



Toward Energy Autonomy in Heterogeneous Modular Plant-Inspired Robots through Artificial Evolution

Frank Veenstra^{1*}, Chloe Metayer², Sebastian Risi¹ and Kasper Stoy¹

¹Robotics Evolution and Art Laboratory, IT University of Copenhagen, Copenhagen, Denmark, ²Centre Genie Industriel, Ecole des Mines d'Albi-Carmaux, Albi, France

OPEN ACCESS

Edited by:

Cecilia Laschi,
Sant'Anna School of Advanced
Studies, Italy

Reviewed by:

Hiroyuki Ishii,
Waseda University, Japan
Ali Sadeghi,
Fondazione Istituto Italiano di
Technologia, Italy

*Correspondence:

Frank Veenstra
frve@itu.dk

Specialty section:

This article was submitted
to Bionics and Biomimetics,
a section of the journal
Frontiers in Robotics and AI

Received: 18 April 2017

Accepted: 18 August 2017

Published: 15 September 2017

Citation:

Veenstra F, Metayer C, Risi S
and Stoy K (2017) Toward Energy
Autonomy in Heterogeneous
Modular Plant-Inspired Robots
through Artificial Evolution.
Front. Robot. AI 4:43.
doi: 10.3389/frobt.2017.00043

Contemporary robots perform energy intensive tasks—e.g., manipulation and locomotion—making the development of energy autonomous robots challenging. Since plants are primary energy producers in natural ecosystems, we took plants as a source of inspiration for designing our robotics platform. This led us to investigate energy autonomy in robots through employing solar panels. As plants move slowly compared to other large terrestrial organisms, it is expected that plant-inspired robots can enable robotic applications, such as long-term monitoring and exploration, where energy consumption could be minimized. Since it is difficult to manually design robotic systems that adhere to full energy autonomy, we utilize evolutionary algorithms to automate the design and evaluation of energy harvesting robots. We demonstrate how artificial evolution can lead to the design and control of a modular plant-like robot. Robotic phenotypes were acquired through implementing an evolutionary algorithm, a generative encoding and modular building blocks in a simulation environment. The generative encoding is based on a context sensitive Lindenmayer-System (L-System) and the evolutionary algorithm is used to optimize compositions of heterogeneous modular building blocks in the simulation environment. Phenotypes that evolved from the simulation environment are in turn transferred to a physical robot platform. The robotics platform consists of five different types of modules: (1) a base module, (2) a cube module, (3) servo modules, and (4,5) two types of solar panel modules that are used to harvest energy. The control system for the platform is initially evolved in the simulation environment and afterward transferred to an actual physical robot. A few experiments were done showing the relationship between energy cost and the amount of light tracking that evolved in the simulation. The reconfigurable modular robots are eventually used to harvest light with the possibility to be reconfigured based on the needs of the designer, the type of usable modules, and/or the optimal configuration derived from the simulation environment. Long-term energy autonomy has not been tested in this robotics platform. However, we think our robotics platform can serve as a stepping stone toward full energy autonomy in modular robots.

Keywords: modular robots, plant-inspired robots, artificial intelligence, energy autonomy, evolutionary algorithms

1. INTRODUCTION

Energy autonomy in artificial systems is beneficial for long-term autonomous behavior in single or multi-robot applications required for, e.g., monitoring and exploration. However, it can be complicated to design energy autonomous systems since this depends on the energy demand and energy acquisition of the robot. In evolutionary robotics, locomotion and object manipulation are among the most prominent objectives for robots (Vargas et al., 2014). Though the same principles in evolutionary robotics can be implemented for energy autonomy in robotic systems, energy autonomy is usually implemented on different robotic systems where the robot is able to utilize energy from light (Noth et al., 2006; Afarulrazi et al., 2011) or microbial fuel cells (Ieropoulos et al., 2003; Philamore et al., 2015). Being able to automatically design robotic systems that are geared toward energy autonomy could give us unintuitive solutions that might be more effective than traditional solutions. Since plants have mastered extracting energy from light in terrestrial environments, they are taken as a source of inspiration. Although plants have many unique features, we do not consider the implementation of many of them since it is either impractical or infeasible to implement in robotic systems. However, an abstraction of plant development is implemented in the form of a generative encoding (section 2.3). Having a modular robotic system that conforms to energy optimization could give rise to self-reconfigurable robots that maximize energy acquisition.

The ultimate mechanisms that drive evolution in organisms are enacted through the external energy influx into an open system that innately works against its thermodynamic equilibrium. Considering the earth as an open system, this energy influx is mainly acquired from the sun in the form of light and to a lesser degree in the form of heat and chemicals from earth. Being able to acquire this energy from the sun to generate complex organisms is thus a vital for evolution. The initial multicellular organisms that roamed the planet acquired energy from their environment in the form of chemicals and light (Reece et al., 2010). These carbon-based life forms were the precursors to plants which became the expert terrestrial organisms for acquiring energy from light. Although plants appear to be slow, they are highly optimized for gathering energy from their environment. Many plants contain track and actuation mechanisms that optimize their productivity by absorbing light more efficiently (Ehleringer and Forseth, 1980). This tracking behavior emerges in the form of circumnutation through *heliotropism* (Graham and Wilcox, 2013). In *phototropism*, a form of heliotropism, the plant actively grows toward a light source as in the case of the sunflower (Atamian et al., 2016; Kutschera and Briggs, 2016). Some other species of plants can adjust their leaves or flowers to bend toward the light with a structure called the *pulvinus*. The pulvinus is a structure below the leaves and flowers that twists and pivots them through adjusting turgor pressure (Song et al., 2014). In general, leaves adjust their angle and move their surface perpendicular to the sun when conditions are optimal in a process called *diaheliotropism*. These are some general adaptive mechanisms through which a plant can adapt their morphology based on specific stimuli. Other morphological traits are

hard-coded in the genome such as *phyllotaxis* (Prusinkiewicz and Lindenmayer, 1990), the arrangement of leaves on a plant. Since the amount of solar modules used in the robotic system is limited, as will become apparent in the next sections, no fair comparison can be made with the complex forms of phyllotaxis seen in plants. However, phyllotaxis is mainly driven by innate factors requiring limited feedback from the environment which is similar to the open-loop control implemented in the generative encoding and evolutionary algorithm.

A mechanism exactly like phototropism is a feat that we are unable to implement in the robot since this would require some form of growth. This could, however, be accomplished with soft robots (Sinibaldi et al., 2014; Heinrich et al., 2016; Sadeghi et al., 2016; Vergara et al., 2017). Although allowing for continual growth show promising bio-inspired applications in robotic systems (Sadeghi et al., 2014), this also brings forth difficulties regarding the reconfigurability and reuse of robotic parts. Our modular robotics approach, therefore, does not allow continual growth but enables reuse and reconfigurability. The bio-inspiration of the implemented robotic platform is motivated by the rotational movements of the pulvinus since this structure allows for movement of the leaves without major morphological change. With an open-loop control system, we can find simple control mechanisms that allow for the optimal energy absorption. While artificial plant systems have been implemented in cellular automata (Hogeweg, 1988; Balzter et al., 1998) as well as virtual creatures (Zamuda and Brest, 2014; Corucci et al., 2016; Veenstra et al., 2016; Zahadat et al., 2016), they have rarely been investigated in a three-dimensional embodied approach other than light-tracking solar panels (Prinsloo and Dobson, 2015). Many evolutionary robotics experiments have focused on acquiring behavior typical of consumers (Sims, 1994a,b; Pfeifer and Bongard, 2006; Vargas et al., 2014), we instead look at how primary energy producers can evolve in artificial systems. Some robotic platforms have been designed to cope with energy autonomy (Greenman et al., 2003; Ieropoulos et al., 2003; Philamore et al., 2015) although research in this area is still limited. Our evolutionary system, instead of modeling nature, aims at implementing feasible evolved designs into real modular robots. Our modules are, therefore, based on a heterogeneous modular design. A robotic module being an independent unit that encapsulates part of the robots functionality (Stoy et al., 2010). The modular robotics approach eases the construction process of different morphologies as well as the capability of changing the morphologies on the spot. By simply implementing the same connection mechanism on each module of the modular robot, different modules can be joined together to form unique and feasible robot phenotypes. Our platform, thus, allows for light absorption in a robotic system that implements actuators with similar degrees of freedom as the pulvinus with the aid of a plant-inspired developmental algorithm.

2. MATERIALS AND METHODS

A simulation environment is used to evolve modular robots and the evolved phenotypes are transferred to the real world. The simulated robots were optimized for harvesting energy from light using simulated solar panels. The physical robot simply

follows the parameters that have been evolved by the simulator. Different environments were simulated that in turn shaped the search space of the robot. Moreover, five different types of modules were designed that could be simulated. These modules could all be connected using the same connection mechanisms. Enabling morphological change and open-loop evolution of the control system enables our platform to evolve unique robot phenotypes that can be transferred to reality. The simulated robots could never use more modules of a type than was maximally allowed. Every evolved robot could, thus, be created in reality. Varying the amount of solar panels that we can implement in our robotic system gave us an idea on the amount of modules, we require to evolve sufficient robotic phenotypes. A modular robot was constructed by attaching one module to another module manually. Since the connection mechanism is based on magnets, the construction process is a simple snap-on procedure.

Virtual Robot Experimentation Platform (V-REP; version 3.4.0) (Rohmer et al., 2013) is used as the robotics simulation platform and the evolutionary algorithm was implemented as a C++ based DLL plugin. The simulation environment consisted of a default floor and was simulated using the bullet dynamics engine (version 2.78). The dynamics settings were set to accurate (default) with a time step of 50 ms. The modules used for the simulator are based on the physical properties of the real modules (Figure 1) approximating similarity in size and weight. The modules consisted of a base module, a servo module, a cube module, and two types of solar modules as will be discussed in the next section. All modules contained connection faces that could either be male or female. The pairing of these two connection faces established a connection between two modules allowing for the composition of a modular robot (section 2.2). A list of the materials used to construct these methods can be found on our website.¹

2.1. Modules

Five different types of modules were used in our approach: a cube module (Figure 1A), a base module (Figure 1B), a servo module (Figure 1C), and two types of solar modules. One of the solar modules simply contained two solar panels that were

joined together (Figure 1D). The other, more elaborative, solar panel module (flower module) was designed with the aim of allowing for more plant-like adaptive behavior in the system (Figure 1E). However, the increased complexity of this flower module also brings about increased complexity in the simulator. Therefore, this flower module is not used in the experiments described in this paper but rather serves the role of informing the reader about the potential future implementations of solar modules. Throughout different modules, all custom parts were 3D printed using polylactic acid (PLA).

The base module (Figure 1B) is composed of a simple custom structure with three female connection faces. Three female connection faces were used to reduce the amount of possible connection faces limiting the search space of the robot. However, more connection sites could make more sense in a future implementation of the base module though this makes the state space landscape more convoluted due to more possible configurations of the modules. This base module was simulated statically meaning that its physical properties were not simulated. Instead, the module was fixed in place ensuring that the structure stays in the same position conforming to the sessile nature of plants. The three female connection sites were 55 by 55 mm and were placed next to each other at a 45° angle. The cube module (55 mm by 55 mm by 55 mm; weight is 300 g) is used as a structural building block for the modular robot to which other modules are attached. This cube has five female connection sites (top, right, left, front, and back) and one male connection site (bottom).

The servo module (80 mm by 55 mm by 55 mm, weight is 160 g) is based on a module from the EMERGE modular robotic platform (Moreno et al., 2017). This module contains three female connection sites attached to the shaft, and one male connection site on the bottom attached to the chassis of the servo motor. It houses a dynamixel AX-18a servo motor. In the experiments, the range of the servo motors is limited to +90° and -90°. The difference of the servo module used in this paper is the attribute of having 4 data channels compared to the 2 data channels implemented in the EMERGE module. Also, infrared sensors that were present in the EMERGE module are not implemented in the servo modules used for this paper. Infrared sensors were not implemented since they take up additional space on the PCBs and consume additional energy. At the location of the infrared sensor in the emerge module, there is simply a 3-mm-wide hole that can

¹List of materials can be found on our website at: <https://sites.google.com/view/emergemodular/projects/energy-autonomy>.

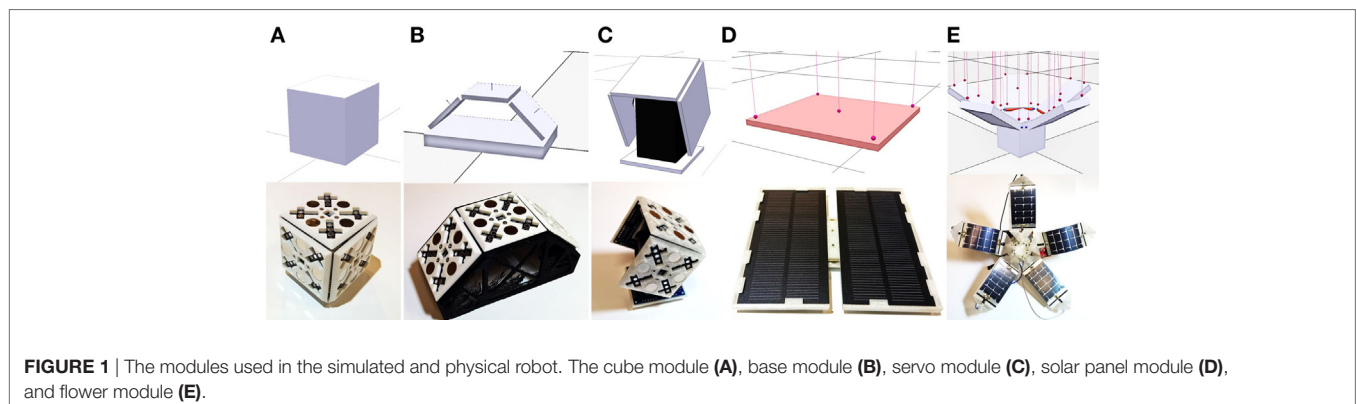


FIGURE 1 | The modules used in the simulated and physical robot. The cube module (A), base module (B), servo module (C), solar panel module (D), and flower module (E).

be used to adjust a screw to tighten the connection between the servo bracket and the servo motor. This tightening can in turn lead to a stiffer joint that is harder to actuate but also harder to move passively. This is beneficial to adjust on the spot, since we do not want an inactive servo joint to buckle under the weight of the robot. Servo motors closer to the base of the robot can, therefore, be tightened more while those at the end can be looser. The dynamixel AX-18a servo motor is controlled via a PID controller that is directly connected to the robotics simulator.

The first type of solar panel module (referred to as solar module) implemented two simple solar panels (141 mm by 124 mm by 6 mm, weight is 66 g) on a 3D printed bracket (**Figure 1D**). Two 0.8 W, 5.5 V monocrystalline silicon solar panels (118 mm by 63 mm) were mounted on the case of the solar module. These two solar panels were connected in series and could output a voltage of 11 V (**Figure 2A**). An additional diode was attached to the solar module to prevent current flowing in the opposite direction. The current acquired from the solar panels was converted to 12 V using a step-up voltage regulator (not shown in the figures) to ensure a 12 V output. This output voltage was connected to two data channels connected to the base module. The solar module was simulated with a basic morphology with similar dimensions and contained five proximity sensors to determine light absorption. These proximity sensors were positioned at each corner and in the middle of the solar module. Each proximity sensor served as a ray tracer simply determining whether there was an object in between the starting position of the ray and a light source. In addition, the difference between the z directional vector of the ray compared to the orientation of the solar panel gives us the angle at which light impacts the solar

panel. This angle of attack linearly influenced the contribution of light absorption to the fitness of an individual.

The flower module (**Figure 1E**) is composed of five SP3-37 flexible solar panels. Each panel is mounted on a 3D printed petal. These artificial petals could not bend making the use of the flexibility in the solar panels redundant. The flexible solar panels, thus, effectively function the same as a regular solar panel. The five petals are connected together through a system of cranks and a small circular platform. A rack and pinion system combined with a small servomotor pushes the platform and actuates the petals. Two hinge joints connect the circular platform to the petals, one at the inner edge of a petal (center of the flower) and the other closer to the middle of the petal. In our case, a hinge joint on the edge of the petals is attached to the platform containing the rack while the hinge joint closer to the middle of the petal is connected to the piston. When the piston moves the rack up and down, the petals actuate and open or close the flower respectively. The petals can open to a certain degree and this position can be optimized depending on the solar intake. The energy harvested is stored in a LiPo battery. This battery was used to power the servomotor actuating the petals making the flower module (not the modular robot) energy autonomous. Though the flower module is not used in the evolutionary runs, it can also be connected to the same power grid of the modular robot in order to charge the 12 V battery. In addition, the flower module contained a MMA842Q accelerometer (3-axis), which enables the flower module to detect movement. This sense of movement can inform the flower module whether it should close or open its petals under harsh or favorable conditions. For example, at night, during heavy rain or in windy scenarios, it would be better for the flower module to close and

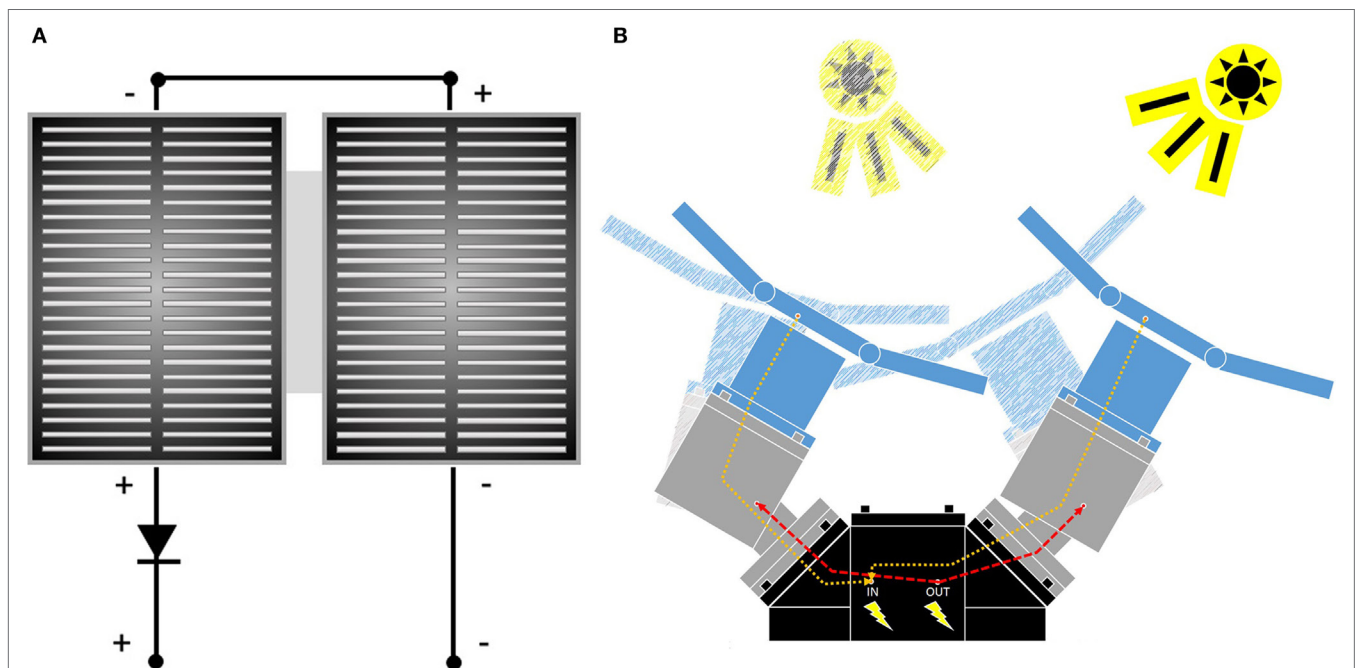


FIGURE 2 | Schematic diagram of the modular robot containing solar panels. The Solar panel module consists of two solar panels that are connected to one another in series (**A**). A diode is attached to prevent current from flowing in the reverse direction. As can be seen in (**B**), the solar panel modules pass power down to the base module where the battery is located.

protect its solar panels. Especially if the solar panels can unfold in an origami-like manner (the original intend of implementing the flexible solar panels), the surface area would be relatively weak highlighting the need for the flower and solar panels to close. The flower module contained five solar panels and five shapes were used to simulate them. Similar to the solar module, each solar panel simulated five proximity sensors that were used to measure the light absorption. This led to the increased computational requirements since 25 proximity sensors and six shapes were used in the flower module. Therefore, the flower module was excluded in the robot simulator.

One base module, one cube module, eight servo modules, five solar modules, and two flower modules were created in reality and the simulator was, thus, restricted to use this amount of modules. This limits the potential resulting phenotypes of the evolutionary algorithm and constrains the search space but enables evolution of feasible modular robots. A direct feedback from simulated modular robots and actual modular robots is thus present. All the morphological parameters of the individual modules were fixed to represent their physical counterparts. The eventual physical modular robot was constructed based on the phenotypes that evolved in the simulation environment. An implementation of the dynamixel protocol (protocol used to communicate with the servo motors) in the plugin allowed for the direct control of the actual robot in the simulation environment. The dynamixel protocol utilizes serial communication to communicate with the dynamixel AX-18a servo motors. Through sending instruction packets, individual motors can be controlled using a single communication bus. The values of the dynamixel servos needed to be transformed into hexadecimal values that indicate parameters, such as the desired speed and position of the servo motors. The solar modules implemented in the evolutionary algorithm did not have any type of actuation. Connecting a few modules together to form a robotic entity enabled current to flow from a power source toward the servo modules while also allowing for the flow of current from the solar panels to the power source (Figure 2B). The solar modules could directly charge a 12 V lithium ion battery that was used though the efficiency of charging the batteries greatly depended on the light saturation of the solar panels.

2.2. Connectivity

The real modules are connected to one another via magnet-based connection faces (Figure 3A). The connection faces contained PCBs with pads and spring pins that enabled electricity to be routed between modules. The connection faces could either be male or female. The male connector faces contained spring pins that were soldered on the pads of the PCB (Figure 3B). The female connection face contains 3 mm pads to ensure a connection with the spring pins on the male connection face. The PCBs included six separate channels through which electricity could be routed. Two of the channels were dedicated to power and ground while the four other channels routed data channels. Two of these data channels were used to enable a current to flow from solar panels back toward the initial power source. 3D printed hulls housed four cylindrical NdFeB (neodymium, iron, and boron) magnets (12 mm diameter, 3 mm depth). Two magnets can hold one another together with a force of roughly 13.5 N. The male connector hull has protrusions that ensure the fit and connection of the male connector to the female connector. The connectivity of the modules in the robotic system is depicted in a block diagram (Figure 4).

In the simulated modules, a force sensor was put in between two connection faces to detect the torque and force between two modules. Since the modules have never disconnected in the physical robot yet, the modules could only break from one another if 100 consecutive threshold violations occurred where the threshold was set to 10,000 Nm of torque and 80,000.0 N of force. The chosen values are incredibly high to ensure that the modules stay connected to one another except in the case of a faulty collision. When the simulated robot simulated modules that inaccurately collided, disconnection between modules could occur. These inaccuracies did not occur in the eventual simulator. As a failsafe, the fitness value of individuals that contained broken force sensors was set to zero. The construction of the modular robot occurred before the simulation started. A generative encoding (section 2.3) translated the genome and created the robot phenotype. This genome consisted of simple morphological parameters and an additional neural network (section 2.4.1) for the servo modules.

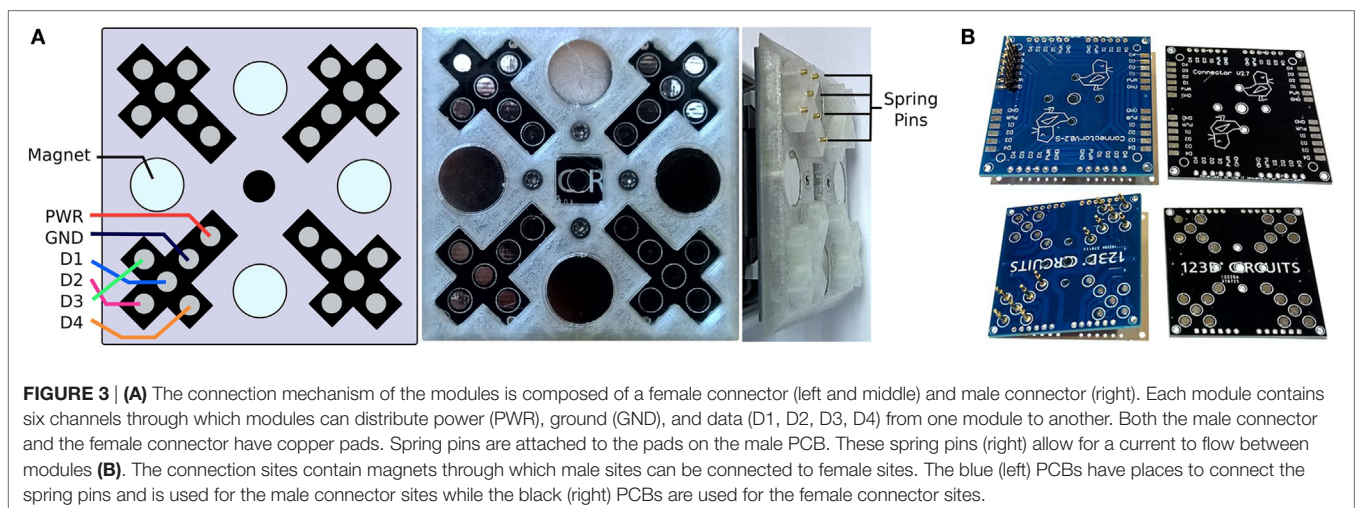


FIGURE 3 | (A) The connection mechanism of the modules is composed of a female connector (left and middle) and male connector (right). Each module contains six channels through which modules can distribute power (PWR), ground (GND), and data (D1, D2, D3, D4) from one module to another. Both the male connector and the female connector have copper pads. Spring pins are attached to the pads on the male PCB. These spring pins (right) allow for a current to flow between modules **(B)**. The connection sites contain magnets through which male sites can be connected to female sites. The blue (left) PCBs have places to connect the spring pins and is used for the male connector sites while the black (right) PCBs are used for the female connector sites.

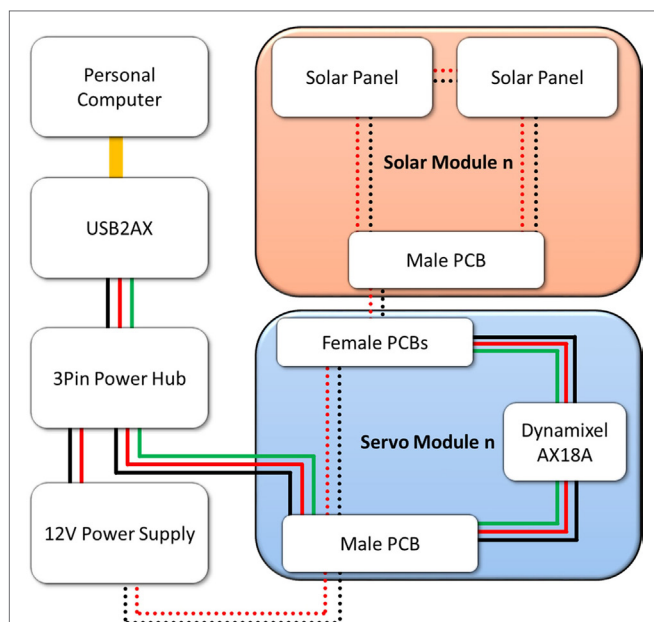


FIGURE 4 | Block diagram of the modular robotics platform. A personal computer is connected to a central 3-pin power hub via a USB2AX interface. A 12 V power supply is also connected to the same 3-pin power hub. The power hub is in turn connected to a male or female connector face of the modular robot. This connection distributes power to all connected modules. The block diagram only shows the Servo and the Solar module. The base and cube module simply transfer all electrical current from their male connection faces to their male connection face and vice versa. Two of the channels are connected separately to the solar panel modules which transfer power from the solar panels to the power source. The red and black connections indicate power and ground, while the green connection represents the communication wire. The dotted lines represent the power and ground connections to the solar panels.

2.3. Generative Encoding

The generative encoding was based on a context sensitive Lindenmayer-System (L-System) (Lindenmayer, 1968; Lindenmayer and Jürgensen, 1992; Prusinkiewicz and Lindenmayer, 1997)—a parallel rewriting system—as implemented in Veenstra et al. (2017). The variables of the L-System represented the different modules of the robot. A variable describes a *state* that is decoded as the morphology, control, and attachment rules of the modules. In the simulation environment, these modules represent the cube, base, servo, and solar modules. One state was dedicated to the base module and was the axiom of the L-System. Another state represented the cube module. Four states represented the servo module and two states contained the parameters of the solar module. Every state can assign unique attachment rules and control parameters. The generative encoding does not necessarily lead to all genotypic parameters to be expressed in the phenotype. Instead, some genotypic parameters could be dormant which allowed for genetic drift to occur across generations. E.g., eight servo modules could be generated in the phenotype if one servo state is expressed eight times in the phenotype, but this phenotype could also be the result of one state being expressed six times and one state being expressed two times in the phenotype. The

phenotypic representation of the modules is determined by the attachment rules of the L-System that were stored in the genome. The attachment rules of the modules included the information of which module is connected to what connection face in which orientation. The L-System was, moreover, context sensitive since a module cannot be placed at an attachment site if another module already occupied it. Modules cannot be created if this caused a collision with other created modules. The different modules were colored in the phenotype based on their states so recursive expression could easily be detected. Five iterations of the L-System were done to create the robot phenotypes starting with the base module as the axiom.

2.4. Evolutionary Algorithm

Our implemented evolutionary algorithm was based on a steady-state genetic algorithm (Syswerda, 1991). The simulations were limited to 46,000 evaluations per evolutionary run. 46,000 evaluations were chosen as a trade-off between computational time and performance. A population size of 92 individuals was used and was simulated for 500 generations. The population size is a multiple of the 23 simulation instances that ran in parallel on a cluster node containing 24 computing cores. One core was dedicated to running the evolutionary algorithm itself while the remaining cores evaluated individuals in V-REP. For each experiment, 12 evolutionary runs were performed with different initial seeds. The evolutionary algorithm was generational and randomly selected one parent from the population to produce an offspring. The initial population consisted of individuals created from randomized genomes. Though the offspring were haploid, a crossover function allowed certain states of another parent to be transferred to the offspring with a 20% chance. In our case, we simulated up to eight states. After crossover occurred, the offspring were mutated with a morphological mutation rate of 0.15 and a control mutation rate of 0.1. The morphological mutation accounted for any aspect of the generative encoding replaced by a random value with a 15% chance. Only the four states of the servo modules contained a neural network and the neural network had several mutation operators that could be activated with a 10% chance. There were four mutation operators working on the neural networks: *change connectivity* that altered the edges between the neural network; *add neuron*; *remove neuron*; and *change neuron* swapping an interneuron for a new interneuron of another type (section 2.4.1). The maximum amplitude of the servo arm ranged from -90° and $+90^\circ$.

2.4.1. Neural Network

In a previous experiment, simple sinusoidal wave functions were implemented to control simulated servo modules (Veenstra et al., 2017). The same sinusoidal patterns are implemented here though they are implemented in a network of neurons. This artificial neural network, implemented in the servo modules, consists of one input neuron, up to six interneurons and one output neuron. In our case, the input neuron is always activated. A recurrent interneuron layer consisted of neurons that simulated a fixed sinusoidal output pattern or a neuron with a binary step function. The phase, amplitude, and frequency of the sinusoidal

neuron could be altered and the mutable parameters of the binary step neuron were the threshold value (between -1.0 and 1.0) and the output weight (between -1.0 and 1.0). The output neuron simply outputs a value between -1.0 and 1.0 based on its inputs. This value is in turn transformed into a value that represents the absolute position of the servo motor of the module. In this neural network, the connections were not weighted, instead, once a neuron gets activated, all the neurons connected to the activated neuron receive the same output. This was implemented due to the small size of the neural network (one input and one output) and to limit the search space. A neural network could be implemented in each state of the servo module. This means that multiple servo modules could express the same neural network if the generative encoding created these neural networks from the same gene.

2.4.2. Fitness Function

The goal of the simulated robots was to absorb light within 5 s of simulation time. The simulation time was limited to 5 s since the modules were able to appropriately adjust the positions of their joints to the light source within the given simulation time. To approximate an actual day cycle in the physical world depends on the transformation of the simulation time which can be stretched according to a given environment. The fitness of each individual was determined by the amount of light that was absorbed by each ray of all the solar panels present on the robot and can be derived from equation (1). The gathered

light was calculated by subtracting the z directional vector of the z axis of each ray on the solar panel from the z directional vector of x axis of the solar panel. An additional cost function was added to represent an arbitrary energy expenditure of the robot.

$$F := \sum_{i=1}^m \rho_i - \sum_{j=1}^n \epsilon_j, \quad (1)$$

where F represents the fitness value obtained by calculating the total amount of light absorption that occurred within the 5 s of simulation time. The “:=” operator represents an update of the fitness value with right-hand side at each time step. ρ represents the amount of energy gathered by each ray i at every time step. For each servo module j , the amount of energy ϵ used by all servo modules is subtracted. An additional cost function was added to compensate for modules that were disconnected due to the breaking of a connection site (not shown in equation). Breaking of modules never occurred in experiments though the initial fitness cost was implemented to ensure we did not reward malfunctioning robots.

2.5. Experimental Setup

The evolutionary runs were performed in four different environments (Figure 5). The first environment contained a light source located directly above the modular robot. The second environment consisted of four walls surrounding the modular

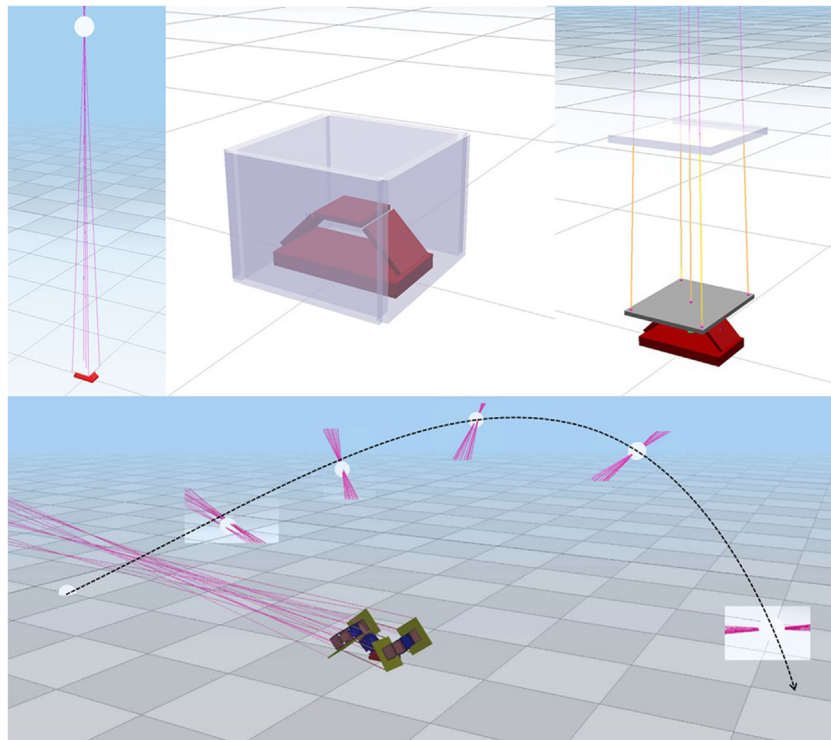


FIGURE 5 | Different Simulated environments: environment with stationary light source (top left); an environment where the initial base module is constrained (top middle); an environment where direct light is blocked by an object (top right); and an environment where the light source is moving in an arc (bottom). The line represents is an approximation of the path the light takes.

robot constricting direct outward growth of the phenotype of the modular robot. The third environment contained an object that blocked the phenotype from receiving direct light absorption. This was done in to motivate outward growth of the modular phenotype. The last environment consisted of a moving light source that mimicked the trajectory of the sun in winter in the northern hemisphere of earth. It was expected that the last environment would promote the evolution of blind control systems that enabled solar panels to tilt toward the sun. The position of the light source was calculated by two sinusoidal functions that were transformed into cartesian coordinates (equation (2)). The “:=” operator represents an update of the left-hand side variable with the term on the right-hand side as it is performed at each time step.

$$\begin{aligned} P_x &:= P_{x_start} \alpha_x + \tau \alpha_x \\ P_y &:= \sin(\tau) \alpha_y, \\ P_z &:= \sin(\tau) \alpha_z \end{aligned} \quad (2)$$

where α represents a specific scaling factor for transforming the position in cartesian coordinates. The x, y, and z positions of the light source are updated at each time step denoted with τ .

In the environment with the moving light source, the energy cost was a custom value. This cost value was either 0.0, 0.1, 0.5, 2.0 or 8.0. For every energy cost, 12 evolutionary runs were done. These values were chosen as they changed the evolutionary trajectory of the different evolutionary runs; whereas above a cost of 8.0, no change in the evolutionary runs could be seen.

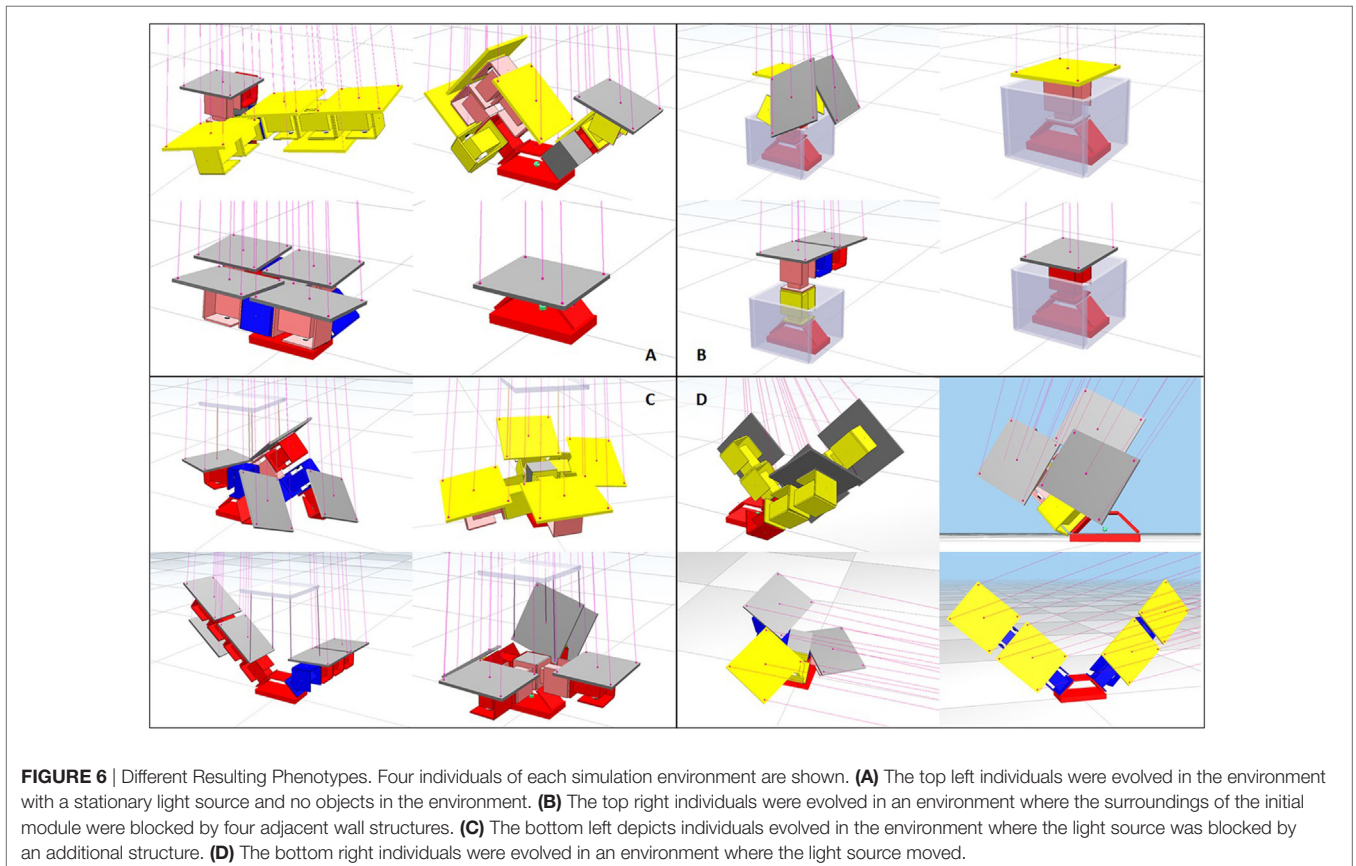
In addition, for each simulated environment twelve evolutionary runs were performed when simulating a maximum of one, two, or five solar modules. This is done to see if movement evolves in environments where a different amount of solar panels can be simulated. It is expected that when simulating a maximum of one solar panel, on average more angular movement can be detected in the solar panels compared to simulating more solar modules. This hypothesis is tested by measuring the difference in angular movement of each solar panel at each time step. This arbitrary measure of angular movement is stored together with the fitness values of the individuals.

3. RESULTS

The results shown in this section are separated in the resulting phenotypes of the eventual population of robots in different environments and the results on the impact of movement and energy cost on evolving the modular robots. The phenotypes give a clear overview of the types of robots that could be evolved while the energy costs explain emergent behaviors seen in the evolved robots.

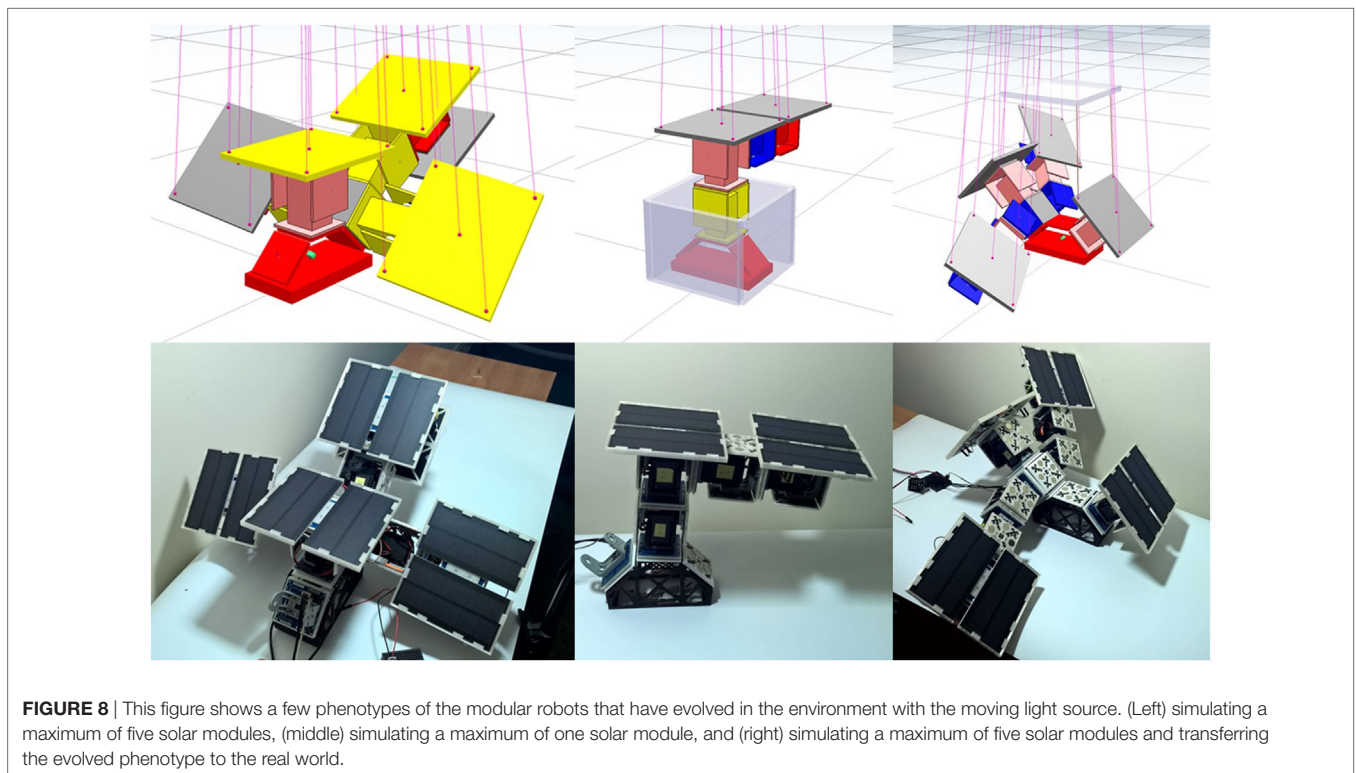
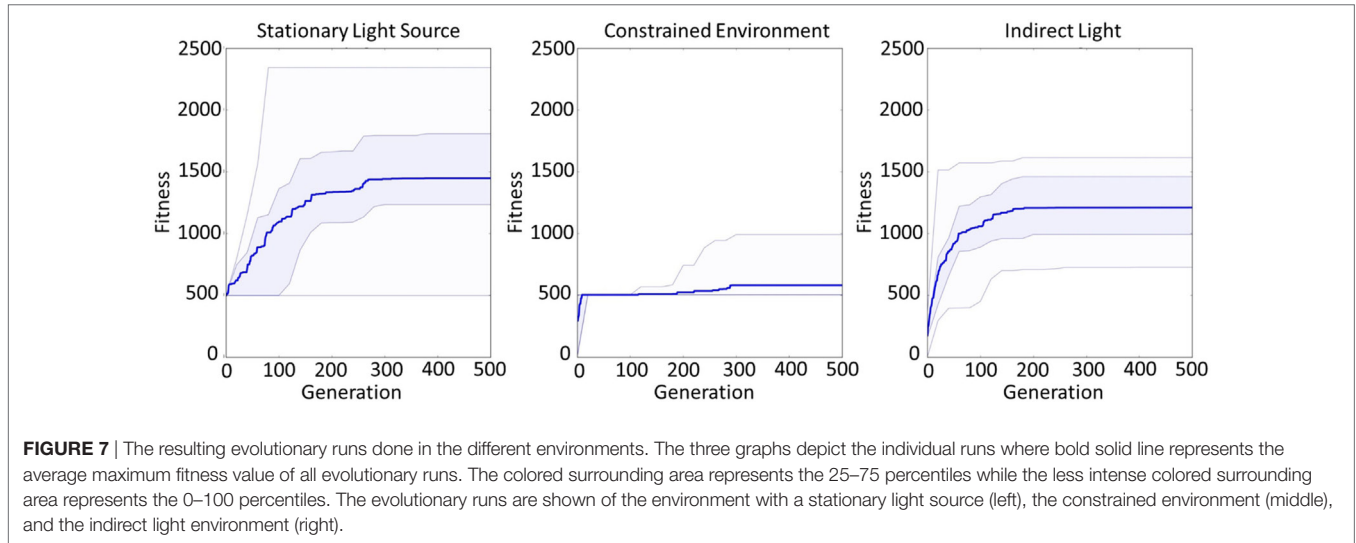
3.1. Phenotypes Evolved in Different Environments

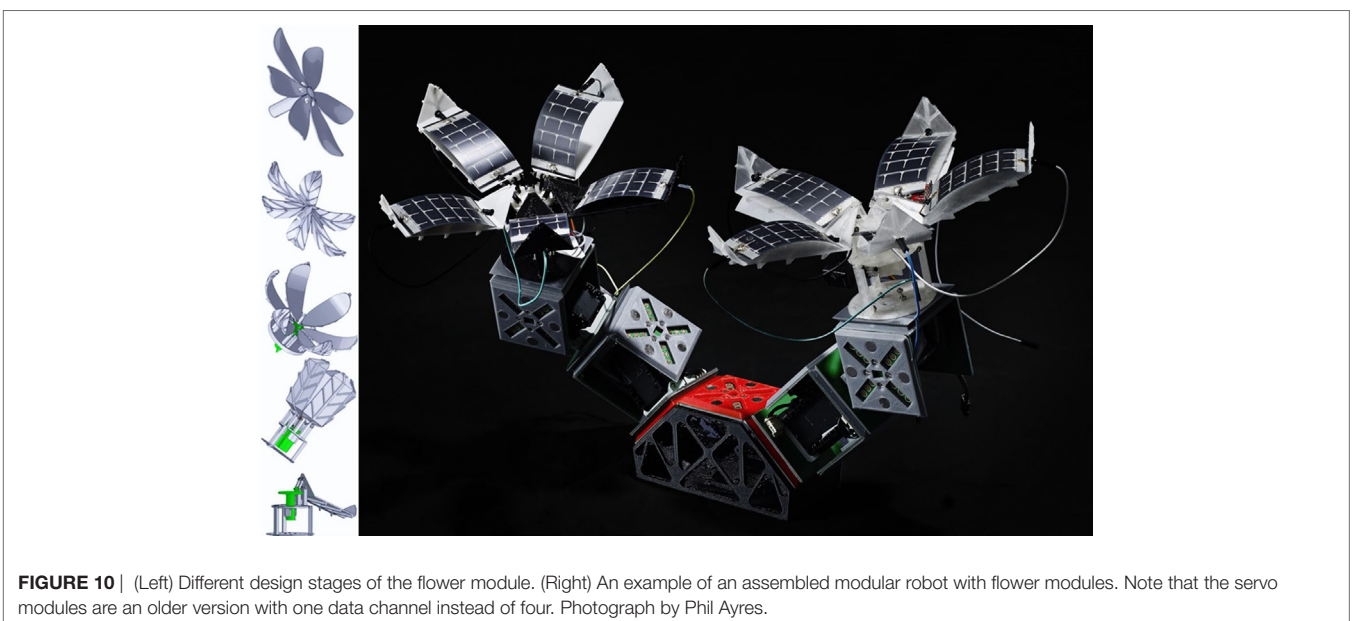
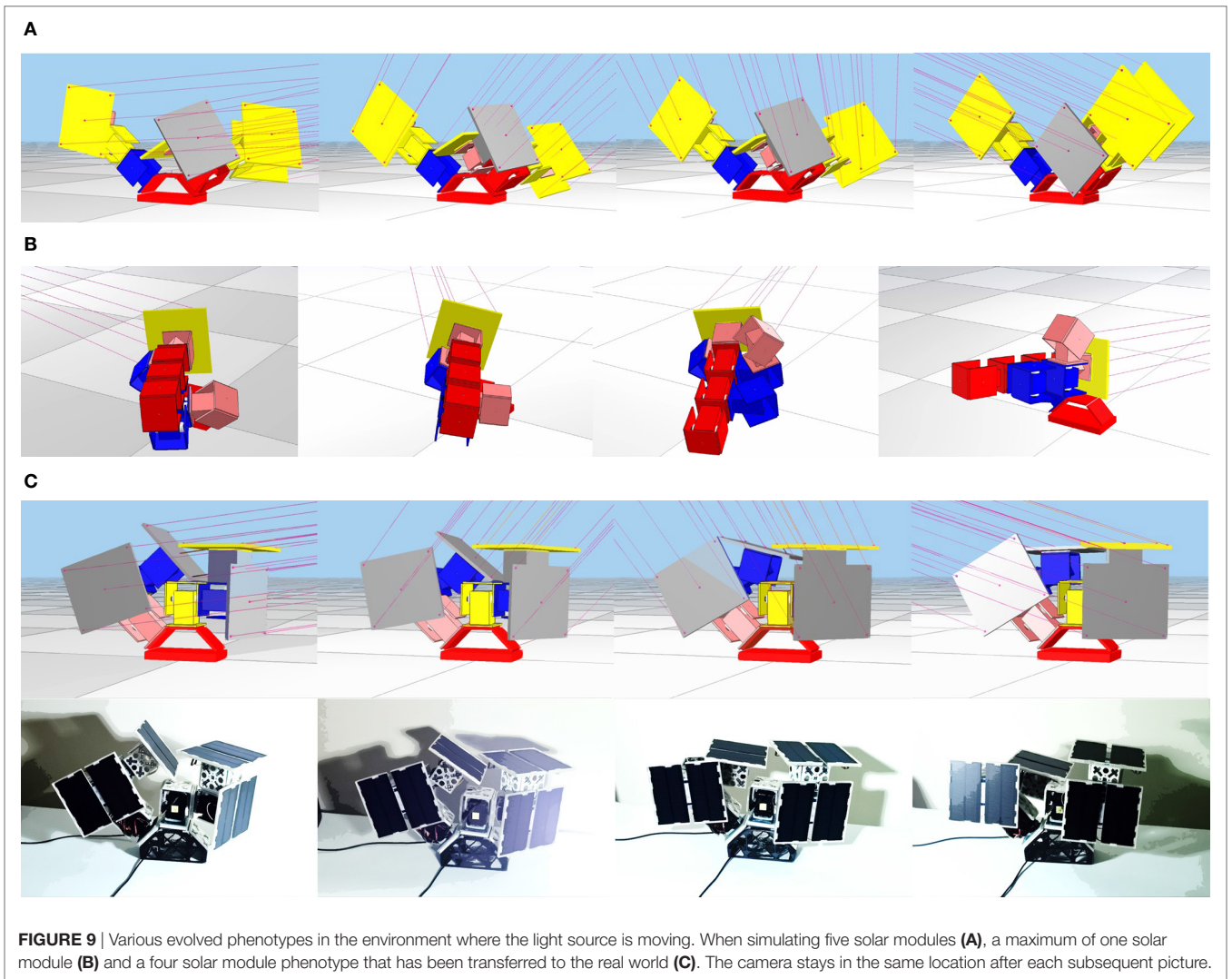
The four types of environments led to major differences in evolved phenotypes (**Figure 6**). The individuals in the environment with the stationary light source evolved simple morphologies



where all solar panels are pointing upwards avoiding collision with one another. The individuals in the constrained environment were more difficult to evolve as can be seen in the graph depicting the evolutionary progressions (Figure 7). In particular, only three individuals in the last generation of the evolutionary run displayed phenotypes that were different from the top right and bottom right individuals seen in Figure 6. It can, therefore, be said that the search space of this environment is a lot more rugged than the search space in the other environments making it harder for the evolutionary algorithm to get out of a local optima. As expected,

the environment where the light source was blocked directly from above, the solar panel modules were located on the outside of the modular robots. Some of these evolved phenotypes were easily implemented in the real world since they did not actuate any servo modules (Figure 8). As can be seen in Figure 9, the eventual population of the modular robots that evolved in the environment with the moving light source was also easy to transfer to the real robot. Eventually, although it is a conceptual model, the modular robot that incorporates the flower modules is depicted in Figure 10.





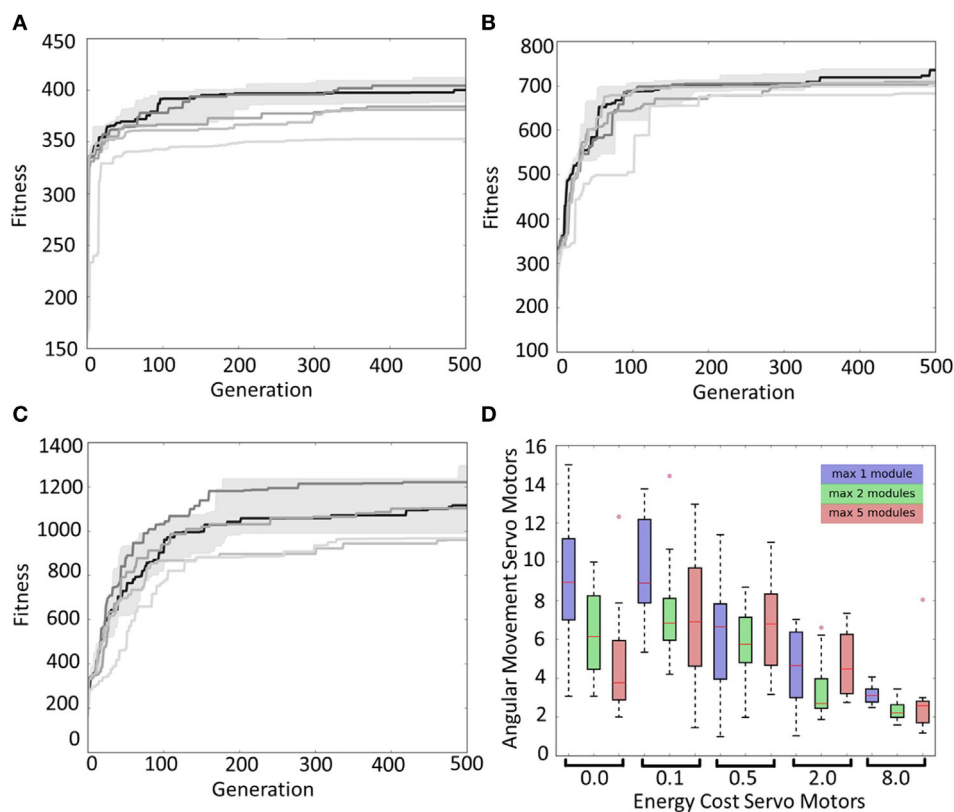


FIGURE 11 | The effects of energy cost on rotations of solar panels when simulating a maximum of 1 solar panel (A), 2 solar panels (B), and 5 solar panels (C). The black line depicts the average of the maximum fitness values of 12 individual runs. The lighter gray lines depict the average of the maximum fitness of 12 individual runs where energy cost was applied to the simulation. The lighter gray lines represent higher energy cost. The box plot (D) depicts the average movement of the solar panels in each simulation with different energy costs. Though a lot of disparity could be seen between runs, a trend can be seen that when energy cost is higher, the resulting phenotypes on average move their solar panels less. This distinction can most clearly be seen when maximum 1 solar panel is being simulated (blue). The different colors of the box plot represent the different amount of maximum solar panel modules that could be generated.

3.2. Movement in Energy Harvesting Modular Robots

As could be seen in the simulation environment with the moving light source, movement can increase the fitness value of evolved robots compared to the evolutionary runs if the cost of moving is low enough (Figure 11). A statistically significant difference can be seen when comparing the average movement angle of all solar panels when comparing maximum and minimum energy cost. The difference in fitness for a high energy cost versus a low energy cost for movement is highest when simulating one module and lowest when simulating five modules. Based on a Mann–Whitney U test, we can see that there is a significant difference in the angular movement of the leaves when a large energy cost is applied when simulating a maximum of 1 (p-value: 0.0000779), 2 (p-value: 0.0000300), and 5 (p-value: 0.0050966) solar panel modules. The difference in angular movement between simulating a maximum a 1 and 5 solar modules without a energy cost was also significant (p-value: 0.0030495). Similarly, the difference of angular movement when simulating a maximum of 1 and 5 solar modules was significant with the highest energy cost (p-value: 0.0070689). When comparing the

maximum of 2 solar modules with the other maximum amount of solar modules, no significant difference could be found. This is due to the fact that the angular movement of the solar panels when simulating a maximum of 2 panels is between simulating 1 and 5 solar panels. These differences indicate that less modules lead to more movement of solar modules. Based on these results, we can say that the amount of solar modules directly influences the amount of movement in the solar panels.

4. DISCUSSION

The aim of this paper was to provide a deeper understanding in how evolution of modular robots could shape robotic entities toward energy autonomy in different environments. The resulting phenotypes of each environment indicate that the simulated individuals adjusted differently to optimize for light absorption. Some of the simulations led to solutions in local optima indicating a rugged search space for the evolutionary algorithm, as was the case of the constrained environment. The other environments typically evolved more varied and more complex phenotypes. This demonstrates that our simulator

is versatile in creating novel robots using a fixed amount of modules in different simulated environments though improvements in the evolutionary algorithm as well as the encoding could be made.

4.1. Light Tracking

When movement was more costly and when more solar panels were available to be implemented in the modular robot, it was expected to see a relative decrease in angular movement of the leaves when compared to runs without implementing cost of movement and simulating less solar panels. Our results indicate that when less solar modules were allowed in the evolutionary runs, more movement could on average be seen in the phenotypes that evolved. The reason that this difference occurs is due to the increased search space of simulating more than one solar module as well as the constraining available positions for solar modules when simulating a limited number of modules. Our results indicate that having more than one solar panel does lead to robots tilting their solar panels toward the light source. However, when only one solar panel is being used, the robot evolved more movement and movement has a clear evolutionary advantage when it tilts the solar module toward the light. Similar to Veenstra et al. (2016), evolution does not necessarily pick up movement of solar panels even when there is no cost for movement attributed to it. A significant difference could, however, be seen when a large cost was implemented for movement in the modular robots. Having more solar modules may be more beneficial than requiring to track a light source. When robots are constructed in the real world, the actual movement costs and solar uptake of the entire robot can be modeled and used as a feedback mechanism to the simulation environment. In this scenario, we could determine if it is better to make small modular robots at different locations only containing a few solar modules, or if it is better to create a single modular robot containing many solar panels.

4.2. Challenges

One major challenge of the presented robotic modular platform is the size and weight of the modules. Since the modules are only connected via magnetized connector sites, the structure will become quite heavy which is especially detrimental at the extremities of the modular robot. Modules connected to the base module would have to cope with more force than the modules at the extremities. Similar to plants, the main stem or tree trunk is usually the strongest and heaviest while its branches are lighter and its leaves even lighter than the branches. Unable to distribute weight accordingly is, however, a common issue in modular robotics and one solution is to remove redundant connection sites. Another solution is to vary the shape, size, and weight of the modules. Such a system might require modules that isolate their mechanical connections and communication. Since every connection face of our modules contains the same amount of magnets, each connection between modules has the same strength. A future implementation could allow for modules at the near the base to be connected with more force and modules at the extremities be connected with less force.

Only a limited number of generations were simulated in the evolutionary runs due to convergence of the evolutionary algorithm. Improvements to the evolutionary algorithm itself would be valuable to cope with this. These improvements could consist of implementing speciation (Stanley and Miikkulainen, 2002), novelty search (Lehman and Stanley, 2011), and/or pareto optimization techniques (Schmidt and Lipson, 2011; Brodbeck et al., 2015). However, doing more evolutionary runs might lead to the discovery of better performing individuals by having an initial population that can be exposed to incremental improvements. The demonstrated robotic platform has potential for integrating interactive evolution (Sims, 1992; Graf and Banzhaf, 1996). A human in the loop could, for example, design the modular robot by simply connecting the modules to form a robot morphology. The simulator could in turn evolve behaviors for that specific robots configuration in the simulation environment similar to Wagay and Bongard (2015) with potential continuous self-modeling (Bongard et al., 2006). Although evolutionary algorithms were used to generate the control and morphology of the modular robots, online adaptation could be a more beneficial strategy, especially for real-world robots. A hybrid approach of initially evolving arbitrary modular robots followed by an online learning strategy could, therefore, lead to more feasible robots. This would be especially valuable if the adaptive control can be localized and its genetic information could be reused similar to the generative encoding.

The generative encoding abstracts the complexity of the genome and has been shown to work well to quickly acquire decent performance. However, as is common in evolutionary algorithms, some runs stagnate in a local optima making it harder for the evolutionary algorithm to discover novel strategies. However, the variety of modular robots that evolved in each environment varied a great deal. It is unknown whether doing more evolutionary runs will increase this variety of decent modular robots. The generative encoding implemented might also be improved through using more implementations that have been used as abstractions of development such as compositional pattern producing networks (Stanley, 2007).

The evolutionary system limited the amount of usable modules to the amount of robotic counterparts in the real world. Simulating more modules could lead to more effective phenotypes when making. However, simulating less modules could potentially lead to good solutions as well when considering that many modules that have been implemented in the evolved robot did not seem to be of any particular advantage (see **Figure 8B**). The implemented modules are also not ideal when optimizing for light harvesting in modular robots. E.g., the servo modules contained a hinge-like joint although it might be better to implement a twisting joint in the environment with the moving light source. Moreover, other modules with less energy demanding actuation, or elastic and rigid structural properties, could increase the performance of the evolutionary algorithm to find more suitable solutions for absorbing light from the environment. One possible new module type could be a soft module similar to Vergara et al. (2017). If a soft module could be used to inflate multiple chambers, directed movement of solar panel modules toward a light source could be implemented with ease.

Such a soft module would be similar to a pulvinus structure seen in many plants. Even the energy gathering system could potentially be combined with modules that gather energy from different sources such as microbial fuel cells. The robotics platform does allow for the effortless integration of new modules in the system—both for the hardware and the software—although explorations of these systems are left for future work.

Four types of environments were used in the evolutionary runs to optimize for light absorption while the fitness function stayed the same across environments. An additional approach could be to evolve a population of individuals initially in one environment and afterward in more advanced environments which can either be done incrementally (Bongard, 2008) or through encapsulation of behavior in specific evolved environments (Lessin et al., 2013). The light source could also be used as a stimulus in mobile modular robots. In this case, the evolutionary algorithm could optimize the robot to locomote toward light as a main stimulus that could potentially give rise to modular Braitenberg vehicles (Braitenberg, 1986). Evolving locomotion would add an additional reality gap that was not present in our sessile modular robots. Additional experiments could also include lattices with many solar panels that greatly increase the uptake of light. As seen in the evolutionary runs, modular robotic structures with many solar panels would not necessarily give any freedom for complex behavior to evolve. Moreover, the robotics platform can be used to evolve any arbitrary task that evaluates compositions of robotic modules. If the same connection faces are used for the modules, any type of module could be implemented. For locomotion, it would be useful to have additional modules that can act as feet or tendons. The robots evolved for different tasks could then be further evolved toward energy autonomy by allowing solar panels to be implemented in the evolved individuals.

4.3. Multi-Robot Systems

Considering food chains in natural ecosystems, organisms higher up the food chain extract the energy that has been harvested by primary energy producers (Reece et al., 2010) simply through consuming and reusing the molecules they produced. Hence, most plants and other photosynthetic organisms are known as primary energy producers. Primary, secondary, tertiary, etc., energy consumers subsequently depend on the acquisition of chemical energy that has been created by their prey. The modular approach allows for a potential implementation of a multi-robot system where a robot specialized in absorbing light could share its energy with other robots, potentially making robot energy autonomy viable in environments that are off the electrical grid. Moreover, humanity's ecological footprint is already unsustainable and in overshoot (Toth and Szigeti, 2016). This calls for an approach toward energy autonomy in robots that will not increase our already existing ecological footprint. Instead of relying on renewable energy sources to be implemented in our energy grid, we can imagine a robotic system where robot entities can specialize in gathering energy from the environment while higher order robots can tap into the power grid of these

robot entities to recharge themselves—possibly tapping into the humanities existing electrical power grid to release their energy. Since light in urban or desert environments is largely wasted since a limited number of plants grow in these locations, robots in these environments can be greatly beneficial if they are able to gather solar energy in say, windows, rooftops, etc.

Since the tasks of our robotic platform is largely undetermined, it is unsure what the eventual robotic entity will look like, how it moves, manipulates objects, and gathers information from the environment. Instead of having robotic entities created for specific functions, we can implement a range of robotic modules that can be optimized for a variety of tasks. In an approach where multiple small modular robots are used to gather energy, an additional modular robot could be used to extract energy from the energy harvesting modular robots simply by connecting a docking module to the solar harvesting modular robot. This docking module could simply be a male connector face that attaches to a female connector face of the energy harvesting robot. This can in turn give rise to an artificial ecosystem whereby some robots are specialized in energy uptake while others can be specialized for different tasks. Such a system could enable symbiotic relationships between modular robots that are optimized to harvest energy and other robots that are able to extract energy from the energy harvesting modular robot. The modular connection mechanisms would allow for the energy to flow directly from one modular robot to another without too much of a hassle. The advantage of energy autonomy in an ecosystem of robots instead of a single robot is that some robots can specialize for very specific tasks not having to worry about spending much energy. This division of labor in multi-robot systems conforms to symbiotic energy exchange that can contribute to the survival of the entire robotic system.

5. CONCLUSION

We have shown how an evolutionary algorithm can evolve light harvesting robots that can easily be transferred to the real world using a modular robotics approach. Furthermore, there is an advantage for evolutionary algorithms to exploit movement when only one solar panel module is simulated but not when more are simulated given the constraints of the simulator. It has been demonstrated that the robotic platform is able to evolve a variety of modular robots based on the various objectives in the environment. This might prove especially advantageous when evolving a multi-robot system that can form a stepping stone toward artificial ecosystem. This artificial ecosystem can consist of multiple modular robotic entities that interact with one another based on energy. The described platform was limited to the available types of modules and the flower module was not used in the evolutionary algorithms. However, a robotic module containing more complex electronics could simplify evolutionary search when as smart morphology and online learning is applied. All of the described experiments and proposed methods can, thus, lead to an automated design of homeostasis in robots that can potentially scale up to artificial ecosystems.

AUTHOR CONTRIBUTIONS

This study was designed, directed, and coordinated by FV, SR, and KS. KS and SR provided conceptual and technical guidance for all aspects of the project. CM helped in designing and constructing the hardware described in the paper. FV designed the simulator as well as the modules and is responsible for performing and analyzing the main experiments. The manuscript was written by FV and commented on by all authors.

REFERENCES

- Afarulrazi, A. B., Utomo, W. M., Liew, K. L., and Zarafi, M. (2011). "Solar tracker robot using microcontroller," in *ICBEIA 2011 – 2011 International Conference on Business, Engineering and Industrial Applications*, Kuala Lumpur, 47–50.
- Atamian, H. S., Creux, N. M., Brown, E. A., Garner, A. G., Blackman, B. K., and Harmer, S. L. (2016). Circadian regulation of sunflower heliotropism, floral orientation, and pollinator visits. *Science* 353, 587–590. doi:10.1126/science.aaf9793
- Balzer, H., Braun, P. W., and Köhler, W. (1998). Cellular automata models for vegetation dynamics. *Ecol. Model.* 107, 113–125. doi:10.1016/S0304-3800(97)00202-0
- Bongard, J. (2008). Behavior chaining: incremental behavior integration for evolutionary robotics. *Artif. Life* 11, 64–71.
- Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. *Science* 314, 1118–1121. doi:10.1126/science.1133687
- Braitenberg, V. (1986). *Vehicles: Experiments in Synthetic Psychology*. Cambridge: MIT Press.
- Brodbeck, L., Hauser, S., and Iida, F. (2015). Morphological evolution of physical robots through model-free phenotype development. *PLoS ONE* 10:e0128444. doi:10.1371/journal.pone.0128444
- Corucci, F., Cheney, N., Lipson, H., Laschi, C., and Bongard, J. C. (2016). "Material properties affect evolution's ability to exploit morphological computation in growing soft-bodied creatures," in *Proceedings of the Artificial Life Conference 2016 (ALIFE XV)*, Cancun.
- Ehleringer, J. R., and Forseth, I. (1980). Solar tracking by plants. *Science* 210, 1094–1098. doi:10.1126/science.210.4474.1094
- Graf, J., and Banzhaf, W. (1996). "Interactive evolution for simulated natural evolution," in *Artificial Evolution, European Conference, AE 95 Brest, France, September 4–6*, Vol. 1063, eds J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers (Berlin, Heidelberg: Springer), 259–272.
- Graham, J. M., and Wilcox, L. W. (2013). *Plant Biology: Pearson International Edition*, 2 Edn. Essex: Pearson Education Limited.
- Greenman, J., Holland, O., Kelly, I., Kendall, K., McFarland, D., and Melhuish, C. (2003). Towards robot autonomy in the natural world: a robot in predator's clothing. *Mechatronics* 13, 195–228. doi:10.1016/S0957-4158(01)00045-9
- Heinrich, M. K., Wahby, M., Soorati, M. D., Hofstadler, D. N., Zahadat, P., Ayres, P., et al. (2016). "Self-organized construction with continuous building material: higher flexibility based on braided structures," in *1st International Workshop on Self-Organising Construction (SOCO), SASO 2016 and ICCAC 2016*, (Ausborg).
- Hogeweg, P. (1988). Cellular automata as a paradigm for ecological modeling. *Appl. Math. Comput.* 27, 81–100. doi:10.1016/0096-3003(88)90100-2
- Ieropoulos, I., Melhuish, C., and Greenman, J. (2003). "Artificial metabolism: towards true energetic autonomy in artificial life," in *Advances in Artificial Life: 7th European Conference, ECAL 2003, Dortmund, Germany, September 14–17, 2003*, eds W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim (Berlin, Heidelberg: Springer), 792–799.
- Kutschera, U., and Briggs, W. R. (2016). Phototropic solar tracking in sunflower plants: an integrative perspective. *Ann. Bot.* 117, 1–8. doi:10.1093/aob/mcv141
- Lehman, J., and Stanley, K. O. (2011). Novelty search and the problem with objectives. *Genet. Program. Theory Pract.* 9, 37–56. doi:10.1007/978-1-4614-1770-5_3
- Lessin, D., Fussell, D., and Miikkulainen, R. (2013). "Open-ended behavioral complexity for evolved virtual creatures," in *Proceeding of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference – GECCO '13*, 335.
- Lindenmayer, A. (1968). Mathematical models for cellular interactions in development. I. Filaments with one-sided inputs. *J. Theor. Biol.* 18, 280–299. doi:10.1016/0022-5193(68)90079-9

ACKNOWLEDGMENTS

Project "flora robotica" has received funding from the European Union's Horizon 2020 research and innovation program under the FET grant agreement, no. 640959. Computation/simulation for the work described in this paper was supported by the DeIC National HPC Centre, University of Southern Denmark, and the Vermont Advanced Computer Core, University of Vermont.

- Lindenmayer, A., and Jürgensen, H. (1992). "Grammars of development: discrete-state models for growth, differentiation, and gene expression in modular organisms," in *Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology*, Chap. 1, eds G. Rozenberg and A. Salomaa (Berlin Heidelberg: Springer), 3–21.
- Moreno, R., Liu, C., Faina, A., Hernandez, H., and Gomez, J. (2017). The EMERGE modular robot, an open platform for quick testing of evolved robot morphologies. *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (New York, NY: ACM), 71–72.
- Noth, A., Engel, M. W., and Siegwart, R. (2006). Flying solo and solar to mars. *IEEE Robot. Autom. Mag.* 13, 44–52. doi:10.1109/MRA.2006.1678138
- Pfeifer, R., and Bongard, J. C. (2006). *How the Body Shapes the Way We Think: A New View of Intelligence*. Cambridge: The MIT Press.
- Philamore, H., Rossiter, J., and Ieropoulos, I. (2015). "An energetically-autonomous robotic tadpole with single membrane stomach and tail," in *Conference on Biomimetic and Biohybrid Systems*, Barcelona, 366–378.
- Prinsloo, G., and Dobson, R. (2015). *Solar Tracking: High Precision Solar Position Algorithms, Programs, Software and Source-code for Computing the Solar Vector, Solar Coordinates & Sun Angles in Microprocessor, PLC, Arduino, PIC and PC-Based Sun Tracking Devices or Dynamic Sun Following Har*. Stellenbosch: Gerro Prinsloo.
- Prusinkiewicz, P., and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. New York, NY: Springer-Verlag, 1–131.
- Prusinkiewicz, P., and Lindenmayer, A. (1997). The algorithmic beauty of plants. *Plant Sci.* 122, 109–110. doi:10.1016/S0168-9452(96)04526-8
- Reece, J. B., Urry, L. A., Cain, M. L., Wasserman, S. A., Minorsky, P. V., and Jackson, R. B. (2010). *Campbell Biology*. San Francisco: Benjamin Cummings.
- Rohmer, E., Singh, S. P. N., and Freese, M. (2013). "V-REP: a versatile and scalable robot simulation framework," in *IEEE International Conference on Intelligent Robots and Systems*, Tokyo, 1321–1326.
- Sadeghi, A., Mondini, A., Del Dottore, E., Mattoli, V., Beccai, L., Taccola, S., et al. (2016). A plant-inspired robot with soft differential bending capabilities. *Bioinspiration & Biomimetics* 12. doi:10.1088/1748-3190/12/1/015001
- Sadeghi, A., Tonazzini, A., Popova, L., and Mazzolai, B. (2014). A novel growing device inspired by plant root soil penetration behaviors. *PLoS ONE* 9:e90139. doi:10.1371/journal.pone.0090139
- Schmidt, M. D., and Lipson, H. (2011). "Age-fitness pareto optimization," in *Genetic Programming Theory and Practice VIII*, Vol. 8, eds R. Riolo, T. McConaghy, and E. Vladislavleva (New York, NY: Springer), 129–146.
- Sims, K. (1992). "Interactive evolution of dynamical systems," in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, Paris, 171–178.
- Sims, K. (1994a). Evolving 3D morphology and behavior by competition. *Artif. Life* 1, 353–372. doi:10.1162/artl.1994.1.4.353
- Sims, K. (1994b). "Evolving virtual creatures," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, New York, 15–22.
- Sinibaldi, E., Argiolas, A., Puleo, G. L., and Mazzolai, B. (2014). Another lesson from plants: the forward osmosis-based actuator. *PLoS ONE* 9:e102461. doi:10.1371/journal.pone.0102461
- Song, K., Yeom, E., and Lee, S. J. (2014). Real-time imaging of pulvinus bending in *Mimosa pudica*. *Sci. Rep.* 4, 6466. doi:10.1038/srep06466
- Stanley, K. O. (2007). Compositional pattern producing networks: a novel abstraction of development. *Genet. Program. Evolvable Mach.* 8, 131–162. doi:10.1007/s10710-007-9028-8
- Stanley, K. O., and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evol. Comput.* 10, 99–127. doi:10.1162/106356002320169811

- Stoy, K., Brandt, D., and Christensen, D. J. (2010). Self-reconfigurable robots: an introduction. *Artif. Life* 18, 237–240. doi:10.1080/09602011.2012.701865
- Syswerda, G. (1991). A study of reproduction in generational and steady state genetic algorithms. *Found. Genet. Algorithm.* 1, 94–101. doi:10.1016/b978-0-08-050684-5.50009-4
- Toth, G., and Szigeti, C. (2016). The historical ecological footprint: from over-population to over-consumption. *Ecol. Indicat.* 60, 283–291. doi:10.1016/j.ecolind.2015.06.040
- Vargas, P. A., Paolo, E. A. D., Harvey, I., and Husbands, P. (2014). “Incremental evolution of an omni-directional biped for rugged terrain,” in *The Horizons of Evolutionary Robotics*, Chap. 11, 237–278.
- Veenstra, F., Faina, A., Risi, S., and Stoy, K. (2017). “Evolution and morphogenesis of simulated modular robots: a comparison between a direct and generative encoding,” in *Applications of Evolutionary Computation: 20th European Conference, EvoApplications 2017*, eds G. Squillero and K. Sim (Amsterdam: Springer International Publishing), 870–885.
- Veenstra, F., Faina, A., Stoy, K., and Risi, S. (2016). “Generating artificial plant morphologies for function and aesthetics through evolving L-systems,” in *Proceedings of the Artificial Life Conference 2016* (Cancun: MIT Press), 692–699.
- Vergara, A., Lau, Y.-S., Mendoza-Garcia, R.-F., and Zagal, J. C. (2017). Soft modular robotic cubes: toward replicating morphogenetic movements of the embryo. *PLoS ONE* 12:e0169179. doi:10.1371/journal.pone.0169179
- Wagy, M. D., and Bongard, J. C. (2015). Combining computational and social effort for collaborative problem solving. *PLoS ONE* 10:e0142524. doi:10.1371/journal.pone.0142524
- Zahadat, P., Hofstadler, D. N., and Schmick, T. (2016). “Vascular morphogenesis controller: a distributed controller for growing artificial structures,” in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*, Augsburg, 273–274.
- Zamuda, A., and Brest, J. (2014). Vectorized procedural models for animated trees reconstruction using differential evolution. *Inf. Sci.* 278, 1–21. doi:10.1016/j.ins.2014.04.037

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2017 Veenstra, Metayer, Risi and Stoy. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.