# Evolving Multiple Gaits for a Robot using Random Boolean Networks

Eric Clark, Christopher Pierce, Josh Pothen , Advisors: Josh Bongard, Christopher M. Danforth, & Peter Sheridan Dodds

Computational Story Lab, Vermont Complex Systems Center, Morphology Evolution & Cognition Lab,
Department of Mathematics & Statistics, & Vermont Advanced Computing Core

## Abstract

The biological brain is a complex, modular structure designed to handle a range of inputs with minimal neuronal hardware. To promote this modularity in simulation, we propose the use of critical random Boolean networks (RBNs) to represent multiple gait patterns in a single data structure for a robot. We used a two-part genetic algorithm to evolve 8-node RBNs, each containing multiple cyclical attractors, in order to show that pre-evolving RBNs for maximal variability greatly improves the evolutionary fitness of the simulated robot. Our results indicate that it is feasible to represent multiple, highly-fit gaits with the cyclic attractors of a single network.

## RBN Generation

A Boolean network is a connected system of nodes, and a RBN is a Boolean Network in which the nodes are connected randomly. Each node in the network can represent two values, {0, 1} which determines whether the node is 'on' or 'off' respectively. The values of the nodes in the Boolean network are determined by each node's interaction with the other nodes in the network.

Since the value of each node in the network at each given timestep is either 0 or 1, and finding the value of any node at a given time step is completely determined by other nodes in the network, every initial condition will eventually repeat after a certain number of iterations. Since these values are fixed, this results in a loop, defined as an attractor of the network. Each network may have several different attractors that each possible initial condition can fall into. These attractors will be the driving force behind the motion of the robot.

Since these attractors will be eventually translated into robotic motion, the goal of evolving these RBNs is to create networks with diverse sets of attractors. Attractor sets that have little variability will be useless in translation to robotic motion, so networks are selected based on high variability. Variation in both the vertical and horizontal directions in the states of attractors is optimal for nontrivial motion, both of these variations are used to select for networks with interesting attractor sets.

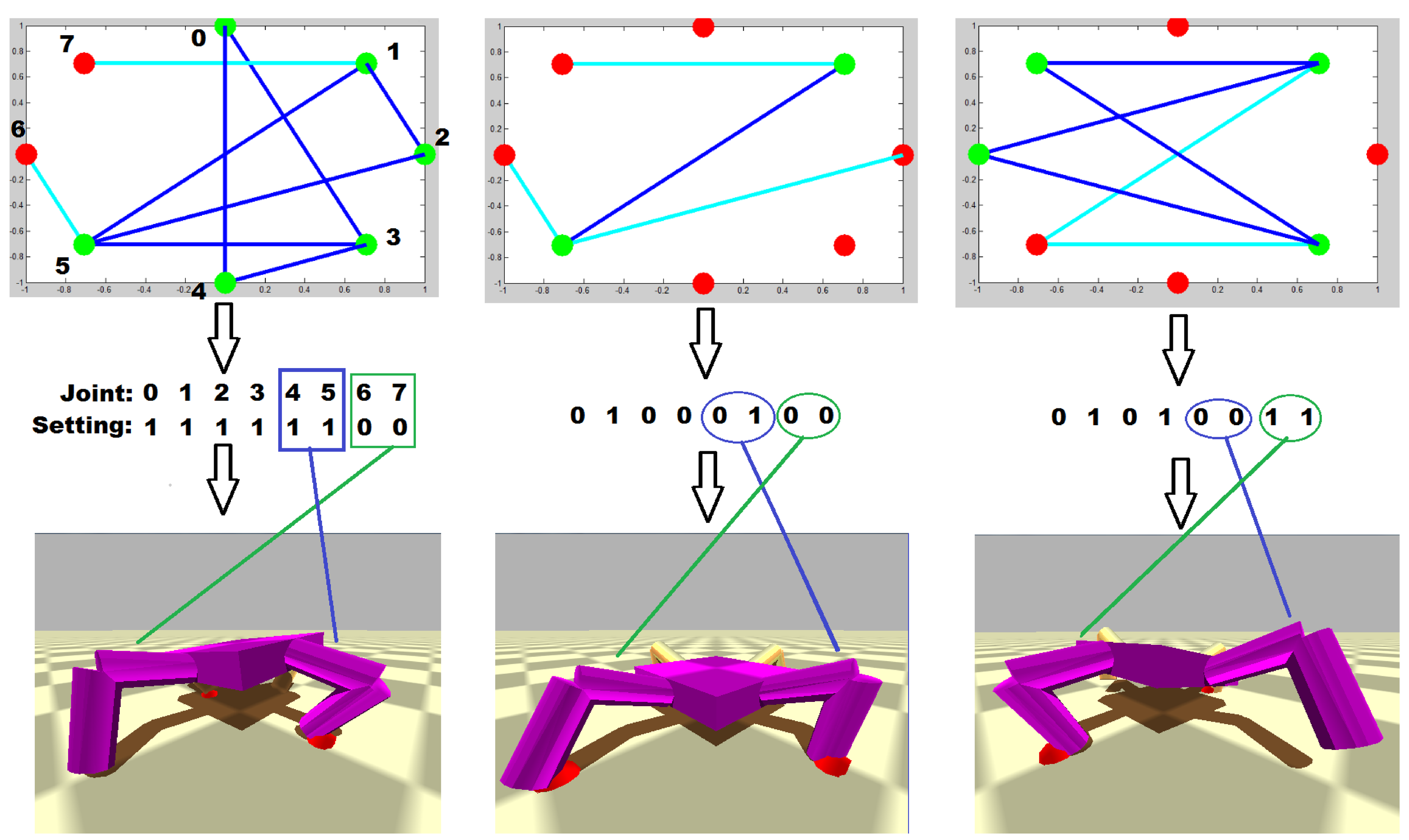
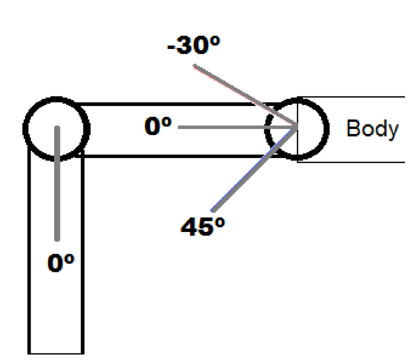



Figure : The top left portion of the figure shows an example state for a 2-regulator RBN. The value at each node is shown as Red or Green (0 or 1 respectively). The top right shows an example of a complete attractor with the associated encoding from the graph. Finally, the bottom shows how this attractor affects the joint angles of two legs. In each case, the last four bits determine the four joint angles shown.

The driving force behind the controls of this robot is the underlying Boolean network. To translate from the RBN output to robotic movement, each joint is assigned a single node in the network to control its motor angle. The default angle for each joint is zero degrees, resulting in a ninety-degree angle between the two sections of each limb. From there, a negative angle corresponds to joint extension while positive angles result in joint flexion.




## Simulated Quadruped

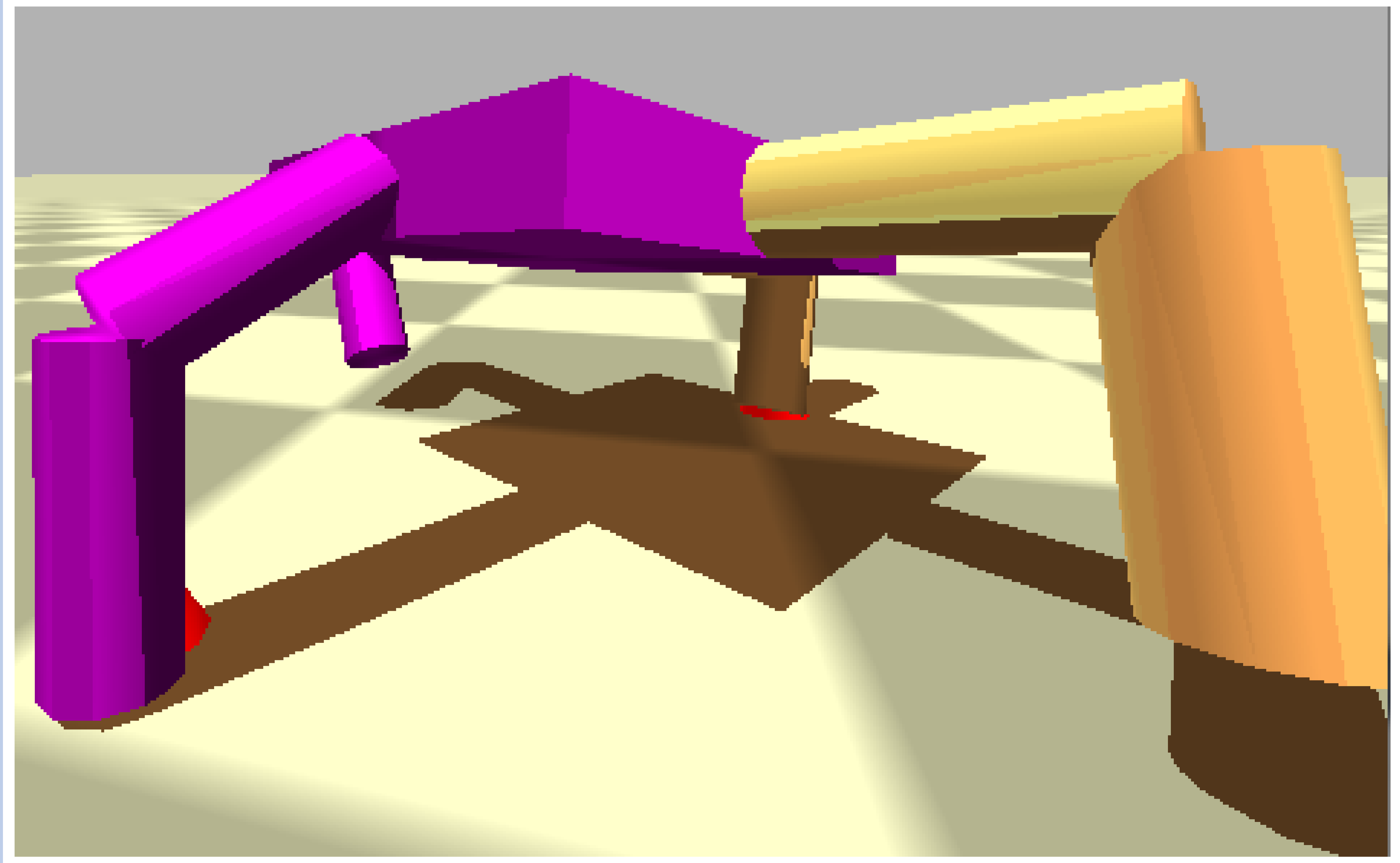


Figure : To illustrate and test this translation scheme from RBN to robot, the open-source Bullet Physics Simulator was employed. The morphology must be established first. We used a radially symmetric quadruped with a rectangular prism for its body. It is intuitively difficult to assign biomimetic gait patterns to this robot due to its obscure shape. Thus without any information on how gaits should look, the final results should not be influenced by experiential biases.

## Evolutionary Algorithm

The attractor sets of each RBN are the driving force for simulating robotic gaits. Sustained motion is a repetitive process in nature. Thus the attractor sets of a network will translate to repetitive robotic motion that can result in a robotic gait. The goal of evolution is to evolve these attractor sets into different types of robotic gaits, and create a diverse network that can perform different tasks.

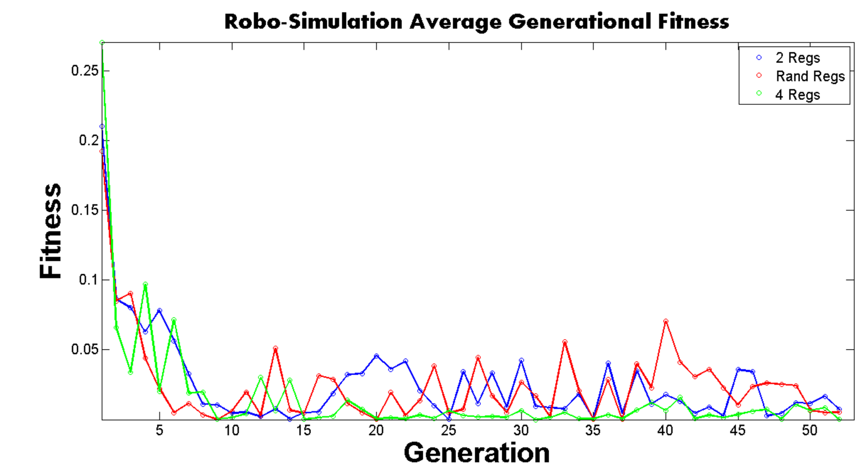
We use a two-part genetic algorithm (GA) to evolve our RBNs. The first step is to evolve a diverse set of critical networks that will be candidates for the initial population of the second GA. The final population from this first GA is then sent into the Bullet Physics simulator to test each attractor set on the simulated robot. The best individuals from the simulator are then taken as the initial population for the second GA. The second GA takes the truncated population of networks from the first GA and evolves based on the fitness obtained in the robot simulator. This is the exploitation phase of the evolutionary process. The goal of the second GA is to select based on the performance of the simulated robot in order to evolve a network that has many different attractors, each of which corresponds to a different type of robotic gait.

| RBN | Robot |
|---|---|
| F = $\frac{V}{\sqrt{N}}$ | F = $(\sqrt{N} * D * V)^{-1}$ |
| N = Number of Attractors | N = Number of Attractors |
| V = Variability | D = Normalized Distance |
| | V = Variability of Motion |

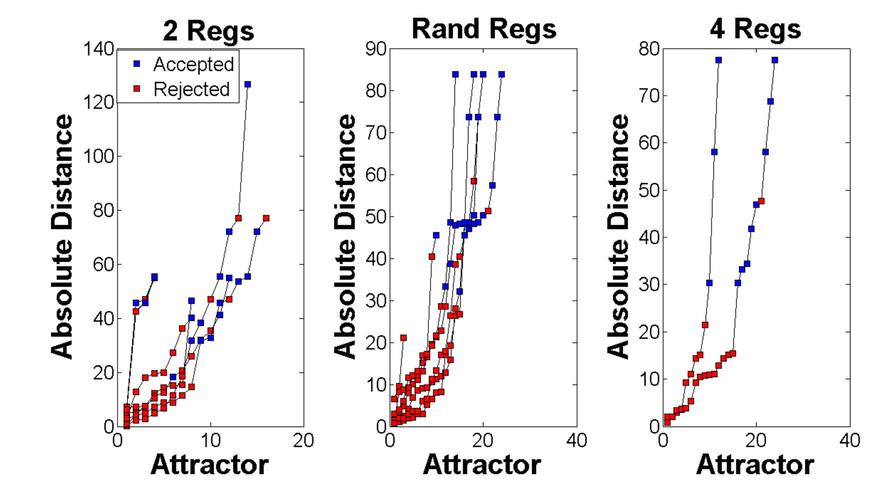
The pre-evolution of the network will (theoretically) drastically reduce the computation time of the costly process of evolving a robot in simulation.

## Results

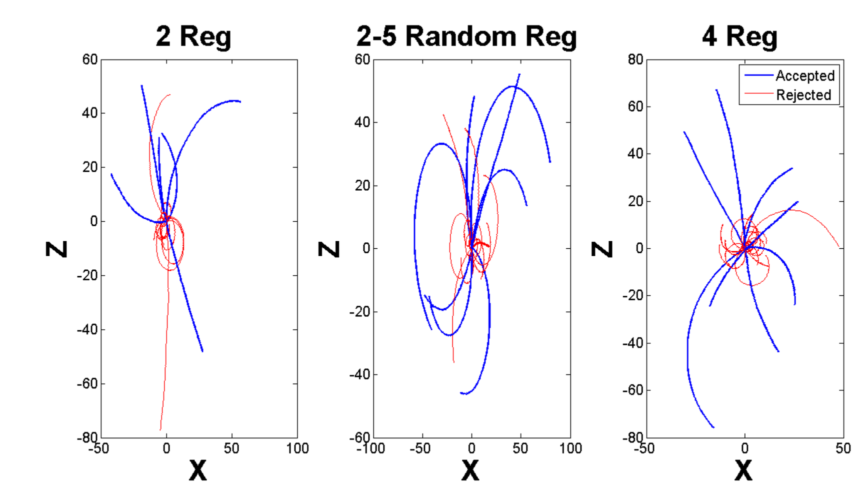
Here, we track the average simulated fitness of the population per generation of the second GA for 2, 4 and a random number from 2 to 5 regulators per node. Individuals in the population had a rapid decrease in fitness for the first few generations and then oscillated between fitness values for the rest of the simulation.




Next we show the absolute distance traveled per attractor per surviving network of the final GA population. Although the population consisted of 50 individuals, replication leads to numerous duplicates which are disregarded for this visualization. Each attractor is colored based on whether it met the fitness threshold of 0.075 after the simulation.




The following shows the trajectories of the attractors of the best final network given 2,4 or a random number of 2 to 5 regulators per node. All simulations begin at the (x,z) origin and move outwards. Trajectories are colored based on meeting the acceptable fitness threshold of 0.075. Green trajectories meet this fitness threshold, while red trajectories do not. Here there is an apparent wide range of types of trajectories due to the different types of acceptable gaits.




## Discussion

In this project 3 distinct types of RBNs were utilized. Two experiments were conducted with static regulators of length 2 and 4, while the third type of RBN used a set of randomly generated regulators between length 2 and 5. Since the first GA driver is designed to evolve toward criticality, the type of RBN is somewhat arbitrary. From these preliminary results, there is no evidence that any one choice of these types regulators is optimal. Results show that each choice will result in networks with diverse attractor sets that can perform different tasks. Future work can continue running these simulations in order to determine if one choice of regulators will significantly outperform the others.

Nevertheless, the final individuals from each experiment exhibit attractors that translate to different types of robotic gaits. These gaits are differentiated due to the clear changes in trajectory along with different types of stability. Each final individual contains a host of different attractors, the best of which correspond to different forms of nontrivial motion. This shows the significance of evolving RBNs to perform repetitive tasks on simulated robots.