The effect of selecting for different behavioral traits on the evolved gaits of modular robots

Babak H Kargar¹, Karine Miras¹ and A.E. Eiben¹

¹Vrije Universiteit Amsterdam, Amsterdam, 1105 1081 HV
b.hosseinkhanikargar@vu.nl

Abstract

Moving around in the environment is a fundamental skill for mobile robots. This makes the evolution of an appropriate gait, a pivotal problem in evolutionary robotics. Whereas the majority of the related studies concern robots with predefined modular or legged morphologies and locomotion speed as the optimization objective, here we investigate robots with evolvable morphologies and behavioral traits included in the fitness function. To analyze the effects we consider morphological as well as behavioral features of the evolved robots. To this end, we introduce novel behavioral measures that describe how the robot locomotes and look into the trade-off between them. Our main goal is to gain insights into differences in possible gaits of modular robots and to provide tools to steer evolution towards objectives beyond 'simple' speed.

Introduction

“A gait is a pattern of locomotion characteristic of a limited range of speeds, described by quantities of which one or more change discontinuously at transitions to other gaits” (Alexander, 2003). This paper is concerned with the possible gaits of modular robots with variable morphologies. To be specific, we investigate an evolutionary robot system where the robots' morphology (body) and controller (brain) are evolved simultaneously. Previous work with this system has confirmed that evolution can discover robots that can move at high speeds. However, inspecting the actual locomotion patterns revealed that many of them moved in weird ways that would be hard to realize in the real world. This motivated the question behind this paper: “Should a gait be described and optimized simply in terms of speed, or should other aspects be considered as well?“.

To date, this question has not received much attention. For instance, (Miras et al., 2018a) investigated the effects of different selection preferences on the morphology and behavior of the robots. Still, the part of their work that focuses on the behavior was minimal. Their work was minimal on discussing the gait and just analyzed the robot’s movement behaviors only by observation. (Miras et al., 2020) also design and develop a set of morphological and behavioral traits. Still, their work was more on the effects of different environments on robots.

In this study, several new behavioral measures are designed. Different fitness definitions based on such measures are tested to see how selection pressure towards various traits can help obtain robots with better gaits. In particular, we compare three different preferences for speed, floor contacts, and robot’s head balance and analyze the trade-off among them. The two main contributions of the paper are 1) the introduction of new behavioral measures that can be used to describe locomotion and gait of the robots and 2) the initiation of a discussion about what a good gait is for modular robots. To this end, we seek answers to the following research questions:

• How does adding behavioral traits in selection preferences change evolved locomotion patterns?

• How do these selection preferences affect the evolved morphologies of the robots?

Related work

The Evolutionary Robotics literature is very focused on the task of locomotion. In some works, researchers tried to co-evolve body and brain together to get more optimized outputs by the robots. On the other side, some others tried to find solutions to have more generic robots with diverse objectives. Still, little attention has been given to describing patterns of locomotion. (Sims, 1994) used a directed-graph-based generative encoding to evolve the morphologies and brains of robots. Lipson et al. in (Lipson et al., 2016) have shown the difficulty of the co-evolution of the body and brain simultaneously. However, (Hornby and Pollack, 2001) in another study tried to use L-System for co-evolution of body-brain and got promising results. Additionally, (Wampler and Popovic, 2009) studied methods for developing efficient morphology and motion behaviors for the robots. (Samuelsen and Glette, 2014) investigated measures that can diversify the morphology of the robots during their evolution. Another study (D’Angelo et al., 2013) investigated the influence of the robots’ size on the quality of the gait for the robots. (Auerbach and Bongard, 2014) in
Interesting research verified the impact of the environment on the morphology complexity. Exciting work has been done in (Doncieux and Mouret, 2010) on exploring generic behavioral similarity measures that rely on sensory-motor values. However, their work was also limited to the number of traits, and they did not evolve the morphologies of the robots. (Miras et al., 2018a) investigated the effect of selection preferences on morphologies and behavioral traits. Their work was mainly focused on morphological characteristics and limited behavioral characteristics. In (De Carlo et al., 2020), it was investigated how different tasks in the same physical environment influence the evolutionary outcome. They compared morphologies and behaviors under two tasks of undirected locomotion and rotation in a flat terrain environment and analyzed the morphological and behavioral traits of the final population.

Quality diversity algorithms are also a recent family of optimization algorithms that search for a large set of diverse but high-performing solutions. In (Mouret and Maguire, 2020) an extension of MAP-Elite algorithm, called Multi-task MAP-Elites that solves multiple tasks when the fitness function depends on the task was used for 6-legged robots and showed exciting results in comparison to CMA-ES algorithms.

**Methodology**

In this investigation, the evolutionary algorithm was used. A framework called Revolve (Hupkes et al., 2018) is utilized for this study, using “Gazebo” as the physics simulator. The following section will explain more about this framework.

![Figure 1: Robot modules: Core-component with controller board (C); Structural brick (B); Active hinges with servo motor joints in the vertical (A1) and horizontal (A2) axes; Touch sensor (T). C and B have attachment slots on their four lateral faces, and A1 and A2 have slots on their two opposite lateral faces; T has a single slot that can be attached to any slot of C or B. The sequence of letters (T or n) in C and B indicates if there is a sensor on the laterals left, front, right, and back (for C only) in this order.](image)

**Robot Framework**

**Morphology:** The phenotype of the robot’s morphology consists of modules based on (Auerbach et al., 2014). These modules are connected in a 2-dimensional pattern, meaning the modules cannot connect to the top or bottom of each other and connect to the sides. Each module type has a specific symbol as a representation of it in the genotype.

**Controller:** A hybrid neural network called recurrent CPG neural perceptron (Miras et al., 2018a) constitutes the robot controller. Each joint in the morphology has an equivalent oscillator neuron activated through a sine wave with Phase offset, Amplitude, and Period. Oscillators are not connected, and each of them may have a direct recurrent connection or may not. The CPG network generates a constant movement pattern for the robot.

**Representation and operators:** Using Generative encoding for co-evolving a robot’s body and brain has shown successful results in the literature. In this study, we use Lindenmayer-system (Lindenmayer, 1968), a generative encoding and a grammatical parallel rewriting system, and collectively encoding elements relative to both body and brain. In particular, we use an L-System implementation proposed by (Miras et al., 2020). An L-System grammar is defined as $G = (V, w, P)$, where

- $V$, the alphabet, is a set of symbols containing replaceable and non-replaceable symbols.
- $w$, the axiom, is a symbol from which the system starts.
- $P$ is a set of production-rules for the replaceable symbols.

Here we present an example of the iterative process of rewriting in an L-System. Each replaceable symbol changed with the symbols of its production rule for several iterations. Given $w = C$, $V = \{C, B, T\}$ and $P = \{C : \{C, B\}, B : \{T\}, T : \{C, T\}\}$, the rewriting proceeds as:

- **Iteration 0:** C
- **Iteration 1:** C B
- **Iteration 2:** C B T
- **Iteration 3:** C B T C T

A distinct grammar that uses the same alphabet defines the genotypes. The alphabet is organized by different types of morphological modules and commands that connect them and commands that make the structure of controllers. For creating a robot, there are two steps. First, in the early development stage, the axiom of the grammar is changed to a more complicated string of symbols, based on the production rules of the grammar. The number of iterations for this stage is limited to 3. The crossovers are performed by taking complete production rules randomly (uniform)
Figure 2: Process of decoding an early-developed phenotype into a late-developed phenotype with morphology and controller. These modules that you see in this figure, described in detail in figure 1.

from the parents. Finally, individuals undergo mutation by adding/deleting/swapping one random (uniform) symbol from a random production rule/position. All symbols have the same chance of being removed or swapped. As for the addition of symbols, all categories have an equal chance of being chosen to provide a symbol. Every symbol of the category also has an equal chance of being selected. An exception is always made to C to ensure that a robot has one core component. This way, C is added as the first symbol of the C production rule and can not be added to any other production rules, neither removed nor moved from the production rule of C (Miras et al., 2018a).

In the second step (late development), this constructed string is decoded to a phenotype. This process of converting to phenotype shows in Fig. 2. The early-development process was removed due to its extensiveness, but it follows as the didactic instance presented before. Two positional references are always maintained in the phenotype during the construction of the late-developed phenotype, one for the morphology (pointing to the current module) and one for the controller (pointing to the current sensor and the current oscillator). The application of the commands to the phenotype happens in the current module, in the morphology. In contrast, it happens in (or between) the current sensor and the current oscillator for the controller. More details about the representation can be found in (Miras et al., 2018a) and (Miras et al., 2018b).

Because there is a possibility that only the rules of one single parent expressed in the final phenotype, and also as it is not unusual that one mutation happens for non-expressed genes, both crossover and mutation probabilities were set high, to 80%, which means there is 80% of chance for each individual to select for mutation or crossover, aiming to minimize this effect.

The maximum number of modules allowed in morphology was set to 25 parts. In the phase of decoding the genotype to phenotype, all upcoming modules that are more than 25 were ignored. Moreover, morphologies without at least one joint or intersecting parts were considered invalid and get zero to any behavioral descriptors.

**Behavioral Descriptors**

In this section, the behavioral traits used in the experiments are described. While we introduce several novel traits, we also found it relevant to use the Displacement velocity hill and Balance have been introduced in (Miras et al., 2020). It must be noted that all such traits are calculated based on the head-center. The head-center is defined as the center of mass (CoM) of the head-link in form of a union of the head and a possible group of non-actuated modules connected directly to it.

**Displacement velocity hill (Speed).** Describes the speed \( (cm/s) \) of the robot along the y axis as defined in Equation [1]:

\[
\text{Displacement velocity hill} (\text{Speed}) = \frac{e_y - b_y}{t}
\]

where \( b_y \) is y-coordinate of robot’s head at the beginning and \( e_y \) is y-coordinate of the robot’s head at the end of simulation and \( t \) is the duration of the simulation.

**Balance.** Balance is the rotation of the robot’s head in the x-y dimension. The rotation of an object in the x-y dimension is defined by pitch, yaw, and roll. In this measure, pitch, yaw, and roll considered as a degree between 0 and 180. perfect balance is defined with zero value for pitch and roll so that the higher the balance, the less rotated the robot’s head is. Balance is defined by Equation (2):

\[
B = 1 - \frac{r + p}{t \times 180 \times 2}
\]

where \( r \) is the accumulated sum of the roll over time and \( p \) is the accumulated sum of the pitch over time, and \( t \) is the duration of the simulation.

The next measures are the measures introduced in this paper.

**Contacts.** Describe the mean number of collisions between the robot and the surface for each part of the robot. Contact is defined by Equation (3):

\[
C = \frac{\text{Collisions}}{t}
\]

where \( C \) is the number of collisions between robot and surface, and \( b \) is the number of robot’s body parts, and \( t \) is the simulation duration.
Steadiness (Steady Movement Deviation). Imagine a straight line between the start position and the end position of the robot’s head during the evaluation. Based on the number of steps that the robot utilized to travel the trajectory, divide this line into n sectors with equal size. Each separation point in this imaginary line is supposed to be the robot’s ideal position in each step. The actual position of the robot will be the position in time step n and could be different with a deviation from the ideal position. The cumulative sum of the distances between this ideal position and the robot’s actual position in each step is what this measure describes. The unit of this measure is centimeters, and how much this value is more close to zero, movement is more steady as well. It means less acceleration toward different directions. Steadiness is defined by Equation (4):

$$SMD = \sum |i_t - a_t|$$  \hspace{1cm} (4)

where \(i\) is the ideal position of the robot and \(a\) is the actual position of the robot, and \(t\) is the duration of the simulation.

Path Angular Error. We use the angular difference of the robot displacement during the evaluation time and Y-axis (north) to define this measure. The values in this measure are always positive, and the unit of this measure is degree. This measure looks like the orientation error in the (Cully and Mouret 2016), but we calculate this value concerning Y-axis because of using directed locomotion in our case. Due to the use of directional speed in fitness, this measure shows the angular error of the robots’ locomotion. This measure is defined by Equation. (5):

$$PAE = \theta$$  \hspace{1cm} (5)

where \(\theta\) is the angle between the y-axis and the straight line between the start position and end position of the robot during the evaluation time.

Effective Movement. This measure has no unit. It is calculated by dividing the displacement of the robot by the traveled distance. It describes how efficient the locomotion is and is defined by Equation. (6):

$$EM = \frac{D}{T}$$  \hspace{1cm} (6)

where \(D\) is the displacement of the robot during the simulation and \(T\) is the traveled distance. This measure range is between 0 and 1, and the higher this value the more efficient the locomotion is. The most efficient route is the one with the least detour.

Head Height Stability. This measure is calculated by the cumulative sum of the robot’s head moves in the z-axis (up and down) in each step. It sums the absolute values of these differences between each step, and its value shows the stability of the head of the robot. The unit of this measure is centimeter. HHS defined in Equation (7).

$$HHS = \sum |p_z(t+1) - p_z(t)|$$  \hspace{1cm} (7)

where \(p_z\) is the position of the robot’s head in the z-axis and \(t\) is the time step of the robot.

Morphological Descriptors

Hinge count. Describes the number of joint modules in the body of the robots.

Number of bricks. Describes the number of brick modules in the body of the robots.

Experimental Setup

Designing a fitness function that captures every aspect of the desired robot behavior is challenging because it is often unclear how the interaction of different traits may result in this desired behavior. An appropriate robot gait does not depend only on the speed of this robot because other qualitative properties may be desired when locomoting. To this aim, we added two other behavioral traits to the fitness function to see how they affect robots’ gait stability. We carried out four different experiments evolving robots in an environment with flat ground. In each experiment, a different fitness function was utilized Equations (8),(9),(10), and (11). All the fitness functions are based on the behavioral traits of the robots, which will be explained in continuation. The first one is just using the directed speed of the robot toward the north. In the second one, we added the mean number of contacts with the surface for each part of the robot to fitness. In the third and fourth experiments, the balance of the robot, mean number of contacts, and speed with two different weights for each of the traits are used.

$$F1 = s$$  \hspace{1cm} (8)

$$F2 = 0.5 \times s + 0.5 \times (\frac{1}{c})$$  \hspace{1cm} (9)
In Equations (8), (9), (10), and (11) $s$ represents speed, $c$ is contacts and $b$ is the robot’s balance.

All the experiments have $\mu = 100$, $\lambda = 50$, and the number of generations was 100. In all experiments, offspring were produced by selecting 50 pairs of parents through binary tournaments (with replacement) and creating one child per pair by crossover and mutation. From the resulting set of $\mu$ parents plus $\lambda$ offspring, 100 individuals are selected for the next generation, using binary tournaments. For all fitness functions, each robot was evaluated in 50 seconds. We repeated ten times each of the experiments and the evolutionary process stopped after 100 generations.

**Results and Discussion**

In this section, we discuss the experimental results from the runs. For this aim, we present line plots and box plots that show each objective or trait changes during the evolutionary generations. Moreover, we depict the paths of the ten best robots of each experiment to show the behavior of the robots in the x-y dimension.

It is essential to mention that around one percent of the robots in the simulation, due to a technical bug related to friction in the simulation, have been gained high speed. They were detected as outliers and removed from the results.

Fig. 3. illustrates how the objectives that we use in our fitness function change during generations. As shown in Fig. 3.b, the robots, even before adding Contacts in the fitness function, tend to reduce contact with the surface. It also can be seen in Fig. 3.c that balance has an inverse relation with speed. This means an increased weight on Speed in the fitness function causes a decrease in the robots’ balance.

Fig. 4. shows the trade-off between the new behavioral traits and weights for Speed, Contacts, and Balance. It can be seen in Fig. 3.a and 4.a; there is a correlation between speed weight in fitness and SMD. We get two attainments from this measure. First, selecting robots with more speed causes dynamic movement with different accelerations through the path. It also tells us how much the robots go out of the way during the walk. In this case, we can say that by increasing the importance of fitness speed, the robots have less steady speed during their move. In Fig. 4.b, it can be inferred that there is a direct relationship between the speed and the PAE. More weight for speed in evolution selecting preferences can help to have less PAE, which means minor deviation through the whole path.

Because the EM is calculated by dividing the displacement of the robots by the traveled distance, so, the lower the value means the robots more goes out of the path during their move. We also found out that by selecting the robots to move faster, we get robots that move more effectively.

In Fig. 4.d, the plot shows the mean Head Height Balance of the robots in generations. As it has shown, there is a direct relation between HHS and the balance of the robot, and how much the weight of the balance increased in fitness, the HHS value decreased. It can also be found that with trying to reduce the robot’s contact with the surface, the robot’s head movement increased.

In Fig. 7, the paths of the ten best robots of each experiment showed. As it indicates in the plots, with the decrease
of the speed weight in the fitness function, displacement of the robots reduced. It also can be seen the robots that were generated in the experiments with more consideration of the balance have more stable movement related to the y-axis. Still, as expected, their speed also reduced.

Another issue that can be seen in the path plots is that reducing the contact with the surface has a significant adverse effect on the path of the robots. As shown in Fig. 7.b, which is related to experiments with more weight for Contacts in the fitness function, the shape of the robot’s path is irregular, and it can be considered as the cost of reducing Contacts.

We also had an overview of the robots’ morphological traits to see how they change relative to the fitness function changes. As illustrated in Fig. 5 and Fig. 6, it is clear that there is a reverse relation between the number of bricks and the number of joins in the robots. An interesting observation in these results is that the robots in experiments that were rewarded for having more speed consist of more hinges and fewer bricks. When contacts and balance were added to the fitness function, the number of bricks increased, and a decrement in the number of hinges appeared. Increasing the number of bricks in experiments with the contact in fitness can be due to the bigger size of the bricks, and they help other parts of the robot not touch the surface.

Figure 4: Upper plots show the progression of the median of the population over all ten evolutionary runs in (a) SMD, (b) PAE, (c) EM, and (d) HHS for different fitness functions can be seen; In the lower ones, boxplots of the population over all ten evolutionary runs in (e) SMD, (f) PAE, (g) EM, and (h) HHS for different fitness functions can be found; It is important to remember that the value that shows more stability for the robot for SMD is zero, for PAE is zero, for EM is one, and for HHS is also zero. For the box plots, Wilcoxon statistical significance test has been done between each two neighbor plots. As shown, NS in the plots means there no significant difference between the two compared experiments, and the numbers show the p-value of the test.
A video that shows some of the best evolved robots for each experiment can be found in the link [1].

Figure 5: Upper plots show the progression of the median of the population over all runs for (a) Number of hinges, and (b) Number of bricks; lower plots, the progression of the median of the population over all runs for (a) Number of hinges, and (b) Number of bricks; in lower plots, box plots of the population over all runs for (a) Number of hinges, and (b) Number of bricks. For the box plots, Wilcoxon statistical significance test has been done between each two neighbor plots. As shown, there is a significant difference between the number of hinges and bricks in the experiment with more weight for speed.

Conclusion

This paper studied the evolution of robot locomotion, providing two main contributions: 1) designing a set of behavioral traits that describe robot gait, and 2) demonstrating how to use these trait descriptors to improve gait quality in different ways. Our results showed that diverse locomotion patterns result from including extra components into the fitness function beyond simply using speed as quality criteria. We observed that various combinations of components for the selection pressure resulted in distinct gaits. Importantly, these distinct gaits presented different benefits, and naturally, a trade-off. For example, by increasing the pressure for a balanced gait, robots became more steady and slower. Notably, when speed is the only component of the selection pressure, the evolutionary search often exploits unsteady gaits to obtain fast robots, disregarding other properties that are also desirable when a robot is locomoting. Although finding equilibrium among speed and other desirable locomotion traits is challenging and problem-dependent, our descriptors have proven helpful for this endeavor.

For future work, we propose designing and studying behavioral traits that describe the robot trajectory and the patterns of movement of their limbs. Furthermore, it may be interesting to experiment with a multi-objective search instead of using rewards consolidated into a single fitness function.

References


Figure 6: Ten best robot's morphology of each experiment

Figure 7: Travelled paths of the ten best robots in x-y dimensions for each experiment during evaluation; (a) is related to F1, (b) is related to F2, (c) is related to F3, and (d) is related to F4


