Evolving Vestibular Bipedal Locomotion with Spring-Mass Tetrahedra

Jeffrey Ventrella

jeffrey@ventrella.com

Abstract

This paper describes an evolutionary simulation for goal-directed bipedal locomotion using a genetic algorithm. It includes tilt-sensors that adjust dynamic springs and point masses forming structures made of flexible tetrahedral modules. Chaotic dynamics from a central pattern generator are tamed and exploited by evolution. A mathematical representation of the vestibular sense of tilt is associated with arbitrary springs, enabling bipedalism. This sense is embodied and evolves with motor control.

Introduction

The three semicircular canals of the inner ear can detect head rotations in three roughly-orthogonal axes. The otolith organs detect linear accelerations. These comprise the vestibular system in vertebrates, and it is essential for balance and equilibrium. There are several systems involved in bipedal locomotion besides the vestibular sense, including the senses of vision and proprioception, and they are tightly-coordinated. Although the internal machinery is complex and intricate, bipedalism comes naturally to humans. For healthy adults, it functions so well we rarely need to give it much thought.

What is the simplest model of vestibular sense and response? This question came up while preparing to design a simulation that uses a genetic algorithm (GA) to evolve goal-directed bipedal locomotion. The simulation uses a primitive physical model; point masses constrained by spring forces, where the spring lengths can oscillate. The reason for using a simple physical model is to reduce the problem-space to a small set of parameters to gain insight into the fundamental factors involved—in the spirit of Occam's Razor. In this paper, the word "vestibular" is used to denote the ability to detect the tilt of a linear object (a sensor) in relation to the direction of gravity.

A key goal is to explore how secondary motions and emergent oscillations in a spring-mass system are exploited by evolution to support energy-efficient locomotion. This is consistent with the view that body dynamics plays a significant role in bipedal locomotion and that it is not only determined by neural control (O'Connor and Kuo, 2009). This is called *passive dynamic walking* (McGeer, 1993).

This exploration is less concerned with biological accuracy than with the kinds of complex adaptive behaviors that emerge, given a relatively simple set of rules. Artificial life simulations can be based on cellular automata, modeled in a 2D plane, or have no spatial representation at all, yet demonstrate remarkable lifelike behaviors (Ray, 1994). The simulation described here uses a custom 3D physics model with a minimal set of physical constants. In a general sense, it is a 3D dynamical system where a GA optimizes the parameters for achieving the shortest distance between two points (the center of mass of a virtual creature and the position of a moving target). It also optimizes for all point masses except those designated as "feet" being on the y-positive side of the x,z plane (the ground). This creates selective pressure for evolving effective vestibular responses to tilt, to avoid falling over.

Flexible Vehicles

This approach is inspired by the work of Valentino Braitenberg, who's book Vehicles (Braitenberg, 1984) has influenced robot design and artificial life research. Braitenberg vehicles demonstrate how a simple four-wheeled robot can exhibit goal-directed behaviors using sensors (two light detectors in front) and actuators (independent speeds of two wheels in back). By wiring-up sensors to actuators in various ways (including neural net-like wiring in-between) many adaptive behaviors can emerge. Unlike Braitenberg's vehicles, these vehicles are not physical; they are virtual, composed of point masses arranged as vertices of tetrahedral modules, and held in place by springs (abbreviated as "tets" in this paper). Tet structures are referred to as "creatures". Figure 1 illustrates locomotion in the style of Muybridge, for three evolved creatures made of 1, 3, and 5 tets (a, b, c). In most illustrations springs and point masses are rendered as lines and dots. Translucent triangles are added to show tetrahedral faces.

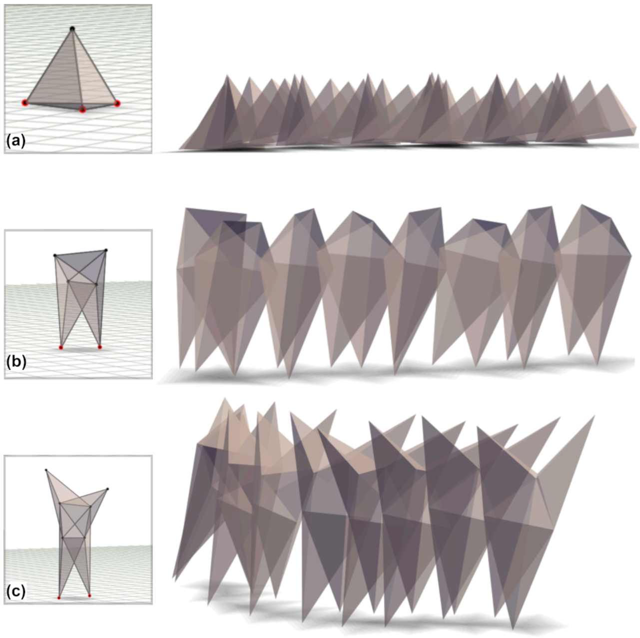

Figure 1: Motions of three evolved creatures.

This model of locomotion can be compared to Sodaplay (Soda Creative Ltd., 2000), a 2D physics game where “muscles” (springs that change in length) are used to construct a variety of playful contraptions and to explore the many ways they can achieve locomotion.

The springs in these creatures have stiffness and damping sufficient to enable vertical postures. For evolution of bipedal locomotion, a simplified model of the vestibular sense of balance is used. It consists of one or more sensors that continually detect the tilt of particular springs away from the horizontal plane (or equivalently: how *parallel* they are to the direction of gravity). To enable evolution of goal-directed behavior (following a wandering target in the environment), another sensor can be set to detect the relative direction to the target position. The continual signal stream from each sensor is used to modify the parameters of a simple central pattern generator (CPG) consisting of multiple sine waves with varying amplitudes and phases (one per spring). These sine waves force the lengths of the springs to oscillate in a vast number of possible ways, most of which are useless for locomotion. The GA is deployed to find the needles in the haystack.

Experiments

The first preliminary experiment tested a single tet for *tripedal* locomotion, using distance traveled as the fitness function (requiring no sensors). It resulted in a springy tet that hops like a three-legged frog (Figure 1a). Adding a sensor that detects the direction to a moving target, and changing fitness to being as close to the target as possible, encourages it to follow the target. And adding another sensor that detects the upward direction, along with a fitness component for staying upright as long as possible, encourages bipedal locomotion. The first example of bipedalism is a tet that hops on two of its point masses. Experiments with more complex structures resulted in several top-heavy creatures with human-like gaits. Their non-feet appendages often shift and sway, which is assumed to be beneficial to bipedal locomotion (otherwise those behaviors would not survive evolution). Many creatures can *almost* lose their balance, exhibiting momentary periods of stumbling and staggering—which is more pronounced when they are placed on an uneven terrain.

These experiments demonstrate how a GA can be used to search a large parameter space where the fitness landscape is made complex by virtue of unpredictable spring oscillations. Bipedal locomotion would be nearly impossible to achieve by manually adjusting the parameters, of which there can be over 100. The fitness function measures three attributes: 1. how closely the creature follows the moving target; 2. how long the creature remains upright before falling onto a non-foot point; and 3. how energy-efficient the motions are. These are explained in more detail in the sections below.

Related Work

An early example of spring physics used for simulated locomotion is a fish model by (Tu and Terzopoulos, 1994). A common model for walking dynamics is based on vaulting over a stiff leg as an *inverted pendulum*. In contrast, (Geyer et al. 2006) show that *compliant* leg behavior (using a spring-mass system) is more general, encompassing running and other gaits. Research in spring-based modeling for bipedal locomotion has contributed to robotics and biomechanics: more than can be referenced here. A few examples include (Blickhan, 1989) and (Selvitella and Foster, 2022). The center of mass (CoM) is often referenced in bipedal robotics in relation to its vertical offset from the center-of-pressure (CoP); the point of foot-ground contact. Most bipedal robots do not have vestibular sensors. Robotics researchers such as (Mergner et al. 2009) and (Mahboobin, 2008) propose vestibular models in addition to CoP and vision, combined as sensor fusion. (Fukuoka et al. 2013) simulated a quadruped with vestibular modulation based on body tilt, allowing automatic gait transitions.

The present model does not make explicit use of CoM. It relies on a GA to find solutions for upright locomotion. Early examples of using evolutionary algorithms for locomotion include (Sims, 1994), (Ventrella, 1994), and (Komosinski and Ulatowski, 1999). (Szerlip and Stanley, 2015) developed an evolutionary system for walking and jumping using the Sodarace construction set.

(Geijtenbeek et al. 2013) present a sophisticated evolutionary model that simulates muscles and neural control, resulting in convincing bipedal locomotion. In contrast, the present model is intentionally abstract, and is similar to a technique described by (Sano and Sayama, 2006).

Virtual Creatures

The mass of a chicken is distributed throughout its body. Figure 2b is a drawing that illustrates an approximation of chicken mass using an irregular spring-mesh that divides the volume into many tessellating tets, having point masses as vertices. If the point masses could bounce when colliding with a ground surface, the whole structure would behave like a rubber chicken tossed onto the floor—bouncing a few times and then coming to rest.

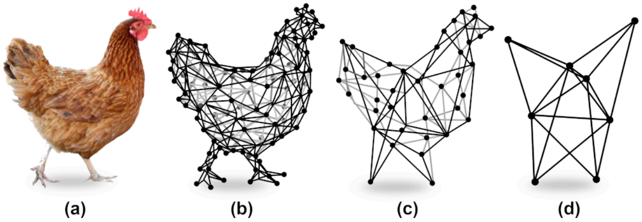

Figure 2: Simplification of mass distribution and connectivity in a chicken.

Achieving realistic dynamics would require fine temporal resolution, associated fine scaling of forces, and sufficient damping. Carefully-tuned spring constants could make it less rubbery and more chicken-like. High-resolution modeling is important for many research, design, and engineering purposes, but it is computationally expensive and prone to instability. Figure 2c illustrates a simpler approximation. And Figure 2d shows what may be the simplest approximation that still maintains some semblance of legs, tail, and head. Compared to a chicken, this is quite abstract. But these 8 point masses and 18 springs provide plenty of emergent complexity for the purpose of the experiments described in this paper.

Ten basic body types were chosen as experimental subjects. (Figure 3). Connecting four point masses with six springs of length 1 results in an approximation of a regular tetrahedron (Figure 3a). Adding an extra point mass outside a tetrahedral face and connecting it to the face with three new springs results in two tets that share 1 face, 3 point masses, and 3 springs (Figure 3b). More tets can be added in this way. Figures 3e and 3j show creatures each made of five tets, with springs forming the edges of a deformed stellated tetrahedron.

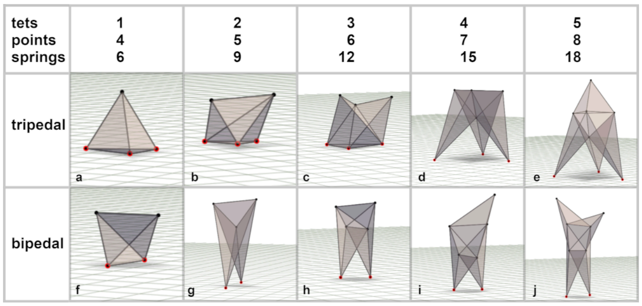

Figure 3. Body types ordered by number of tets and locomotion type.

Spring forces are determined by a stiffness constant and a damping constant. Point masses are constrained by these spring forces, and also by two external forces: gravity and soft collisions with the ground surface.

Motor Control

Creatures have no brains; just a CPG that can evolve. Each creature is born with a regular heartbeat: a master sine wave oscillation that drives the CPG. The frequency determines the gait period, and remains constant throughout the creature's life, but it varies among creatures as determined by genes. The amplitude and phase of the oscillation vary uniquely among springs as determined by genes. They affect how each spring's length oscillates. More precisely: the spring's *rest length* (*equilibrium length*) is actually what oscillates, and the associated point masses continually adjust as determined by spring forces. Without any input from sensors, the resulting oscillations serve as the CPG for all locomotion gaits. The addition of sensors dynamically modifies these parameters, creating opportunity to adapt to changes in the environment.

Some Types of Movements

A paper by Braitenberg titled "Some Types of Movements" was published in the first Artificial Life conference proceedings (Braitenberg, 1989). He begins by comparing small aquatic animals (having simple symmetry like his vehicles) to larger animals with articulated appendages: their motor systems must manage more degrees of freedom. "They pose fascinating problems of interpretation…because we do not know in what terms the states and changes of the motor system are represented in the brain." He ends with, "It is my contention that the brain deals with movements in a naive way".

This is taken as permission to build a *naive* model—to discover how a motor control system might evolve in a toy simulation where the fitness criteria are abstracted from the gnarly details of bodily mechanics. Octopuses have been observed "walking" across the ocean floor on two tentacles, sometimes as a defensive posture (Amodio et al. 2021). The talents of these remarkable mollusks have inspired robotic designs (Calisti et al. 2014). Initially the goal in this project was to simulate bones with muscles as antagonistic pairs (Figure 4a) but this was abandoned for something more general. Figure 4b illustrates a vertebrate leg, an octopus tentacle, and the vertex of a tet. These can be generalized as appendages with end-effectors that make contact with a ground surface, where friction affords horizontal movement.

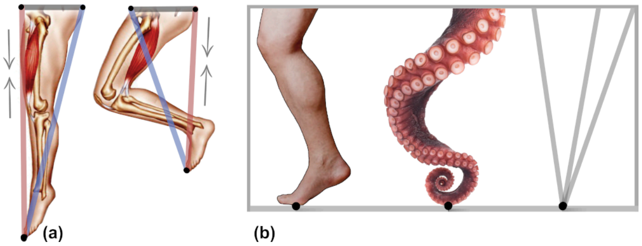

Figure 4: Antagonistic pairs of muscles (a). Ground-contact of vertebrate leg, octopus tentacle, and tet (b).

How exactly do the springs in this model move a creature's "feet"? For illustration, Figure 5a, shows the effect of two edges of a unit equilateral triangle (green and blue) changing their lengths in unison, oscillating between 0.5 and 1.5, which causes vertex y to move vertically. Figure 5b shows a tet as seen from above with its upper three edges oscillating in length. Their oscillations are out of phase with each other in such a way as to cause the top vertex to trace out an approximate ellipse.

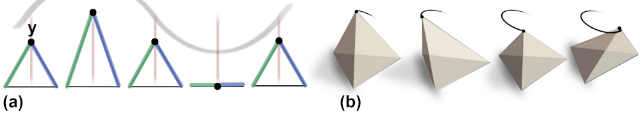

Figure 5: lengths of 2 triangle edges oscillating in unison (a); lengths of three tet edges oscillating with different phases (b).

By allowing each spring length to oscillate with its own unique amplitude and phase, a huge variety of motions are possible—including the characteristic alternating steps of walking. This brings up an important question: can springs *conflict* with each other as they *try* to force point masses around? Indeed they can, and almost always do. Imagine if the triangle in Figure 5a were made of stiff springs, and the top edges oscillated with a greater amplitude (e.g., between 0.4 and 1.6). At their shortest length they would be over-stretched by the forces of the bottom spring, and the bottom spring would suffer compression from the forces of the two upper springs, causing a build up of potential energy. The tet is more prone to conflicting forces, and a creature with 18 springs is even more prone. Without coordination, pressure can build and cause mayhem. But here the GA is used for mathematical optimization: to weed-out the most self-injurious of the population. To encourage more energy-efficient locomotion, a penalty is applied to the fitness function proportional to the average magnitude of spring forces. Thus, excessive spring force lowers fitness.

Animal morphology and motor control evolve together. The present model makes some accommodation: although predefined body topologies are used, there are up to 9 genes that can modify the proportions of the body, allowing morphology to co-evolve with motor control. Spring stiffness is also determined genetically, within a certain range (specified below). Spring stiffness applies uniformly to all the springs in a creature.

Sensors

Imagine a line segment extending from your left shoulder to your right shoulder. That line should be perpendicular to the direction to a target you are walking towards. (It also helps if it is perpendicular to the direction of gravity, otherwise you may get dizzy). Now imagine two direction vectors originating from your head (Figure 7a); one vector points forward in relation to your head's coordinate system, and the other vector points rightward. They are perpendicular to each other. Rotating your head causes these vectors to tilt.

Instead of using a creature's direction vectors for detecting tilt (because there aren't any), the directions of predetermined springs in the creature's body are used. An example is shown in Figure 7b. One spring extends from a to b, and another from c to d. Springs provide the scaffolding on which to place sensors. They are embedded in the body and are thus subject to the bumps and surges of true embodiment, like the inner-ears of all bipedal animals. The utility of these sensors evolves in tandem with locomotion.

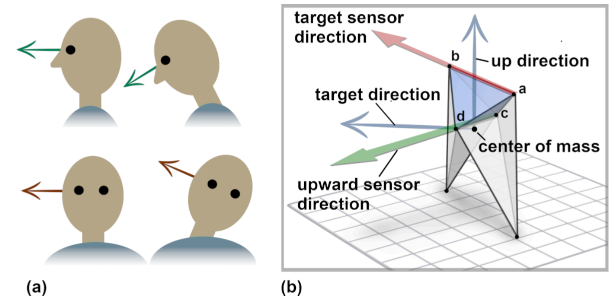

Figure 7: Head rotations with front and right direction vectors (a); using spring directions as sensors (b).

Spring vectors are normalized to be used as sensors; they serve as directions in the global coordinate system. They continually signal the degree to which they are parallel or antiparallel to the directions toward specified points in the environment, using a dot product. For example, a sensor can be configured to detect the direction from the creature's center of mass to the zenith (i.e., the upward direction, opposite of gravity). Another sensor can be configured to detect the direction from the creature's center of mass to a wandering target in the environment. Sensors that are configured for these two purposes are referenced as "upward sensors" and "target sensors".

If you take the dot product $\boldsymbol{o}$ of a sensor's direction with the direction to its associated environmental point, you get a value in the range of -1 to 1 (because all vectors involved are normalized). A value of -1 means the sensor direction is pointing *away* from the environmental point; a value of 1 means it is pointing *towards* it; and a value of 0 means the sensor direction is *perpendicular* to the direction to the point.

The creature in Figure 7b has the body plan of Figure 3h. The "core" tet is shown in light blue, having vertices abcd. In a regular tetrahedron, opposing edges are perpendicular to each other. This is highlighted with red and green arrows in the figure. They are approximately horizontal when the creature is standing upright. A sensor can be associated with *any spring* in the creature and it will produce valid signals, but with varying degrees of fidelity. The two spring directions shown in Figure 7b are optimal for this body type. Since there is more risk of tilting forward/backward (*pitch*) than left/right (*roll*), an upward sensor is placed on the cd spring to aid in equilibrium. A target sensor is placed on the ab spring, to aid in moving towards the target.

The upward sensor offers a crude model of vestibular sensing, but what exactly does the target sensor model? Perhaps vision, but only in a very loose sense. These sensors are generalized to operate as components in an exploratory cybernetic design. Any position in global space can be referenced by a sensor. The output of each sensor modifies *every* spring simultaneously. These outputs are *not* averaged, but treated separately. This associates specific dynamic locations in the environment with motor behavior. All sensors are continually operating, creating opportunity for sensor fusion by evolution.

Actuators

Are these springy sensors up to the task? Can a wobbly spring provide a reliable signal for a creature to establish equilibrium? Perhaps wobbling springs with sensors adapt through evolution to operate in sensorimotor loops with other wobbling springs. The hypothesis is that the parameters determining spring dynamics simultaneously adapt for locomotion and for sensor/actuator function.

To ease over-reactivity to noise, the signal stream $\boldsymbol{s}$ from each sensor passes through a hysteresis filter to smooth-out spikes before being used to modify the oscillation parameters. It is a global value that is continually updated as follows: $\boldsymbol{s}=\boldsymbol{s}\times\boldsymbol{w}+\boldsymbol{o}\times(1-\boldsymbol{w})$, where $\boldsymbol{o}$ is the dot product and $\boldsymbol{w}$ (smoothing weight) is 0.5.

Modifying these parameters is like smoothly tuning knobs on an oscilloscope to adjust the amplitudes and phases of several sine waves (see Figure 11). Each spring has two associated genes that determine the baseline amplitude and phase—these remain constant throughout the creature's life. Each sensor determines real-time adjustments to these amplitudes and phases. The amount of adjustment is determined by genes.

Dynamically adjusting the amplitude of a sine wave is straightforward, but shifting the phase is tricky because it has the effect of momentarily changing the speed of the oscillation. The rate of change should not shift the sine wave by a whole cycle in a short amount of time. The smoothing filter helps to ease this effect. Any remaining spasms that are disruptive to locomotion are weeded out by the GA.

Simulation

The simulation is modeled as a continuous 3D space with a horizontal ground plane at the origin. The positive y axis designates the upward direction. Point masses have position and velocity, and zero volume. They all have a mass of 1. At initialization, the point masses of a creature are arranged in a predefined configuration, with proportions determined genetically. They are positioned so that the designated feet are on the ground surface. The topology of spring-to-point-mass connections is also set. Springs are massless. The rest lengths of the springs are determined at initialization according to the distances between associated point masses. After the simulation starts running, various forces are continuously added to their velocities, including gravity (a constant force in the downward direction of size g), and spring forces (scaled by stiffness k, which varies genetically between creatures). There is a continual damping of all velocities (scaling by $d<1$). A small amount of heat (noise) is added in the form of random forces to the x, y, z components ranging from $-h$ to h.

If a point mass falls below the ground plane there is a soft collision, as follows: 1. an upward force is applied with a magnitude equal to the amount of vertical penetration below the ground plane; 2. the vertical component of the velocity is dampened (scaled by $v<1$); and 3. friction is applied to the horizontal component. The horizontal friction force is a unit vector in the opposite direction of velocity, scaled by f. If the horizontal speed of the point mass is less than f then the horizontal component of the velocity is set to zero, having the effect of static friction. The physics-related parameters are:

gravity $g = 0.0001$
spring stiffness $k = 0.005$ to 0.015
damping $d = 0.01$
heat $h = 0.0001$
vert collision damping $v = 0.02$
horiz collision friction $f = 0.004$

These parameters are not associated with any particular real-world materials or forces; they are approximations. They are fixed for *all* experiments. Having more parameters than this was avoided for simplicity, maintainability, and repeatability.

The ground collision values v and f can be adjusted to approximate different kinds of ground surfaces. In general, higher values make the ground more "sticky".

The simulation is updated in regular time steps. A typical experiment runs at an animation rate of about 60 time steps per second, on a MacBook M1 Pro. At this rate, gait periods range from about 1 to 5 cycles per second. The simulation is coded in plain vanilla javascript, and the visuals are rendered in a browser with html5 canvas. A demo, source code, and video are available on the author's website[1].

The Moving Target

The target is a point mass located on the ground surface. Figure 8a shows a top-down view of a creature and the target at initialization. The target starts at a random location on a circle of radius $r=16$ (initial target distance), and the creature is centered at the origin. The direction to the target position relative to the direction of the target sensor is shown.

If the creature is able to get close enough to the target so its distance is less than 4 units (the repulsion radius), the target becomes mobile; there is a repulsion force causing the target to move away from the creature, and this force is greater when the creature is closer. It also turns red for visualization. There is a small horizontal randomizing force causing the target's movements to be unpredictable and unique for every simulation run. A damping constant is applied to the target velocity. These parameters were tuned so that the creature must first approach the target (requiring at least basic locomotion) and then—if it reaches the repulsion radius—it must turn in order to follow the target as it moves randomly and evades the creature. Several variations of these parameters were tested, and a set of values was chosen that provide sufficient challenge for the creature in these experiments.

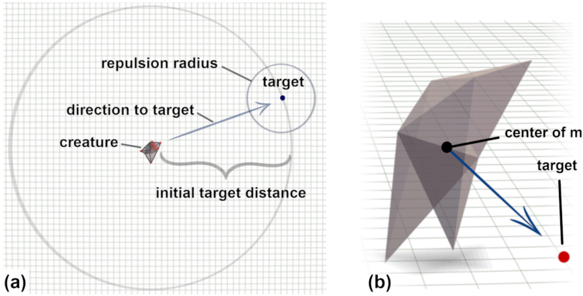

Figure 8: Top-down view of creature and target at initialization (a); Evolved 4-tet creature in repulsion zone (b)

The fact that the target is initialized at random locations and wanders randomly has the effect of adding *noise* to the environment. This discourages over-fitting; where the population converges on a narrow solution. It helps the population find more general solutions to turning so as to approach the target.

Genetic Algorithm

The genetic algorithm operates on a population of $p=100$ individuals. Each individual has an array of real values ranging from 0 to 1. These are the genes of the individual. The population is initialized with randomized genes. Every individual is given an initial fitness value of 0. The first n genes in the array are mapped to phenotype values, each of which has a pre-defined range. Phenotype values determine the evolvable attributes of a creature. The number of genes n that code for phenotype values depends on the complexity of the individual (e.g., a creature with more springs requires more parameters—up to 155).

A variation of *elitist* selection is used, whereby the most fit individuals persist over evolutionary time. And in fact this is the rule: the population is not replaced entirely with its offspring to advance each generation, as with the standard algorithm. Instead, it is updated (iterated) one individual at a time (typically 20,000 iterations). Here are the steps for a single GA iteration:

1. Selection. *Tournament selection* is used to choose two relatively-fit individuals as parents: a set of $t=5$ individuals are picked randomly from the population, and the most fit among

---

[1] http://ventrella.com/vestibular-bipeds

them is selected as the first parent. This process is repeated to get the second parent.

2. Reproduction. These two parents generate a single offspring individual via crossover with a chance of mutation. Crossover rate c is 0.1 and mutation rate m is 0.01. Small mutations are favored. If a mutated gene value falls outside of the interval, it is wrapped around.

3. Replacement. The offspring replaces (kills-off) the *least-fit individual* in the population. If there are multiple individuals sharing the lowest fitness value, then a random one is chosen for replacement.

4**. Testing**. The new individual is put into the simulation and tested. The simulation runs for 20,000 time steps, resulting in a fitness value for that individual. This fitness value *may or may not* be higher than the fitness of the individual that it replaced, but it is higher *on average*. In some tests, the fitness value may fall below 0. This does not pose a problem since selection is determined by *relative* fitnesses.

5**. Fitness Decay**. The fitness values of *all* individuals are decreased by a small amount ($i = 0.0001$) at each iteration. If the value of i were set to 0 then elitism would be in full force. Setting it lower is preferred because over time the uniform gradual decrease of fitness causes older elites to become less able to compete in tournament selection. The old fade and the young occupy the higher fitness range. This helps avoid premature convergence.

Tournament number t and fitness decay i are key values. Note that if tournament number t is 1 then the population *can still converge*, but slowly. A tournament number of 1 is essentially the same as choosing parents at random without any fitness weight. But by replacing the least-fit individual (which hence replaces its fitness with a new value that is higher *on average*), the population still converges. These values can be adjusted to affect the convergence rate.

It was found that the early innovators in a population are typically reckless and sometimes just plain lucky. For instance, a very unstable creature at the start of the simulation might "accidentally" bounce around near the target and remain there long enough to get a high score for "following the target", but the manner in which this was accomplished has nothing to offer for future generations. Fitness decay is analogous to making room for younger innovators to improve upon the old ways.

The parameters just described could be tweaked further and tested for better GA performance, but this design is sufficient for the purpose of these experiments. The goal is not to find the most optimal solution but to discover and analyze evolved strategies that are illuminating and informative.

Fitness Function

As mentioned earlier, there are three components to the fitness function (labeled here as w, u, and q):

1. Following the Target. The distance between the creature and the target is recorded at every time step, and the average is calculated at the end. Subtracting this value from the initial target distance r and then dividing that by r results in the fitness component $w<1$. If the creature spends the majority of the time at a distance greater than r, then $w < 0$. The better the creature is at following the target, the closer to 1 w becomes.

2. Staying Upright. For bipedal creatures, the number of time steps in which the creature remains upright is recorded. It is divided by the max number of simulation time steps (20,000) to get the fitness component u. If the creature stays upright for the entire duration then $u = 1$.

3. Being Energy-Efficient. All spring force magnitudes are accumulated at every time step. This accumulated value is divided by the duration of the simulation to factor out time, and scaled by a penalty parameter e to get the fitness penalty q. This value is larger for uncoordinated creatures whose springs are fighting with each other. A creature with no oscillating springs would result in an epsilon value (never zero because gravity and ground collisions are always in effect). The value q is subtracted from fitness as a penalty.

For simplicity, the fitness components w and u take equal weight in all experiments: fitness = $(w+u)/2 - q$

Here are the parameters associated with the GA:

population size	p	= 100
num genes used	n	= 59 to 155
tournament number	t	= 5
crossover rate	c	= 0.1
mutation rate	m	= 0.01
init target dist	d	= 16
fitness decay	i	= -0.0001
force penalty	e	= 0 to 1

Experiments and Observations

As a preliminary test, fitness was defined as the creature's distance from the starting point at the conclusion of the simulation. The single-tet body type was tested for tripedal locomotion (requiring no sensors). The GA was updated 2000 times. The creature with the highest fitness was animated in the simulation. It exhibited something akin to the gait of a hopping frog (Figure 1a).

Next, a target sensor was added to one of the bottom springs, and fitness was set to *closeness to target*. As hoped, the highest-fit creature resulting from this test was able to approach the target and then turn to follow it closely, sometimes passing over it completely, requiring near-180-degree turns, which wasn't always graceful. This test typically resulted in the target sensor being at the leading edge of locomotion, perpendicular to the direction of the target. But in some cases it was in an unexpected orientation, as shown with the red segment in Figure 9a. The reason is not clear, but it does imply a degree of versatility.

If you balance a tetrahedron on an edge, it will likely tip over onto a face. What does it take to keep the balance? To test for bipedalism, the tet was given a target sensor and an upright sensor, shown in Figure 9b as red (target) and green (upright). As with all bipedal experiments, it was initialized standing on two point masses designated as feet. The number of GA iterations was increased to 10,000 to allow time for the population to find its footing. Sometimes the results were fruitless. (As a very rough estimate, about half the tests were successful). The expectation was that evolved creatures would adopt a walking gait, but GA runs almost always resulted in a hopping gait in a direction *parallel* to the bottom spring, as shown in Figure 9b. It was still able to turn, but not very skillfully. The reason is not entirely clear. Figure 9c shows a top-down view of the footprints of this creature hopping towards the target, with poor turning ability.

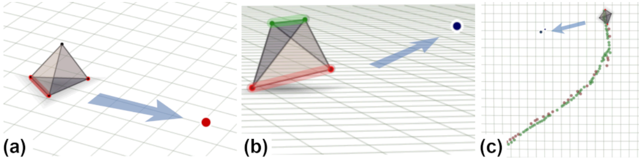

Figure 9 Tripedal tet with unexpected sensor orientation (a); bipedal tet with hopping gait (b); top-down view of bipedal tet following target poorly, with footprints (c).

The next subject tested was the bipedal 3-tet creature shown in Figure 3h. Tests had the expected result of a human-like walking gait, although comparatively stiff due to its simplicity and obvious lack of an upper-body. Figure 10a shows a time series graph of the *average* fitness of a typical population for all 3 components of the fitness function. The graph represents 20,000 GA iterations. The vertical range is from 0 to 1.

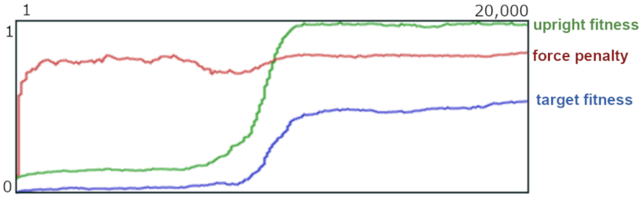

Figure 10: Fitness components (population average) over 20,000 GA iterations, for the 3-tet bipedal creature.

The green line shows *upright* fitness $\boldsymbol{u}$. It typically jumps quickly and approaches the value of 1 and hovers just below it: once the population has mastered bipedalism, most creatures remain upright throughout the simulation. The blue line shows *target* fitness $\boldsymbol{w}$ (this value can start negative if many creatures are moving away from the target, but it typically becomes positive early on as wanderers are weeded-out). Target fitness typically lags behind upright fitness, presumably because a creature must be upright in order to approach the target, and the simulation is terminated as soon as a creature falls. The red line shows force penalty $\boldsymbol{q}$. Note that this line is mapped differently: since $\boldsymbol{q}$ is negative and typically small it is translated to the top of the graph and scaled by a factor of 50, so it can be seen more easily—as a visual overlay. It starts low because many creatures have conflicting, chaotic spring forces, which the GA quickly culls.

Motor Symmetry

Although these body plans have symmetry, the motor control parameters have no inherent symmetry. In the beginning, most creatures tested either fall over, rotate in place, trace out spirals, or bounce erratically. Selective pressure for initial locomotion towards the target encourages moving in a relatively straight path. This in turn encourages the motor control parameters to converge on a symmetrical solution. In a sense, each body plan prefers its own kind of motor control symmetry, and evolution complies (to varying degrees and in different ways). But symmetry is not always achieved entirely, and never perfectly, which is why there is usually a detectable limp. Sometimes *limping* appears to be a strategy in itself, supported by asymmetrical contributions from various springs.

Figure 11a shows the baseline pattern of sine waves for the CPG of a random un-evolved bipedal 5-tet creature *without* modification from sensors. It was recorded over the span of 1000 time steps. There are 18 sine waves (one for each spring in the creature). Figure 11b shows the pattern for the most fit individual (Figure 1c) at the end of 20,000 GA iterations. Frequency is lower (gait period longer), many springs have subdued amplitudes, and there is more phase alignment. Figure 11c shows the same creature's pattern but in this case it includes modifications from the sensors, which deform the sine waves. Also, in Figure 11c the *actual* spring lengths are shown. These lines roughly conform to the rest lengths, but they have a lot of wiggle. This visualizes the dynamic of point mass motion caused by spring forces. Note that the actual spring lengths are the values selected for by evolution.

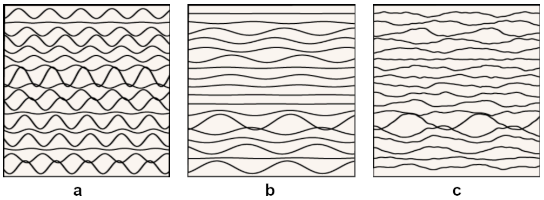

Figure 11: CPG for un-evolved bipedal 5-tet *without* sensor modification (a); similar pattern for an *evolved* creature (b); pattern for the same creature showing *actual* spring lengths including sensor modifications (c).

Balancing Act

One interesting 4-tet bipedal creature frequently lost its balance, but had a remarkable ability to correct itself after a period of staggering which could last for several gait cycles. This demonstrates an ability to recover in a general way with variation. To put it to the test, it was placed on an uneven terrain (a variation having periodic undulations in the vertical component with phases shifted randomly at initialization). Figure 12a shows the creature with footprints added to illustrate moments of staggering and slipping. Even on the uneven terrain it was (sometimes) able to reach the target.

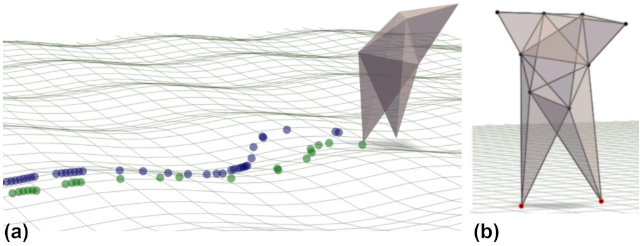

Figure 12: A 4-tet creature with footprints showing staggering on an uneven terrain (a); An 8-tet creature (b).

More Complex Bodies

All body types in Figure 3 were tested, with mostly positive results. The 1-tet and 2-tet bipeds (Figure 3f and 3g) were found to be poor performers, perhaps due to having limited degrees of freedom. The two tripods (Figure 3d and 3e) came up with unique solutions to locomotion. They were not given upright sensors under the assumption that having three feet is inherently stable. However, being unable to correct for tilt, they frequently fell over.

Figure 12b shows an experimental creature made of 8 tets (having 10 point masses and 25 springs). The stubby arm-like appendages displayed plenty of movement, which appeared "purposeful". A conclusion is that this technique could be scaled in terms of creature complexity.

Curiosities, Questions, Further Work

There are several observations, curiosities, and untested questions worth mentioning, even if briefly:

One creature displayed a curious 2-period gait cycle. Presumably, the emergent dynamic of one gait cycle changed the conditions for the sensors in the next cycle, resulting in alternating motions.

Extremely wobbly target sensors can still be effective, particularly when the wobble is mostly on a plane perpendicular to the direction to the target.

Even with randomly initialized target positions, populations tend to favor particular directions for turning, which impedes generalized solutions to turning. Longer GA runs usually help the population generalize to some degree, but in many cases creatures are not able to turn in one direction, and always choose a single direction even though it is inefficient.

The single tripedal tet often suffers from ground stickiness, associated with horizontal static friction—possibly because its feet drag more than those of creatures with different anatomies. Altered ground collision settings can help with different anatomies.

Immobilizing the six springs of the core tet was found to stabilize the more complex creatures. But having all springs oscillate is preferred in these experiments, assuming there are enough GA iterations for these motions to subdue and/or become more coordinated.

Initial experiments using multiple upright sensors were not conclusive. One hypothesis has not been tested yet: "one-legged" locomotion may be possible by including multiple upright sensors, allowing responses to different axes of tilt, analogous to what the vestibular sense accomplishes in coordination with other senses, while hopping on a pogo stick.

Conclusion

A structure of 8 point masses connected by 18 springs exhibits unpredictable chaos from nudging a single point mass. Evolution is a master at taming chaos, as witnessed in the amazing feats of bipedalism (think of a juggler on a tightrope). Achieving the characteristic strut of a chicken is outside the scope of this exploration, but these experiments may shed some light on a few of the basic principles involved in bipedalism, for future research. And it might inspire some new techniques for animating characters.

References

Amodio, P., Josef, N., Shashar, N., and Fiorito, G. (2021). Bipedal locomotion in Octopus vulgaris: A complementary observation and some preliminary considerations. Ecol Evol. Mar 5; v.11(9).

Blickhan, R. (1989). The spring-mass model for running and hopping. Journal of Biomechanics. Volume 22, Issues 11–12, pages 1217-1227

Braitenberg, V. (1984). Vehicles: Experiments in synthetic psychology. MIT Press, Cambridge, MA.

Braitenberg, V. (1989). Some Types of Movements. In Langton, C. Artificial Life I. MIT Press, Cambridge, MA.

Calisti, M., Corucci, F., Arienti, A., Laschi, C. (2014). Bipedal Walking of an Octopus-Inspired Robot. In: Duff, A., Lepora, N.F., Mura, A., Prescott, T.J., Verschure, P.F.M.J. (eds) Biomimetic and Biohybrid Systems. Living Machines. Lecture Notes in Computer Science, vol 8608. Springer.

Fukuoka, Y., Habu, Y., and Fukui, T. (2013). Analysis of the gait generation principle by a simulated quadruped model with a CPG incorporating vestibular modulation. Biological Cybernetics volume 107, pages 695–710

Geijtenbeek, T., van de Panne, M., and van der Stappen, F. (2013). Flexible muscle-based locomotion for bipedal creatures. ACM Transactions on GraphicsVolume 32 Issue 6 Article 206, pages 1–11

Geyer, H., Seyfarth, A., and Blickhan, R. (2006). Compliant leg behaviour explains basic dynamics of walking and running. Proceedings of the Royal Society B: Biological Sciences. vol 273, pages 2861–2867.

Komosinski, M., and Ulatowski, S. (1999). Towards a Simulation of a Nature-Like World, Creatures and Evolution. Lecture Notes in Computer Science, vol 1674, pages 261-265.

McGeer, T. (1993). Dynamics and Control of Bipedal Locomotion. Journal of Theoretical Biology. vol 163, pages 277-314

Mergner, T., Schweigart, G., and Fennell, L. (2009). Vestibular humanoid postural control. Journal of Physiology - Paris 103, pages 178–194

Mahboobin, A. (2008) Computational and Robotic Models of Human Postural Control. Doctoral Dissertation, University of Pittsburgh.

O'Connor, S.M. and Kuo, A. D. (2009). Direction-dependent control of balance during walking and standing. Journal of Neurophysiology. vol 102 Issue 3, pages1411-1419. American Physiological Society

Ray, T. 1994. Evolution, complexity, entropy and artificial reality. Physica D: Nonlinear Phenomena, volume 75, Issues 1–3, pages 239-263

Sano, K. and Sayama, H. (2006). Wriggraph: A kinetic graph model that uniformly describes ontogeny and motility of artificial creatures. Artificial Life X, pages 77-83. MIT Press, Cambridge, MA.

Selvitella, A. M., and Foster, K. L. (2022). The Spring-Mass Model and Other Reductionist Models of Bipedal Locomotion on Inclines. Integrative and Comparative Biology, Volume 62, Issue 5, November 2022, pages 1320–1334.

Sims, K. (1994). Evolving 3D morphology and behavior by competition. In Artificial Life IV. 28–39. MIT Press, Cambridge, MA.

Soda Creative Ltd. 2000. Sodaplay. https://en.wikipedia.org/wiki/Soda_Constructor

Szerlip, P., and Stanley, K. O. (2015). Indirectly Encoding Running and Jumping Sodarace Creatures for Artificial Life. Artificial Life. 21 (4): 432–444.

Tu, X. and Terzopoulos, D. (1994). Artificial fishes: Physics, locomotion, perception, behavior. Proc. ACM SIGGRAPH, pages 43–50.

Ventrella, J. (1994). Explorations in The Emergence of Morphology and Locomotion Behavior in Animated Characters. In Artificial Life IV. MIT Press, Cambridge, MA.