Architectural Models of Bacterial Chemotaxis

Adaptive Systems – Project Report

Candidate # 81222 (Ken Webb)
April 15, 2004

Abstract

E coli bacteria follow chemical gradients towards food and away from noxious chemicals. This bacterial *chemotaxis* is controlled by an adaptive protein network that senses chemical levels in the environment, and controls movement through rotary motors attached to long flagella. The flagella and motors are also constructed out of proteins, and as with the controlling protein network, are therefore under evolutionary genetic control. It is important in any computer system, including biological simulations, to start with a good architecture. This paper presents two complete architectural models, one very high level that explores externally observable behavior, and a second that explores some key difficult areas in more depth. This work is a first step that lays the groundwork for more comprehensive simulations, with the long-term goal of building an adaptive system in which both the behavior and the structure are under evolutionary control.

Introduction

This paper will discuss the various ways in which bacteria adapt to their environment, as they swim towards food and away from toxic substances, a process called *chemotaxis*. It will focus on several modeling tools. It will take a whole-cell architectural perspective, and demonstrate the benefits of doing so.

E Coli bacteria adaptively control combinations of forward-movement runs with direction-changing tumbles, using protein networks that compare previous and current environmental conditions. This control mechanism triggers rotary motors in the bacterial membrane that turn elongated flagella that bundle together to produce a propeller-like device. Adaptive elements include: evolution of the protein network control mechanism, optimization of rigidity and flexibility in the flagella, maintenance of flagella at the correct length, ability to repair damaged flagella, and evolution of the rotary motor the only wheel to have evolved naturally.

Using techniques learned in the Adaptive Systems course, one would model a single swimming bacterium by combining a physics engine such as Open Dynamics Engine (ODE) (Smith, 2003), a continuous-time recurrent neural network (CTRNN) (Beer, 1995; Beer, 1996; Slocum, Downey, and Beer, 2000), and a genetic algorithm (GA) (Mitchell, 1997). ODE provides a simulation of the environment and the body of the organism. The CTRNN

uses the body's sensors and controls its movement. The GA can be used to evolve the CTRNN.

Once the CTRNN is properly trained, and able to control the organism within the simulated ODE environment, it is loaded onto a robot operating in the real world. This allows the model organism, and especially its controller, to be tested in the real world. It is probably impossible to create a 100% accurate environment within the computer.

This approach however cannot capture what I believe to be one of the most important aspects of the bacterial chemotaxis. This is the role of proteins at both the structural and behavioral levels, under the adaptive control of genetics over an evolutionary time scale, over a separate developmental time scale, and moment by moment as it finds its way through its environment. The CTRNN does not represent behavior at the level of protein networks. A robot, or ODE simulation of a robot, does not use proteins (or some protein-like entity) as basic structural building blocks.

A bacterium, especially as it moves, is an ideal model organism to use in this exploration of architectural approaches to chemotaxis. As a single cell, it is relatively simple, and yet evolution has crafted a surprising amount of complexity into it. It includes a molecular control system that uses a network of proteins to allow it to adaptively follow a chemical gradient, and it also uses proteins to build the structure that allows it to move.

The work described in this paper is a first step toward learning how to simulate an adaptively swimming bacterium at the chemical level. Two simulations will be discussed. The first simulation, written in NetLogo, captures the overall behavior at a high architectural level and shows how multiple bacteria can move toward and congregate at a food source. The second simulation, written in C using ODE, captures the basic structure of a single bacterium and demonstrates how these structural elements dynamically interact to produce simple forward movement.

Motivation

The main motivation for this work was to pre-explore the overall architecture and some difficult (risky) areas in preparation for my dissertation. The two main practical research questions pursued in this paper are:

1. What is a good architecture to use for a whole-cell simulation of bacterial chemotaxis?
2. Can ODE be used to simulate the physics within the simulation, what are its limitations, how much non-ODE code will have to be written to get the job done, and what other tools will be required?

I see these as the two areas of highest risk, and have therefore chosen to write two separate simulations to pre-explore these areas. Because I have only just started to learn ODE, I have seen question 2 as representing the greatest risk, and have chosen accordingly to spend the bulk of my time on that.

I view other areas that will have to be dealt with during the dissertation, as of lower priority. For example, genetic algorithms (GAs) can be used to evolve various aspects. Because I took a comprehensive course last year on evolutionary techniques complete with several very demanding C++ programming projects using GA and genetic programming (GP), I do not see this as a high priority risk. In the final project in that course, I successfully evolved protein parameters (enzyme kinetic constants) resulting in an optimally homeostatically "healthy" cell. The appendix includes C++ code I developed during that course that could be used to evolve the parameters in a future more comprehensive simulation of bacterial chemotaxis in which, unlike in the current simple simulation, evolutionary control would be necessary.

For adaptive control, a CTRNN might be a good choice. But, as this paper will argue, artificial protein networks are more biologically plausible as the controlling element. "From the standpoint of a living cell, the closest approximation to a neural network is probably found in the pathways of intra-cellular signals" (Bray, 2003). These "systems of interacting proteins act as neural networks trained by evolution to respond appropriately to patterns of extracellular stimuli" (Bray, 1995). This conclusion is also supported by earlier work with my own CellAK cell simulation environment (Webb, 2003; Webb & White, 2003; Webb & White, 2004a; Webb & White, 2004b). A CTRNN can only control behavior. Because of its focus on enzymes, other proteins, cell bilayers, and other active objects, CellAK can control both behavior and structure. I see the use of CellAK for control purposes as of relatively low risk.

Biology background

An *e coli* bacterium is a cylindrically-shaped single cell that is about 2 μm long by about 1 μ m in diameter. Extending behind the cell is a set of flagella, typically six in number.

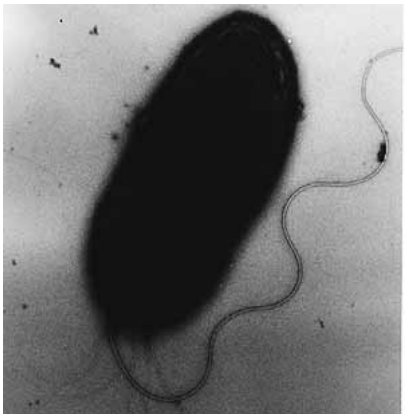

Figure 1: Bacterium with one flagellum (source: www.nigms.nih.gov/news/science_ed/cyto4.html).

Each flagellum consists of three parts – a rotary motor, a flexible hook, and an elongated filament. Flagella are typically placed at intervals along the side of the cell.

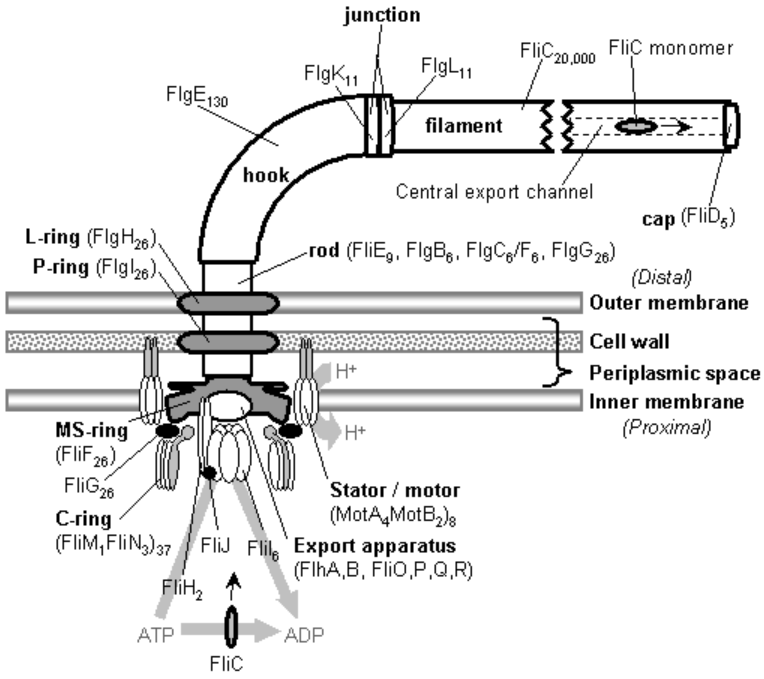

Figure 2: Flagellum showing basal body complex with motor, hook, and filament. The various proteins that make up each component are shown in non-bolded text (source: Matzke, 2003).

The motor, made up of some twenty genetically-specified proteins (Bray, 2001), is embedded within the outer membrane of the bacterium. Using a protein gradient for energy, it continuously rotates in either a counter-clockwise (CCW) or clockwise (CW) direction, at a speed of up to 300 revolutions per second (rps). It can reverse direction in less than one-tenth of a second. It generates enough torque to turn the hook and filament which are rigidly attached.

The filament, constructed out of repeating flagellin proteins, is a hollow tube typically 5-10 μm long and 14-20 nm (0.014-0.020) μm in diameter (Bray, 2001). It is semi-rigid, but flexible enough to curve into a variety of shapes. It takes on one shape when rotated CCW, and adopts one of three characteristic shapes when rotated CW. These shapes effect how it interacts with the water.

The hook, made out of a polymer of one specialized type of protein, flexibly connects the motor and the filament. The axis of rotation of the motor is perpendicular to the membrane along the length of the cell, while the filament extends behind the membrane. The hook therefore converts the rotation by about 90°.

The bacterial cell is able to reconstruct a damaged filament, and will adaptively maintain it at an optimal length. Flagellin proteins are manufactured, under genetic control, within the cell body. They are then transferred, with the help of other proteins, into the motor, through the hook, and along the filament to its distal (far) end, where other active proteins help to assemble them into the growing filament structure. All of this happens while the flagellum is rotating at 300 rps.

The cell's behavior depends on all of these structural elements working correctly. When the several flagella all rotate CCW, each interacts with the water around it. The combined motions create a fluid flow or vortex that tends to pull the filaments together into a bundle. The bundle takes on a shape that resembles a corkscrew, and functions like a propeller on a ship (Purcell, 1997).

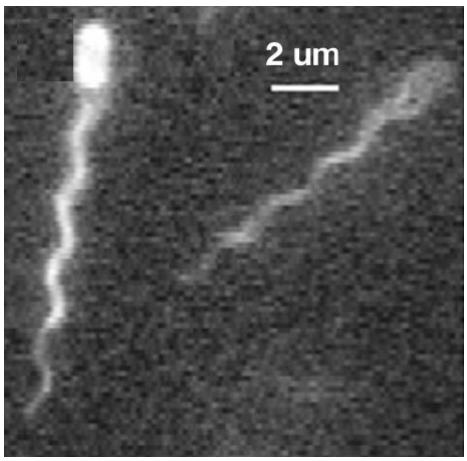

Figure 3: E Coli bacteria with normal-shaped (corkscrew) flagellar bundles (source: Turner et al., 2000).

The rapidly rotating bundle pushes against the water, which, as an opposite reaction, pushes the cell body forward. An e coli bacterium can move at up to 30 μm per second, about 15 times its body length.

There appears to be an autopoietic (Maturana and Varela, 1980; Beer, 2004) interaction between the flagella and the water. The flagella shape the water by controlling the fluid flow, while the water brings the flagella together into a corkscrew shape. Each participates in creating the shape of the other. Together they momentarily create a single organized unity that generates motive force, and that also generates an approximate boundary around the space of this unity.

The cell's forward motion is referred to as a *run*, and will typically last a few seconds. A run ends when one or more motors reverses itself. This reversal changes the rotating shape of the filament, and disrupts the bundle. As a result, the cell moves chaotically for a fraction of a second. This motion is referred to as a *tumble*. When the motors again reverse themselves, the bundle reforms, and the bacterium sets off at high speed in a new direction. An *e coli* cell continuously toggles between these two movement states. A run moves it forward some distance. A tumble changes its direction.

Figure 4: E Coli bacterium with unbundled filaments (source: Turner et al., 2000).

Using a combination of runs and tumbles, a bacterium can follow a chemical gradient, a process called chemotaxis. A run has been observed to last longer if it causes the bacterium to move up a chemical gradient toward a food source such as glucose, thus taking it closer to the nutrient-rich region in the environment. If the bacterium finds itself moving up a gradient of a noxious chemical such as phenol, its runs will be shorter in length (more tumbles).

The motors are controlled by a network of enzymes and other proteins within the cell body and membrane of the bacterium. The types of proteins are specified by the genome as a result of evolutionary processes. The average levels of these proteins are adaptively set based on current environmental conditions, through the attachment of various trigger chemicals to promoter sites along the DNA genome. As proteins inevitably degrade, they are adaptively replaced at a rate that depends partly on current average conditions in the environment.

At a second-by-second time scale, the bacterium adapts to changes in the chemical gradient using chemotaxis receptor proteins embedded in the membrane. When a food molecule such

as aspartate binds to the exterior binding site of the protein, a cascade of events is triggered. The protein deforms, which causes movement of other proteins inside the cell. A complex series of protein interactions ensues which ultimately affects the motor. This process includes a comparison of a "memory" of the previous state of the environment, with its current state.

Thus the cycle is complete. The bacterium senses the environment for specific chemicals, compares the current concentration with the concentration it "remembers" from a short time ago (up to a few seconds), adjusts its control of the motors, moves the flagella CCW or CW, runs or tumbles accordingly, senses the environment at its new location, and so on.

Proteins, under adaptive genetic control, do all of the work in this cycle. Receptor proteins sense the environment. A network of internal enzymes and other proteins interact based on this signal. The result of these interactions controls the motors, hooks and filaments of the flagella, which are themselves all constructed from proteins. Other protein pathways adaptively construct and repair flagella.

Behavior model, from an external observer's perspective

What is a good architecture to use for a whole-cell simulation of swimming bacteria? To help answer this research question, I have developed a model using the NetLogo programming language and graphical environment.

NetLogo is a parallel version of Logo, itself a descendent of Lisp, and like its predecessors it was originally developed at MIT. A NetLogo simulation includes a cellular-automaton (CA) grid that typically represents a 2D environment, and a number of active agents (turtles in logo-speak) that move around within the environment, continuously interacting with each other and/or with the environment. It is a relatively easy tool to use, and allows rapid development of prototypes.

I developed a minimal NetLogo model of bacterial chemotactic behavior and movement within a 2D environment. The environment can include various static distributions of food and noxious chemicals. Each position (NetLogo patch) in the CA grid represents a level for each of three chemicals. Up to 100 bacteria are represented as NetLogo active agents. Each time step each bacterium samples the chemical levels at its current position, compares these with the levels it internalized during the previous time step, and stochastically chooses a behavior that depends on this comparison.

The probabilities are such that if the gradient is up-hill toward a food source, it will tend to undergo longer runs of forward movement. If there is no difference in chemical level, it will be more likely to tumble which will result in its following a new course starting on the next

time step. If the gradient is leading it towards a noxious chemical, it will tend to tumble even more often.

I designed eight different experiments, each with different levels and distributions of the chemicals aspartate and glucose (foods) and phenol (noxious). The experiments are summarized in Table 1.

Table 1: Descriptions of each of the eight experiments.

Num	Experiment	Description
1	No chemicals	level = 0 for all 3 chemicals tests default behavior
2	Constant level aspartate	level = n for aspartate, = 0 for other 2 chemicals tests default behavior
3	Rectangular gradient	level varies from 0 to 1 (max) for aspartate in the y direction tests the run behavior; center is low value of aspartate
4	Three diffusing chemicals	three circular gradients (2 food, 1 noxious) with regions of no chemical; tests default, run and tumble behaviors
5	Long gradient	level varies from 0 to 1 (max) for glucose in the y direction tests the run behavior
6	Long uniform	level = n for aspartate, = 0 for other 2 chemicals tests default behavior
7	Long combination	level = 0 in lower half; = n in upper half for aspartate compares default with gradient run behavior
8	Brownian motion	allows above behaviors to be compared with simple Brownian motion

All experiments that contain chemicals were also designed to demonstrate the concentration of bacteria, as they all congregate around a food source. These concentrations result from the simple stochastically-determined processes of runs and tumbles.

Statistics were accumulated and demonstrate a difference in the mean length of a run between the *Long gradient*, the *Long uniform*, and the *Three diffusing chemicals* experimental conditions (Table 2). Whether or not these results are statistically significant has not yet been determined. For each of the Long gradient and Long uniform conditions, ten 100-step trials and five 1000-step were conducted. For the Three diffusing chemicals condition, five 100-step and five 1000-step trials were conducted. The mean run length and the standard deviation are given for each trial. Each trial involved 100 simulated e coli bacteria.

Table 2: Experiment results.

Experiment	TimeSteps	Mean	Std
Long gradient	100	1.923	3.599
Long gradient	100	2.088	4.203
Long gradient	100	2.094	3.818
Long gradient	100	2.164	4.077
Long gradient	100	2.028	3.836
Long gradient	100	2.082	3.786
Long gradient	100	2.109	3.867
Long gradient	100	1.949	3.811
Long gradient	100	1.951	3.625
Long gradient	100	1.967	3.469
Long gradient	1000	2.037	3.697
Long gradient	1000	2.040	3.678
Long gradient	1000	2.009	3.674
Long gradient	1000	2.065	3.765
Long gradient	1000	2.030	3.710
Long uniform	100	1.405	1.851
Long uniform	100	1.386	1.831
Long uniform	100	1.442	1.836
Long uniform	100	1.350	1.790
Long uniform	100	1.410	1.846
Long uniform	100	1.374	1.803
Long uniform	100	1.366	1.822
Long uniform	100	1.424	1.851
Long uniform	100	1.444	1.869
Long uniform	100	1.378	1.832
Long uniform	1000	1.442	1.884
Long uniform	1000	1.434	1.867
Long uniform	1000	1.442	1.878
Long uniform	1000	1.436	1.881
Long uniform	1000	1.436	1.878
Three diffusing chemicals	100	1.382	2.585
Three diffusing chemicals	100	1.438	2.574
Three diffusing chemicals	100	1.381	2.568
Three diffusing chemicals	100	1.296	2.350
Three diffusing chemicals	100	1.436	2.604
Three diffusing chemicals	1000	1.457	2.060
Three diffusing chemicals	1000	1.455	2.042
Three diffusing chemicals	1000	1.457	2.081
Three diffusing chemicals	1000	1.446	2.054
Three diffusing chemicals	1000	1.435	2.048

As expected, and in agreement with the behavior reported for real bacteria, the mean run length was longer for bacteria following a food gradient (Long gradient condition) than for those engaging in the default behavior in an environment with a uniform food distribution (Long uniform condition).

Figure 5 shows the movement of a typical single bacterium in the Long combination experimental condition. There is no chemical in the bottom half of the grid. In the top half

there is an increasing gradient from 0 to 1 (max). The bacterium is initially positioned at the bottom of the grid facing upwards (in the positive y direction). The larger number of tumbles and subsequent shorter run lengths when no chemical is present, is evident from the more erratic movements in the lower half of the grid. The longer run lengths and fewer tumbles when there is a chemical gradient, can be seen in the upper half of the grid.

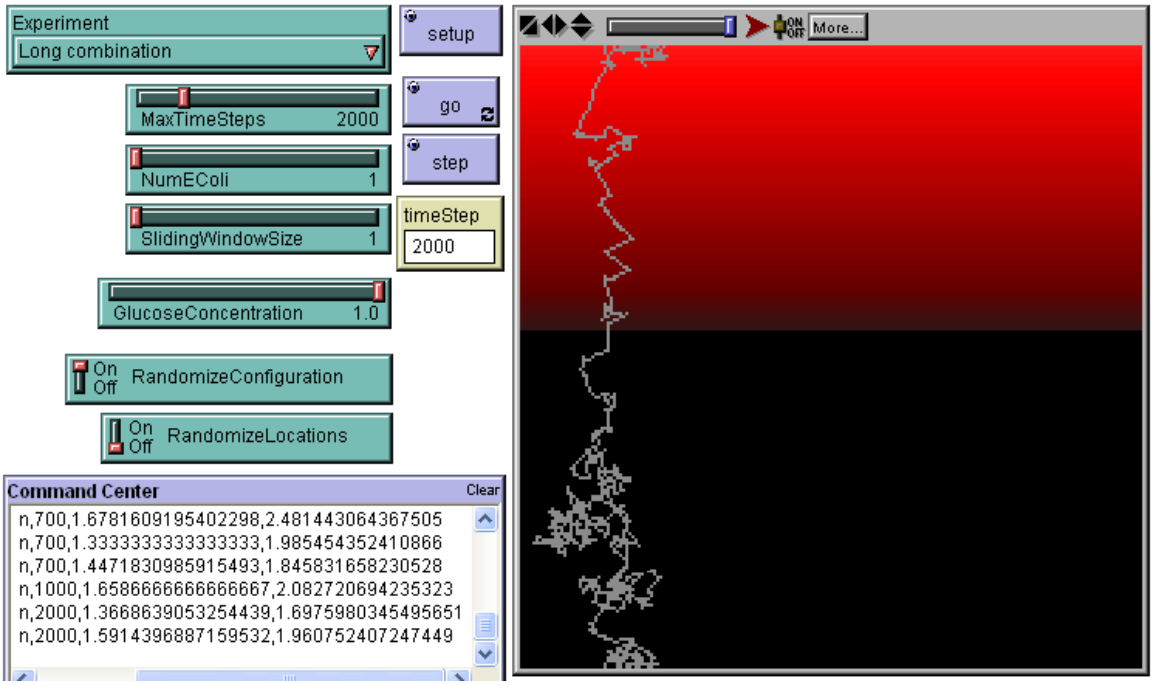

Figure 5: Movement of a single bacterium in the *Long combination* experimental condition.

In Figure 6, a single bacterium is following a gradient away from a noxious chemical concentration, and toward a food source. It then continues to move around but remains within the region containing a concentration of aspartate (food).

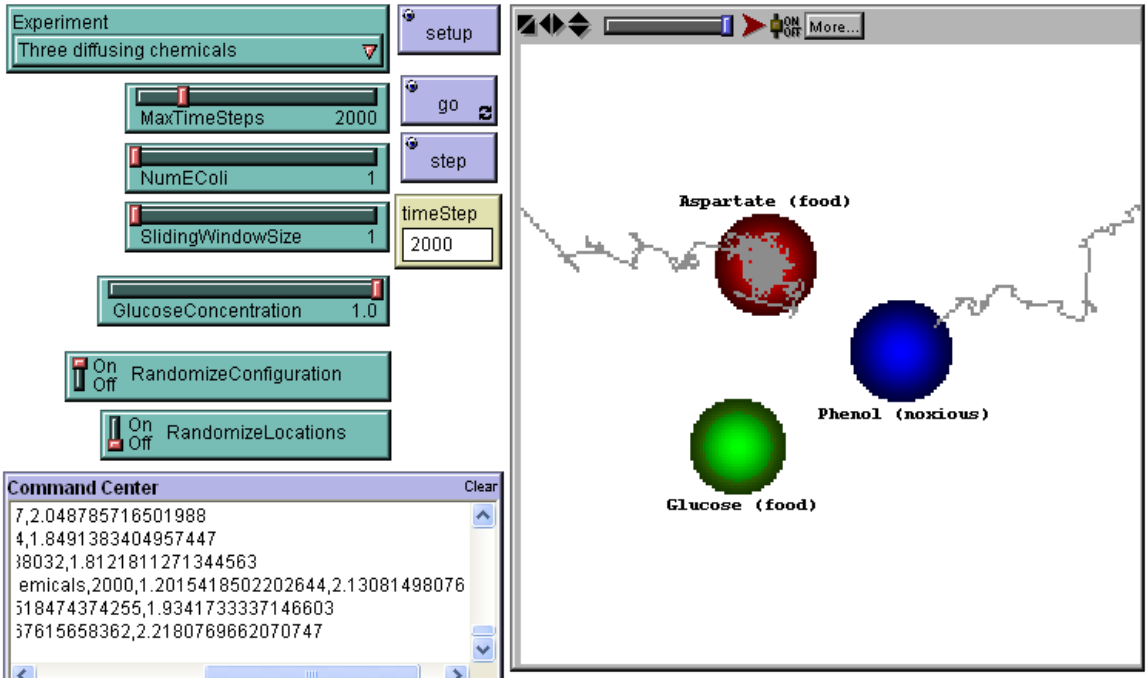

Figure 6: Single bacterium following a gradient away from a noxious chemical concentration, and toward a food source.

The NetLogo model groups all parameters (movement probabilities) into a single chromosome structure. There are currently nine probabilities, each a value between 0.0 and 1.0. There is one group of three for each of the three chemicals (`Asp` for aspartate, `Glu` for glucose, `Phe` for Phenol). The three for each chemical are the probabilities of a run or tumble based on whether the chemical level in the environment is greater than (`Gt`), equal to (`Eq`), or less than (`Lt`) the currently internalized level, the level found at the previous location. The chromosome could be evolved to more optimal values using a GA. It would, however, not be practical to use a GA in NetLogo for performance reasons. The actual values used in the experiments were hand optimized. Experiments using slightly different chromosome values yield means and standard deviations that are different from those reported in Table 2. One fitness function for a GA would be to find a chromosome configuration that would give the greatest difference in mean and standard deviation between the Long gradient and Long uniform experimental conditions.

In the behavior model, a bacterium has no internal structure. It is displayed as a square or other simple small shape constructed out of pixels. The elongated flagellum cannot be modeled. The bacterium does have some internal behavior, but at a very high stochastic level, which excludes any explicit dependency on the flagella. The relationship with the environment is similarly very simplistic. The environment contains chemicals, but no physics. There is no concept of matter that has mass, shape and size, and no concept of forces that can

cause things to move around. These are largely limitations with NetLogo itself. NetLogo's role has been to help get an architecture in place, and to observe the expected external behavior. We will next look at a technology that *can* provide internal structure, and *can* allow realistic physical interactions between objects and with the environment.

A structural model, with minimal dynamics

Open Dynamics Engine (ODE) is a physics engine. It allows the simulation of rigid objects, joined together by joints. Forces such as gravity can be applied to the world as a whole and all objects within it, or to individual objects. When objects collide, they bounce off each other in keeping with their size and shape. Every time step ODE recalculates all the forces and the resulting velocities, positions and rotations of all objects in the world. The objects can then be displayed on the screen. The result is the appearance of realistic dynamic movement.

Can ODE be used to simulate the physics involved in modeling a bacterium chemotaxis? To take a first step in answering this research question, I have constructed a minimal physical model. The cell body of the e coli is represented as a capped cylinder. The motor or basal body is represented as a sphere. A filament consists of n (default: 50) boxes strung together. All 52 objects are connected by joints. The motor has a torque that rotates it at a specified angular velocity. Gravity is set to 0 to allow the bacterium to float above the surface. There is only one flagellum attached at the center of the back end of the bacterium, instead of the six typically attached along the side. The flexible hook is not modeled, but would be needed if flagella were attached to the side.

Figure 7 is a close up view of the tail end of the e coli cell body. It shows the motor (the sphere), and the first few filament segments once friction has a chance to start rotating the segments. The direction of angular velocity of the motor is shown by an arrow.

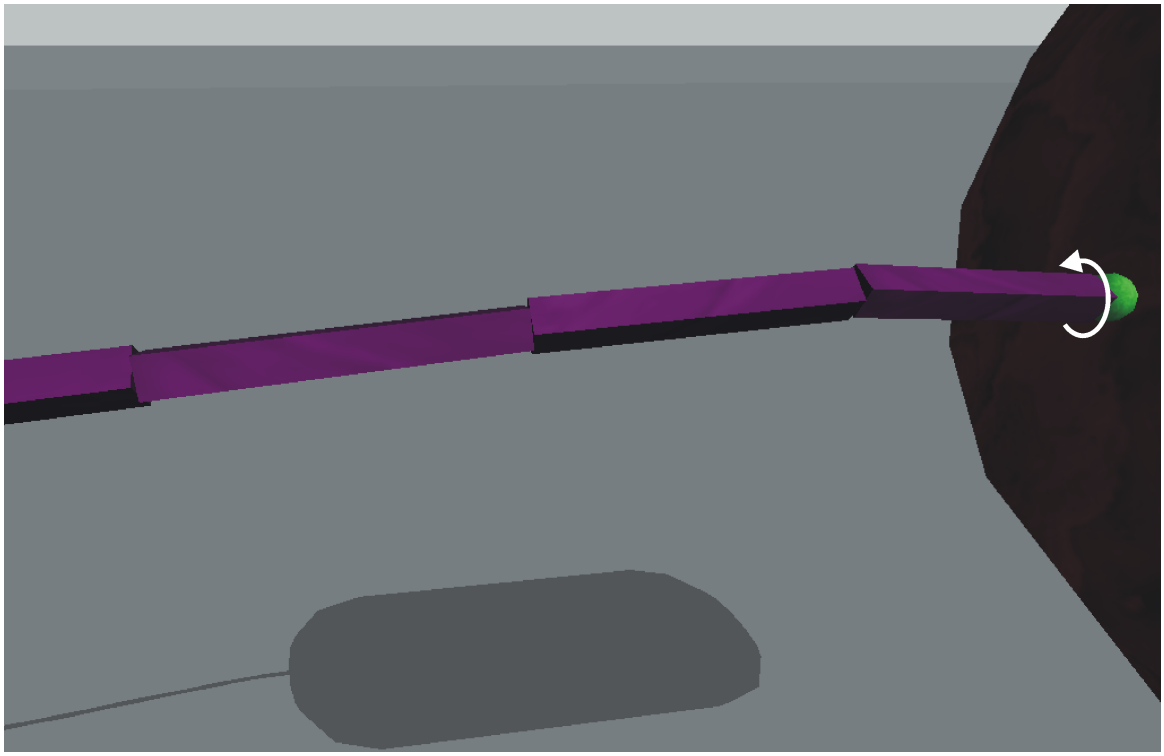

Figure 7: Close up view of the bacterium with the motor and several filament segments.

After a considerable amount of experimental tinkering, the bacterium can now move its flagellum in a roughly helical pattern, the shape adopted by a single flagellum in a real bacterium. The flagellum stays elongated behind the bacterium. The most distal (50^{th}) filament segment stays within a range of 7.5 to 9.9 μm from the motor, where 10 μm is the maximum. Viewed from the side it takes on a roughly sine wave appearance. The bacterium moves by comparing this distance with a set of two thresholds. If the distance is greater than the upper threshold, a force is applied pushing it in a forward direction. This simulates the *run* behavior.

Figure 8 shows the entire organism. The filament segments have organized themselves (by way of the physics) into a helical shape, shown in the figure as a sine wave. The bacterium is undergoing a *run*.

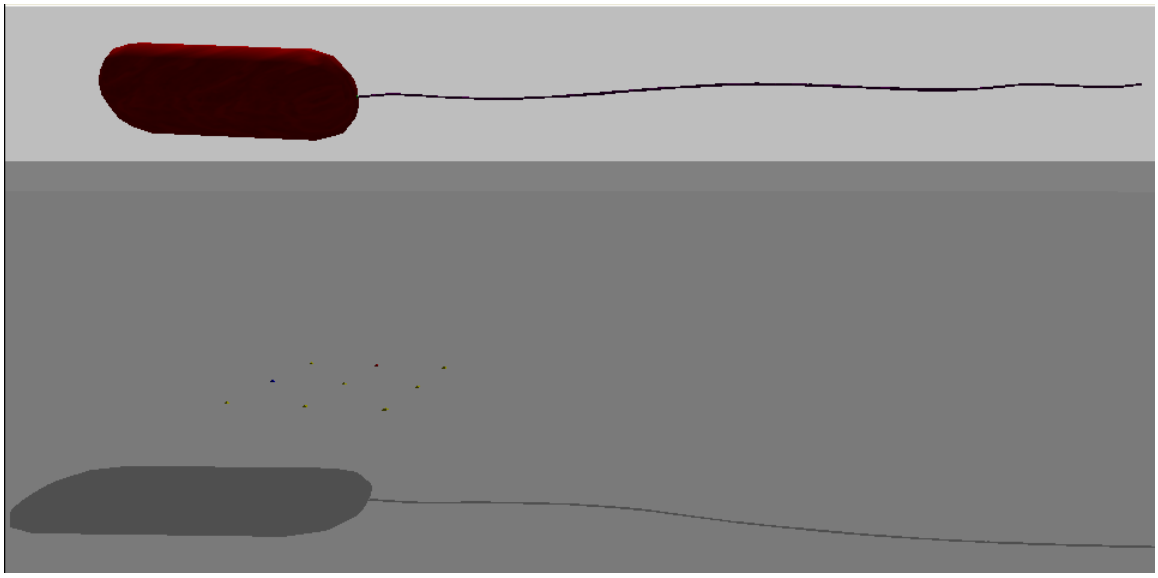

Figure 8: The bacterium undergoing a *run*.

If the distance is less than the lower threshold, a force is applied to rotate it toward the side. This simulates a *tumble* behavior. Figure 9 shows a different shape for the filament segments as it tumbles. The shadows in the lower part of each figure show the filament from a different angle. The combination of views from two angles gives an idea of the 3D shape.

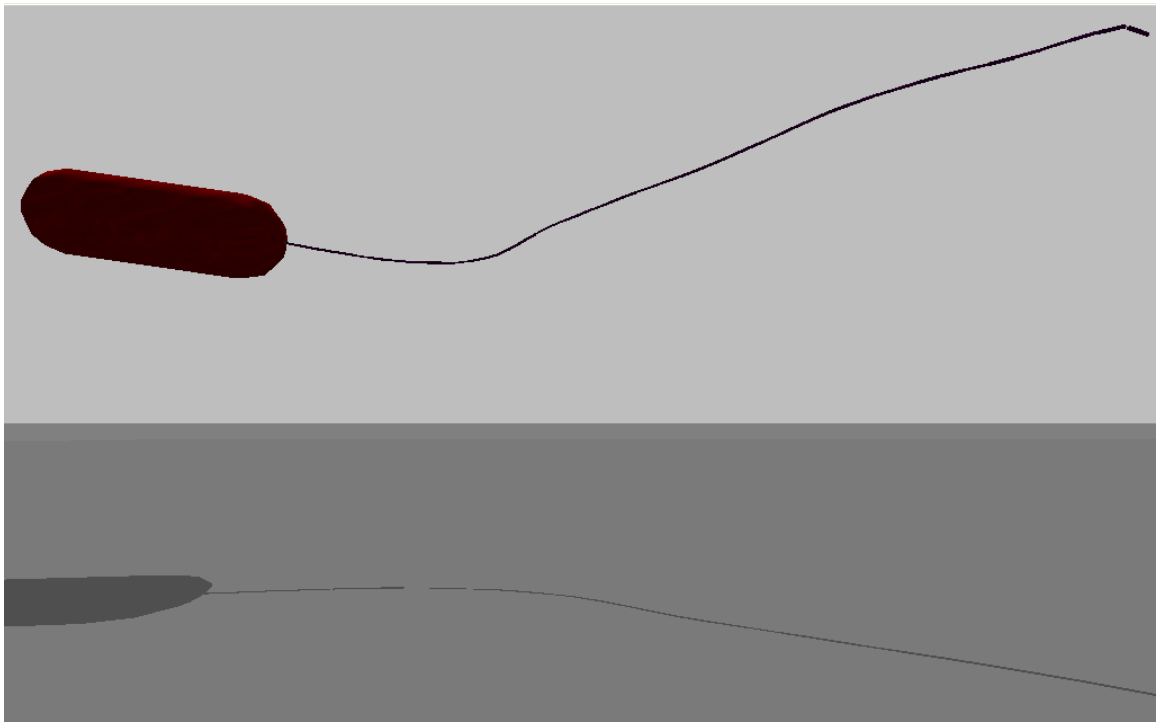

Figure 9: The bacterium undergoing a *tumble*.

Thus, when the flagellum most resembles the corkscrew shape observed in real bacteria as they propel themselves rapidly forward during a run, the ODE simulated bacterium moves

forward. When the flagellum takes on another shape, the simulated bacterium tends to tumble, changing its direction. The ODE physics engine automatically rotates the bacterium in a direction opposite to the rotation of the motor, as is observed with real bacteria, and as is predicted from Newton's third law of motion concerning equal and opposite action and reaction.

Figure 10 shows the path traced out by the center of the e coli cell body during one simulation trial. The abrupt turns are caused by the pseudo-tumbling described above. The x, y, z coordinates are saved at each n^{th} time step. These could be analyzed in more detail by loading them into a mathematical analysis package such as matlab.

Figure 10: Bacterium trajectory during one trial.

This is not a very satisfactory way of getting bacterial movement. Yes, there is a causal relationship between movement of the flagellum and movement of the whole bacterium. But, no, it does not make direct use of the physics; it doesn't just happen because of the way the physics works. The problem is that ODE does not include a hydrodynamics capability, and implementing this is a project in itself.

I have included a continuous-time recurrent neural network (CTRNN) as part of the simulation. It mediates the effect of the chemical environment on the amount of torque applied to the motor. Figure 11 shows the bacterium beginning to chemotactically move through a gradient of the aspartate food chemical. The level of aspartate increases as the bacterium moves to the right in the diagram, shown by the darkening color of the spheres.

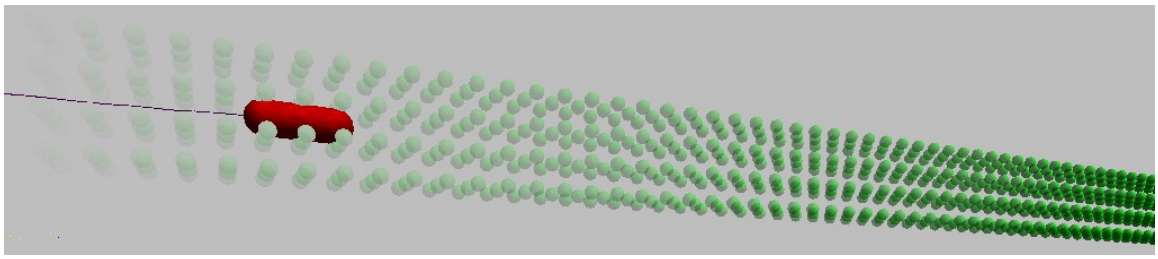

Figure 11: Bacterium moves up an aspartate gradient.

The CTRNN, shown in Figure 12, is very simple. It contains one input neuron and one output neuron. The input senses the single chemical aspartate in the environment. The output produces a torque that can be applied to the single protein motor currently in the simulation.

A CTRNN with two neurons has 10 parameters (Beer, 1995), 1 input I, 2 initial output values y, 2 constant time decay values tau, 2 constant bias values theta, and three weights (2 self-weights and 1 inter-neuron weight). All ten parameters are given specific initial and constant values, as can be seen in the C code (ctrnn.cpp) included in the appendix.

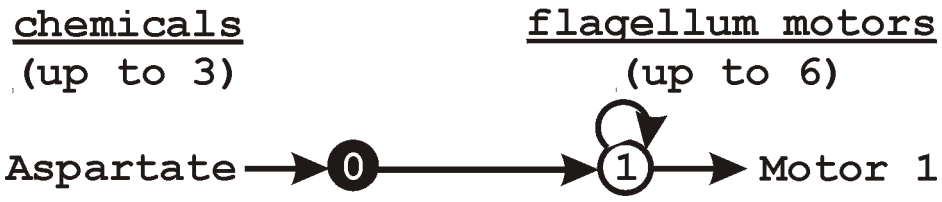

Figure 12: CTRNN with one input and one output.

The CTRNN provides a way for the bacterium to adapt, on a second by second basis, to changing conditions in the environment.

Discussion

Figure 13 shows many of the most important aspects of bacterial chemotaxis. I will briefly describe this architectural diagram, and then discuss and compare the NetLogo and ODE chemotaxis simulations by referring to this architectural plan.

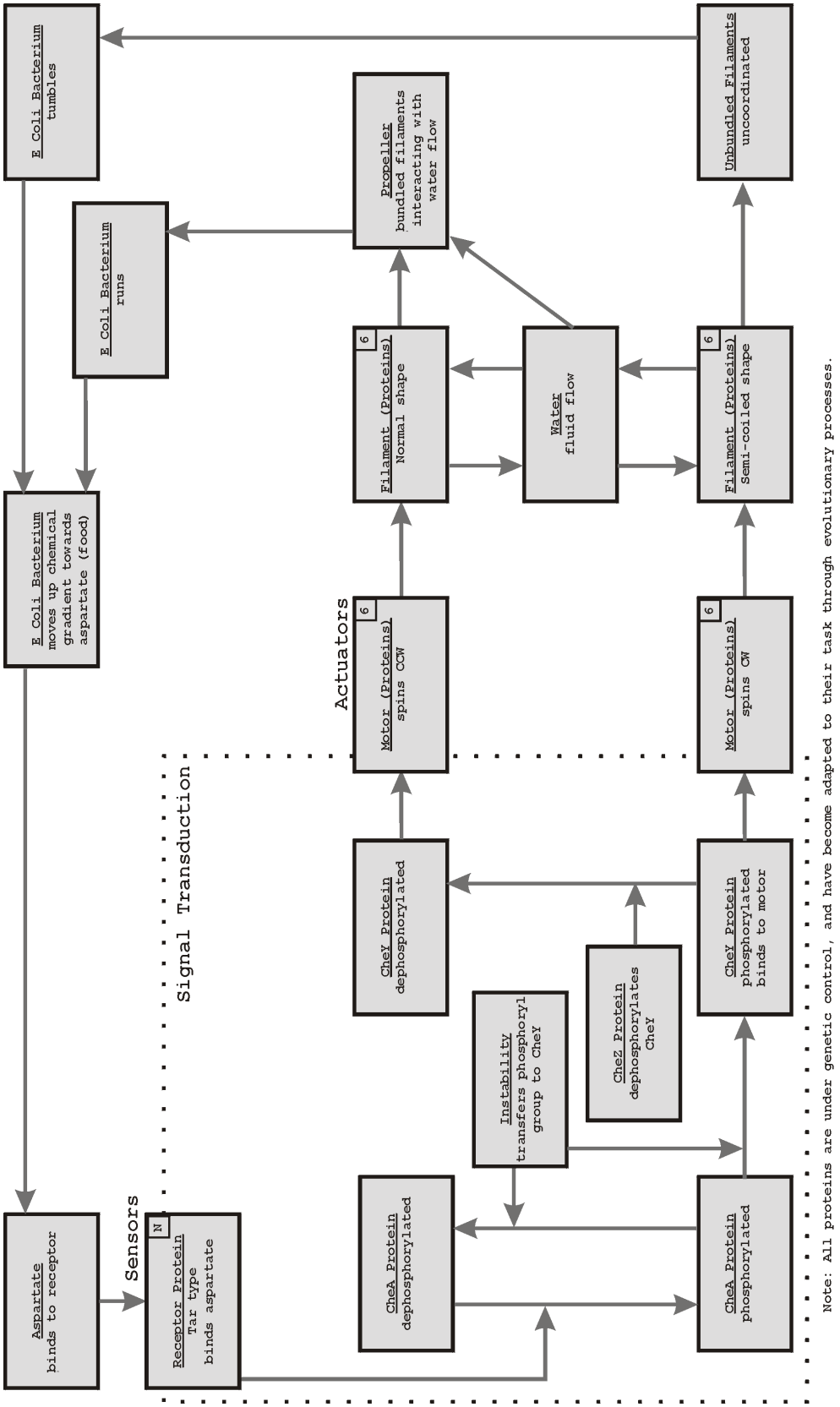

Figure 13: Architectural plan showing major cycles involved in bacterial chemotaxis.

The underlined top line within each box is the name of an object. The arrows are causal connections. The rest of the text within each box is either the state of that object, or a brief description of the action it takes. Some of the boxes have numbers in the top right-hand corner. These indicate a typical number of objects of that type, or N just to indicate some number greater than one. Many of the objects on the diagram are proteins, all of which are under adaptive genetic control. Some of these are part of the signal transduction pathway that adaptively converts sensor signals into actuator (motor) commands. Other proteins function as structural building blocks.

Aspartate is a small food molecule found in the e coli's watery environment. When an aspartate molecule binds to a Tar-type receptor protein (a sensor) embedded within the e coli membrane, it triggers the phosphorylation (chemical addition of a phosphoryl group) to CheA proteins. A phosphorylated CheA protein is very unstable (decays within 10 seconds), and quickly looses its phosphoryl group to a CheY protein, that then moves across the bacterium to bind to the inner surface of one of the six flagellar motors. This causes the motor to reverse its normal behavior, and start to turn in a clockwise direction. The CheZ protein regulates this by dephosphorylating the CheY protein, causing it to unbind from the motor, and causing the motor to resume its default counter-clockwise movement. Each motor turns a filament, which, depending on the direction of rotation, takes on a normal or semi-coiled shape. Normal-shaped filaments interact with the flow of water to produce a propeller. Semi-coiled filaments also interact with water, but produce uncoordinated movements. The propeller moves the e coli bacterium rapidly forward in a run. The unbundled filaments cause a brief tumble. Runs and tumbles work together to move the e coli bacterium up a chemical gradient towards a concentrated area of the aspartate food. The cycle repeats itself indefinitely, and typically lasts a few seconds. Delays within the cycle, induced partly by signal transduction reactions, are part of what gives the bacterium a memory of environmental (aspartate concentration) conditions during the previous seconds. On the other hand, the fast decay of phosphorylated proteins allows it to adapt very quickly (within seconds) to changing environmental conditions (Alberts, 2002).

The NetLogo implementation models the complete cycle, but with very few of the details. The sensor and signal transduction pathway are contained within the few lines of the `behave` function. Signal transduction is represented as a set of probabilities. The motor, filament, water, propeller, and unbundled filaments objects are tersely summarized within the decision to call either the `behaviorRun` or `behaviorTumble` function. The `behaviorRun` function moves the e coli forward one unit (`fd 1`), while `behaviorTumble` changes direction by a random amount.

The ODE implementation also models the complete cycle, but in more detail. It uses a CTRNN to produce the function achieved by the signal transduction pathway in a real bacterium. The output of the CTRNN drives the motor in either the CCW or CW direction. The ODE physics engine then moves the filament segments, which form one of two types of shape or position, elongated or more spread out. This positional shape is made to cause the bacterium to run or tumble, which moves it along.

Future Work

The number of flagella should be increased to six, a typical number found in real *e coli*. This introduces various issues not found when there is only one flagellum exactly in the center of the tail end of the bacterium. An additional component, a flexible hook, is required to convert the sideways spin of the motors(around the x axis) into rotation of the flagella (about the y axis). Figure 14 shows approximately what a bacterium with six flagellum will look like, based on some early trials using ODE.

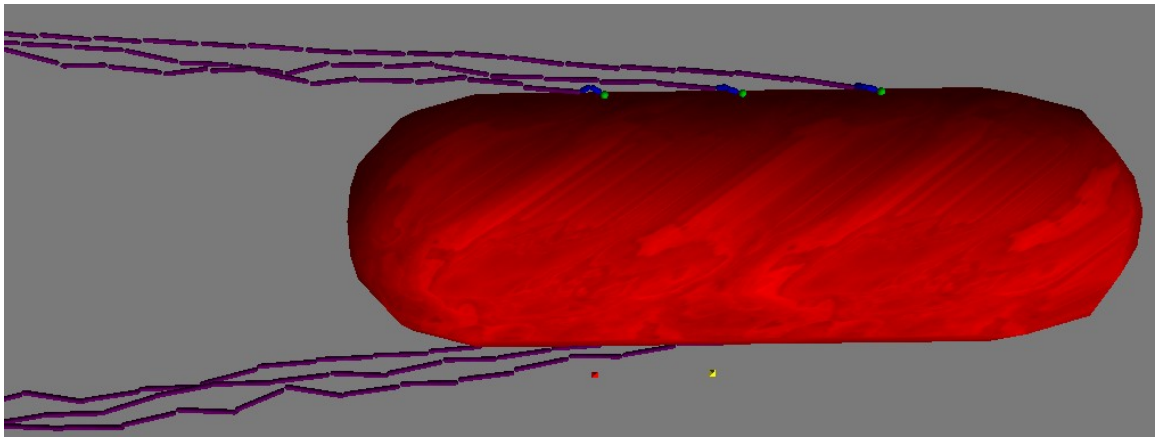

Figure 14: e coli bacterium with six filaments projecting from its side.

Figure 15 shows a close up of the motor (sphere), hook (two short cylinders), and first few filaments (elongated cylinders) would look like. In ODE, the filaments need to be initially placed some distance away from each other, to prevent their becoming fused.

The bacterial cell body, motors, hooks, and filaments, all rigid bodies within ODE, are connected to each other using simple joints. They should be reimplemented as springs, a more advanced form of joint.

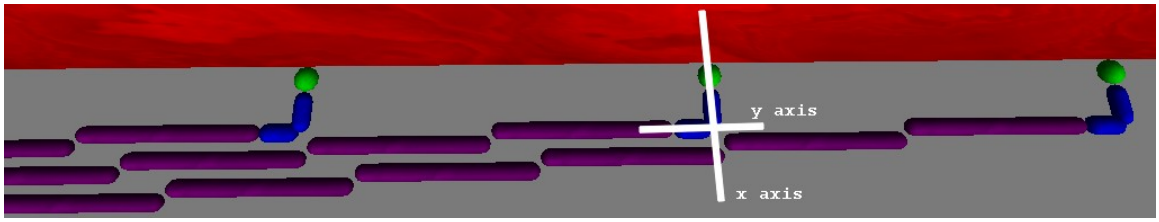

Figure 15: Close up of e coli bacterium with six flagella (three shown here).

With the resulting more significant amount of structure, it will be more important to make use of a genetic algorithm to optimize the CTRNN or CellAK enzyme kinetic parameters. The CTRNN would be qualitatively the same as at present, but would have six output neurons, one to control each motor. It could also be extended to respond to the levels of three chemicals (three input neurons).

The physics needs to be enhanced by adding a hydrodynamics capability. Various researchers have explored the simulation of fish in a water environment. M. G. Triantafyllou (Clark, 2001; Triantafyllou, 1995) have investigated the close inter-relationship between bluefish tuna and water. Tuna are able to swim considerably faster than scientists originally thought possible. They do this by adaptively moving their tails to create additional vortices that they can then make use of. This is similar to the autopoietic interaction between grouped bacterial flagella and water described in this paper. Terzopoulos and colleagues (Terzopoulos et al., 1994) have modeled the 3D form and appearance of fish as "full-featured autonomous agents with sensory, motor, and control systems" (Pfeiffer and Scheier, 1999, p.263), and make use of simulated hydrodynamics. The issue is complicated by the fact that bacteria experience water in a different way because of their small size. To them, water has more the consistency of molasses (Purcell, 1997), a phenomenon referred to as hydrodynamics with a low Reynolds number.

The main future focus will be to integrate the present work with previous work on modeling of stationary cells and other biological systems. CellAK (Webb and White, 2003; Webb and White, 2004b) is an architecture for developing top-down (analytical) simulations of natural systems based on the object-oriented (OO) concepts of containment hierarchies and inheritance hierarchies. This approach has been shown to be capable of modeling an autopoietic system (Webb and White, 2004a). Follow-on work (Webb, 2003) has confirmed that this architecture can be extended to allow bottom-up (synthetic) construction of such a system. This construction could be done using adaptive evolutionary approaches, operating on the internal tree structure of the simulation. The entire ODE-based simulation described in the present paper could be implemented using this approach. The CTRNN would be replaced by protein networks, the basic agents of behavior in CellAK. These are based on an artificial

chemistry. This work could include a comparison of CTRNNs and proteins (enzymes, transport proteins, signal transduction proteins, etc.). Interestingly, Randall Beer, a major proponent of the use of CTRNNs, is also interested in chemotaxis, and the role of proteins in cell behavior (Drennan and Beer, 2004). Where a CTRNN uses weights, enzymes use kinetic constants, both of which are evolvable parameters. In addition to their control function, the proteins could also be used to adaptively construct the motors, hooks and filaments as shown in Figure 13.

Conclusion

The work described in this paper has explored the overall architecture and certain key difficult (risky) areas, in preparation for a more comprehensive simulation of bacterial chemotaxis. I will conclude by returning to the two practical research questions posed in the motivation section. The architecture presented in is a reasonable plan for going forward. It was successfully implemented by both the NetLogo and C++/ODE/CTRNN models. ODE can be used to simulate the physics in the simulation, but additional code needs to be written to deal with its limitations, especially its lack of support for hydrodynamics. It is hoped that integration of the work described here with previous work on modeling of stationary cells (CellAK), will lead to additional progress towards the long-term goal of building an adaptive system in which both the behavior *and* the structure are under evolutionary control.

References

Alberts, B. (2002). *Molecular Biology of the Cell, 4th ed*. New York: Garland Science. [bacterial chemotaxis, p.890-892.

Becker, W. (1996). *The World of the Cell*. Menlo Park, CA: Benjamin/Cummings. [p. 706-709 has an introductory section on The Bacterial Flagellum]

Beer, R. (1995). On the Dynamics of Small Continuous-Time Recurrent Neural Networks. *Adaptive Behavior 3*, 469-509.

Beer, R. (1996). *Toward the evolution of dynamical neural networks for minimally cognitive behavior*.

Beer, R. (2004). Autopoiesis and Cognition in the Game of Life. To appear in *Artificial Life*.

Bray, D. (1995). *Protein molecules as computational elements in living cells*. Nature 376, 307-312.

Bray, D. (2001). *Cell Movements – From Molecules to Motility, 2nd ed*. New York: Garland.

Bray, D. (2003). *Molecular Networks: The Top-Down View*. Science 301, 1864-1865.

Cappuccinelli, P. (1980). *Motility of Living Cells*. London: Chapman and Hall.

Chiel, H.; and Beer, R. (1997). The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *TINS 20*, 553-557.

Clark, A. (2001). *Mindware*. New York: Oxford U.P. [talented tuna, p.143]

Drennan, B; and Beer, R. (2004). A Model for Exploring Genetic Control of Artificial Amoebae. To appear in *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems*. Boston, MA.

Kim, M.; Kim, M.; Bird, J.; Park, J.; Powers, T.; and Breuer, K. (2003). Macro-scale PIV Experiments on Bacterial Flagellar Bundling. *5th International Symposium on Particle Image Velocimetry*, Busan, Korea, Sept. 22-24, 2003.

Maturana, H.; and Varela, F. (1980). *Autopoiesis and Cognition*. Dordrecht: D. Reidel.

Matzke, N. (2003). *Evolution in (Brownian) space: a model for the origin of the bacterial flagellum*. http://Evolution of the bacterial flagellum.htm

Mitchell, M. (1997). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.

NetLogo itself: Wilensky, U. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.

Pfeifer, R.; and Scheier, C. (1999). *Understanding Intelligence*. Cambridge, MA: MIT Press. [fish, hydrodynamics, p.263-5]

Powers, T. (2002). Role of body rotation in bacterial flagellar bundling. *Physical Review E 65*, 040903-1 -4.

Purcell, E. (1997). The efficiency of propulsion by a rotating flagellum. *Proc. Natl. Acad. Sci. USA 94*, 11307-11311.

Slocum, A.; Downey, D; and Beer, R. (2000). *Further Experiments in the Evolution of Minimally Cognitive Behavior: From Perceiving Affordances to Selective Attention*. [especially section on Short-Term Memory]

Smith, R. (2003). *The Open Dynamics Engine (ODE).*

Terzopoulos, D.; Tu, X.; and Grzesczuk, R. (1994). Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life 1*, 327-351.

Triantafyllou, M.; and Triantafyllou, G. (1995). An efficient swimming machine. *Scientific American 272*, 64-71.

Turner, L.; Ryu, W.; and Berg, H. (2000). Real-Time Imaging of Fluorescent Flagellar Filaments. *Journal of Bacteriology 182*, 2793-2801.

Webb, K. (2003). *Programming Techniques Project*. Brighton: University of Sussex.

Webb, K.; and White, T. (2003). UML as a Cell and Biochemistry Modeling Language. *Carleton University Cognitive Science Technical Report 2003-05*. http://www.carleton.ca/iis/TechReports . Accepted for publication in *BioSystems* journal.

Webb, K.; and White, T. (2004a). Combining Analysis and Synthesis in a Model of a Biological Cell. *SAC '04*, March 14-17, 21004, Nicosia, Cyprus.

Webb, K.; and White, T. (2004b). Cell Modeling using Agent-based Formalisms. Accepted for *AAMAS 2004*.