reduced using existing symmetries in the structure. This method can be applied for self-recovery and fast re-optimization after structural changes due to hardware failure or voluntary morphological modifications, for example. A test situation could be the deployment of self-reconfigurable modular robots in an unknown environment where they would quickly need to re-learn some efficient gaits after reconfiguring during a time-critical task.

Our work is driven by the two following hypotheses:

- **Hypothesis 1:** the use of bio-inspired functional patterns and symmetries to generate the architecture of a CPG network controller for locomotion significantly increases the speed of convergence towards an acceptable¹ solution in terms of forward velocity and collision, compared to using a fully open CPG network.
- **Hypothesis 2:** the quality of the solution (in our case the velocity after convergence and the potential internal collisions between modules) is not significantly modified in comparison to a fully open optimization.

This paper is organized as follows. In section II, we review the existing approaches for controlling modular structures with dynamically changing morphologies. We then introduce in section III the robotic test platform Roombots as well as the basic control architecture used. Afterwards we describe

¹In our case, *acceptable* means capable of moving at a reasonable velocity above some minimum threshold.

our method to find relevant sub-structures inside any modular configuration (section IV) and explain how a reduced control network can be generated based on this differentiation and on the concept of distance-based symmetry (section V). We validate our approach using three structures in simulation (section VI) and discuss our results (section VII) before concluding (section VIII).

II. RELATED WORK

Modular robots offer the advantage of morphology that can change depending on external factors (e.g. changes in the environment) or internal ones (e.g. sudden hardware failure). This flexibility brings an additional challenge in comparison with monolithic robots in terms of design of efficient controllers. Moreover the increase in the number of degrees of freedom with each module added to the structure makes it difficult to hand-design specific gaits. Monolithic robots can also have to cope with a change in their morphology due to hardware issues, requiring as a consequence a re-design of their locomotion controller. The optimization of the set of parameters to generate efficient locomotion is often time consuming.

Since the early work by Yim [25] on the caterpillar locomotion of Polypod robots, several approaches have been proposed for the control of the locomotion of structures made of modular robots. For example, Shen et al. [18] proposed a hormone based method to control the locomotion of CONRO robots, Stoy et al. [22] used role-based control and cellular automata, and Yu et al. [26] described a consensus based approach for the locomotion control of 2D modular robots. CPGs, implemented as systems of coupled oscillators, have been applied for locomotion control by several researchers for distributed locomotion control and various techniques have been investigated [8, 12, 13, 7, 20]. The main drawbacks of the presented approaches is that they consider a fixed morphology and require the manual design of the CPG network, which might prove to be a tedious task for large structures. Some authors [19, 1, 11] used evolutionary methods and co-evolution to make the robot discover its own morphology, or used genetic algorithms to evolve possible gaits for given structures [9]. Those methods are often computationally demanding and time consuming, making them difficult to transfer on-board and on-line. More recently, accelerated learning methods have been investigated [3, 4, 5] based on a distributed and morphology independent learning process. The main difference with our approach is that we propose to optimize beforehand the control network itself instead of approximating the learning reward for the different possible actions. Christensen et al. [2] described a control framework to generate full body behavior based on the decomposition of the structure into bio-inspired parts (like muscle or bones) with pre-defined function (e.g. muscles can contract). The control is then done at the level of those sub-parts, abstracting away their individual components. Although this approach is similar in essence to our method, the main difference is that we propose an automatic detection

of bio-inspired joints and symmetries in any arbitrary structure instead of considering predefined structures built from known sub-parts.

III. PLATFORM AND CONTROL FRAMEWORK

A. Roombots

We test our techniques on a simulated model of our self-reconfigurable modular robot Roombots (RB) [21]. Compared to other SRMR, we chose to use RB because of the large variety of gaits that can be obtained with few modules, thanks to their three degrees of freedom capable of both oscillation and continuous rotation. A RB module is composed of four half-spheres linked together using revolute joints with continuous rotation capabilities. Using four-way symmetric compact Active Connection Mechanisms (ACMs, up to 10 per module) each RB module can autonomously connect to and disconnect from another module or a passive connector embedded in the environment. A single RB module weighs around 1.4kg and any of its joints can provide sufficient torque to lift at least one additional RB module.

B. Control architecture

We considered as locomotion controller a network of coupled non-linear oscillators mimicking the Central Pattern Generators found in many vertebrates [8]. The control inputs for this CPG are the amplitude A_i , the offset X_i , and the phase lags ψ_{ij} of each oscillator i connected to oscillator j . We use one common frequency for all oscillators ($\nu = 0.2$ Hz, according to [14]), bi-directional couplings follow the rule such that $\psi_{ij} = -\psi_{ji}$ and all coupling weights are set to 2. We set the CPG output to produce oscillatory joint angle signals. The coupled phase oscillators are implemented by the following coupled differential equations:

$$\dot{\phi}_i = 2\pi \cdot \nu \cdot \sum_j w_{ij} \cdot r_j \cdot \sin(\phi_j - \phi_i - \psi_{ij}) \quad (1)$$

$$\dot{r}_i = a_i (A_i - r_i) \quad (2)$$

$$\dot{\theta}_i = r_i \cdot \sin(\phi_i) + X_i \quad (3)$$

where i and j are the indexes of the oscillator, θ_i is the oscillator output controlling the position set point of the degree of freedom (dof) number i , r_i is the signal amplitude, and ϕ_i the phase. Each oscillator i has a maximum of three parameters that are subject to optimization: the desired amplitude A_i , offset X_i and the phase lag ψ_{ij} to the following neighbor j . More information about CPGs can be found in [8].

In order to find the most efficient gait for each structure, we use a population-based algorithm based on Particle Swarm Optimization (PSO) [10, 6] to generate the set of CPG control input parameters. In this work, we used simulated (off-line) gait optimization in the simulation software Webots [24].

IV. BODY/LIMB FINDER

In many vertebrates, the body and limbs are clearly differentiated and play different roles in the chosen locomotion strategy. In order to benefit from this definition of specific sub-structures, we developed an automatic centralized algorithm, called *Body/Limb Finder* (BLF), to automatically identify body and limbs in an arbitrary modular structure. This structure is represented as an undirected graph in which each node represents a module and each edge represents a connection between two modules (as illustrated in Fig. 2 top right). The main idea of our approach is that the removal of the body from a given structure will lead to several disconnected elements that represent the limbs. Additionally, the body can be further decomposed into a linear part (or chain part) and/or a cyclic part. A cyclic part is defined as a closed loop of connected modules. The actuation strategy will vary depending on the type of body part considered. In the scope of our bio-inspired control approach, we have introduced a set of rules to identify relevant articulations within the structure: we are able to differentiate between spines, hips, knees and ankles. A special control pattern for each of those units will be introduced in section V. We first present the theoretical aspects related to the detection of those different elements inside a given structure and then describe the validation of our method using a statistical approach.

A. Theory

1) *Body/Limbs differentiation*: The BLF algorithm is primarily based on the notion of *bi-connected components* (bcc). A bi-connected component of a graph is a graph with no articulation vertices, meaning no vertices that, if removed, would lead to a disconnected graph. The BLF algorithm is composed of three main steps (illustrated in Fig. 1):

a) *Step 1: decomposition into bi-connected components*: We first obtain the different components of the graph. This gives us the linear parts (i.e. bcc composed of less than 2 nodes) and the cycles (i.e. bcc composed of strictly more than 2 nodes), if any.

b) *Step 2: finding the cyclic parts of the body*: We use the following rule to find the cyclic parts of the body. The cyclic parts of the body correspond to groups of modules that are fully linked together, meaning that at least two paths exist between any pair of the group. For each cycle found at step 1, we check the connectivity of the graph resulting from the removal of this cycle: if the remaining graph is still connected then the cycle is not part of the body.

c) *Step 3: finding the linear parts of the body*: For this step we consider the different nodes which compose the 2-nodes bcc found at step 1. We select the nodes using the following rules (the 2 conditions have to be validated):

- i Clustering power: if the removal of the node leads to a number of components for the remaining graph strictly

- greater than 2 then the node is a linear part of the body².
- ii Articulation: the node must be an articulation of the graph.

After that, we calculate the shortest path between the selected nodes and we include it in the linear part of the body (minus the intersection with the nodes found at step 2). The limbs are the disconnected components remaining after the removal of the previously found body.

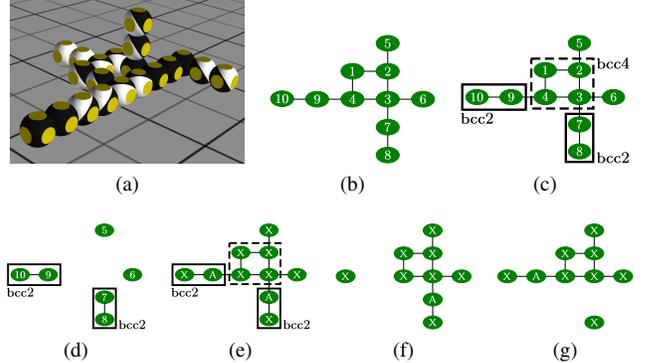


Fig. 1: The different steps of the body limbs finder illustrated on *struct10* (a), a structure with 10 modules used later in our experimental validation. (b) We start by converting the robot structure into an undirected graph in which each node represents a module and each edge represents a physical connection. (c) At step 1, we detect the bi-connected components (bcc) in the graph and sort them depending on the number of nodes they contain. (d) At step 2, we detect whether the previously found cycles (i.e. the bcc containing more than 3 nodes) are part of the body by testing if their removal leads to a disconnected graph: since we obtain 4 sub-graphs after the removal of bcc4, it is part of the body. (e) At step 3, we detect in the remaining bcc the articulation nodes, indicated with an A, and check whether they lead to more than two sub-graphs or not (f and g): since the removal of those articulations leads to only two sub-graphs, they are not part of the body.

2) Articulation rules:

a) *Spine*: Every joint inside the linear part of the body is part of the spine, except for hip joints. We chose to consider the cyclic parts of the body as unactuated.

b) Hip:

- A hip joint can belong either to a limb or to the body.
- A hip is a joint at the frontier between a limb and the body. This joint must therefore have at least one neighboring joint being part of the body.
- We want a hip joint to be as proximal as possible, which means that a hip joint defined as part of the body will be preferred over a hip joint belonging to a limb.

c) Knee:

- A knee joint must be part of a limb. There is only one knee per limb.

²We need to impose this condition because of the case of "long" linear limbs: in a linear limb composed of strictly more than 2 nodes, the central node has a clustering power of 2 and is an articulation, but it is not part of the body.

- A knee joint is at the center of the limb, between the pod and the hip (we define the pod as the most distal joint of the limb).
- If a hip joint and a foot joint are connected to each other with only one additional joint, no knee joint can be defined in the limb.
- If possible, a knee joint should be at equal distance from a foot joint and a hip joint. If such a joint cannot be found, we would choose the more proximal joint situated at the center of the limb as the knee joint.

d) *Ankle*: The rules to define an ankle joint are the same as the one describing a knee joint but considering the limb is starting at the knee joint. There is only one ankle per limb.

This set of rules, as well as the result from the BLF algorithm, is illustrated on Fig. 2 on a test structure with 9 modules shaped as a quadruped. The unclassified degrees of freedom are considered as *locked*.

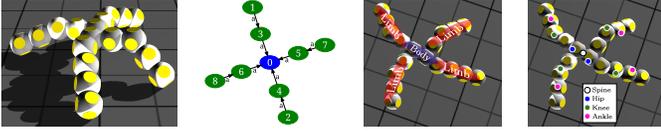


Fig. 2: Top left: A quadruped structure composed of 9 RB modules. Top right: the corresponding graph representation, used in the BLF. Bottom left: the detected body and limbs. Bottom right: the articulations detected using the set of rules described in subsection IV-A2.

B. Results

Since the notion of body is difficult to define and to characterize, we manually evaluated the "recognition rate" of our algorithm applied to *twenty* randomly generated structures. The number of modules per structure varies from 12 to 32. We considered the RB modules presented in section III as a test platform, but our method could be applied to other modular robots. Given that the goal of our method is to improve the locomotion control of a structure, we discarded unusable structures, for example, the ones with no limbs at all. The results matched our manual tagging for all the tested structures (see Fig. 2 for an example).

V. AUTOMATIC GENERATION OF REDUCED CPG NETWORKS

In this section, we describe the rules applied to design the control network of a modular structure using the coupled oscillators introduced in section III depending on the results from the BLF. Additionally, we introduce the notion of *distance-based symmetry* using a labeling function for the connection between modules and show how those symmetries can be used to further reduce the number of parameters in the control network.

A. Articulation network

The CPG network we derive from the articulations found using the BLF is inspired by the typical bone connection network present in many vertebrates (for example, the knees are usually connected to the hips and the hips to the spine). Each spine, hip, knee and ankle joint is driven by a single oscillator. The other degrees of freedom are considered as locked. We also assume that only the linear parts of the body are actuated (the cyclic parts being blocked) and that each spine is composed of a single joint (the most central one of the linear part, tie being solved at random) driven by a single oscillator. If more than one linear part is present in the body, each is controlled using a single oscillator. The parameter boundaries for the amplitude of an oscillator depend on the type of articulation. The coupling rules between the oscillators are the following (illustrated in Table IV):

- The spine oscillators are fully coupled together.
- The hip oscillators are fully coupled together. They are further coupled to the closest spine oscillator in the structure.
- The knee oscillators are only coupled to the corresponding hip oscillator (the one located in the same limb). If no hip is present in the limbs, the knee oscillator will act as a substitute and the same hip coupling rules will apply to it.
- The ankle oscillators are only coupled to the corresponding knee oscillator (the one located in the same limb).

Using this technique, the maximum number of parameters depends only on the number of limbs in the structure and is independent from the number of modules per limb. We consider only three parameters for each oscillator: the amplitude, the offset and the phase shift between the different oscillators. In this formula, we also consider bi-connected connection between the oscillators. If n represent the number of limbs, n_s the number of spine oscillators, and n_h the number of hip oscillators, the number of parameters $P_{reduced}$ can be computed as follows:

$$P_{reduced} = \sum_i^n 2(\delta_i^a + \delta_i^k + \delta_i^h) + 2 \times \sum_i^{n_h} (\delta_{h \in Body}) + 2n_s + \sum_i^n 2(\delta_i^a + \delta_i^k) + n_h(n_h - 1) + n_s(n_s - 1) + \sum_i^{n_h} 2(\delta_i^s) \quad (4)$$

where δ_i^a , δ_i^k , and δ_i^h equal 0 or 1 depending if the limb i contains an ankle, a knee or a hip, respectively, $\delta_{h \in Body}$ equals 0 or 1 depending if the hip is inside the body or not, and δ_i^s is equal to 0 or 1 if the hip is connected to a spine or not.

For a fully open network controlling a structure with m joints, each of them represented by one oscillator coupled to its closest neighbor, the total number of parameters P_{open} is equal to

$$P_{open} = 2 \times m + \sum_i^m (\delta_i^c) \quad (5)$$

where δ_i^c is the number of connections between oscillator i and its neighbors.

B. Distance-based symmetry

In order to further reduce the number of parameters required in our control network, we use geometrical symmetries between the limbs in the structure. If two limbs are considered symmetric, the corresponding oscillators share the same amplitude and the same offset (the phase shift remaining open to avoid restricting the possible gait patterns). To capture the geometrical organization of a given structure, a label for the connection between modules has to be defined. The label expresses the physical relationship between modules by encoding the source connector, the destination connector and the rotation between the 2 modules. We propose to use *Cantor polynomials* as a labeling function. Cantor polynomials are the only polynomials of degree 2 bijective from \mathbb{N}^2 to \mathbb{N} . They are defined by the following formulas:

$$f: \begin{array}{l} \mathbb{N}^2 \mapsto \mathbb{N} \\ (a, b) \rightarrow \frac{[(a+b)^2 + 3a + b]}{2} \end{array} \quad (6)$$

and

$$g: \begin{array}{l} \mathbb{N}^2 \mapsto \mathbb{N} \\ (a, b) \rightarrow \frac{[(a+b)^2 + 3b + a]}{2} \end{array}, \quad (7)$$

where $\forall (a, b) \in \mathbb{N}^2 \quad f(a, b) = g(b, a)$.

In order to obtain a bijection from \mathbb{N}^n to \mathbb{N} we only need to compose f or g by itself. In the case of the RB platform, we need to associate a unique integer to any given set of 3 values representing respectively the source connector (we called it a), the destination connector (we called it b) and the type of connection (we called it c). Thus, the labeling function that we are going to use is defined as follows:

$$l: \begin{array}{l} \mathbb{N}^3 \mapsto \mathbb{N} \\ (a, b, c) \rightarrow \frac{[(f(a,b)+c)^2 + 3f(a,b)+c]}{2} \end{array}$$

We can now associate any tuple (*source, destination, type*) with a unique positive integer. The only issue with this labeling system is that we have to take into account the orientation of the connection (a bijective function cannot be symmetrical and, as a consequence, if we switch the source connector and the destination connector, the computed label will be different).

The RB platform is equipped with 10 connection ports. Nevertheless the connectors placed on one outer hemisphere are equivalent considering a rotation of $\pi/3$. Similarly those two hemispheres can also be flipped (which corresponds to flipping the module) without modifying the functional characteristics of the connection. As a consequence the range for the connection ports is reduced to $[0..4]$ instead of $[0..9]$.

The distance-based symmetries in a structure are determined using the information provided by the body/limb finder applied to the graph representing the structure and labeled using the

previously mentioned labeling function. Only limbs of the same length are compared. The use of symmetries inside the structure is coupled with the information about the localization of the joint with respect to the body. The connections between modules are sorted into different groups depending on their distance from the body. The labels of the connection inside each limb are iteratively compared among groups: only fully identical limbs are considered symmetric.

VI. EXPERIMENTAL RESULTS

We considered three RB structures as test cases to evaluate our method. The first structure is a quadruped made of 5 modules with all limbs symmetrical (called *quad5-sym*, shown in Fig. 3 on the left). The second structure is the same quadruped but with a limb connected to the spine with a different orientation, so that only 3 limbs are now symmetric (called *quad5-unsym*, depicted in Fig. 3 on the right). The last structure is a pseudo random asymmetric structure made of 10 modules (called *struct10*, shown in Fig. subfig:struct10webots). The first two structures were chosen to represent bio-inspired structures, with the distinction between fully symmetric and partially symmetric one. We decided to use *struct10* to test our method on a much larger structure in which no intuitive gait could be engineered and also for which a fully open optimization requires a significant amount of time to converge.

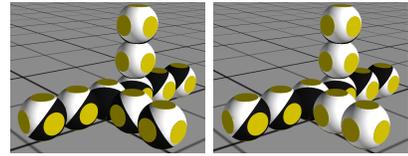


Fig. 3: Two of the three test structures: *quad5-sym* (left) and *quad5-unsym* (right). The difference between the two structures is that one of the modules (the one at the bottom right of the picture) is connected using a different orientation.

We compare an optimization of the parameters of the corresponding CPG network for each structure in the following four conditions:

- 1) Fully open optimization (FO): all the parameters of the network are considered open. One oscillator per dof is used. For each oscillator the amplitude is only constrained to $[0; \pi]$.
- 2) Body Limbs Finder reduced network (BLF): we use the technique described in subsection V-A to generate a reduced network for the structure. The amplitude parameter is constrained depending on the type of articulation considered (see Table I).
- 3) BLF network and symmetry finder (BLF-SYM): additionally to using the reduced network generated by the BLF, we consider symmetries as described in subsection V-B to further decrease the number of parameters to optimize. The amplitude parameter is constrained depending on the type of articulation considered (see Table I).

4) Symmetry finder (SYM): we applied distance-based symmetries between the limbs to reduce the number of parameters in the fully open CPG network controlling the structure. This step requires the use of the BLF to determine body and limbs in the structure, but, contrary to the previous case, no specific network structure is derived from this detection.

In terms of search space, the BLF, BLF-SYM, and SYM cases are sub-sets of the FO case. The number of parameters for each structure in the different cases are summarized in Table II. The parameters used for the PSO optimization for each case can be found in Table III. The corresponding CPG networks are depicted in Table IV.

TABLE I: The boundaries for the amplitude parameter depending on the type of articulation considered.

	Spine	Hip	Knee	Ankle
Min	0	0	0	0
Max	$\frac{2}{3}\pi$	$\pi/2$	$\pi/6$	$\pi/6$

TABLE II: The number of optimized network parameters for the three case structures in the four different conditions. The number in parenthesis indicates that the network is the same as one previously defined, and as a consequence, that it was not used.

	quad5-sym	quad5-unsym	struct10
FO	44	44	90
BLF	21	21	26
BLF-SYM	15	17	(26)
SYM	26	30	(90)

TABLE III: The fixed parameters for the PSO optimizations for the different structures.

Parameters	quad5-(un)sym	struct10
No. particles	80	160
No. iterations	800	
maximum velocity	0.6	
social factor	2.05	
cognitive factor	2.05	
constriction factor	0.729	
exp. duration t_{total}	30s	

We ran the PSO optimization twenty times with different initial random populations for the three structures *quad5-sym*, *quad5-unsym*, and *struct10*. For the latter, only the FO and BLF networks were tested, since no apparent symmetries are present in the structure. The fitness function f chosen for the optimization process takes into account the displacement of the structure and penalizes collisions between modules:

$$f = \frac{d}{t_{total}} \times c \quad (8)$$

where d corresponds to the displacement of the robot during the total experiment time t_{total} and c is a penalization factor used in case of self-collision equal to 0.001 if there is a collision and 1 otherwise. c was determined experimentally.

At each iteration and for each of the twenty optimization runs, we only consider the solution with the highest fitness.

We then computed the mean value of these sets of twenty best solutions and repeated the process for the three structures considered. The results are depicted on Fig. 4.

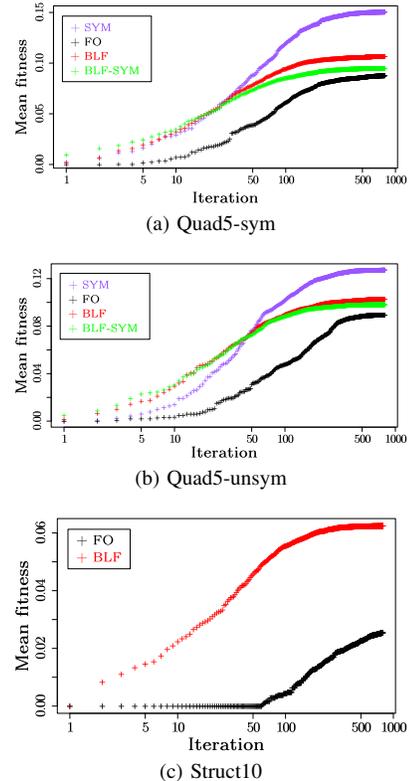


Fig. 4: The mean value of the fitness function over twenty runs for (from top to bottom) the *quad5-sym*, the *quad5-unsym* structure, and the *struct10* structure. The results are displayed in semi-log scale.

In order to compare the results of the best particles obtained at each iteration using the different network topologies, we performed single factor ANOVA tests (we tested the homoscedasticity of the residuals using the Levene’s test and we assumed they were normally distributed). The results are summarized in Table V.

VII. DISCUSSION

Our first hypothesis was that the fact of using a reduced CPG network generated using bio-inspired rules would significantly reduce the number of iterations needed to obtain an acceptable gait for a given structure. As we can see in Table V, the solutions generated using the reduced CPG network dominated the fully open population at least to the 30th iteration for the *quad5-sym* and *quad5-unsym* structures. For the bigger structure, we can clearly notice that restricting the search space by introducing automatically generated prior knowledge and boundaries to the parameters positively impacts the results: the fitness values are significantly better up to the 200th iterations and the convergence is significantly faster. We also observed that out of twenty runs of the *struct10* FO cases, we only obtain four valid solutions that converge to a set of parameters

TABLE IV: The different CPG networks for the three tested structures. In the fully open case, the circles represent the generic oscillators. For the BLF and BLF-SYM cases, the limbs are represented in green, the body in orange, and the shape coding is as follows: the spine oscillators are circles, the hip oscillators are squares, the knee oscillators are hexagons, and the ankle oscillators are crosses. For the BLF-SYM and SYM cases, the symmetric oscillators are indicated with the same stripe type.

Structures	FO	BLF	BLF-SYM	SYM
quad5-sym				
quad5-unsym				
		FO		BLF
struct10				

TABLE V: Iteration number after which no significant difference (with a $p - value < 0.05$) can be found between the samples of best individuals corresponding to the four network topologies. The values marked with a star indicate the iteration number before which no significant difference (with a $p - value < 0.05$) can be found between the samples of best individuals. The three numbers correspond to the three structures, respectively, from left to right, *quad5-sym*, *quad5-unsym*, and *struct10*. Before this iteration number, the ordering between the different network topologies can be seen on Fig. 4. *None* means that the two samples tested were not significantly different. *X* means that the test was not performed because the networks were not tested.

	BLF	BLF-SYM	SYM
FO	32/96/202	30/88/X	80/224/X
BLF	-	92/None/X	57/84*/X
BLF-SYM	-	-	35/82*/X

that did not induce self-collisions, emphasizing the need for a more robust method. If we select those solutions and compare them to four randomly chosen solutions from the BLF set, we observed that the BLF solutions are significantly better at the beginning of the optimization process (until iteration 71) before being dominated by the FO solutions (as illustrated in

Fig. 5), which remains consistent with our first hypothesis. Similarly, we observed in all cases that no significant differences could be found between our three proposed methods and the standard FO case at convergence, which remains consistent with our second hypothesis.

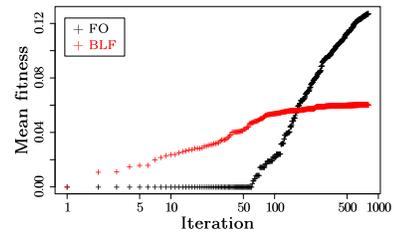


Fig. 5: In black, the mean value of the fitness function for the four optimization runs of the *struct10* structure in the FO cases in which no collisions were observed in the optimization best solution. In red, the mean fitness value of four randomly chosen optimization solutions in the BLF case.

The reduced networks seems to be less sensitive to local minima resulting from the complex optimization landscape, as illustrated on Fig. 4c. The results we obtained are as expected, since reducing the search space is known to have a positive effect on the speed of convergence, but through our study, we

managed to validate our hypothesis and to quantify for how many iterations it is still valid.

One typical test situation for our method would be some hardware failure of a self-modular robot during a time critical mission: the robot is then forced to reconfigure into a new shape and to re-learn how to move. It can, for example, connect to a remote cluster service to ask for new possible gaits but it cannot wait until the full convergence (meaning hours of computation for large structures). A similar scenario could involve a monolithic robot having to deal with a change in its morphology after some hardware issue. Our approach could be used in those two cases to characterize the new configuration of the robot and to propose corresponding reduced CPG networks to speed-up the optimization of the gait. With our proposed technique, after only five iterations (around one minute of optimization on average on our computer cluster³) we manage to provide a gait with a fitness value of 0.017, 0.024, and 0.016 (BLF, BLFSYM, and SYM) against only 0.005 in the FO case for *quad5-sym* (at least 3 times less on average, as depicted on Fig. 6). Similarly, for the *quad5-unsym*, the fitness after five iterations is almost an order of magnitude bigger with the reduced network (minimum 0.017) in comparison with the fully open case (0.002), as illustrated on Fig. 7. Similar trends can be observed at iterations 25, 50 and 100 (5, 10 and 20 minutes of computation, respectively), as shown on Fig. 6 and Fig. 7. The solutions found in the FO case in the early exploration phase were often heavily penalized because of the self-collision induced by the large boundaries set for the CPG parameters.

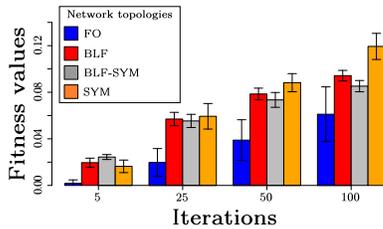


Fig. 6: The mean fitness values and standard deviation at iteration 5, 25, 50, and 100 for the different network topologies applied to the *quad5-sym* structure.

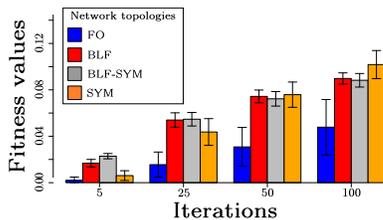


Fig. 7: The mean fitness values and standard deviation at iteration 5, 25, 50, and 100 for the different network topologies applied to the *quad5-unsym* structure.

We can also observe on Fig. 4a and 4b that the reduced

³Our cluster is composed of forty 2.00GHz quad-core Intel Xeon E5504 processors.

networks generated using the distance-based symmetry technique (SYM) obtain better results relatively to the two other reduced networks (BLF and BLF-SYM). This can be explained by the fact that the amplitude for the oscillators has larger boundaries than in the two other cases. A qualitative analysis of the resulting gait showed that in the SYM case, as well as on the FO case, the structure tends to rely much on almost rolling movement of some joints to increase its momentum. On the contrary, in the BLF and BLF-SYM cases, the structure tends to have a smaller amplitude of oscillation and favor animal-like displacement of the limb, making the obtained gaits more hardware friendly. The different solutions are illustrated in the video attachment.

Regarding the portability of the solutions to the hardware platform, we set the parameters of the simulation environment according to the results of a work that used meta-optimization on the RB robot in order to reduce the reality gap between simulation and hardware [14]. This should ensure that the gaits we obtained in this study remained consistent when transferred to a hardware platform.

VIII. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an automated method to generate a reduced CPG control network for the locomotion control of modular robots. We based our approach on the recognition of bio-inspired patterns in the structure playing the role of spine, hip, knee or ankle. Each of them are driven by a single oscillator with particular boundaries for the optimization parameters and specific coupling rules with the neighboring oscillators, with the goal of reducing the optimization time needed to find acceptable gait. We further reduced the number of parameters required in the optimization by automatically considering the existing symmetries in the structure.

By comparing the results obtained with three different structures, two quadrupeds and one pseudo-random structure composed of 10 modules, we noticed that our method leads to significantly better results during the first iterations, making the goal of re-optimizing a locomotion strategy (for example to cope with an unexpected change in the morphology of the robot due to a hardware failure) online and on-board a reachable goal.

Despite our method being generic, our preliminary study involved a restricted number of structures and was focused on a particular robotic platform. We are planning in the future to further extend our work to different types of modular robots and to increase the number of modules per structure to emphasize the gain induced when using our method.

ACKNOWLEDGMENT

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research Robotics. We gratefully acknowledge the help of Luc Guyot and Sébastien Gay for their support regarding the simulation environment and the optimization process.

REFERENCES

- [1] Josh Bongard, Victor Zykov, and Hod Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006.
- [2] David Johan Christensen, Jason Campbell, and Kasper Stoy. Anatomy-based organization of morphology and control in self-reconfigurable modular robots. *Neural Computing and Applications*, 19(6):787–805, 2010.
- [3] David Johan Christensen, Alexander Spröwitz, and Auke Jan Ijspeert. Distributed online learning of central pattern generators in modular robots. In *From Animals to Animats 11*, pages 402–412. Springer, 2010.
- [4] David Johan Christensen, Jørgen Christian Larsen, and Kasper Stoy. Adaptive strategy for online gait learning evaluated on the polymorphic robotic locokit. In *Evolving and Adaptive Intelligent Systems (EAIS), 2012 IEEE Conference on*, pages 63–68. IEEE, 2012.
- [5] David Johan Christensen, Ulrik Pagh Schultz, and Kasper Stoy. A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots. *Robotics and Autonomous Systems*, 2013.
- [6] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73, 2002.
- [7] Alessandro Crespi and Auke Jan Ijspeert. Online optimization of swimming and crawling in an amphibious snake robot. *Robotics, IEEE Transactions on*, 24(1):75–87, 2008.
- [8] Auke Jan Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, 2008.
- [9] Akiya Kamimura, Haruhisa Kurokawa, E Toshida, Kohji Tomita, Satoshi Murata, and Shigeru Kokaji. Automatic locomotion pattern generation for modular robots. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 714–720. IEEE, 2003.
- [10] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.
- [11] Hod Lipson and Jordan B Pollack. Towards continuously reconfigurable self-designing robotics. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1761–1766. IEEE, 2000.
- [12] Daniel Marbach and Auke Jan Ijspeert. Co-evolution of configuration and control for homogenous modular robots. In *Proceedings of the eighth conference on intelligent autonomous systems (IAS8)*, pages 712–719, 2004.
- [13] Daniel Marbach and Auke Jan Ijspeert. Online optimization of modular robot locomotion. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 1, pages 248–253. IEEE, 2005.
- [14] R Moeckel, Y. Perov, A. The Nguyen, M Vespignani, S Bonardi, S. Pouya, A. Sproewitz, J. Van den Kieboom, F. Wilhelm, and AJ Ijspeert. Gait optimization for roombots modular robots - matching simulation and reality. In *Proceedings of IEEE/RSJ IROS 2013, Tokyo, Japan, November 3*. IEEE, 2013.
- [15] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. *Swarm intelligence*, 1(1):33–57, 2007.
- [16] S. Pouya, J. van den Kieboom, A. Sproewitz, and A.J Ijspeert. Automatic gait generation in modular robots: to oscillate or to rotate? that is the question. In *IROS*, pages 514–520, 2010.
- [17] Michael JD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162, 1964.
- [18] Wei-Min Shen, Behnam Salemi, and Peter Will. Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots. *Robotics and Automation, IEEE Transactions on*, 18(5):700–712, 2002.
- [19] Karl Sims. Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372, 1994.
- [20] Alexander Sproewitz, Rico Moeckel, Jérôme Maye, and Auke Jan Ijspeert. Learning to move in modular robots using central pattern generators and online optimization. *The International Journal of Robotics Research*, 27(3-4):423–443, 2008.
- [21] A Spröwitz, R Moeckel, M Vespignani, S Bonardi, and AJ Ijspeert. Roombots: A hardware perspective on 3d self-reconfiguration and locomotion with a homogeneous modular robot. *Robotics and Autonomous Systems*, 2013.
- [22] Kasper Stoy, Wei-Min Shen, and Peter M Will. Using role-based control to produce locomotion in chain-type self-reconfigurable robots. *Mechatronics, IEEE/ASME Transactions on*, 7(4):410–417, 2002.
- [23] Kasper Stoy, David Brandt, and David Johan Christensen. *Self-reconfigurable robots: An introduction*. MIT Press, 2010.
- [24] Webots. <http://www.cyberbotics.com>. URL <http://www.cyberbotics.com>. Commercial Mobile Robot Simulation Software.
- [25] Mark Yim. New locomotion gaits. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2508–2514. IEEE, 1994.
- [26] Chih-Han Yu, Justin Werfel, and Radhika Nagpal. Coordinating collective locomotion in an amorphous modular robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2777–2784. IEEE, 2010.