A Bacterial Algorithm for Surface Mapping Using a Markov Modulated Markov Chain Model of Bacterial Chemotaxis

by

Alaa Amin Kharbouch

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2006

Author ....
Department of Electrical Engineering and Computer Science
ary 30, 2006

Certified by........
Alan V. Oppenheim
Ford Professor of Engineering
Department of Electrical Engineering and Computer Science
isor

Certified by............
Maya R. Said
:tronics
)ervisor

Accepted by...
Smith
Chairman, Department Committee on Graduate Students
Department of Electrical Engineering and Computer Science
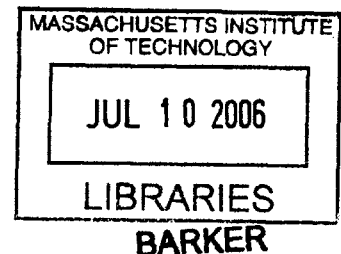
A Bacterial Algorithm for Surface Mapping Using a Markov Modulated Markov Chain Model of Bacterial Chemotaxis

by

Alaa Amin Kharbouch

Submitted to the Department of Electrical Engineering and Computer Science
on January 30, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Bacterial chemotaxis is the locomotory response of bacteria to chemical stimuli. *E. coli* movement can be described as a biased random walk, and it is known that the general biological or evolutionary function is to increase exposure to some substances and reduce exposure to others. In this thesis we introduce an algorithm for surface mapping, which tracks the motion of a bacteria-like software agent (based on a low-level model of the biochemical network responsible for chemotaxis) on a surface or objective function. Towards that end, a discrete Markov modulated Markov chains model of the chemotaxis pathway is described and used. Results from simulations using one- and two-dimensional test surfaces show that the software agents, referred to as *bacterial agents*, and the surface mapping algorithm can produce an estimate which shares some broad characteristics with the surface and uncovers some features of it. We also demonstrate that the bacterial agent, when given the ability to reduce the value of the surface at locations it visits (analogous to consuming a substance on a concentration surface), is more effective in reducing the surface integral within a certain period of time when compared to a bacterial agent lacking the ability to sense surface information or respond to it.

Thesis Supervisor: Alan V. Oppenheim
Title: Ford Professor of Engineering
Department of Electrical Engineering and Computer Science

Thesis Supervisor: Maya R. Said
Title: Visiting Scientist, Research Laboratory of Electronics

Acknowledgments

I would like to thank my advisor, Al Oppenheim, for his great insight and empowering guidance. Al, I am truly grateful for your enthusiasm and support throughout this rewarding experience, and I am more confident going on any intellectual adventures with you in ground control. I would also like to thank my co-advisor, Maya Said, for her mentorship and encouragement. This work would not have been possible without her efforts and expertise.

To the members of the digital signal processing group, past and present: Alecia, Charles, Charlie, Daniel, Dennis, Joe, Matt, Melanie, and Tom, thanks for making the DSPG a comfortable home and a stimulating environment. Special thanks go to Petros, for putting up with all my questions and for the many pieces of great advice, and Sourav for his help and the many great discussions. I'd also like to thank Joonsung for being a great officemate, and Ross for the good conversation and the many laughs while we shared the terminal room. Eric, thank you for always being a friendly face, the go-to man in the group, and a frequent lifesaver.
I'd like to thank Zahi, my officemate, onetime roommate, and longtime friend and confidant. Thanks for the countless times you came to the rescue.

I owe a debt of gratitude to the great faculty at the Georgia Institute of Technology, who have inspired me with their style, enthusiasm and experience. I'd especially like to thank Prof. James McClellan, Prof. Jay Alberts and Prof. Anthony Yezzi for the influence they've had on my career.

I am fortunate to have a lot of family and many relatives who were a tremendous source of love and support. Thank you all. To my friends Nabil, Pranav, Hanan, Rujuta and Zaid, thanks for all the good times and for making the more difficult times easier. I thank Deema for her refreshing perspective, caring spirit, and the positive impact she's had on my life. Rami, thanks for being a big part of my broader education, and always being there.

To my family, thank you for your love and emotional support. I'd like to thank my little brothers Ayyad, Adel and Asem. I'm proud of you all. To my parents, Amin and Maha, thank you for your patience and all the sacrifices you made. You taught me many lessons in selflessness.

Contents

List of Figures

11

List of Tables

Chapter 1

Introduction

Many algorithms draw their inspiration from phenomena in nature. For example, genetic programs and genetic algorithms borrow ideas from evolution and Mendelian genetics such as selection, reproduction and mutation to produce programs and solutions to problems [13, 27]. Simulated annealing is a local search optimization method employing monte carlo techniques, based on an analogy with the thermodynamic process by which cooling metals find a low energy state [12]. The concept of self-similarity and fractals, which occur in several contexts in nature and man-made systems, serve as a basis for novel approaches and directions in signal processing and communications [28]. Artificial neural networks use interconnected processing elements to form an information processing and storage system, often with the capacity for adaptive learning, much like neurons in a biological nervous system [29]. Other examples include work with solitons [25] and quantum signal processing (QSP) [8]. In a similar spirit, this thesis will explore the possibility of exploiting the talent of $E.\,coli$ bacteria at seeking out higher or lower concentrations of certain substances for the problem of surface mapping. The locomotory response of bacteria to chemical stimuli is referred to as $chemotaxis$ [19, 10].

15

1.1 Surface Mapping Based on Chemotaxis

We are often interested in uncovering features of an unknown function of one or more independent variables, or finding an approximation to it. However, in many problems in science and engineering the observations we can make of the functions are limited [20, 16]. These constraints on the possible sampling of the function or surface can arise from the nature of the application or from the physical limitations involved in making measurements in a real-world application. In many machine learning applications an algorithm will attempt to learn a function from examples, i.e. from knowledge of the values the function takes at a limited number of points in the function domain. However, we are interested in the cases where such information about the function is not globally available, and we do not have direct access to the true value of the function at any point. This shares some similarities with the premise behind many problems and applications that use hidden Markov models, where a Markov chain model is assumed as an underlying stochastic process which affects the observable output, but whose states are not directly observable [21]. Another problem in which this constraint applies is the estimation of a probability distribution function of a random variable. The value of the distribution function cannot be observed, but rather the result of probabilistic experiments influenced by the probability distribution function are available, as instances of random variables with that distribution, or measurements of a quantity modeled as such, are instead used to estimate it. The surface mapping algorithm proposed in this thesis tracks a software agent, which simulates a model of the protein network that controls chemotaxis, as it explores the domain space of the function of interest and samples it locally. This is formulated by analogy to the way an *E.coli* cell travels in a three-dimensional volume where the concentration varies as a function of the spatial coordinates. Since an *E.coli* cell uses the concentration values it encounters to bias its swimming behavior, one may be able to find an approximation to the concentration surface based on the observed trajectory of one or more bacteria on that surface. The algorithm described in this thesis monitors the behavior of this agent, which is influenced by the surface it is

navigating, to form an estimate of the function.

Some optimization algorithms such as Ant Colony Optimization and Particle Swarm Optimization are based on analogies with the behavior of certain organisms [6]. Optimization algorithms often involve two competing components: exploration and exploitation [15, 22]. Exploitation is the use of the information encountered so far in the search process to guide the algorithm to solutions with a higher or lower value of the objective function as desired, whereas exploration emphasizes a choice of subsequent trial points that helps explore a wider area of the search space, which is crucial to the discovery of a global optimum in problems with a multitude of local extrema. For example, gradient search methods shift the balance completely towards exploitation, as the direction of the evaluated gradient of a surface at one point completely determines the choice for the next trial solution. The mechanism by which *E.coli* search a three-dimensional environment using information about changing concentration of certain substances potentially suggests a mechanism for algorithmically exploring the domain of an objective function. While optimization algorithms attempt to find a global maximum or minimum of an objective function that can be evaluated at any point in the search space, we will employ a chemotactic strategy to map or find an estimate of an unknown scalar-valued function (i.e. a function that can be sampled by the surface mapping agent, but whose value at any point in its domain cannot be used directly in the formation of the estimate). Although the chemotaxis system allows the bacterium to fare better in an environment with a varying abundance of nutrients and beneficial substances, its purpose is not necessarily to enable the bacterium to quickly find a location where there is an absolute maximal exposure to the desired substance.

1.2 Bacterial Chemotaxis

The biological network in *E.coli* bacteria that controls chemotaxis is one of the most well-studied signaling pathways in cell biology [11]. Bacteria move in their environment in an informed way, seeking out higher concentrations of attractants and lower

concentrations of repellents [5]. While concentrations of a certain substance may vary spatially, the bacterium can only sample the concentration locally at its current position. As the cell moves in a certain direction, spatial concentration gradients are perceived as temporal gradients [14].

The swimming behavior of an *E.coli* cell is controlled mainly by its helical tails, or flagella, which can turn clockwise or counterclockwise. The bacterium alternates between a run mode and a tumble mode. A run is a period of smooth swimming, which results when the flagellar motors turn counterclockwise and propel the bacterium forward. Runs are interrupted by tumbles, which are random re-orientations of the cell direction with little or no displacement that occur when the flagellar motors turn clockwise [26]. When looking at a typical path an *E.coli* cell takes, approximately straight lines of motion would represent runs, and the changes in direction between connected lines represent the effect of tumbles. The movement of an *E.coli* cell in its environment can be modeled as a biased random walk [1], since the cell compares external environmental conditions at different time instances and adjusts its swimming behavior accordingly. If an increasing gradient is sensed, tumbling is suppressed and the cell is more likely to move further in that favorable direction. If decreasing attractant is sensed, tumbling is favored and the cell will tend to turn away from that direction.

After a prolonged absence of attractant, the swimming of an *E.coli* cell can be modeled as a random walk, and the run and tumble durations will be (approximately) exponentially distributed with a mean λ_r and λ_t respectively. One can think of the binary state switching between runs and tumbles as a two-state continuous-time Markov chain [3]. The flagellar motors can be in an anticlockwise (run) state or a clockwise (tumble) state, with λ_r or λ_t as a constant probability per unit time of a transition, or average switching frequency. Assuming this simplified model, in the general case with a changing input concentration, these parameters are time-varying and the chemotaxis network effectively has the ability to modulate these instantaneous average switching frequencies between the states or, equivalently, the expected time until the next transition out of the current state. If the bacterium is exposed to the

same concentration for a sufficiently long period of time, its internal chemical states reach a steady state and consequently the average switching-frequency parameters controlling its swimming behavior also reach a steady value. If attractant is suddenly added and the concentration is increased, the cell will respond to the stimulus and swimming behavior will change, most likely resulting in longer runs. However, if the new perceived attractant concentration persists and remains constant for an extended period of time, the swimming behavior of the cell will return back to its pre-stimulus condition. This property of the chemotaxis network is called *adaptation*, which means that the steady state swimming behavior does not vary significantly with the absolute value of the steady state input concentration. The bacterium therefore responds to *changes* in the input stimulus, and the absolute value of the attractant concentration does not play a major role in the observed flagellar motor response.

1.3 Outline of the Thesis

Chapter 2 presents some background on the biological mechanism responsible for bacterial chemotaxis. That chapter also begins the specification of the software agent that plays a central role in our surface mapping algorithm, by introducing a model of the protein network that is responsible for the chemotactic response in *E.coli*. The model was simulated to verify that certain aspects of the simulated response to an input concentration signal that are crucial to the surface mapping algorithm match those from experimental results and findings about *E.coli* cells. In Chapter 3, we describe how the surface mapping agent moves to different locations on a surface, and present the major steps of the surface mapping algorithm, which we refer to as the Bacterial Algorithm for Surface Mapping (BASM). Chapter 4 contains further specifications for and results from various BASM simulations using one- and two-dimensional surfaces. We also demonstrate that the bacteria-like agent, when given the ability to reduce the value of the surface at locations it visits (analogous to consuming a substance on a concentration surface), is more likely to reduce the surface integral to a lower value within a certain period of time when compared to a bacterial

19

agent lacking the ability to sense surface information or respond to it.

Chapter 2

A Biological Model of Bacterial Chemotaxis

The proposed surface mapping algorithm, inspired by the swimming response of *E. coli* cells in a chemical environment, uses an agent-based approach to form an estimate of a surface. The behavior of the surface mapping agent is largely described by a biological model of the protein network underlying chemotaxis. In this chapter we review the biological background relating to the inner workings of the chemotaxis system, as reported in the literature. We then present the model of the chemotaxis biochemical network, developed in [23] and based on the Markov modulated Markov chains framework, which is integral to our software agent and consequently our surface mapping algorithm. Some liberty is taken to make modifications to the parameters of the model and diverge from the biological basis to produce more practical or interesting behavior in the surface mapping context.

2.1 Chemotaxis Biological Network

The biological network controlling the chemotaxis response to aspartate, a type of attractant, consists of a small set of proteins, namely Tar, CheA, CheB, CheR, CheY, CheW and CheZ [4]. Figure 2-1 illustrates the components of the chemotaxis network and the interactions among them.
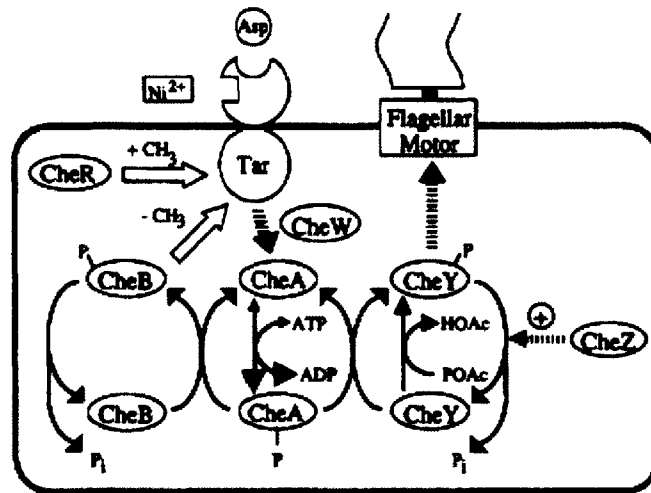
Figure 2-1: Illustration of the components of the chemotaxis network, reproduced from Bray *et al* [4].

The following three types of reactions are relevant to the operation of the chemotaxis network:

1. **Phosphorylation reactions:** A phosphorylated protein is one with a phosphoryl chemical group attached to it. A phosphorylated CheA, CheB or CheY protein is considered to be activated. An active protein is able to perform certain functions an inactive protein cannot, including the activation of another protein. The phosphorylated forms of these proteins are denoted by CheAp, CheBp and CheYp.

2. **Binding reactions:** The Tar protein is a type of receptor. Information about the external concentration of the attractant is conveyed through the binding of aspartate molecules, also called the *ligand* molecules, to Tar. Binding to a ligand leads to a conformational (structural) change of the receptor, affecting other reactions involving the receptor.

3. **Methylation reactions:** A number of methyl groups (CH_3) can be attached to the Tar receptor. Methylation of the receptor effectively increases the rate of CheA phosphorylation. CheR adds methyl groups, while the active form of CheB (CheBp) removes methyl groups from the receptor.

22

2.1.1 The Excitation Pathway

The Tar receptor, CheA and CheW form what is referred to as the *receptor complex*. The Tar receptor, with the help of CheW, affects the phosphorylation activity of CheA [26]. The receptor complex can be in an active state or an inactive state, and the CheA protein is able to phosphorylate only in an active receptor complex. The probability that a certain receptor complex will be in the active state depends on the methylation state and whether or not it is bound to a ligand (aspartate in our case). Ligand binding decreases the probability that the receptor complex is active, while a higher number of attached methyl groups increases the activity probability.

A series of phosphorylation reactions carries the effect of the external ligand concentration sensed at the receptor to the flagellar motor [4]. The flagellar motor is made up of about 40 proteins [17]. In the absence of active CheY, the default rotation direction of the motor is counterclockwise. The phosphorylated form of CheA can activate both CheB and CheY. As more CheYp proteins bind to the flagellar motor, it is more likely to turn clockwise, so active CheY promote tumbling. CheBp and CheYp can de-phosphorylate (i.e. lose their phosphoryl group), and CheZ catalyzes the removal of phosphoryl groups from CheYp.

To illustrate the immediate response to a change in chemical stimulant, consider an example scenario: As a bacterium runs in a certain direction, a sudden decrease in input attractant concentration would decrease the number of receptors bound to an aspartate molecule. This increases the rate at which CheA proteins phosphorylate, which then transfer these added phosphoryl groups to CheY proteins. More CheYp molecules then bind to the flagellar motors, inducing more tumbling, therefore resulting in runs of shorter duration (on average) in that direction [4].

2.1.2 The Adaptation Pathway

Adaptation in the chemotaxis pathway occurs through the methylation and de-methylation of the receptor by CheR and active CheB, respectively. Since CheAp can pass its phosphoryl group to CheB, an increase in active CheA will lead to an increase in active

CheB. CheBp can de-methylate the receptor, therefore leading to a reduced activity of the receptor complexes on average. The system thus relaxes back to a state with a lower number of active CheA, CheY and CheB.

The receptor methylation state modification reactions occur on a slower timescale than the activation or phosphorylation reactions, and therefore allow for a transient response to a change in stimulus before the system relaxes or adapts back to its steady state activity.

2.2 The Markov Modulated Markov Chains Model of Chemotaxis

The Markov Modulated Markov Chains (3MC) framework is a new approach to modeling biological signaling networks [23]. The model consists of v nodes, where each node represents a k-state discrete-time Markov chain. A Markov modulated Markov chains network is defined as one where the state of one chain X_r at time index n, denoted by $X_r[n]$, can affect the transition probability from state i to j of another chain X_p at time n, denoted by $p_{ij}^{X_p}[n]$. The interaction between the two nodes in the network is represented by:

$$p_{ij}^{X_p}[n] = f(X_r[n]) \tag{2.1}$$

where $f(\cdot)$ is referred to as the *modulating function*. In this framework, each chain represents a component of a biochemical network. The transition probabilities describe the effect of chemical reactions and other interactions between the components, and incorporate any relevant biochemical parameters [24]. Since the model uses a probabilistic description, it offers the advantage of being able to compute state occupancy probabilities for the nodes that are equivalent to results produced from a deterministic model of the system (such as those obtained from a model using non-random differential equations), and perform stochastic simulations that describe the stochastic nature of the underlying chemical or biological processes.

To perform a stochastic simulation of a 3MC network such as the one described

above, a group of identical Markov chains corresponding to the node X_p are independently simulated. At each time instance a chain from this group is paired with a chain in a group corresponding to the node X_r from a concurrent simulation. The state of the selected X_r chain determines the transition probability to be used for the X_p chain according to the modulation relation in equation (2.1). A random number is generated for each chain and compared against the time-varying transition probabilities to decide if any state transitions occur. The state evolution of a Markov chain in the X_p or X_r group following this procedure defines a random process $X_p[n]$ or $X_r[n]$ respectively. Each simulation of a Markov chain constitutes a realization of the random process, and each realization of $X_p[n]$ is dependent on the realizations of $X_r[n]$.

In this thesis we use the *apriori* Markov Modulated Markov Chains (a3MC) model approximation of the chemotaxis network from [23]. With this approximation, equation (2.1) is replaced by

$$p_{ij}^{X_p}[n] = f_a(Pr(X_r[n] = m)) \tag{2.2}$$

where $Pr(X_r[n] = m)$ is the probability that chain X_r is in the mth state at time index n. The a3MC modulating function is derived from the state dependency described by the 3MC modulating function $f(\cdot)$, and is used to compute the transition probabilities in an a3MC stochastic simulation using state occupancy probabilities. In more general cases where more than one node interacts with X_p, $f_a(\cdot)$ can be a function of several state occupancy probabilities [23]. The essential difference in the a3MC case is that the state evolution in a stochastic simulation of the X_p chain is now independent of any realization of the random process $X_r[n]$. This allows the use of a stochastic simulation with chain X_p or the computation of state probabilities for it, using only the time-evolving state probability vector for the X_r chain. When calculating the

state probability vector

$$\mathbf{p}_{X_p}[n] = \begin{bmatrix} Pr(X_p[n] = 1) \\ Pr(X_p[n] = 2) \\ \vdots \\ Pr(X_p[n] = k) \end{bmatrix}^T, \qquad (2.3)$$

the recursive relation for conventional Markov chains still applies, with the exception that the transition matrix is time-varying. The transition matrix $\mathbf{A}_{X_p}[n]$ is populated by state transition probabilities which could generally be dependent on time-varying state probabilities of other chains in the network, as specified by the a3MC influence relations of the form of equation (2.2). More specifically:

$$\mathbf{p}_{X_p}[n+1] = \mathbf{p}_{X_p}[n]\mathbf{A}_{X_p}[n] \qquad (2.4)$$

and

$$\mathbf{A}_{X_p}[n] = \begin{bmatrix} p_{11}^{X_p}[n] & \cdots & p_{1k}^{X_p}[n] \\ \vdots & \ddots & \vdots \\ p_{k1}^{X_p}[n] & \cdots & p_{kk}^{X_p}[n] \end{bmatrix} \qquad (2.5)$$

A full stochastic simulation of all the components in the chemotaxis network is therefore not necessary if we are interested in a stochastic simulation of one (or more) component(s).

The states and interactions of the a3MC chemotaxis model from [23] are shown in Figure 2-2. Interactions are indicated by the dashed lines and each protein is represented by a Markov chain. The model approximates the Tar receptor as having only two possible methylation states (methylated and unmethylated). We use a time step of $dt = 10^{-3}s$ corresponding to the time interval between successive updates of the Markov chain states. With this value for the time step we ensure that none of the calculated transition probabilities for any of the chains exceed 1 or are less than 0 at any time index. The model assumes that the state of one molecule is independent of all other molecules of the same type. For example, every CheY protein has some

26

probability of being in the active state yp at time n, which does not change with information about the state of other CheY molecules in the system.
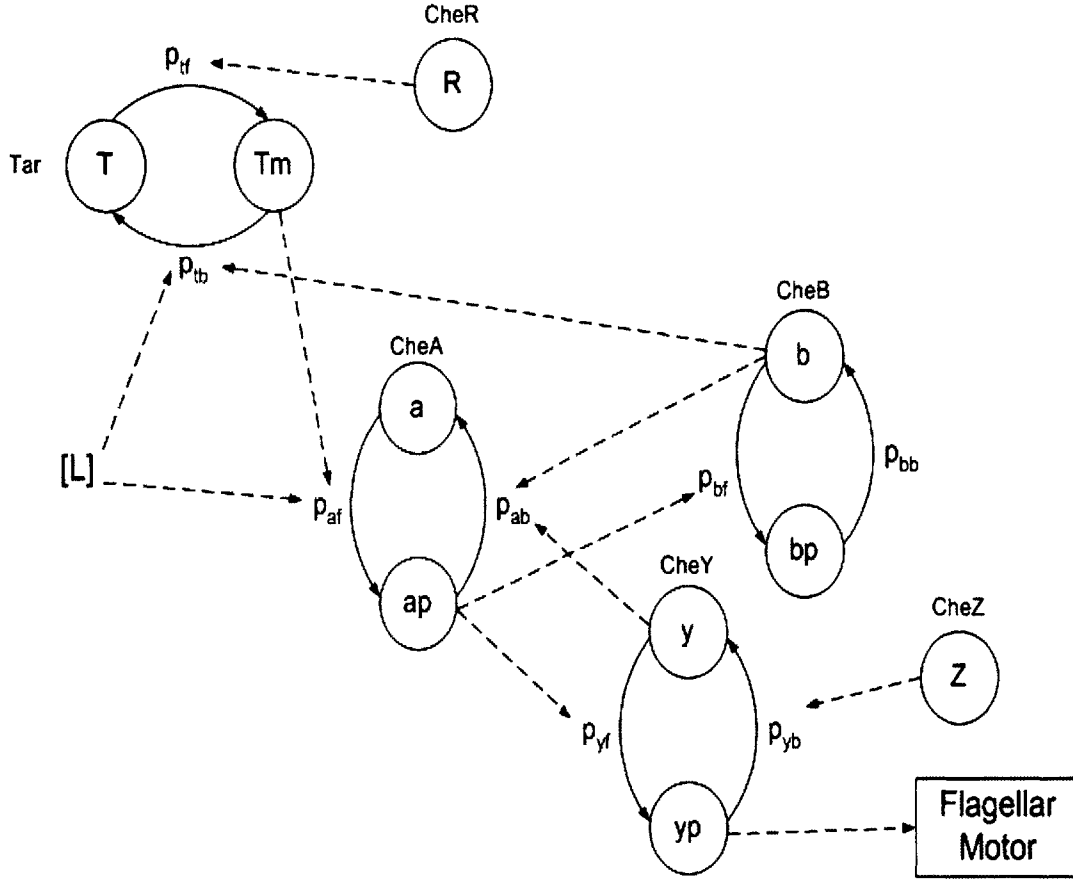
Figure 2-2: States and interactions in the Markov modulated Markov chains model of chemotaxis.

2.2.1 Effect of the External Ligand Concentration Input

In Figure 2-2 the input attractant concentration is denoted by [L]. The attractant concentration determines the fraction, on average, of Tar receptors in an *E. coli* cell that are bound to an aspartate molecule. At steady state, for some concentration L, this can be shown to be [4]:

$$\beta(L) = \frac{L}{L + K_D} \qquad (2.6)$$

27

where K_D is the dissociation constant of the binding and unbinding reactions between the receptor and aspartate. We use a K_D of $1\mu M$ from [4]. In the a3MC model, $\beta(L)$ is interpreted as the probability that a receptor is bound to a ligand (aspartate in this case), and we refer to it as the *receptor occupancy*. From Figure 2-2 we can see that the receptor occupancy affects transition probabilities of the CheA and Tar receptor chains, both of which are part of the receptor complex. The ligand binding and unbinding reactions occur on a much faster timescale than the other reactions described in the model, so we use the approximation that the ligand binding reactions reach steady state equilibrium instantaneously. This is coupled with a zero order hold approximation to the external ligand concentration input, so the receptor occupancy is updated at the beginning of every time step, and is assumed to be constant until the beginning of the next time step. The input concentration at time index n is denoted by L_n. We also assume that the probability that a receptor is ligand bound is independent of the methylation state of the receptor.

2.2.2 Interactions and Transition Probabilities

The parameters of the modulating function $f(\cdot)$ are determined from the chemical rate constants and total intracellular concentrations of the different proteins obtained from Spiro *et al* [26] and Barkai and Leibler [2]. The values used are listed in Tables 2.1 and 2.2. A receptor complex in an *E.coli* cell can be either in an active or inactive state at any point in time. The a3MC model uses the receptor activity probabilities in Table 2.3 that are approximated from the probabilities for all five methylation states in [18]. As the activity probabilities show, the receptor is more likely to be in an active state when it is methylated, and less likely to be in the active state when it is bound to a ligand (an aspartate molecule). Table 2.4 contains the transition probabilities for the a3MC model depicted in Figure 2-2, which are computed using the rate constants, total protein concentrations, receptor activity probabilities and the modulating function. $p_T[n]$ and $p_{Tm}[n]$ denote the probabilities that the Tar chain is in the unmethylated and methylated states respectively. $p_y[n]$ and $p_{yp}[n]$ are the probabilities that the CheY chain is in the unphosphorylated and phosphorylated

state respectively, and similar notation is used for the state occupancy probabilities of the CheA and CheB chains. Only CheA proteins associated with active receptor complexes are able to phosphorylate, so it is not surprising that the transition from the inactive to the phosphorylated state of CheA depends on the receptor occupancy, the receptor methylation state probability and the conditional activity probabilities. The model also uses the assumption from [2] that active CheB can only de-methylate receptors that are in the active conformation. We demonstrate how such constraints are incorporated and the reasoning involved in obtaining the transition probabilities through an example. To find the backwards transition probability for the Tar chain, corresponding to the demethylation of the receptor complex by CheBp, we need to determine the probability that a methylated receptor is active:

$$Pr(\text{active}|\text{methylated})$$

$$=Pr(\text{active}|\text{methylated and ligand bound})Pr(\text{ligand bound})$$

$$+ Pr(\text{active}|\text{methylated and not ligand bound})Pr(\text{not ligand bound})$$

$$=Pr(\text{active}|\text{methylated and ligand bound})\beta(L_n)$$

$$+ Pr(\text{active}|\text{methylated and not ligand bound})(1 - \beta(L_n))$$

$$=\alpha_4\beta(L_n) + \alpha_2(1 - \beta(L_n))$$

which is the term that appears in the expression for $p_{tb}[n]$ in Table 2.4.

$$
\begin{aligned}
k_{tf} &= 7.9992 \times 10^4 Ms^{-1} \\
k_{tlf} &= 7.9992 \times 10^4 \times 1.43 Ms^{-1} \\
k_{tb} &= 7.9992 \times 10^4 Ms^{-1} \\
k_{tlb} &= 7.9992 \times 10^4 Ms^{-1} \\
k_{af} &= 45s^{-1} \\
k_{ab} &= 8 \times 10^5 Ms^{-1} \\
k_{ay} &= 3 \times 10^7 Ms^{-1} \\
k_{bb} &= 0.35s^{-1} \\
k_{yb} &= 20s^{-1}
\end{aligned}
$$

Table 2.1: Reaction rate constants.

$$
\begin{array}{l}
[A_{tot}] = 8\mu M \\
[B_{tot}] = 1.7\mu M \\
[Y_{tot}] = 20\mu M \\
[R_{tot}] = 0.3\mu M
\end{array}
$$

Table 2.2: Total intracellular concentrations used in the a3MC model.

Receptor complex state	Activity probability
Non-methylated and not ligand-bound	$\alpha_1 = 0.07$
Methylated and not ligand-bound	$\alpha_2 = 0.88$
Non-methylated and ligand-bound	$\alpha_3 = 0$
Methylated and ligand-bound	$\alpha_4 = 0.74$

Table 2.3: Receptor activity probabilities.

$$
\begin{aligned}
p_{tf}[n] &= \left(k_{tlf}\beta(L_n) + k_{tf}(1 - \beta(L_n))\right)[R_{tot}]dt \\
p_{tb}[n] &= \left(\alpha_4\beta(L_n) + \alpha_2(1 - \beta(L_n))\right)[B_{tot}]p_{bp}[n]k_{tb}dt \\
p_{af}[n] &= \left(\alpha_4 p_{Tm}[n]\beta(L_n) + \alpha_1(1 - \beta(L_n))p_T[n] + \alpha_2(1 - \beta(L_n))p_{Tm}[n]\right)k_{af}dt \\
p_{ab}[n] &= k_{ab}[B_{tot}]p_b[n]dt + k_{ay}[Y_{tot}]p_y[n]dt \\
p_{yf}[n] &= k_{ay}[A_{tot}]p_{ap}[n]dt \\
p_{yb}[n] &= k_{yb}dt \\
p_{bf}[n] &= k_{ab}[A_{tot}]p_{ap}[n]dt \\
p_{bb}[n] &= k_{bb}dt
\end{aligned}
$$

Table 2.4: Transition probabilities for the a3MC model of chemotaxis.

Much like other existing computer models, the a3MC model cannot faithfully reproduce some aspects of the chemotaxis pathway that are demonstrated experimentally [7, 17]. However, for our purposes the model adequately captures the dynamics of the network and its response to external attractant concentration inputs. Figure 2-3 shows the result of suddenly changing the concentration to 1mM after the system has reached a steady state with a zero input concentration, from a simulation of the model. We see that after the initial decrease in the probability that a CheY protein is in the phosphorylated state, it returns to the same steady state level after an extended constant exposure to 1mM of ligand concentration, therefore the system adapts between zero and 1mM input levels. The accompanying plot from the simulation in Figure 2-4 shows how the system eventually compensates for the activity-inhibiting effect of increased ligand binding with increased methylation of the receptors, similar to the observed adaptive response in *E.coli* cells [26]. The exhibited adaptive response between different input levels will be important for our intended use of the a3MC model. At high (or low) concentrations, through adaptation, the ability of an *E.coli* cell to respond to further increases (decreases) in receptor occupancy (an increasing function of input concentration) is not diminished [2]. This return to pre-stimulus behavior after an initial transient response, and habituation to an increased background or average input level, will allow the a3MC model-based surface mapping agents to keep seeking out areas with higher input levels, and be sensitive to more localized changes in the landscape of the objective function.

2.2.3 Flagellar Motor

E.coli bacteria have about 5 to 10 flagellar motors per cell [5]. In the a3MC model, illustrated in Figure 2-5, each flagellar motor is represented by an 8-state chain adapted from the model in [17], where each state represents a different number of CheYp proteins bound to the motor. This model assumes that a motor in any of the states 1 through 5 will turn counterclockwise (run mode), otherwise it is in clockwise or tumbling mode. The chemical rate constant values used in the a3MC model of the flagellar motor are obtained from [17] and listed in Table 2.5. Table 2.6 contains the
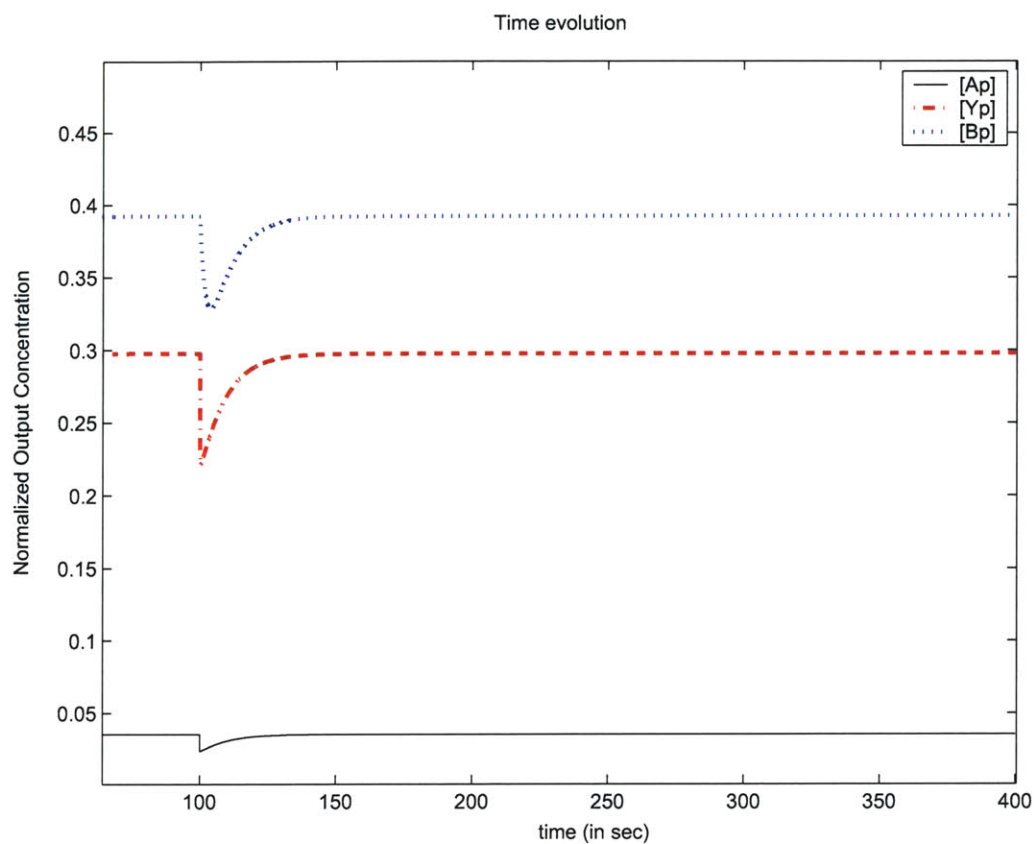
Figure 2-3: Response of phosphorylated state probabilities for CheA, CheB or CheY to a step input, the attractant concentration changes from zero to 1mM at 100 sec.

transition probabilities for the model. We define the *bias* as the probability that a motor will be in any of the 5 states which correspond to a run. Figure 2-6 shows the corresponding bias time plot for the step attractant input response plot of Figure 2-3. Since CheYp promotes tumbling in the motor, the bias increases with the decreasing CheY activity probability. The transient decrease in the probability that a CheY protein is phosphorylated leads to a decrease in the forward transition probabilities (p_{mf1} through p_{mf7}) given in Table 2.6, and the probability that the motor will be in any of the first five states increases.
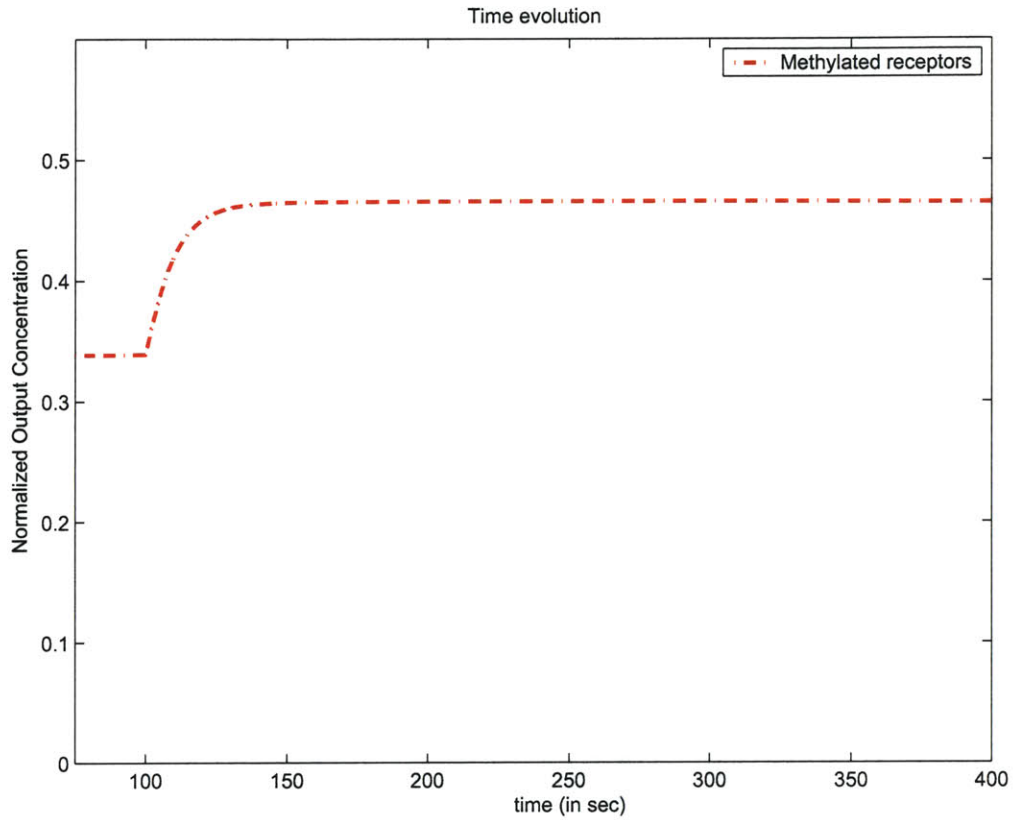
Figure 2-4: Probability that a receptor is in the methylated state during a step input, the attractant concentration changes from zero to 1mM at 100 sec.

$$
\begin{aligned}
k_{mf1} &= 7 \times 10^6 Ms^{-1} \\
k_{mf2} &= 6 \times 10^6 Ms^{-1} \\
k_{mf3} &= 5 \times 10^6 Ms^{-1} \\
k_{mf4} &= 4 \times 10^6 Ms^{-1} \\
k_{mf5} &= 3 \times 10^6 Ms^{-1} \\
k_{mf6} &= 2 \times 10^6 Ms^{-1} \\
k_{mf7} &= 1 \times 10^6 Ms^{-1} \\
k_{mb1} &= 1.43s^{-1} \\
k_{mb2} &= 2.86s^{-1} \\
k_{mb3} &= 4.29s^{-1} \\
k_{mb4} &= 5.72s^{-1} \\
k_{mb5} &= 7.15s^{-1} \\
k_{mb6} &= 8.58s^{-1} \\
k_{mb7} &= 10.01s^{-1}
\end{aligned}
$$

Table 2.5: Flagellar motor rate constants.

$$p_{mf1}[n] = k_{mf1}[Y_{tot}]p_{yp}[n]dt$$

$$p_{mb1}[n] = k_{mb1}dt(1 - k_{mf2}[Y_{tot}]p_{yp}[n]dt)$$

$$p_{mf2}[n] = (k_{mf2}[Y_{tot}]p_{yp}[n]dt)(1 - k_{mb1}dt)$$

$$p_{mb2}[n] = k_{mb2}dt(1 - k_{mf2}[Y_{tot}]p_{yp}[n]dt)$$

$$p_{mf3}[n] = (k_{mf3}[Y_{tot}]p_{yp}[n]dt)(1 - k_{mb2}dt)$$

$$p_{mb3}[n] = k_{mb3}dt(1 - k_{mf2}[Y_{tot}]p_{yp}[n]dt)$$

$$p_{mf4}[n] = (k_{mf4}[Y_{tot}]p_{yp}[n]dt)(1 - k_{mb3}dt)$$

$$p_{mb4}[n] = k_{mb4}dt(1 - k_{mf2}[Y_{tot}]p_{yp}[n]dt)$$

$$p_{mf5}[n] = (k_{mf5}[Y_{tot}]p_{yp}[n]dt)(1 - k_{mb4}dt)$$

$$p_{mb5}[n] = k_{mb5}dt(1 - k_{mf2}[Y_{tot}]p_{yp}[n]dt)$$

$$p_{mf6}[n] = (k_{mf6}[Y_{tot}]p_{yp}[n]dt)(1 - k_{mb5}dt)$$

$$p_{mb6}[n] = k_{mb6}dt(1 - k_{mf2}[Y_{tot}]p_{yp}[n]dt)$$

$$p_{mf7}[n] = (k_{mf7}[Y_{tot}]p_{yp}[n]dt)(1 - k_{mb6}dt)$$

$$p_{mb7}[n] = k_{mb7}dt$$

Table 2.6: Transition probabilities for the a3MC model of the flagellar motor.
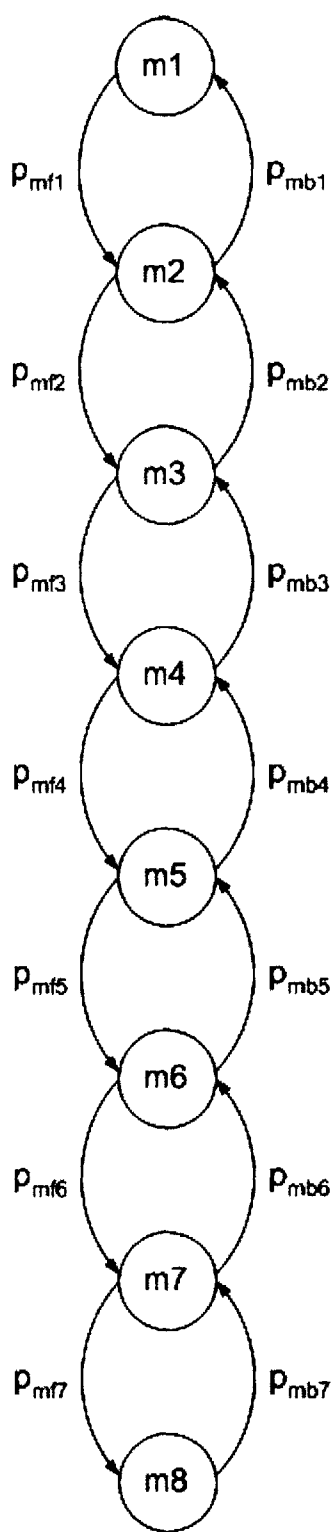
Figure 2-5: a3MC model of flagellar motor.
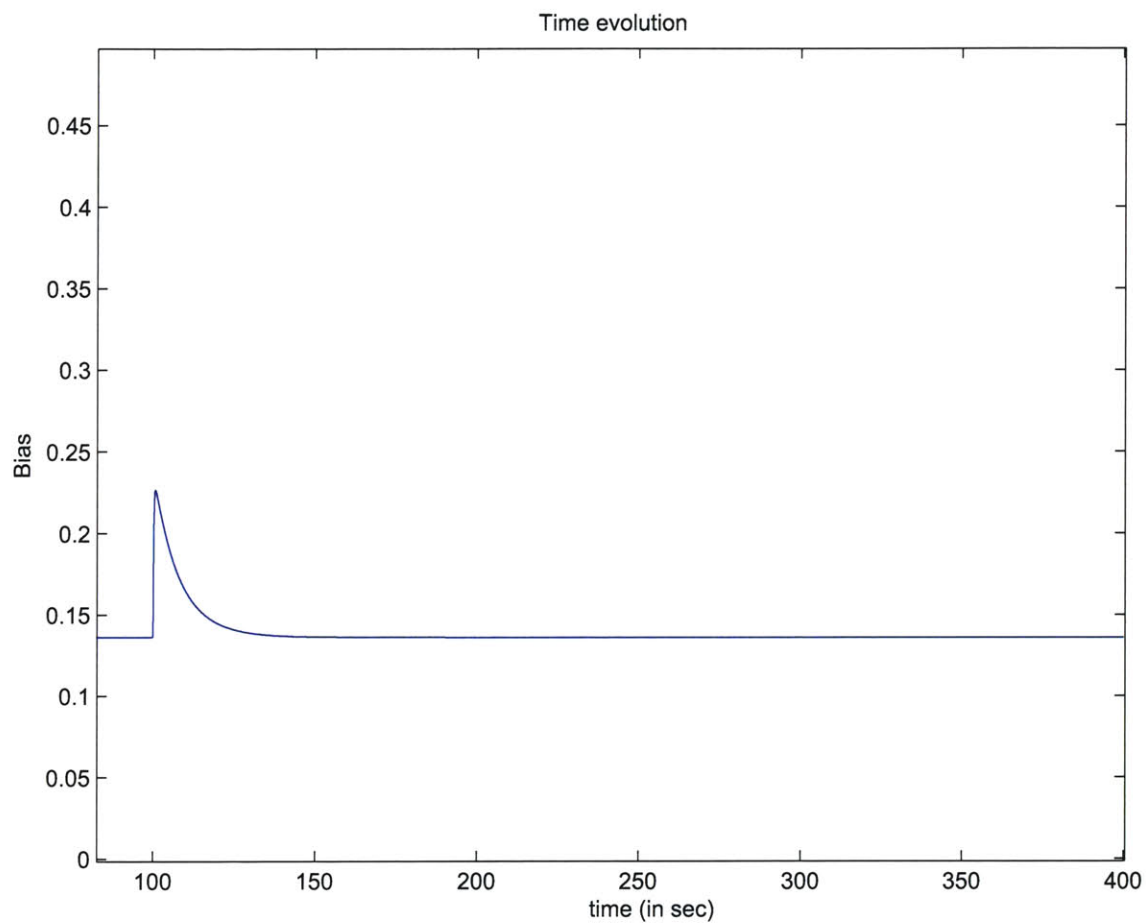
Figure 2-6: Motor bias in response to a step input, at the time step corresponding to 100 seconds the attractant concentration changes from zero to 1mM.

2.3 Modifications to the a3MC Model

This section discusses the modifications that are made to the parameters of the biological a3MC model of chemotaxis in order to make it more suitable for use in the surface mapping algorithm. To obtain a larger expected change in observable behavior (mainly through run and tumble durations or frequencies) with a changing input concentration, some of the biological parameters in the model are changed to the values listed in Tables 2.7 and 2.8. The factors multiplying the original rate constants are also picked to ensure that the model still adapts between a zero input and a 1mM input. Additionally, the rate constants for the flagellar motor are increased to the values listed in Table 2.9 to obtain a larger steady state bias. This allows the surface mapping agent to spend a greater fraction of its total time changing location and exploring the surface due to the increase in run durations.

We can see from the bias response using the new parameters to a 1mM step concentration input in Figure 2-7 that a larger change in bias is obtained when compared to the result using the original parameters in Figure 2-6.

Receptor complex state	Activity probability
Non-methylated and not ligand-bound	$\alpha_1 = 0.15$
Methylated and not ligand-bound	$\alpha_2 = 0.95$
Non-methylated and ligand-bound	$\alpha_3 = 0$
Methylated and ligand-bound	$\alpha_4 = 0.4$

Table 2.7: Modified receptor activity probabilities for use with the surface mapping algorithm.

$$k_{tf} = (7.9992 \times 10^4 \div 1.2)Ms^{-1}$$
$$k_{tlf} = (7.9992 \times 10^4 \div 1.2 \times \sqrt{8.433})Ms^{-1}$$
$$k_{tb} = (7.9992 \times 10^4 \times 1.2)Ms^{-1}$$
$$k_{tlb} = (7.9992 \times 10^4 \times 1.2 \div \sqrt{8.433})Ms^{-1}$$

Table 2.8: Modified rate constants for use with the surface mapping algorithm.

$$k_{mb1} = 1.43 \times 2.5 s^{-1}$$
$$k_{mb2} = 2.86 \times 2.5 s^{-1}$$
$$k_{mb3} = 4.29 \times 2.5 s^{-1}$$
$$k_{mb4} = 5.72 \times 2.5 s^{-1}$$
$$k_{mb5} = 7.15 \times 2.5 s^{-1}$$
$$k_{mb6} = 8.58 \times 2.5 s^{-1}$$
$$k_{mb7} = 10.01 \times 2.5 s^{-1}$$

Table 2.9: Modified flagellar motor rate constants for use with the surface mapping algorithm.
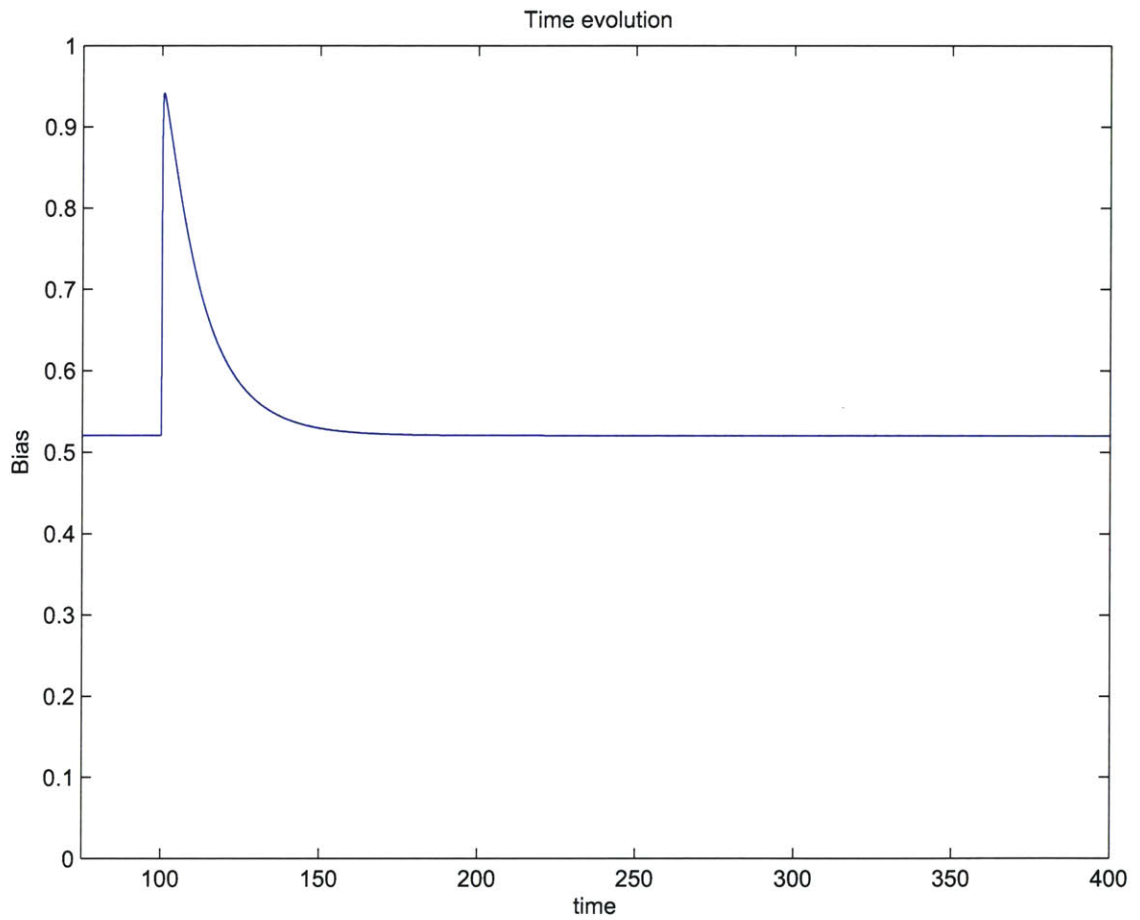
Figure 2-7: Motor bias in response to step input after the modifications to the model parameters, at 100 sec the attractant concentration changes from zero to 1mM.

Chapter 3

Bacterial Algorithm for Surface Mapping

In this chapter we present a surface mapping algorithm, which we refer to as the Bacterial Algorithm for Surface Mapping (BASM). The algorithm simulates trajectories of a bacteria-like software agent in the domain space of a surface represented by a function $C(\cdot)$, analogous to the movement of $E.coli$ cells in a volume where they are exposed to a concentration of attractant that changes with their location. Although this thesis will often discuss the mapping of a concentration function $C(\cdot)$, whose arguments are spatial coordinates, they are generally not restricted as such. The surface mapping agent, which we will refer to as a *bacterial agent*, switches between run and tumble modes to explore the surface, using the surface values at different locations for guidance. We describe how the bacterial agent uses the surface values as a time-varying input to an a3MC stochastic simulation of the model of the chemotaxis network in Chapter 2 in its decision-making. In general, a maximum-seeking bacterial agent tends to spend more time in the regions where the surface has a higher value or a peak, so the behavior of the agent can be informative in cases where an explicit or analytic expression of the function is not known or unavailable. The algorithm computes an estimate of the surface based on the movement of the bacterial agent, which we refer to as the *density* function. This thesis focuses on BASM simulations that map one- and two-dimensional surfaces.

3.1 Stochastic Simulation of Flagellar Motor Chains

An important aspect of the exploration of the surface by the bacterial agent is the flagellar motor. With the input concentration to the a3MC model at any time index set as the surface value at the current location, the state probabilities for all the chains in the model are calculated according to the prescription in Table 2.4 and equation (2.4). An a3MC stochastic simulation is performed with 9 flagellar motors, each of which can be in any of 8 states. The states are updated at every time step, with the transition probabilities depending on the probability of the Y chain being in the phosphorylated state (yp) as specified by Table 2.6. The motors are simulated independently from each other, and using the voting hypothesis in [9] and [26], if more than half of them are in a run state, the bacterial agent runs, otherwise it tumbles. Using multiple motors and the voting hypothesis has an amplification effect on the individual motor bias to obtain the cell bias, i.e. the probability that the bacterial agent will be in a run state. To obtain the cell bias we find the probability that 5 or more of the 9 motors will be in a run state:

$$\text{cell bias} = \sum_{k=5}^{9} \binom{9}{k} b^k (1-b)^{9-k} \tag{3.1}$$

where b is the individual motor bias. As the relation in Figure 3-1 shows, a motor bias greater than $\frac{1}{2}$ will be increased towards 1, and a bias less than $\frac{1}{2}$ will be pushed down towards zero.
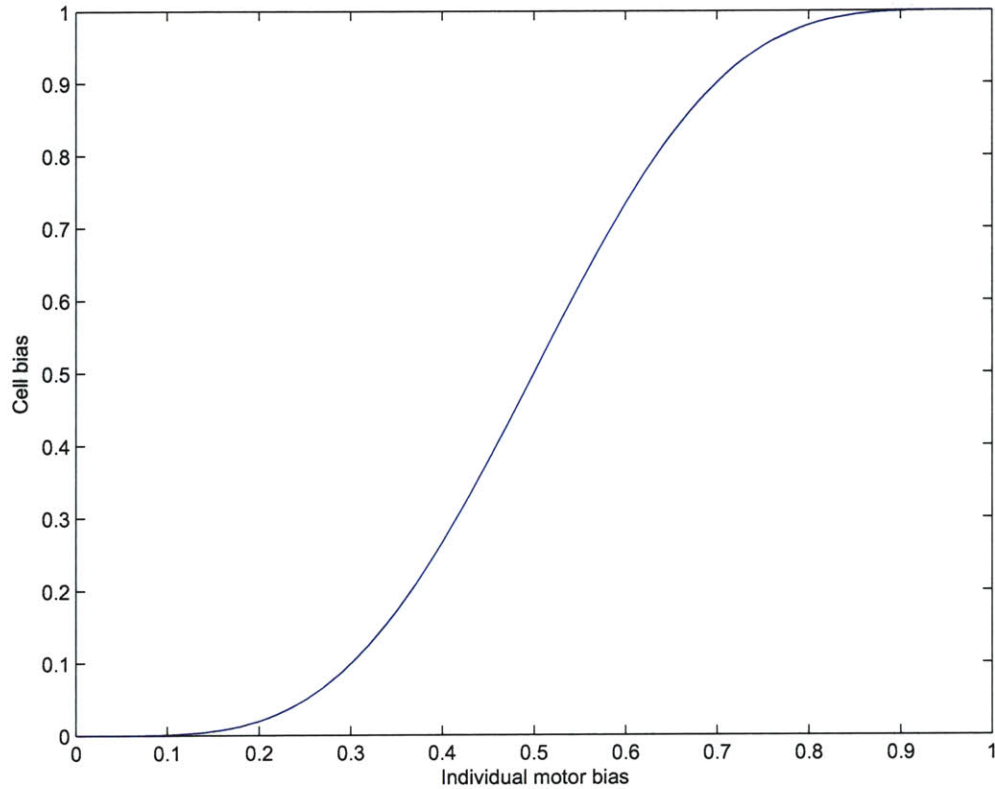
Figure 3-1: Probability that a bacterial agent using 9 motors will be in run mode as a function of the single flagellar motor bias.

3.2 Surface Mapping Algorithm

We describe the motion and position of the bacterial agent in a two-dimensional search space at a time index n in terms of the following parameters:

- ($x[n], y[n]$): position or current coordinates.

- $\theta[n]$: angle or direction the bacterial agent is facing.

- The state of the motors. The bacterial agent can either be in run mode (flagella rotating counterclockwise) or tumble mode (flagella rotating clockwise) between time indices n and $n+1$. During a run, the position changes but the direction does not, and during a tumble the direction changes while the position remains the same.

41

3.2.1 Tumbles

The simulations assume a simple model for the effect of a tumble. In BASM, for each tumble the bacterial agent rotates counterclockwise or clockwise with equal probability (in contrast with flagellar rotation, which is always in the clockwise direction for tumbles). The rotation direction is constant throughout the tumble, and a constant radial speed rs (in rad/s) is used to determine the total change in direction due to the tumble. For every time step during which the bacterial agent is in tumble mode, the angle or direction of the bacterial agent at time n is adjusted as follows:

$$\theta[n+1] = \theta[n] + sgn(U) \times rs \times dt \tag{3.2}$$

where dt is the time in seconds that corresponds to a time step for the discrete-time Markov chains, and U is a random variable generated at the start of the tumble distributed uniformly in the interval [-1,1].

3.2.2 Runs

Once an *E.coli* cell is running, its speed moving forward is approximately constant throughout the run, and does not vary significantly from one run to another [14]. While *E.coli* move in three dimensions, the algorithm carries this over to the two-dimensional case, as in [19], using a constant running velocity v.

In the two-dimensional surface mapping simulations, a run changes the position of a bacterial agent as follows:

$$x[n+1] = x[n] + \cos(\theta[n]) \times v \times dt \tag{3.3}$$

$$y[n+1] = y[n] + \sin(\theta[n]) \times v \times dt \tag{3.4}$$

where v is the running velocity in (length unit)/s.

In the one-dimensional case, the interpretation of the direction angle $\theta[n]$ is less straightforward. In BASM, the bacterial agent still moves with speed v, but the angle

42

$\theta[n]$ determines whether the displacement is in the positive or negative x direction. More specifically:

$$x[n + 1] = x[n] + sgn(\cos(\theta[n])) \times v \times dt \qquad (3.5)$$

3.2.3   Algorithm Steps

At the beginning of the simulation, the bacterial agent position, direction, internal state probabilities, and motor state (run or tumble) are set to an initial value or state. At every iteration or time step n (each corresponding to a time duration dt) the algorithm performs the following steps:

1: Determine the current position and direction ($x[n]$,$y[n]$ and $\theta[n]$) using previous position, direction and previous overall motor state(whether the agent was in run or tumble mode at time index $n - 1$) according to equations (3.2)-(3.5).

2: Determine the current ligand input concentration for the agent as $g[n] = C(x[n])$ or $g[n] = C(x[n], y[n])$ in the case of one- or two-dimensional surfaces respectively.

3: Use previous concentration input $g[n - 1]$ and previous internal states (state occupancy probabilities for previous time step $n - 1$) to calculate current internal state probabilities (for time index n).

4: Use previous internal states to check for transitions in any of the 9 motors, and determine new motor state (whether the bacterial agent is in run or tumble mode at time index n).

5: Go to 1 for next time step.

3.3   Calculation of the Density Function

The algorithm produces a histogram indicating the relative amount of time the bacterial agent spent near every position. We refer to this as the density function, which varies with the spatial coordinates. Several simulations are performed and the results from the different trajectories produced are averaged. Since the bacterial agent tends to seek out higher concentrations, areas with higher densities indicate an increased

likelihood of higher concentrations near these locations.

For one-dimensional surface mapping simulations a uniform grid of the spatial dimension is formed, where each point on the grid is defined as:

$$x_k = k \times v \times dt \tag{3.6}$$

for integer k, such that the points are separated by vdt. Every simulation produces a density function $D[k]$, which is a histogram indicating the total number of time steps during the entire simulated movement trajectory where the bacterial agent was in the run state, and in a position that falls within the bin (of size vdt) centered at x_k. At the beginning of a simulation, $D[k]$ is set to to zero for all k within a limited range. At every time step, if the flagellar motors are in a run mode, the current position $x[n]$ is rounded to the nearest value of x_k, and the density function $D[k]$ is incremented at the corresponding k.

Every run of BASM simulates the sample motion trajectory of the bacterial agent for 1000 seconds (or $\frac{1000}{dt}$ time steps) and produces an associated density function. An average density function is then calculated using the results of N_1 simulations:

$$D_{avg}[k] = \frac{1}{N_1} \sum_{i=1}^{N_1} D_i[k] \tag{3.7}$$

where i denotes the simulation number.

For two-dimensional surface mapping simulations a two-dimensional uniform grid of the spatial dimensions is formed, with each point on the grid written as (x_{k_1}, y_{k_2}) where

$$x_{k_1} = k_1 \triangle_x \tag{3.8}$$

$$y_{k_2} = k_2 \triangle_y \tag{3.9}$$

such that adjacent points on the grid are separated by $\triangle_x$ and $\triangle_y$ distance units in the horizontal and vertical dimensions respectively. Every simulation produces a density function $D[k_1, k_2]$, which is a histogram indicating the total number of time

44

steps during the entire simulated movement trajectory where the bacterial agent was in the run state, and in a position that falls within the bin (of size $\triangle_x \triangle_y$) centered at (x_{k_1}, y_{k_2}). At the beginning of a simulation $D[k_1, k_2]$ is set to zero for all k_1 and k_2 within a limited range. At every time step, if the flagellar motors are in a run mode, the current position coordinates $x[n]$ and $y[n]$ are rounded to the nearest values of x_{k_1} and y_{k_2} respectively, and the density function $D[k_1, k_2]$ is incremented at the corresponding k_1 and k_2.

Every run of BASM simulates the sample motion trajectory of the bacterial agent for 2000 seconds (or $\frac{2000}{dt}$ time steps) and produces an associated density function. An average density function is then calculated using the results of N_2 simulations:

$$D_{avg}[k_1, k_2] = \frac{1}{N_2} \sum_{i=1}^{N_2} D_i[k_1, k_2]$$
(3.10)

where i denotes the simulation number.

Chapter 4

Simulations and Results

In this chapter we evaluate the surface mapping algorithm by running simulations on one- and two-dimensional test functions and examining the results. We observe the effect of changing the parameters v and rs, as well as a few modifications to the BASM algorithm on the obtained averaged density function, which serves as an estimate of the surface. The true value of the test function is accessible only to the bacterial agent. The density function is computed based only on the behavior of the bacterial agent in the surface mapping simulations, and can be compared against the test functions used in this chapter. We also demonstrate that the bacterial agent, when given the ability to reduce the value of the surface at locations it visits (analogous to consuming a substance on a concentration surface), is more effective in reducing the surface integral within a certain period of time when compared to a bacterial agent lacking the ability to sense surface information or respond to it.

4.1   One-Dimensional Surface Mapping Simulations

In this section we present the results from one-dimensional BASM simulations using two test concentration surfaces:

$$C_1(x) = 10^{-4} \exp(-\frac{3}{4}|x|) \tag{4.1}$$

47

$$C_2(x) = (5 \times 10^{-5}) \left| \frac{\sin(\frac{1}{4}x)}{\frac{1}{4}x} \right| \tag{4.2}$$

The unimodal function $C_1(x)$ and the multimodal function $C_2(x)$ are plotted in Figure 4-1. The results presented in this section are all produced by running BASM and using $N_1 = 40$ simulations.
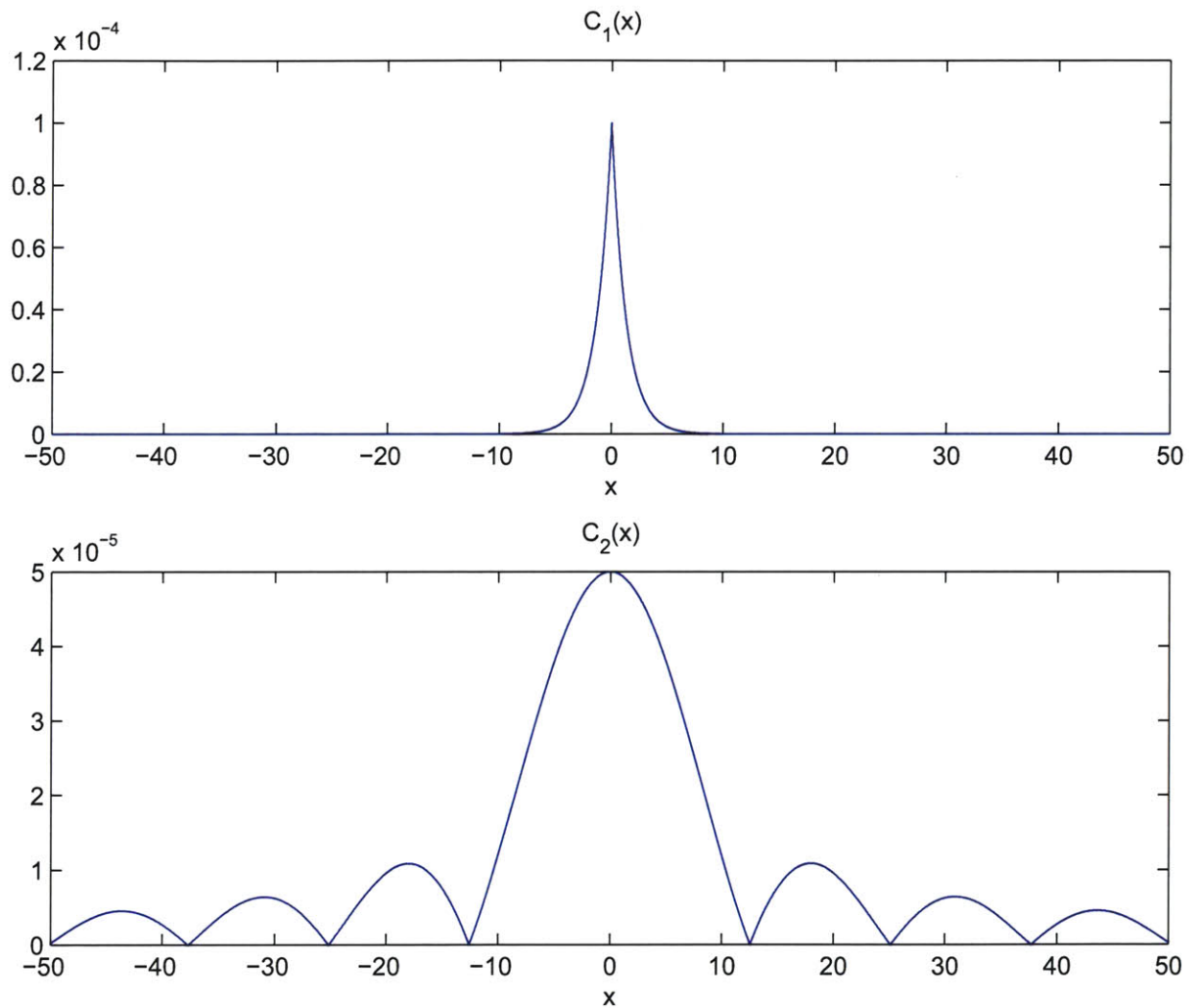

Figure 4-1: The one-dimensional test surfaces $C_1(x)$ and $C_2(x)$.

At the beginning of the ith simulation, the starting position $x[0]$ along the x-axis is set to the value of a random number X_{init}^i, which is distributed uniformly between -10 and 10. Furthermore, the bacterial agent is equally likely to be facing the positive or negative x direction, as $\theta[0]$ is set to 0 or π with equal probability.

For BASM simulations on the unimodal concentration surface $C_1(x)$, a default run speed v of 0.75 s^{-1} and a tumbling rotation speed rs of π rad/s are used. For the multimodal function $C_2(x)$ we use $v = 0.6s^{-1}$ and $rs = \pi$ rad/s. The averaged density functions obtained are shown in Figure 4-2 for $C_1(x)$ and 4-3 for $C_2(x)$. We see that the density function shares some broad characteristics with the surface and can shed light on some properties of it, such as the approximate maxima and minima locations. Both density functions are approximately symmetric, and the density function from the simulations using the multimodal surface captures the main lobe and the adjacent side lobes of lower height.

To remove some of the extraneous noise-like detail from the average density function, it is passed through an FIR smoothing filter. A 2001-point averaging filter was used. The smoothed density function is then scaled to approximately integrate to unity over a finite interval, for comparison against the similarly normalized test surface. Figures 4-4 and 4-5 show a plot of the smoothed density function against the test surfaces and the approximation error as an absolute difference. The global maximum of the smoothed density function for the unimodal surface simulations was found to be at $x = -0.228$, less than a quarter of a distance unit away from the true maximum of the surface. For the multimodal surface simulations, the maxima and minima locations are estimated by finding the highest or lowest value respectively that the smoothed density function takes in a reasonable range around the true maximum/minimum locations (usually between the two adjacent extrema). Tables 4.1 and 4.2 summarize the estimated maxima and minima locations computed using the smoothed density function from the multimodal surface simulations.
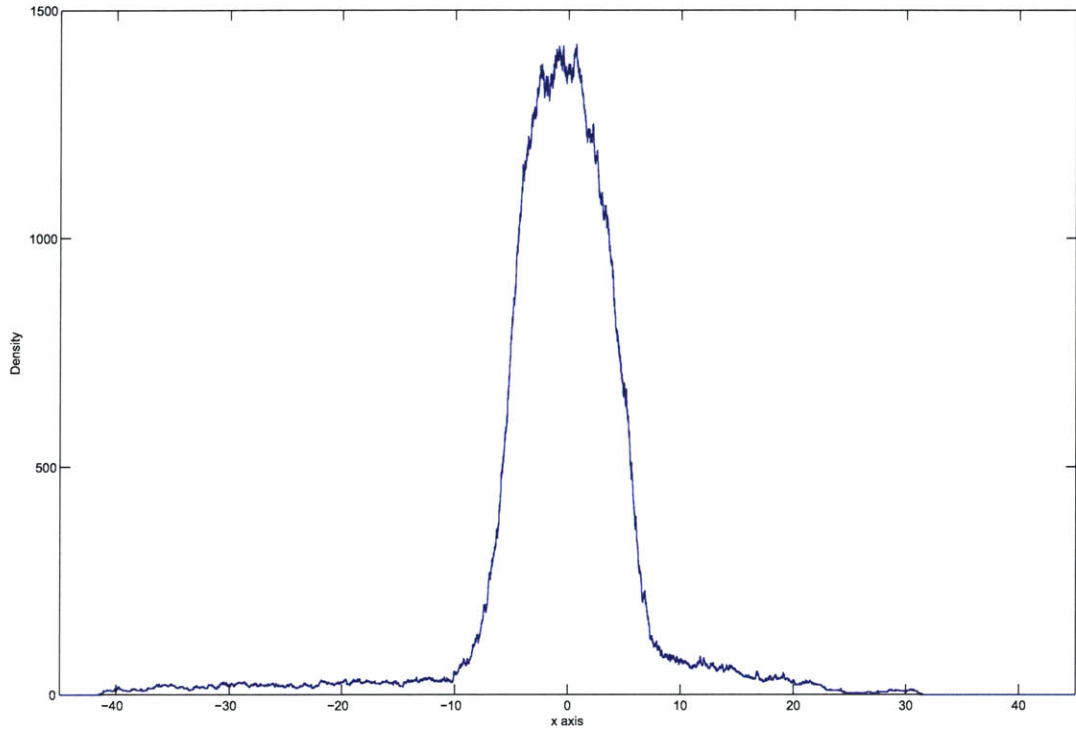
49

Figure 4-2: Average density function result from one-dimensional surface mapping simulations using $C_1(x)$, $v = 0.75s^{-1}$, $rs = \pi$ rad/s.

Maxima locations	Measured location	Error
-10π	-28.694	2.7219
-6π	-17.893	0.9566
0	4.667	4.667
6π	17.343	1.5066
10π	34.285	5.3309

Table 4.1: Maxima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \pi$ rad/s.

Minima locations	Measured location	Error
-8π	-25.955	0.8223
-4π	-13.373	0.8066
4π	13.306	0.7396
8π	26.085	0.9523

Table 4.2: Minima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \pi$ rad/s.
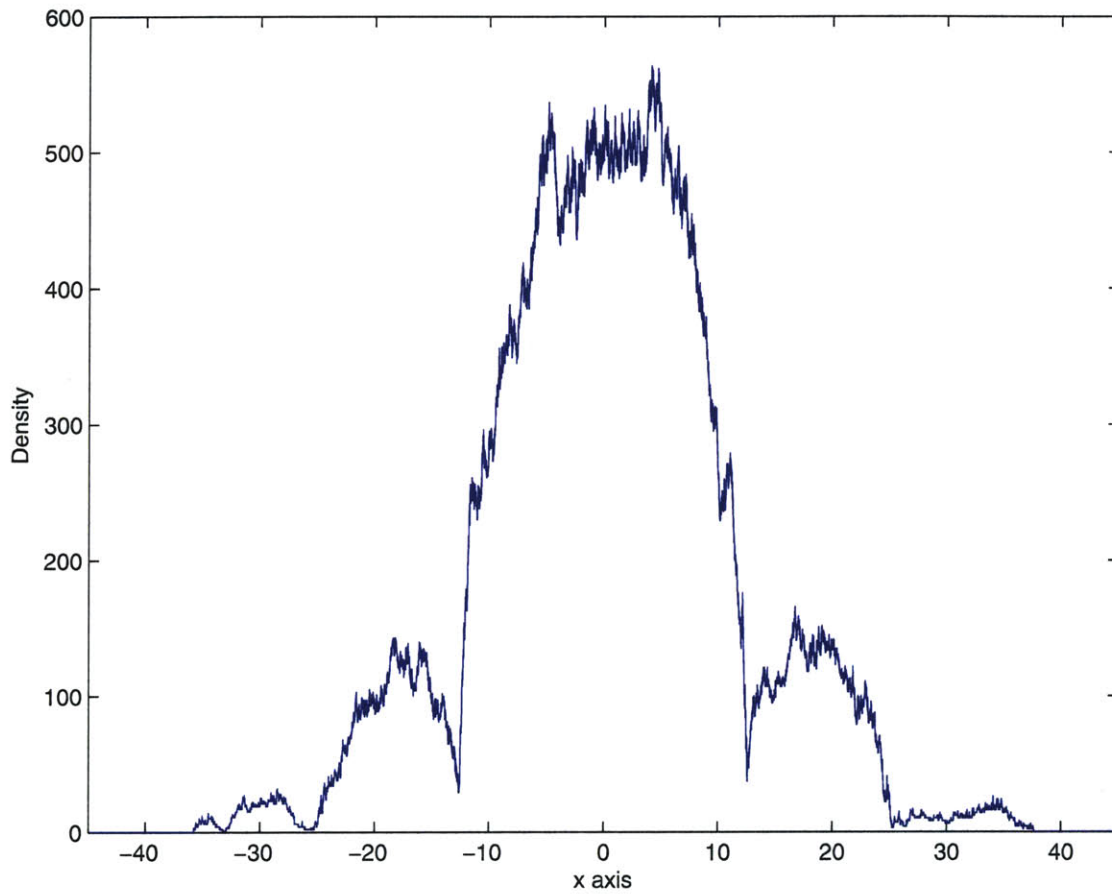
50

Figure 4-3: Average density function result from one-dimensional surface mapping simulations using $C_2(x)$, $v = 0.6s^{-1}$, $rs = \pi$ rad/s.
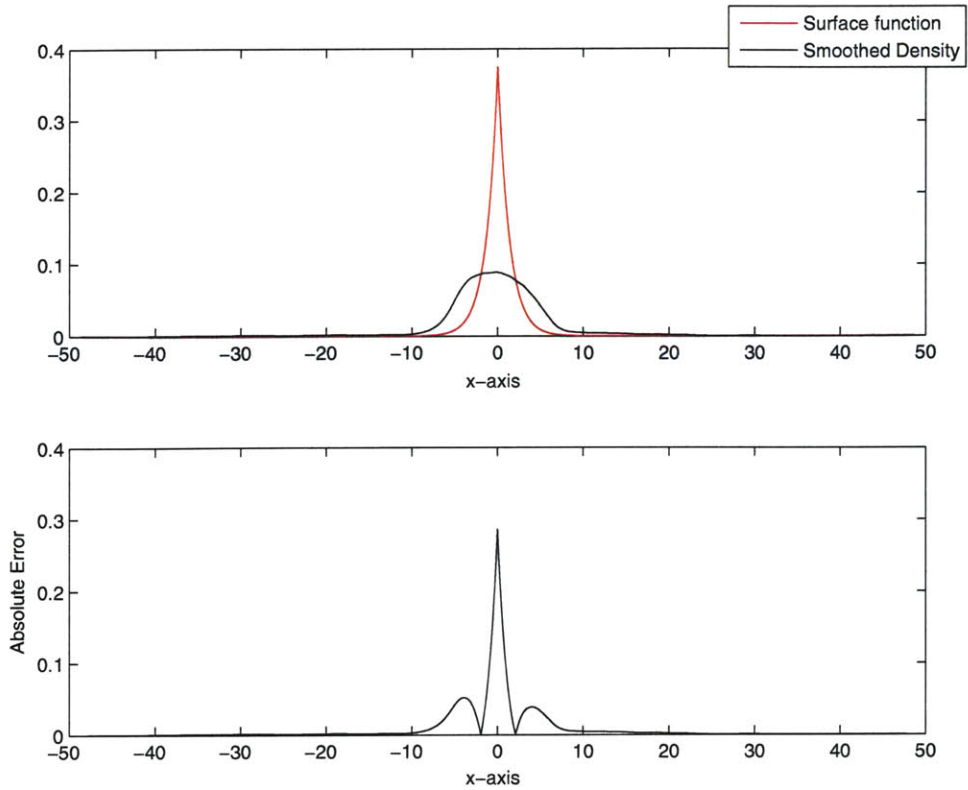
Figure 4-4: Smoothed density function and absolute approximation error. $C(x) = C_1(x)$, $v = 0.75s^{-1}$, $rs = \pi$ rad/s.
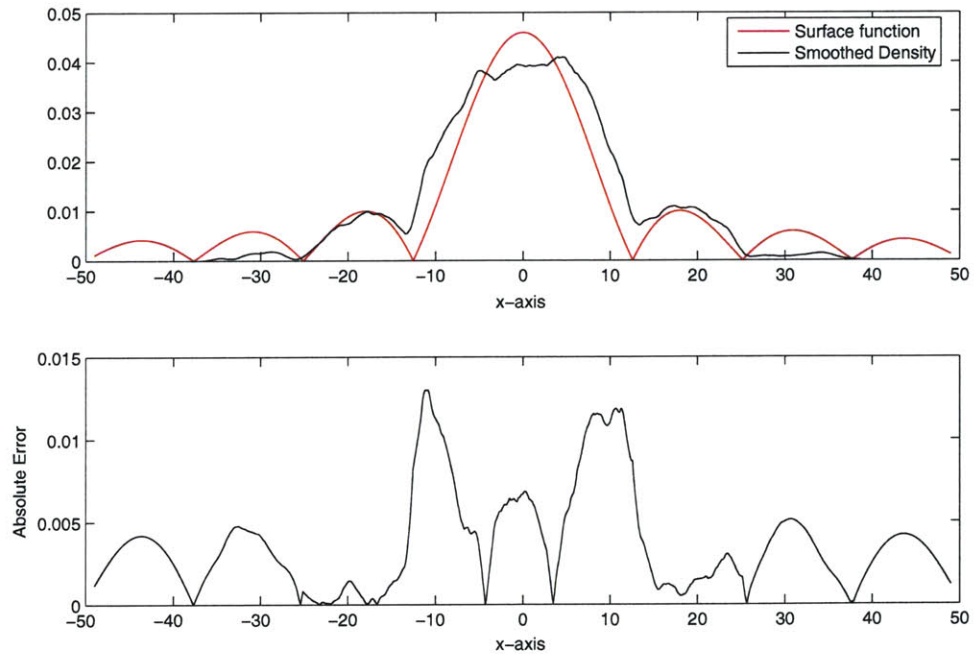

Figure 4-5: Smoothed density function and absolute approximation error. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \pi$ rad/s.

Figures 4-6 to 4-9 show the results obtained when the running speed v is increased for the $C_1(x)$ and $C_2(x)$ simulations to $1.25s^{-1}$ and $1s^{-1}$ respectively. We can see that the density function is more spread out, as the higher run speed allows the bacterial agent to spend more time exploring areas further away from both its starting position and the global maximum at zero. We also observe that the mapping of $C_2(x)$ captures the main lobe of the sinc function as well as three of the side lobes on either side of it. Tables 4.3 and 4.4 summarize the estimated maxima and minima locations computed using the smoothed density function from the multimodal surface simulations.
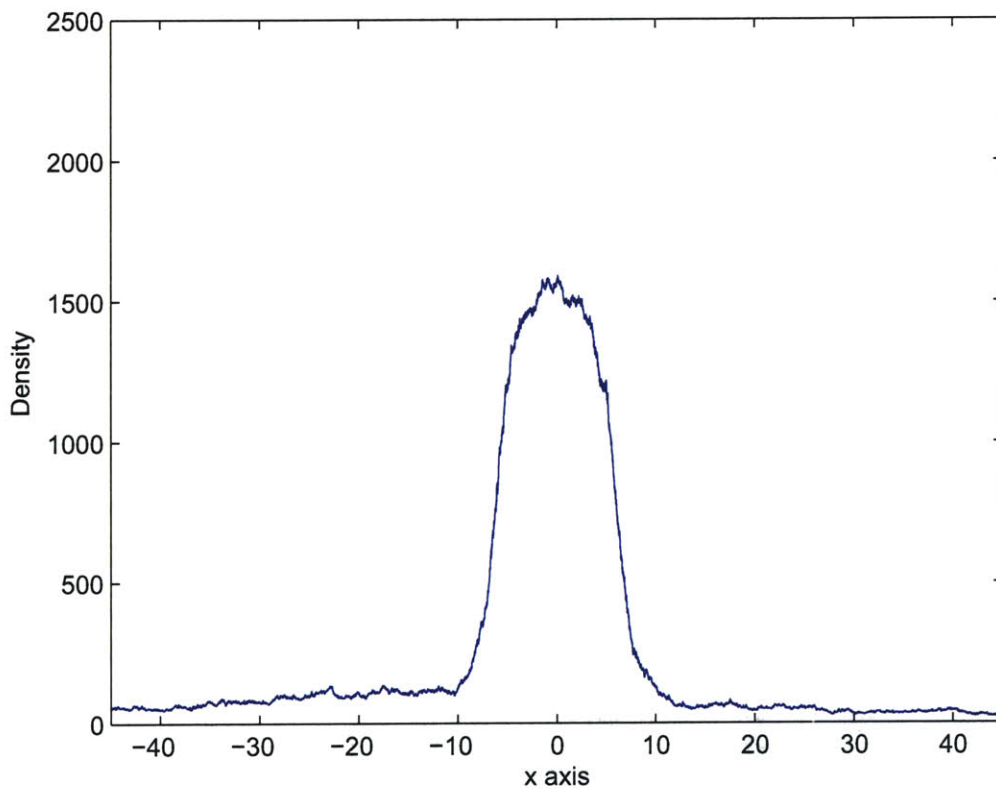

Figure 4-6: Average density function result from one-dimensional surface mapping simulations using $C_1(x)$, $v = 1.25s^{-1}$, $rs = \pi$ rad/s.
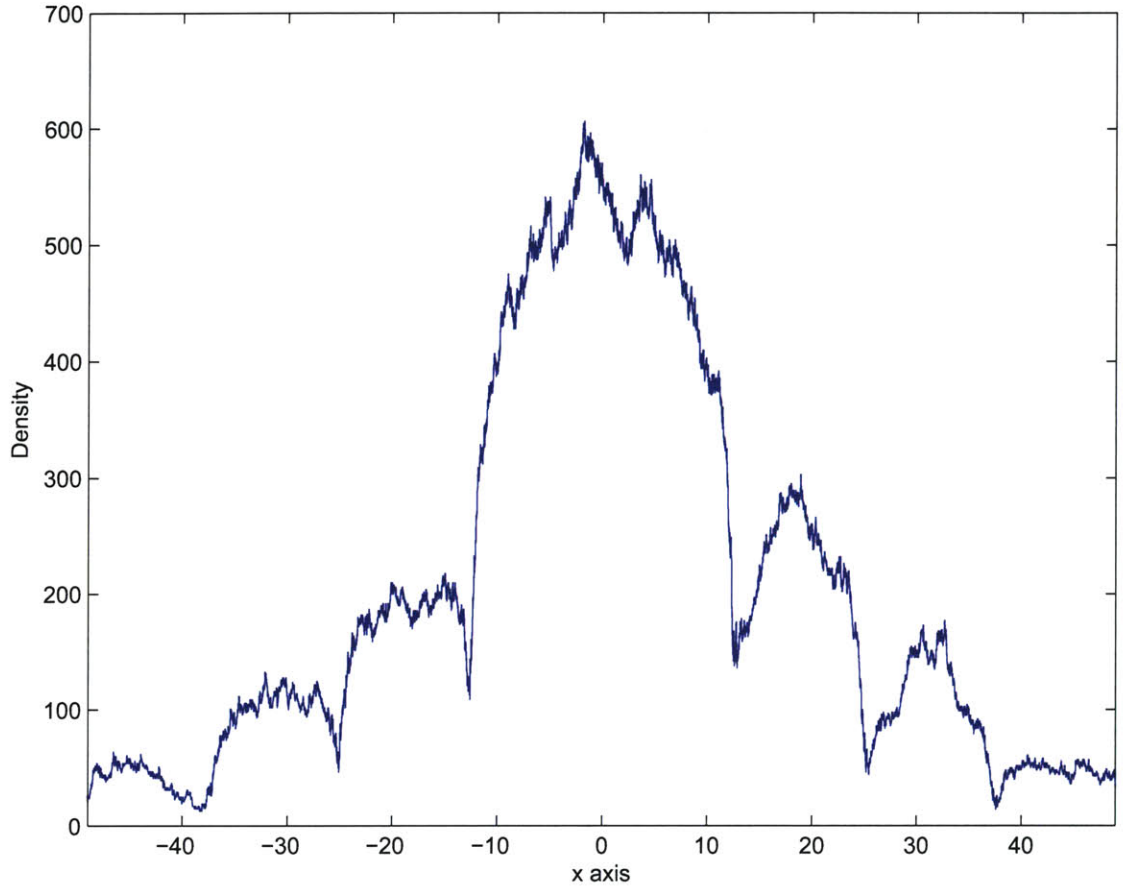
Figure 4-7: Average density function result from one-dimensional surface mapping simulations, using $C_2(x)$, $v = 1s^{-1}$, $rs = \pi$ rad/s.

Maxima locations	Measured location	Error
-14π	-45.647	1.6647
-10π	-30.288	1.1279
-6π	-14.887	3.9626
0	-1.251	1.251
6π	18.211	0.6386
10π	31.887	0.4711
14π	41.139	2.8433

Table 4.3: Maxima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 1s^{-1}$, $rs = \pi$ rad/s.

Figure 4-8: Smoothed density function and absolute approximation error. $C(x) = C_1(x)$, $v = 1.25s^{-1}$, $rs = \pi$ rad/s.

Minima locations	Measured location	Error
-12π	-38.603	0.9039
-8π	-25.637	0.5043
-4π	-13.246	0.6796
4π	13.384	0.8176
8π	25.886	0.7533
12π	38.029	0.3299

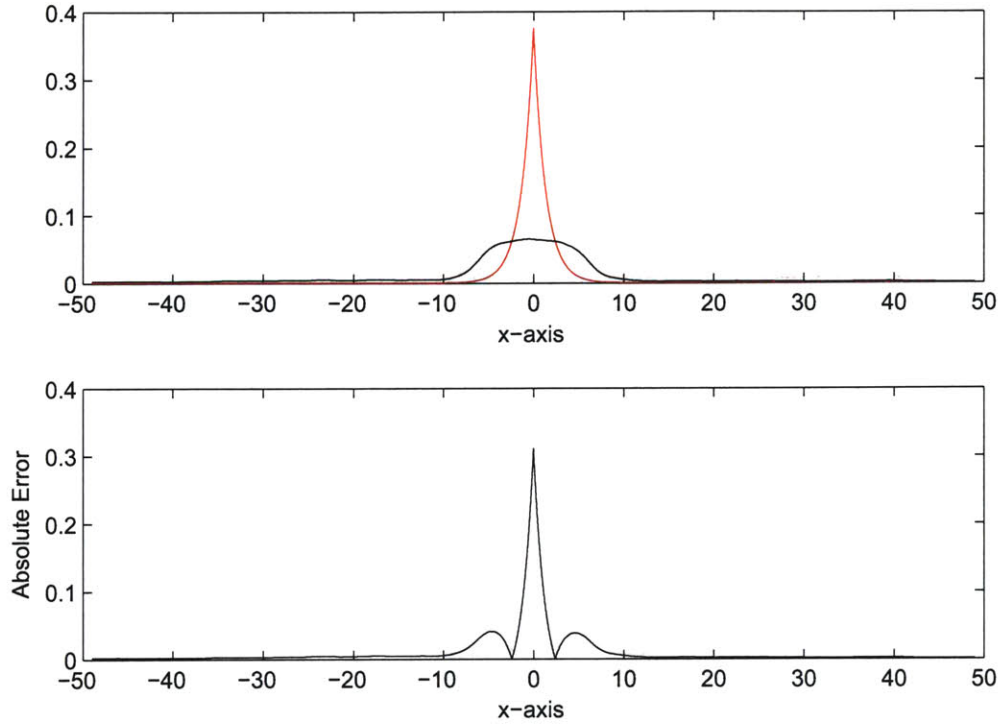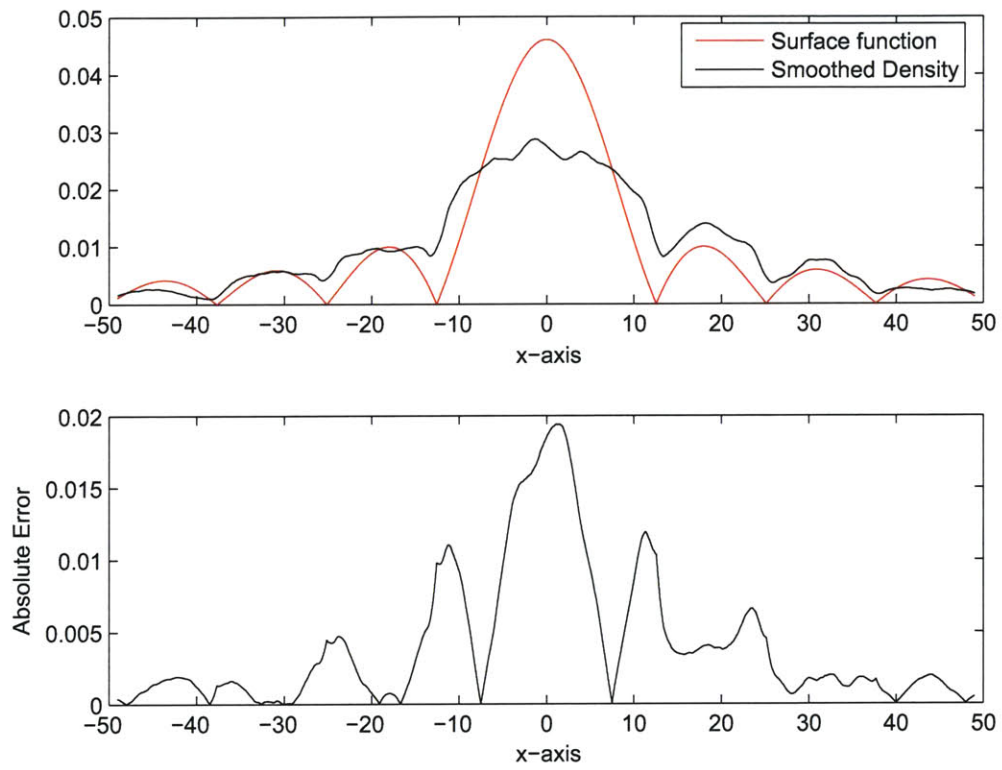Table 4.4: Minima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 1s^{-1}$, $rs = \pi$ rad/s.

Figure 4-9: Smoothed density function and absolute approximation error. $C(x) = C_2(x)$, $v = 1s^{-1}$, $rs = \pi$ rad/s.

Figures 4-10 to 4-13 show the results obtained when the rotation speed rs is increased for the $C_1(x)$ and $C_2(x)$ simulations to $\frac{3\pi}{2}$ rad/s and $\frac{5\pi}{4}$ rad/s respectively. We see that this has the opposite effect on the density function than increasing v, as the density suggests that the bacterial agent did not spend a significant amount of time outside a small range around the global maximum. This is expected as the high tumbling rotation speed results in more frequent switches in running direction and the bacterial agent is therefore not likely to travel far in one direction. Tables 4.5 and 4.6 summarize the estimated maxima and minima locations computed using the smoothed density function from the multimodal surface simulations.
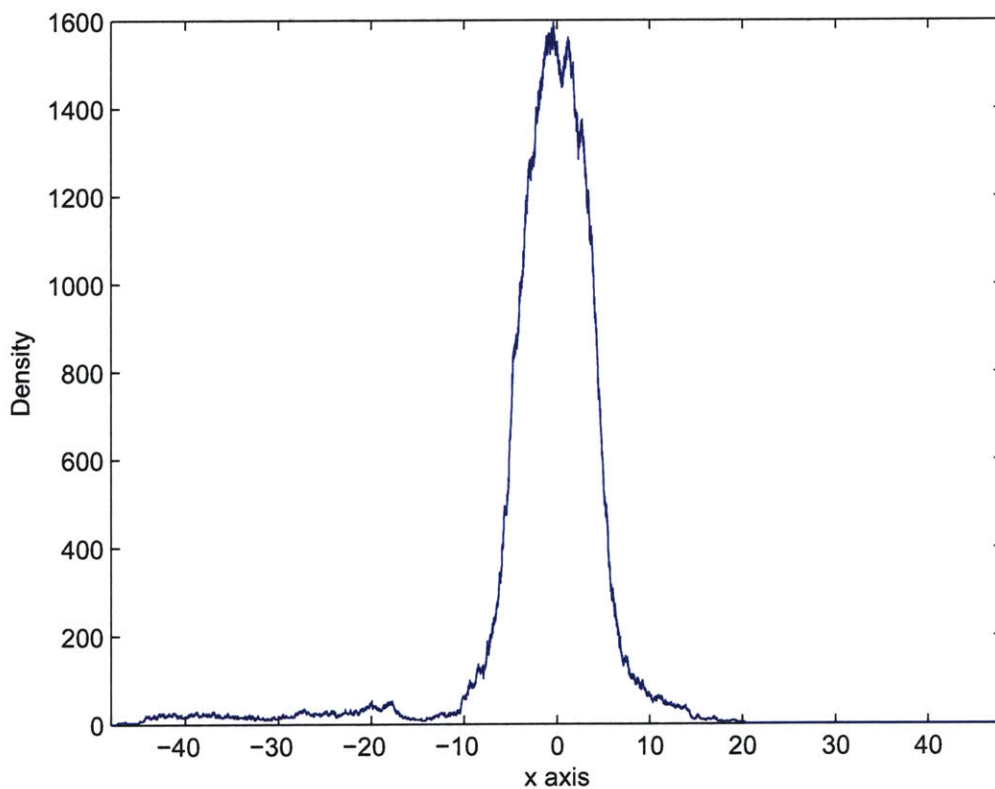

Figure 4-10: Average density function result from one-dimensional surface mapping simulations using $C_1(x)$, $v = 0.75s^{-1}$, $rs = 1.5\pi$ rad/s.
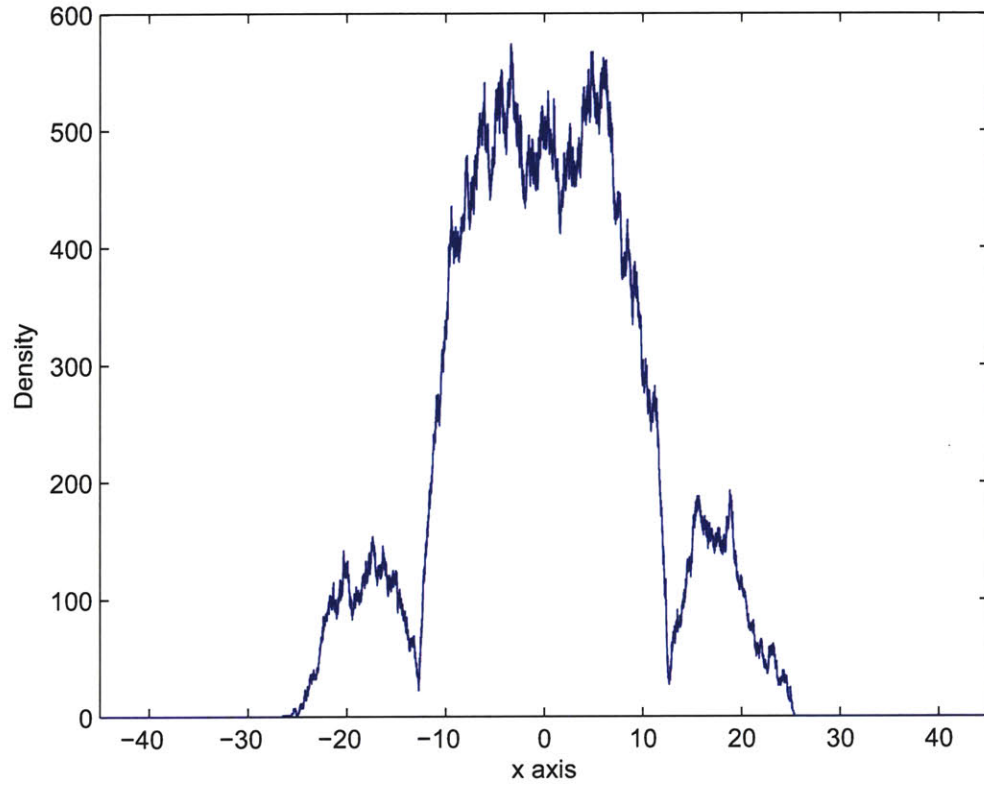
Figure 4-11: Average density function result from one-dimensional surface mapping simulations, using $C_2(x)$, $v = 0.6s^{-1}$, $rs = 1.25\pi$ rad/s.

Maxima locations	Measured location	Error
-6π	-16.903	1.9466
0	-5.430	5.430
6π	16.08	2.7696

Table 4.5: Maxima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \frac{5}{4}\pi$ rad/s.

Minima locations	Measured location	Error
-4π	-13.325	0.7586
4π	13.350	0.7836

Table 4.6: Minima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \frac{5}{4}\pi$ rad/s.
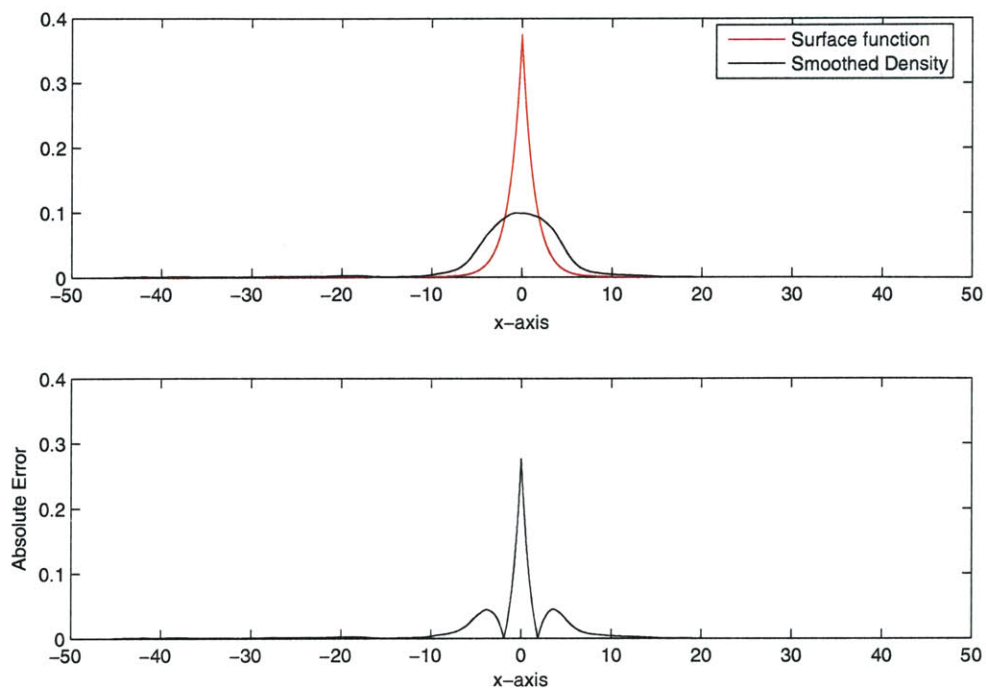
58

Figure 4-12: Smoothed density function and absolute approximation error. $C(x) = C_1(x)$, $v = 0.75s^{-1}$, $rs = 1.5\pi$ rad/s.
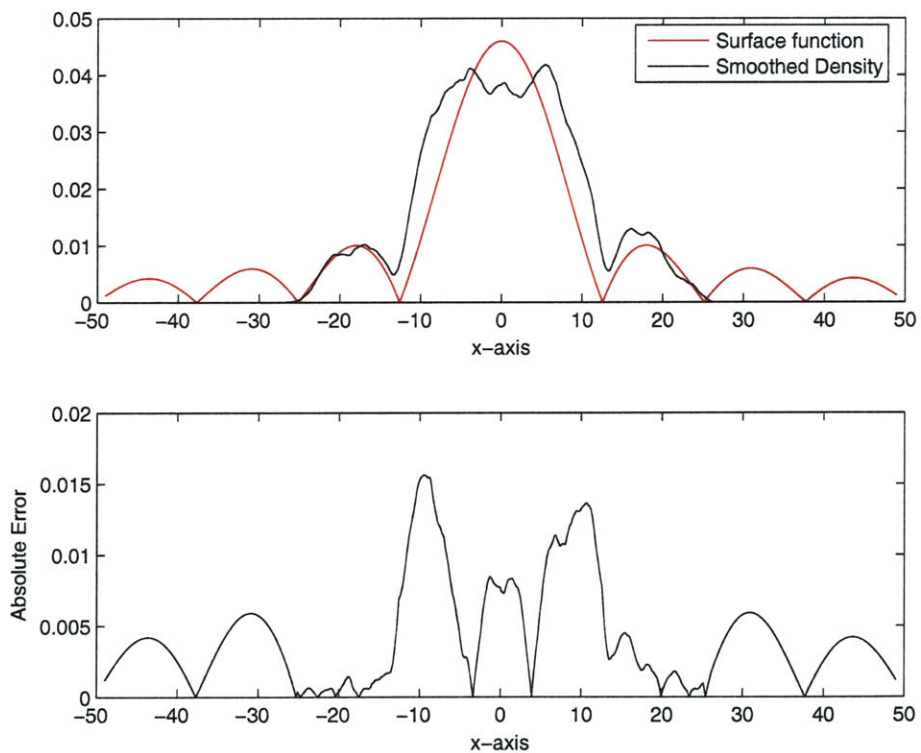

Figure 4-13: Smoothed density function and absolute approximation error. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = 1.25\pi$ rad/s.

4.1.1 Variations on the Algorithm

To better understand the effect of using multiple motors and the amplification effect discussed in section 3.1, we ran simulations where only one flagellar motor is simulated. The state of that motor completely determines whether the bacterial agent runs or tumbles. The results of the simulations are shown in Figures 4-14 to 4-17. We can see that the quality of the surface mapping suffers, and this suggests that the use of more than one motor can allow the bacterial agent to more effectively use the information it encounters about the surface to guide its exploration. Tables 4.7 and 4.8 summarize the estimated maxima and minima locations computed using the smoothed density function from the multimodal surface simulations.
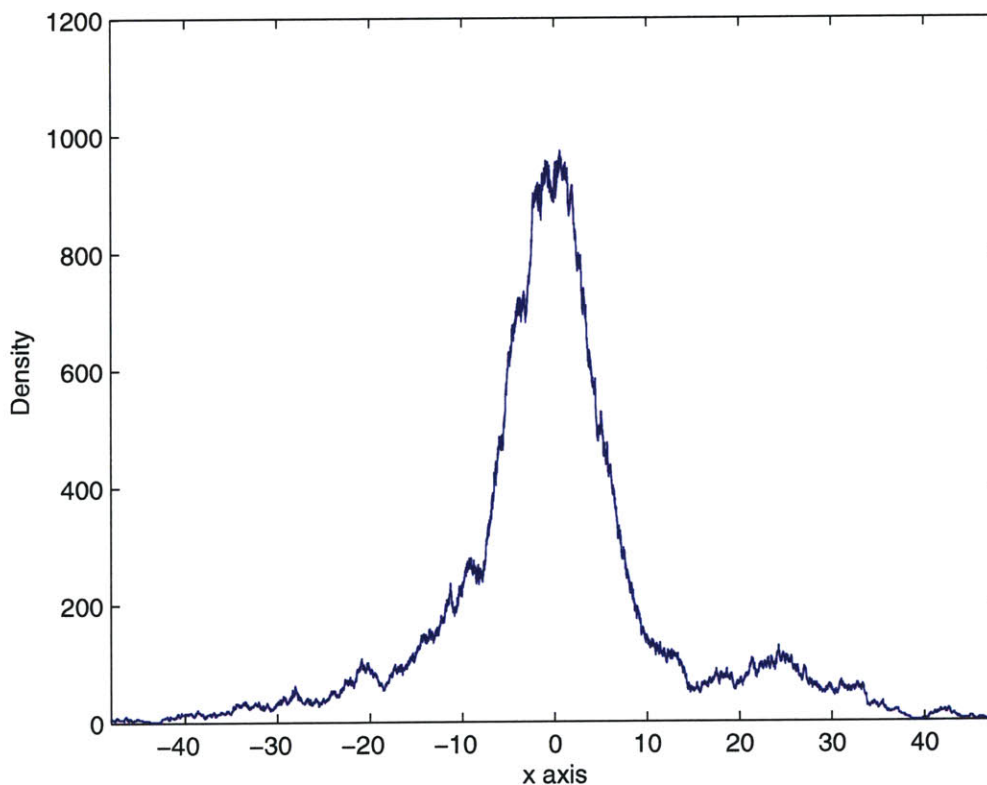

Figure 4-14: Averaged density function result using the single flagellar motor variation of BASM and $C_1(x)$, $v = 0.75s^{-1}$, $rs = 1\pi$ rad/s.
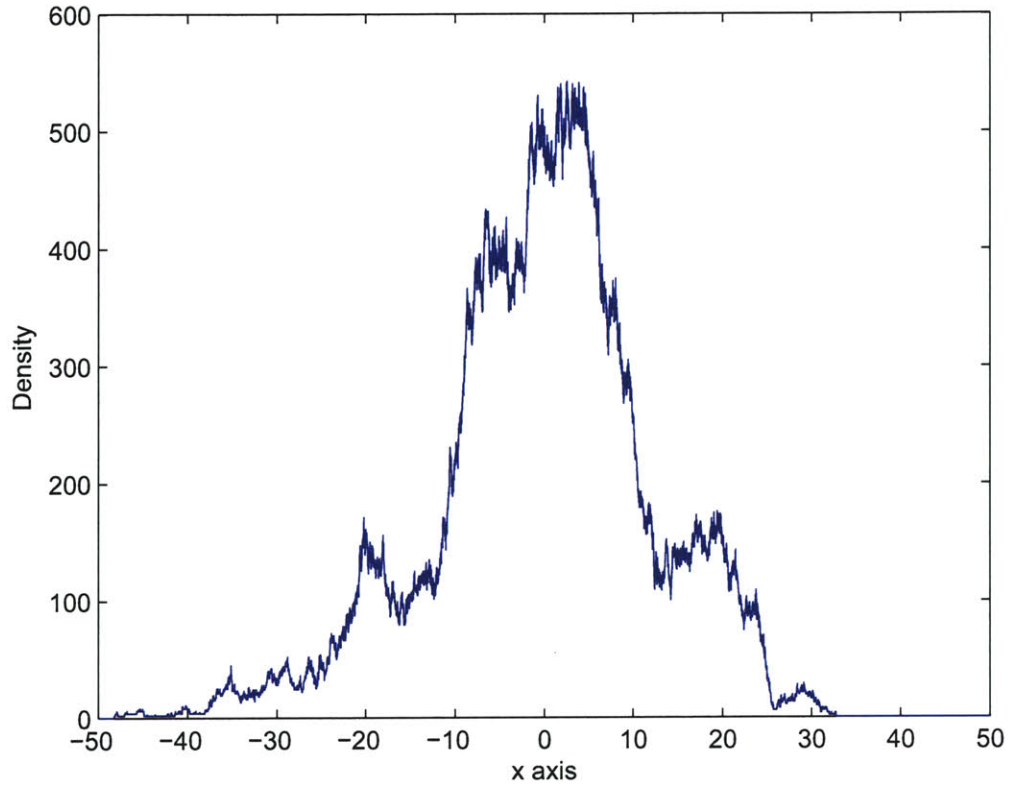
Figure 4-15: Averaged density function result using the single flagellar motor variation of BASM and $C_2(x)$, $v = 0.6s^{-1}$, $rs = 1\pi$ rad/s.

Maxima locations	Measured location	Error
-6π	-19.751	0.9014
0	3.463	3.463
6π	19.089	0.2394
10π	29.199	2.2169

Table 4.7: Maxima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \pi$ rad/s, using one flagellar motor.

Minima locations	Measured location	Error
-4π	-15.799	3.2326
4π	13.333	0.7666

Table 4.8: Minima locations estimated from smoothed density function. $C(x) = C_2(x)$, $v = 0.6s^{-1}$, $rs = \pi$ rad/s, using one flagellar motor.

Figure 4-16: Smoothed density function and absolute approximation error. $C(x) = C_1(x), v = 0.75s^{-1}$, $rs = \pi$ rad/s, using one flagellar motor.
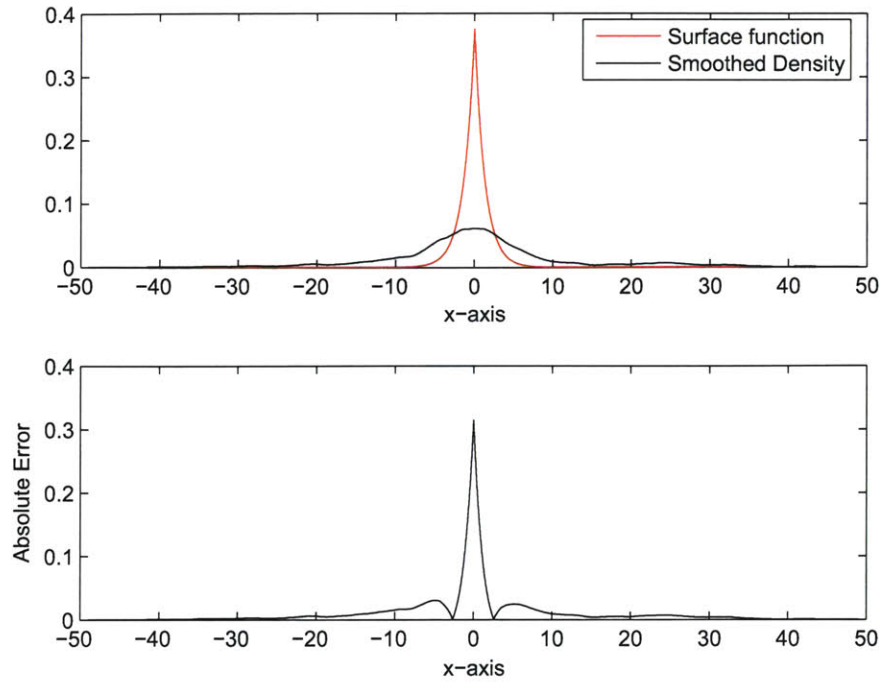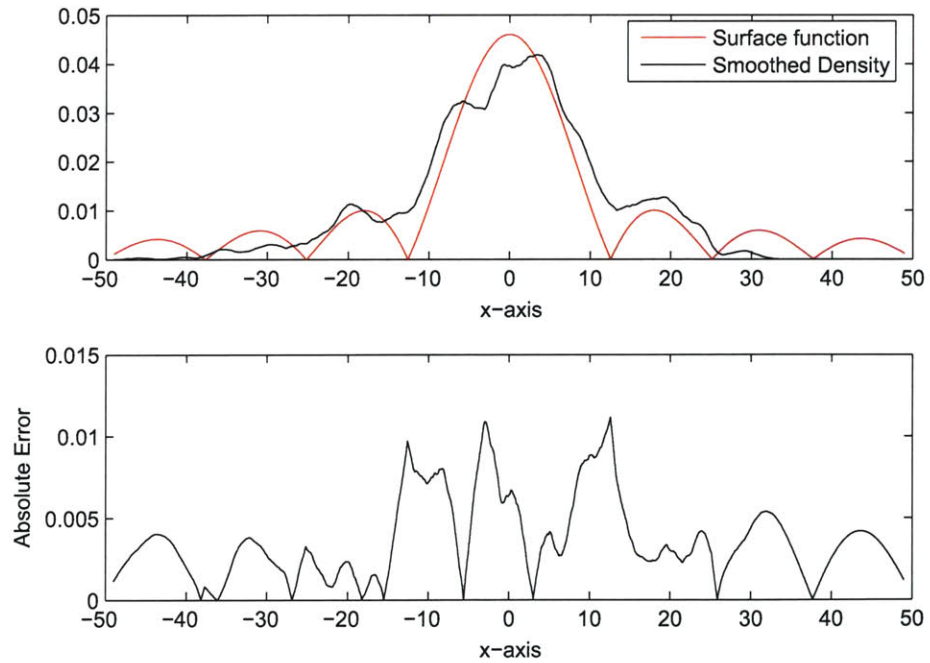

Figure 4-17: Smoothed density function and absolute approximation error. $C(x) = C_2(x), v = 0.6s^{-1}$, $rs = \pi$ rad/s, using one flagellar motor.

Since subsequent tumbles may result in rotations in opposite directions, it is possible for one tumble to undo the effect of a previous tumble. Therefore, an increase in tumbling time within some interval, likely due to a perceived decrease in the value of the surface $C(\cdot)$ in a certain direction, will not guarantee a switch in running direction. We experimented with a tumbling variation where the rotation direction is consistent, and equation (3.2), the update equation for $\theta[n]$ due to a tumble, is replaced with

$$\theta[n+1] = \theta[n] + rs \times dt \tag{4.3}$$

This leads to a more straightforward relation between tumbling and switches in running direction, as the bacterial agent switches running directions after spending enough time in the tumble state to correspond to an angular rotation of π. The results from simulations implementing this modification are shown in Figures 4-18 and 4-19. We also combined this variation with a modification to the a3MC model where the rate constants for the flagellar motor in Table 2.9 are doubled. This results in a larger bias, increasing the durations of runs and decreasing the durations of tumbles. The average density function obtained from surface mapping simulations using $C_1(x)$ is shown in Figure 4-20. The flat density near zero suggests that switches in running direction rarely occurred in that interval, and this may be attributed to the longer run times resulting from the second modification. The density function is also smooth, and it drops off from its maximum sharply, suggesting that the switches in direction occurred mostly within a small interval centered a distance of 10 away from the global maximum. The tumbling direction modification contributes to this increased predictability of the bacterial agent behavior, because it eliminates the uncertainty in one aspect of the process, namely the relation between the total time spent tumbling in the time interval between two switches in running direction. Figure 4-21 shows the state probability for the CheY chain and the concentration signal the agent perceives during a typical run of this experiment. The concentration input signal is defined as $g[n] = C(x[n])$. The CheY activity probability and the time-varying input signal both

63

oscillate with an almost consistent period. This highly predictable mode, approaching a deterministic behavior, is lost if only a single flagellar motor is used or either of the above mentioned modifications are not applied, as suggested by results from other simulations. The long uninterrupted runs due to the increased bias from doubling of the motor chain rate constants, the single motor bias amplification effect of the multiple motors and the voting hypothesis, and the elimination of the random aspect of tumbling rotation direction through the simplified tumbling update are all crucial for this effect.

Figure 4-18: Average density function result using the tumbling direction variation of BASM and $C_1(x)$, $v = 0.75s^{-1}$, $rs = 1\pi$ rad/s.
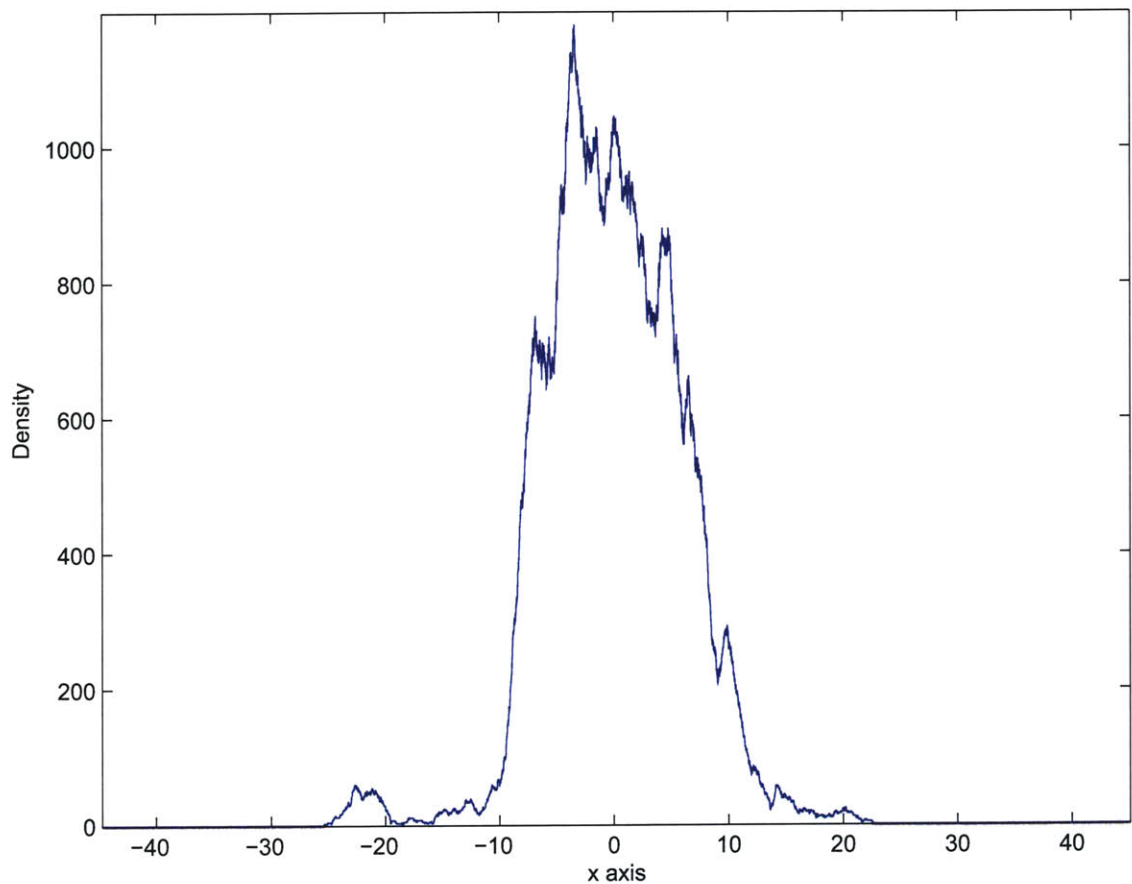
64
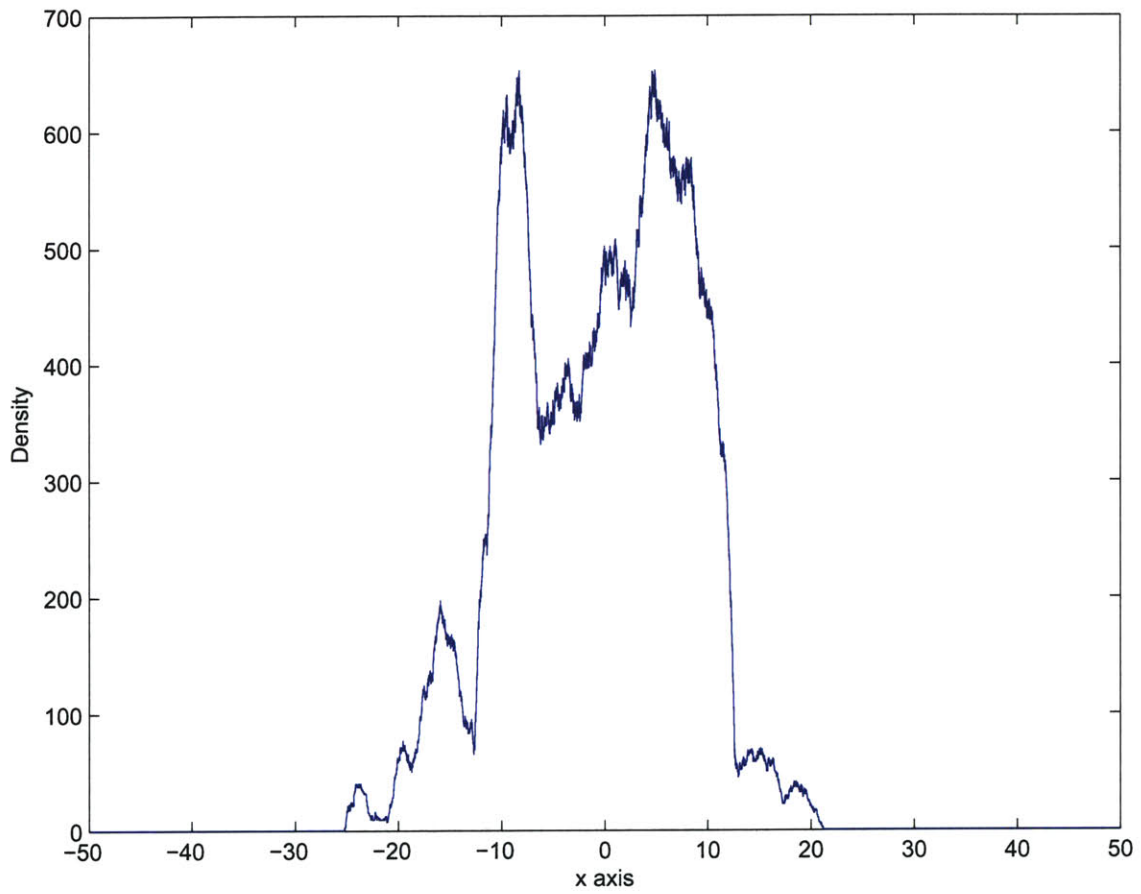
Figure 4-19: Average density function result using the tumbling variation of BASM and $C_2(x)$, $v = 0.6s^{-1}$, $rs = 1\pi$ rad/s.
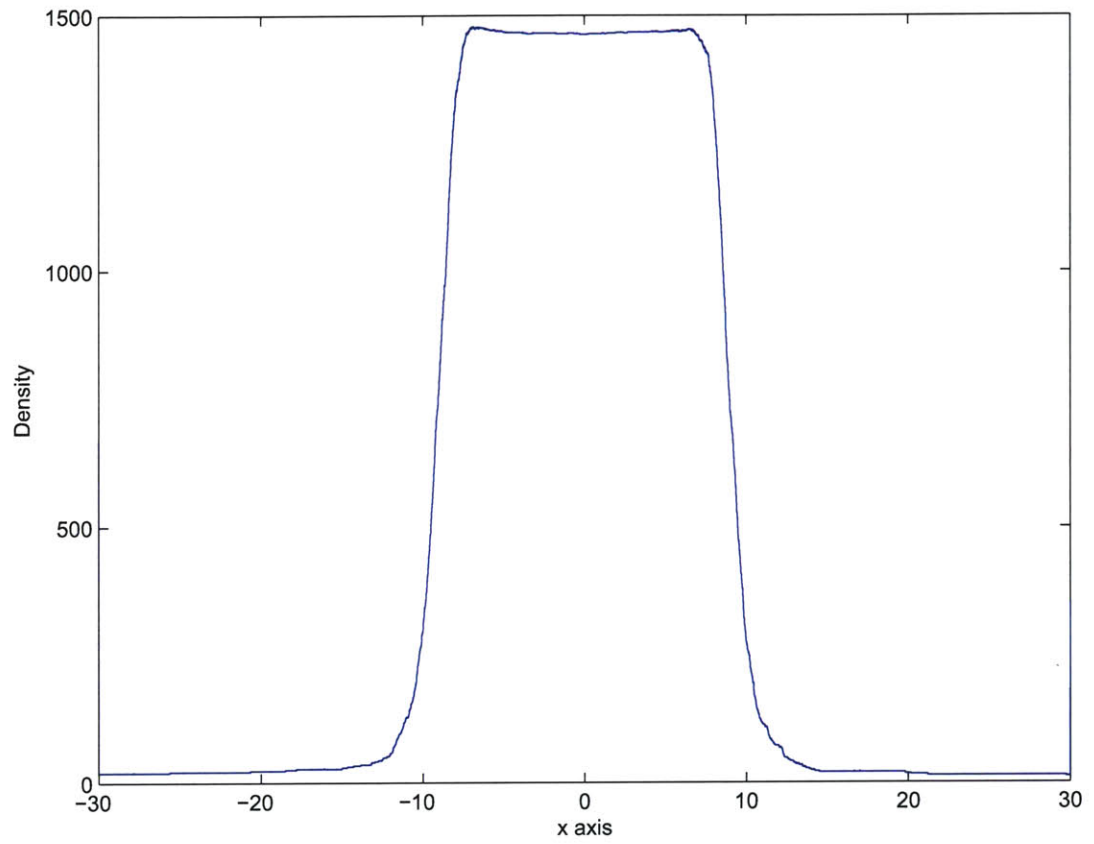
Figure 4-20: Average density function result using the tumbling variation of BASM, modified rate constants and $C_1(x)$, $v = 0.75s^{-1}$, $rs = 1\pi$ rad/s.

Figure 4-21: Internal state activity probability for CheY and the concentration at the location of the agent versus time for a typical surface mapping simulation implementing the tumbling direction modification, and using modified flagellar motor rate constants and the $C_1(x)$ surface.

4.2 Two-Dimensional Surface Mapping Simulations

In this section we present the results from BASM simulations using two test concentration surfaces that are two-dimensional extensions of $C_1(x)$ and $C_2(x)$:
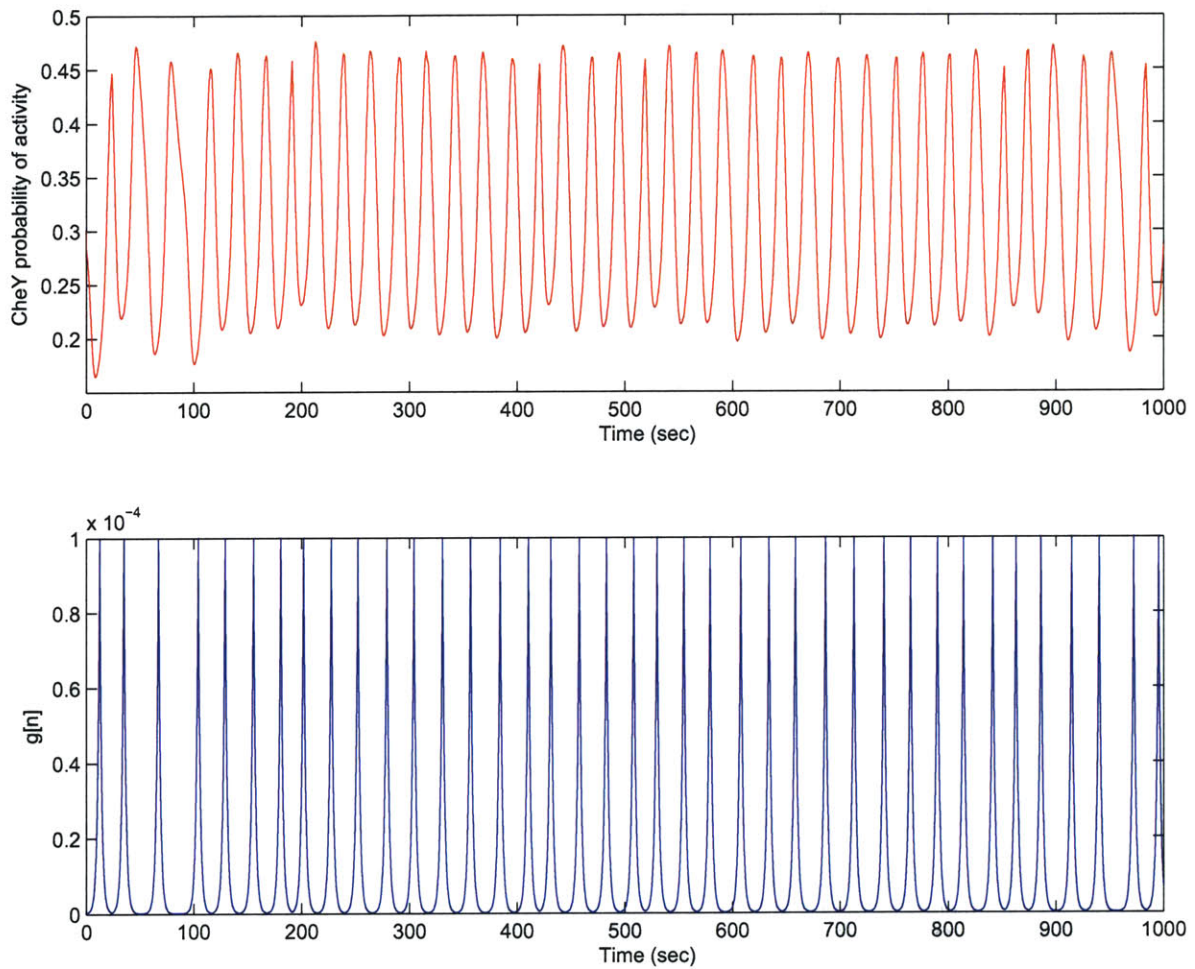
$$C_3(x, y) = 10^{-4} \exp(-\frac{1}{2}\sqrt{x^2 + y^2}) = (10^{-\frac{4}{3}})(C_1(\sqrt{x^2 + y^2}))^{\frac{2}{3}} \qquad (4.4)$$

$$C_4(x, y) = (5 \times 10^{-5}) \left| \frac{\sin(\frac{1}{4}x)\sin(\frac{1}{4}y)}{\frac{1}{16}xy} \right| = (5 \times 10^5) C_2(x) C_2(y) \qquad (4.5)$$

The unimodal function $C_3(x, y)$ and the multimodal function $C_4(x, y)$ are plotted in Figures 4-22 and 4-23. The results presented in this section are all produced by running BASM with the spatial spacing parameters $\triangle_x$ and $\triangle_y$ set to 2 and using $N_2 = 80$ simulations.

In the two-dimensional simulations, the initial position of the bacterial agent is always set to be a distance R away from the global maximum at the origin:

$$x[0] = R\cos(\Theta_i) \qquad (4.6)$$

$$y[0] = R\sin(\Theta_i) \qquad (4.7)$$

where Θ_i is a random number generated for the ith simulation distributed uniformly between 0 and 2π. The initial angular direction $\theta[0]$ is also set to an independent random number Φ_i with the same distribution. The results presented in this section are from simulations that use an R of 7.
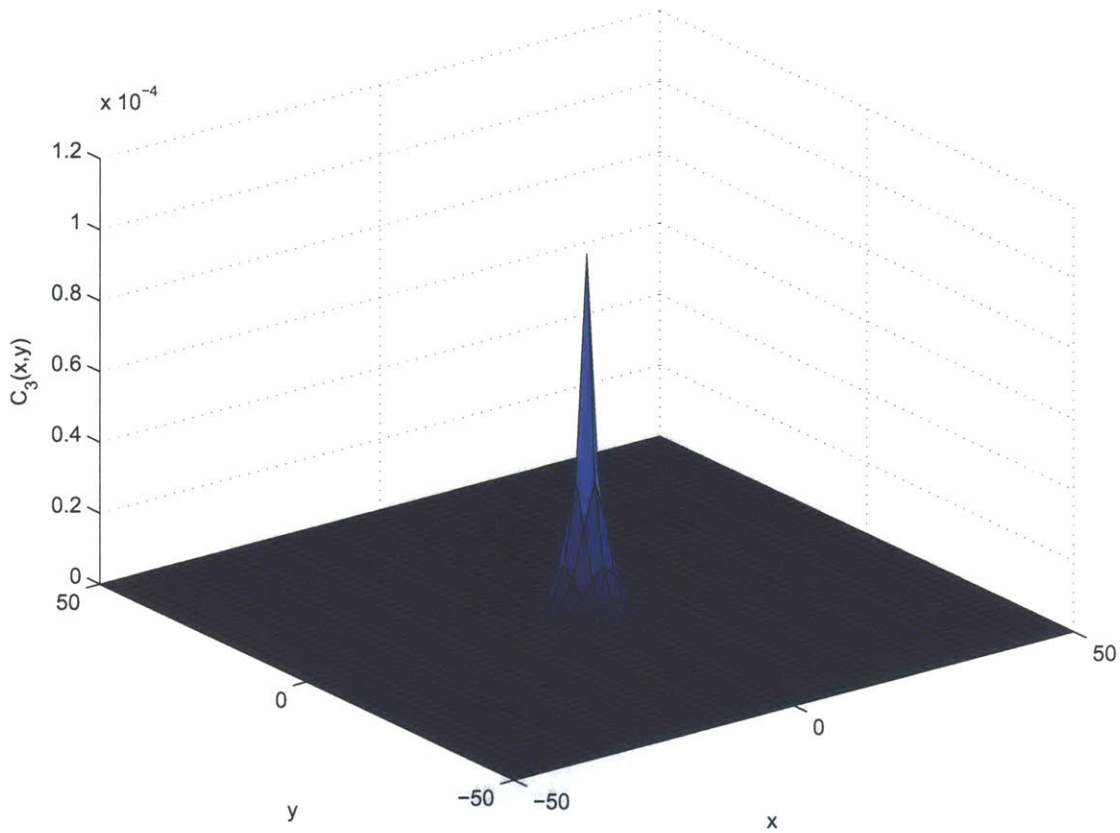
Figure 4-22: The two-dimensional unimodal test surface $C_3(x, y)$.

A default run speed v of $0.3\ s^{-1}$ and a tumbling rotation speed rs of 10π rad/s are used. The average density functions obtained are shown in Figure 4-24 for $C_3(x, y)$ and 4-25 for $C_4(x, y)$. In the unimodal case, there is a single clear peak in the density function. In the multimodal mapping, we can see the main lobe of the two-dimensional sinc function, as well as the four neighboring side lobes. The oscillating nature of the function is conveyed through the density function, and it is evident that the majority of the highest peaks lie along the x- and y-axes.

Figure 4-23: The two-dimensional multimodal test surface $C_4(x, y)$.

Figure 4-24: Average density function result from two-dimensional surface mapping simulations using $C_3(x, y)$.

Figure 4-25: Average density function result from two-dimensional surface mapping simulation, using $C_4(x, y)$.
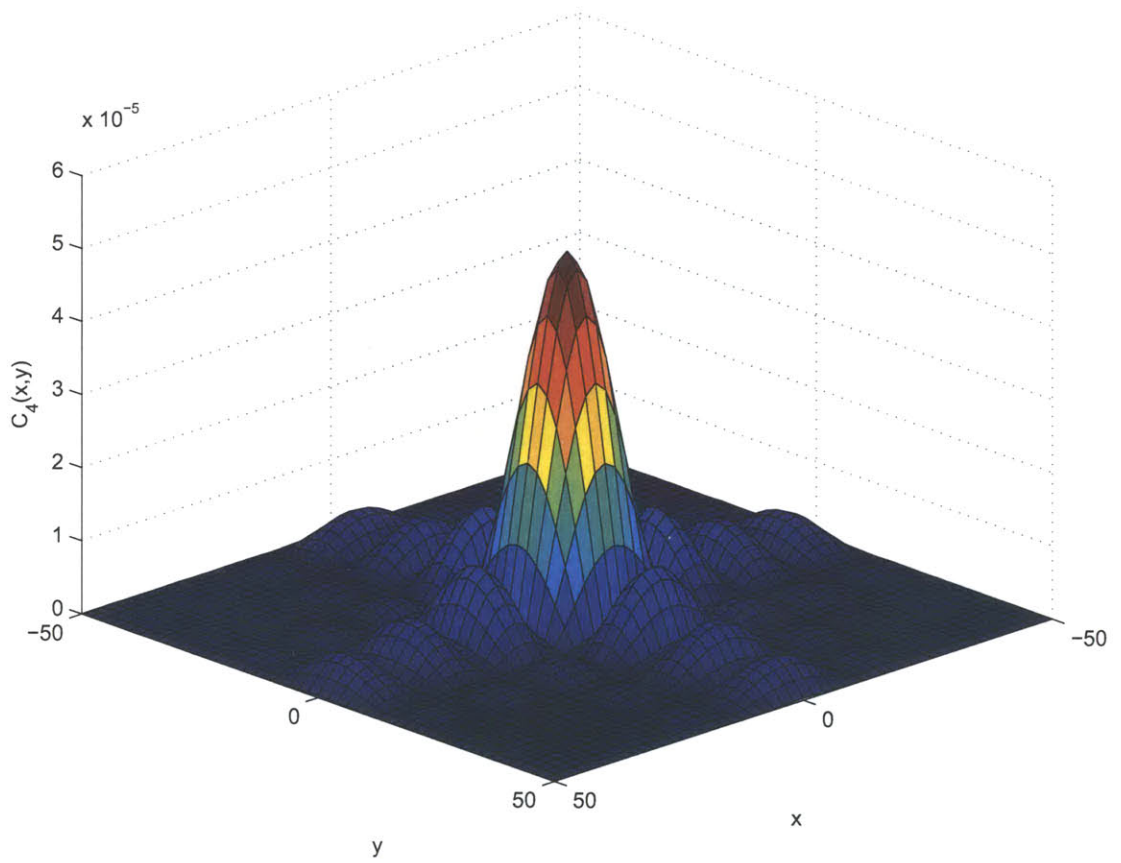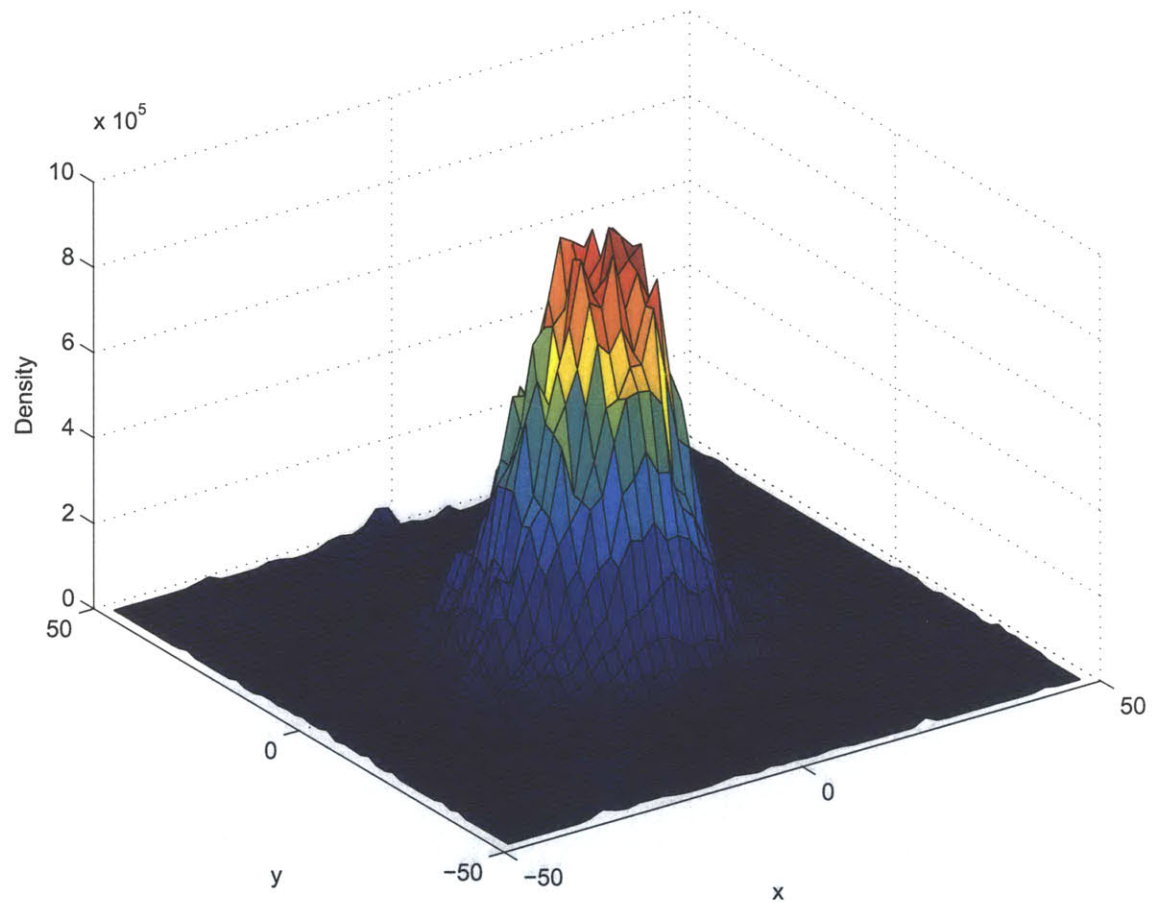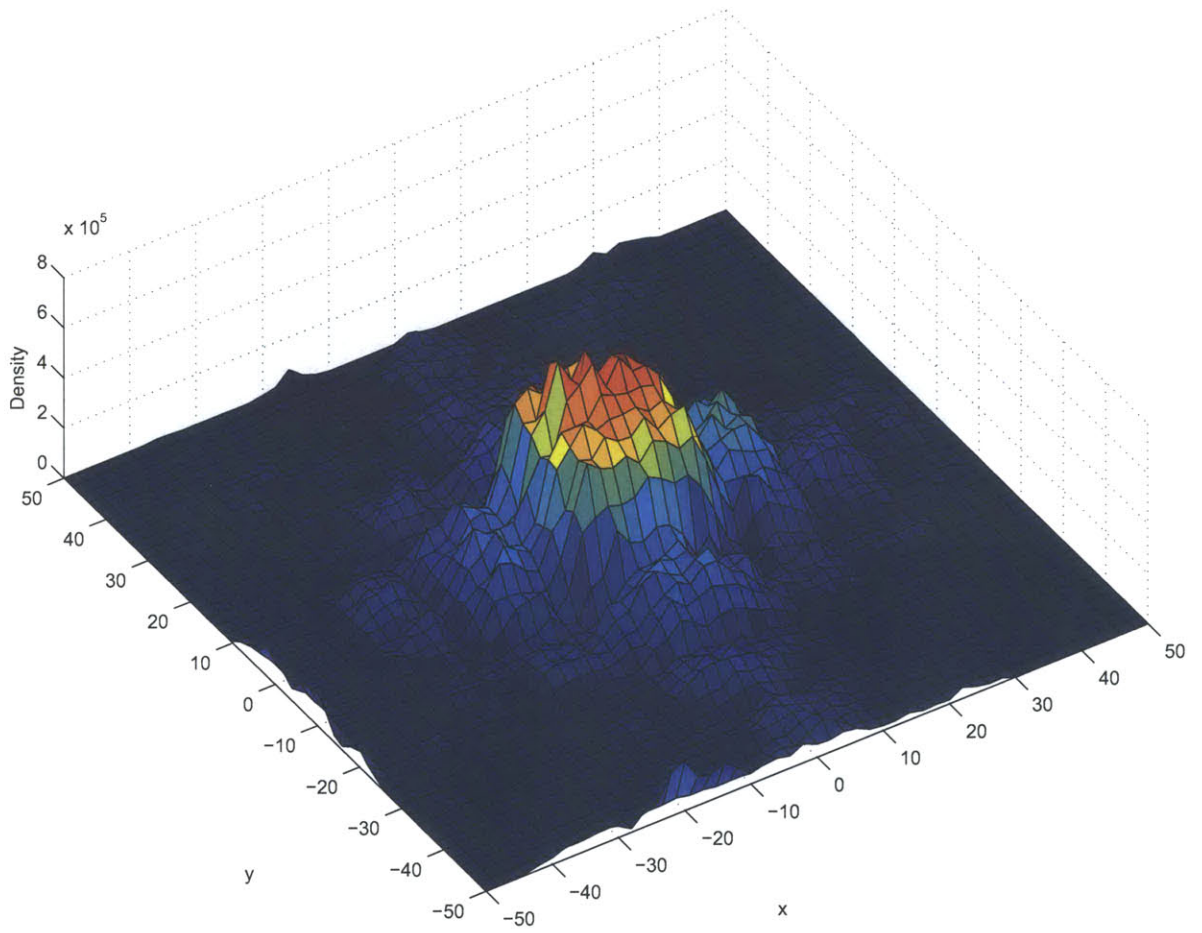
4.3 Range of Bacterial Agent Input Values

An important consideration when choosing a suitable value for the parameters of the BASM algorithm is the range of values that the surface $C(\cdot)$ can take at all locations of interest. As discussed in section 2.2.1, the a3MC model incorporates the effect of the time-varying input $g[n]$, where $g[n] = C(x[n], y[n])$ in the case of a two-dimensional surface and $g[n] = C(x[n])$ for the one-dimensional case, by calculating the receptor occupancy $\beta(g[n]) = \frac{g[n]}{g[n]+K_D}$. The bacterial agent can only experience the effect of the input through the nonlinear function $\beta(\cdot)$, so the choice of K_D can be crucial. All the test functions used so far have been scaled to match the size of the dissociation constant $K_D = 10^{-6}$ obtained from the literature on $E.coli$ chemotaxis. Consider the one-dimensional test surface:

$$C_5(x) = \cos\left(x\frac{2\pi}{20}\right) \tag{4.8}$$

This cosine function oscillates between 1 and -1 with a period of 20 distance units. However, this function takes on negative values which would not yield a meaningful receptor occupancy probability. To remedy this problem, we can introduce an offset variable O, that is added to the value of the surface input encountered by the bacterial agent before the receptor occupancy is calculated. O can be set to +1, the additive inverse of the minimum value of $C_5(x)$. A value of $K_D = 10^{-1}$ seems reasonable for use with $C_5(x)$, since the minimum concentration encountered will lead to a receptor occupancy of 0 percent, and the maximum possible receptor occupancy will be approximately 95.24 percent with the additive offset. Figure 4-26 shows the results of simulations using these parameters with $v = 1.25s^{-1}$, $rs = \pi$ rad/s, and the initial position of the bacterial agent set to an independent random number uniformly distributed between -50 and 50 for each simulation. A constant was subtracted from the density function such that it sums to 0, and both the smoothed density function and the surface are normalized such that their absolute value integrates to unity over the -50 to 50 interval.

Figure 4-26: Smoothed density and absolute error from one-dimensional surface mapping simulation using $C_5(x)$, $K_D = 10^{-1}$, $v = 1.25s^{-1}$, $rs = \pi$ rad/s.

Some prior knowledge about the surface goes a long way towards setting up surface mapping simulations with useful results. However, one cannot expect that such predictions about the surface will be completely reliable. The value of K_D might be chosen based on inaccurate expectations or assumptions about the surface. Another problem is that we may not be able to guarantee that the surface will not take on negative values lower than some threshold. For that scenario, we can adjust our algorithm such that O is an adaptive parameter, updated whenever an input lower than O is encountered by the bacterial agent. Figure 4-27 illustrates the added steps using a flowchart. We examined the case where, using the same surface $C_5(x)$, the dissociation constant is quadrupled, i.e. $K_D = 4 \times 10^{-1}$ and the offset was initially set

74

to 0.5. The results are shown in Figure 4-28, the smoothed density function obtained is comparable to the density function in the case with $K_D = 10^{-1}$ in terms of how closely it matches $C_5(x)$. The algorithm therefore still performs well with a range of possible settings for K_D and initial values for O.

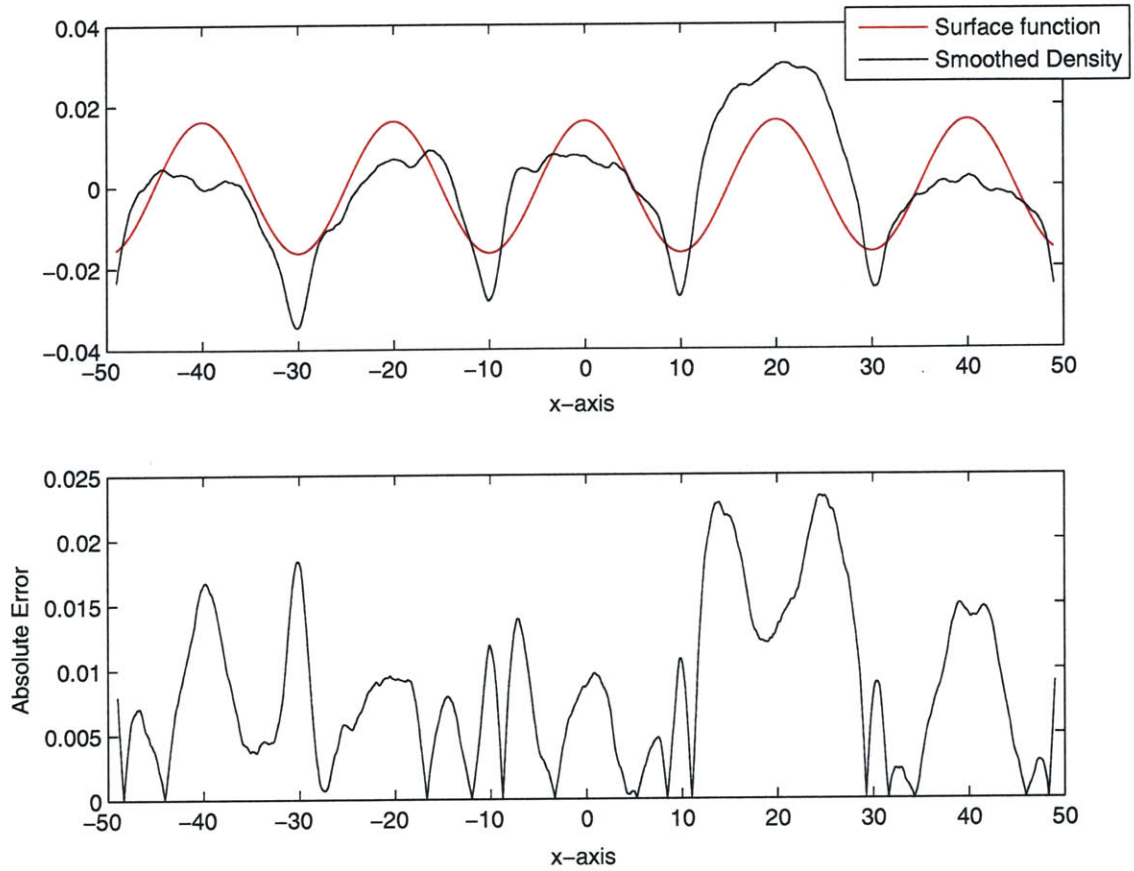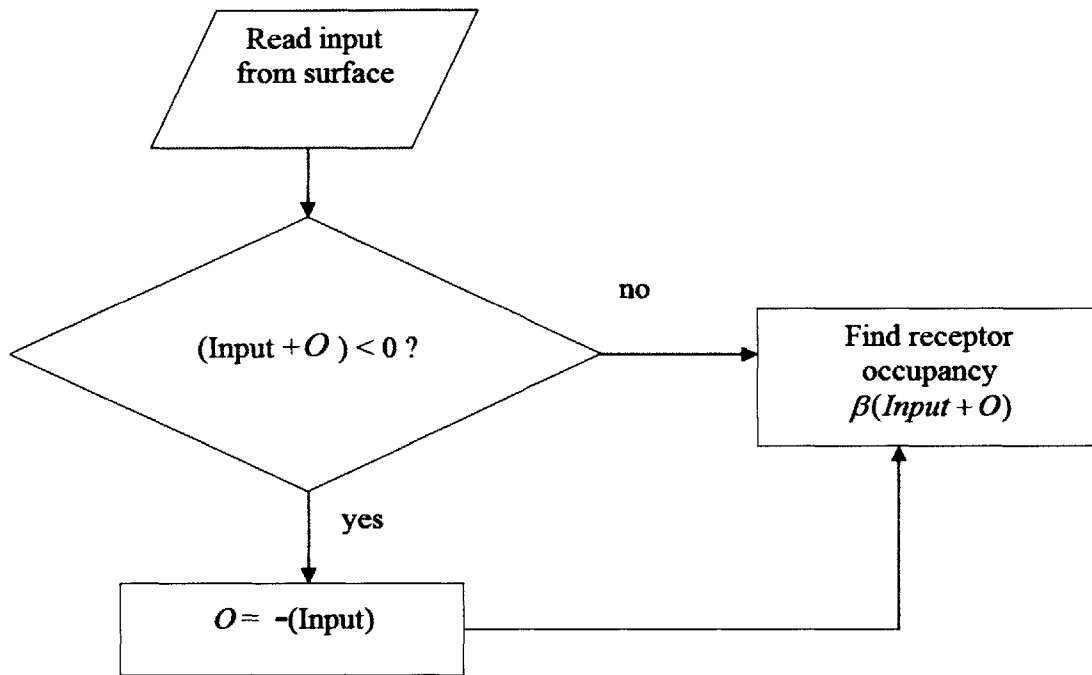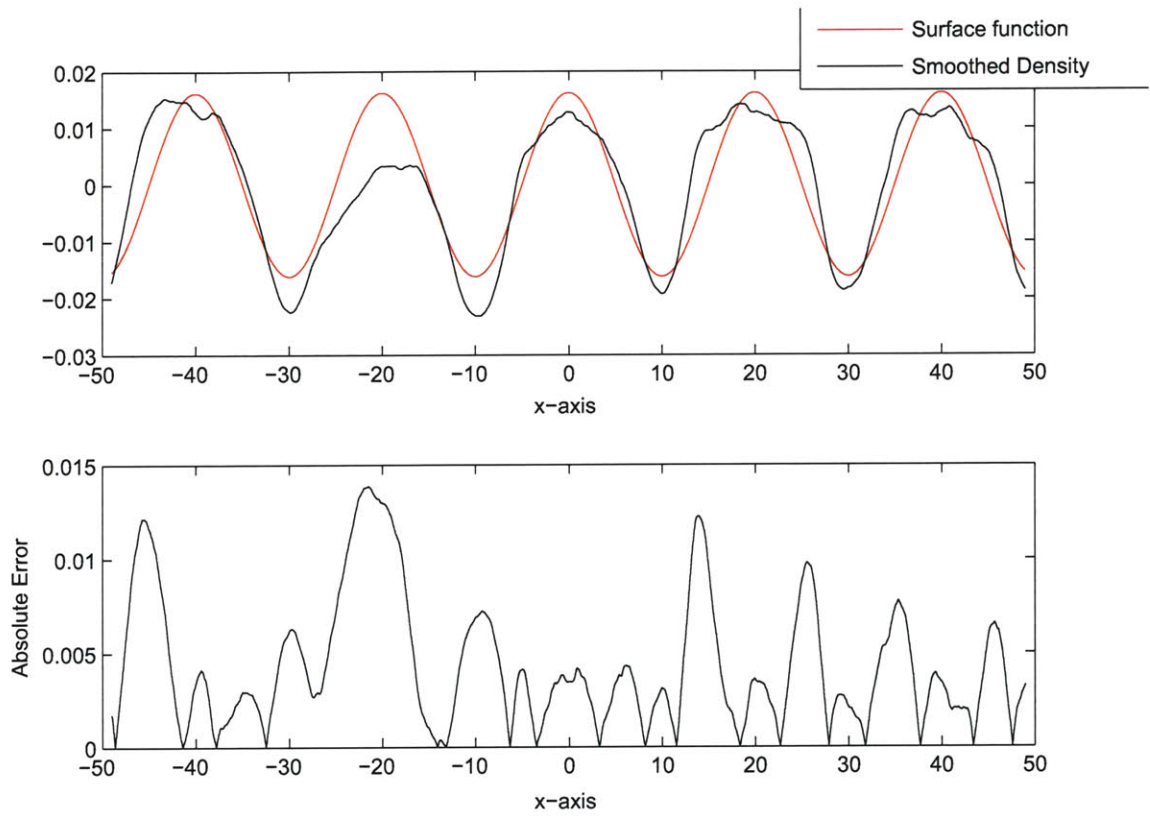Figure 4-27: Adaptive offset parameter implementation.

Figure 4-28: Smoothed density and absolute error from one-dimensional surface mapping simulation using $C_5(x)$, $K_D = 4 \times 10^{-1}$, $v = 1.25s^{-1}$, $rs = \pi$ rad/s.

4.4 Surface Flattening

In some applications, it may be advantageous to use bacterial agents that actively modify the function landscape as they use the information they gather about the surface to bias their exploration. For example, consider a group of mobile active sensors that transmit a signal that can be detected or measured by other sensors. Such active sensors that move in a physical setting according to the BASM algorithm and use total sensor signal strength at their position for their input can play the role of bacterial agents exploring a sensor coverage surface. The bacterial agents can seek out areas with a lower signal strength, while simultaneously affecting the (time-varying) sensor coverage surface with their movement. This will effectively lead to a sensor coverage function that is more evenly distributed across the spatial coordinates, or said differently, a flattening of the surface that indicates a local time-averaged sensor coverage or similar metric. In this section, the surface mapping algorithm is taken a step further, by allowing the bacterial agents to consume or reduce the value of a function at the locations they visit. The effects can be observed as the bacterial agents are allowed to modify the surface they are navigating instead of just passively sampling it. They essentially flatten the concentration surface (to approximately zero) as they visit the areas with higher concentrations more often and reduce the total amount of the substance.

In the one-dimensional surface flattening algorithm, a uniform grid of the x-axis is formed as before:

$$x_k = k \times v \times dt \tag{4.9}$$

for $|x| \leq x_{max}$ or equivalently, $|k| < \frac{x_{max}}{vdt}$. The time-varying concentration function is denoted by $C[k, n]$, where k is a spatial index and n is the time index. At the beginning of the simulation $C[k, n]$ is initialized to a sampled version of the concentration surface $C(x)$:

$$C[k, 0] = C(x_k) \tag{4.10}$$

For every time step, if the flagellar motors are in a run mode, the current position $x[n]$

is rounded to the nearest value of x_k and the discrete-space concentration function $C[k, n]$ is reduced at the corresponding k by a factor of γ:

$$C[k, n + 1] = \gamma C[k, n] \tag{4.11}$$

The surface value the bacterial agent reads at that location is the value of the time-varying surface (interpreted as a concentration) at the nearest grid point, i.e. $C[k, n]$. Any position updates due to a run that would lead to a value of x[n] outside the range $|x| < x_{max}$ are prevented. We set x_{max} to 90 for the one-dimensional simulations.

We investigated whether the bacterial agent is more effective at reducing the total amount of a substance than an unbiased version of the random walk, implemented using a bacterial agent that always senses a complete lack of attractant everywhere (zero concentration), and is therefore not influenced by the concentration surface. Two sets of simulations were performed, and for each simulation, the total amount of remaining substance, denoted by $S[n]$, was used as a metric of how fast the two algorithms flatten the surface. The surface sums from 20 simulations are averaged to obtain an estimate of the expected amount of remaining substance at time n as follows:

$$S_{avg}[n] = \frac{1}{20} \sum_{i=1}^{20} S_i[n] = \frac{1}{20} \sum_{i=1}^{20} \sum_{k=-\frac{x_{max}}{vdt}}^{\frac{x_{max}}{vdt}} C_i[k, n] \tag{4.12}$$

where i denotes the simulation number. We used $C_2(x) = (5 \times 10^{-5}) \left| \frac{\sin(\frac{1}{4}x)}{\frac{1}{4}x} \right|$ as the surface, a v of 0.75 s^{-1}, a tumbling rotation speed rs of π rad/s, and a reduction factor γ of 0.8. Figure 4-29 shows the obtained average running sum of the surface for the surface biased and unbiased versions of the random walk.

4.4.1 Surface Flattening with Two Bacterial Agents acting Simultaneously

In this section we explore the case where two bacterial agents are simultaneously navigating and modifying the surface. Similarly, the ability of two bacterial agents
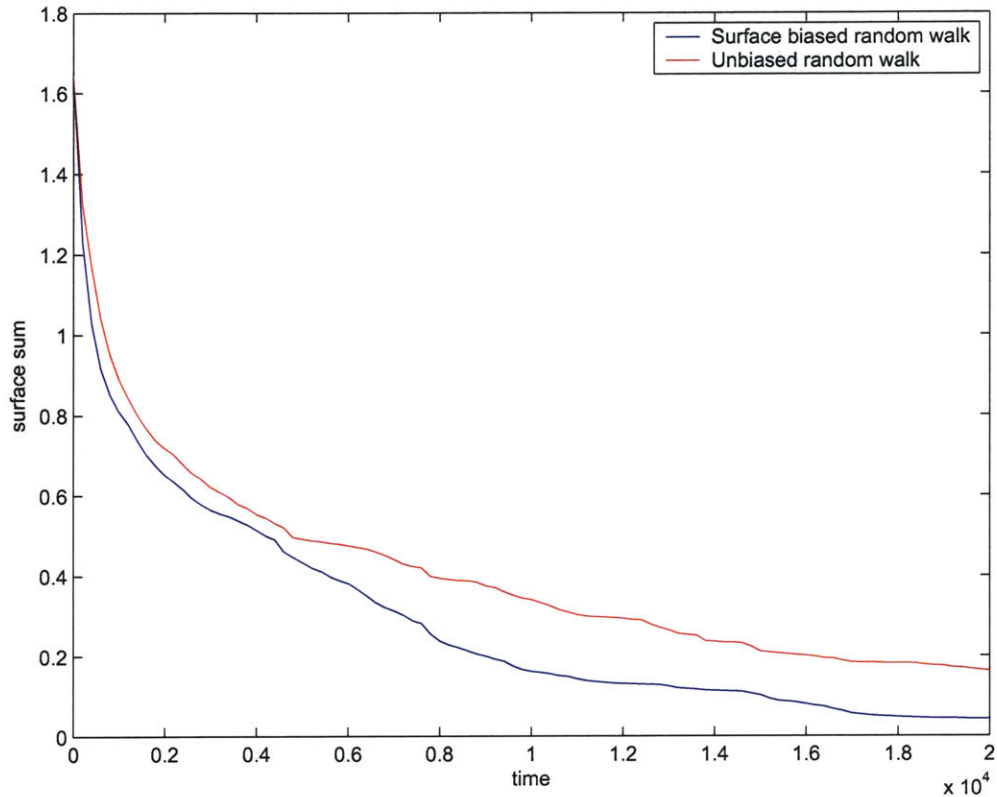
Figure 4-29: Average remaining total substance on the one-dimensional concentration surface from surface flattening simulations using the surface biased and unbiased versions of the random walk.

working in concert to flatten the surface was compared with the performance of two bacterial agents that always perceive a zero concentration and perform an unbiased random walk. The result is shown in Figure 4-30, where again the biased random walk is consistently more likely to be ahead in the race to reduce the total amount of substance remaining (the surface sum).

4.4.2 Two-Dimensional Surface Flattening

In this section the surface flattening algorithm is extended to two-dimensional surfaces. We again attempt to gauge how much the bacterial agent benefits from sensing concentrations locally and moving accordingly in such an experiment. In this case, the time-varying surface is a function of two spatial indices k_1 and k_2, related to the
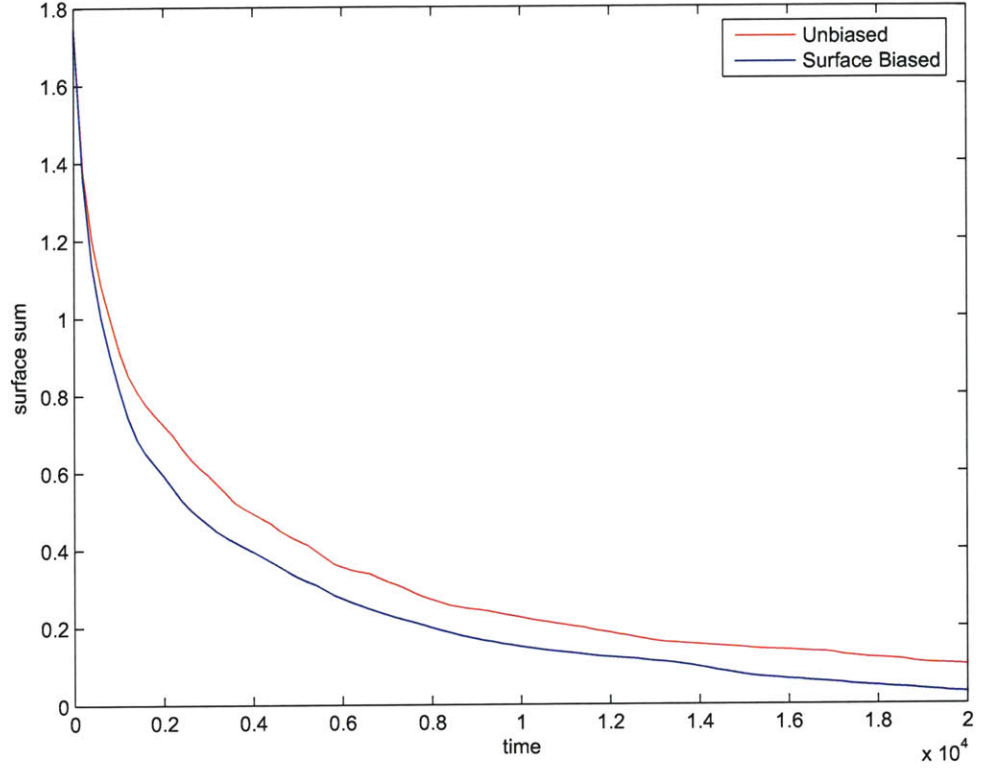
Figure 4-30: Average remaining total substance on the concentration surface using the surface biased and unbiased versions of the random walk, from simulations where two bacterial agents simultaneously modify the surface.

continuous spatial variables as follows:

$$x_{k_1} = k_1 \times 50 \times v \times dt \qquad (4.13)$$

$$y_{k_2} = k_2 \times 50 \times v \times dt \qquad (4.14)$$

This acts as a two-dimensional grid of finite size for $|x_{k_1}| \leq x_{max}$ and $|y_{k_2}| \leq y_{max}$. At the beginning of the simulation $C[k_1, k_2, n]$ is initialized to a sampled version of the concentration surface $C(x, y)$:

$$C[k_1, k_2, 0] = C(x_{k_1}, y_{k_2}) \qquad (4.15)$$

For every time step, if the flagellar motors are in a run mode, the current position coordinates $x[n]$ and $y[n]$ are rounded to the nearest values of x_{k_1} and y_{k_2} and the sampled concentration function is reduced at the corresponding indices by a factor of γ:

$$C[k_1, k_2, n + 1] = \gamma C[k_1, k_2, n] \tag{4.16}$$

Any position updates due to a run that would lead to a value of x[n] or y[n] outside the range $|x| \leq x_{max}$, $|y| \leq y_{max}$ are prevented. x_{max} and y_{max} are set to 90 for the two-dimensional simulations.

The surface sums from 20 simulations are averaged to calculate an estimate of the expected amount of remaining substance at time n, as follows:

$$S_{avg}[n] = \frac{1}{20} \sum_{i=1}^{20} S_i[n] = \frac{1}{20} \sum_{i=1}^{20} \sum_{k_1 = -\frac{x_{max}}{50vdt}}^{\frac{x_{max}}{50vdt}} \sum_{k_2 = -\frac{y_{max}}{50vdt}}^{\frac{y_{max}}{50vdt}} C_i[k_1, k_2, n] \tag{4.17}$$

We used $C_4(x, y) = (5 \times 10^{-5}) \left| \frac{\sin(\frac{1}{4}x)\sin(\frac{1}{4}y)}{\frac{1}{16}xy} \right|$ as the surface, a v of 5 s^{-1}, a tumbling rotation speed rs of 5π rad/s, and a reduction factor γ of 0.8. Figure 4-31 shows the obtained average running sum of the surface for the surface biased and unbiased versions of the random walk.
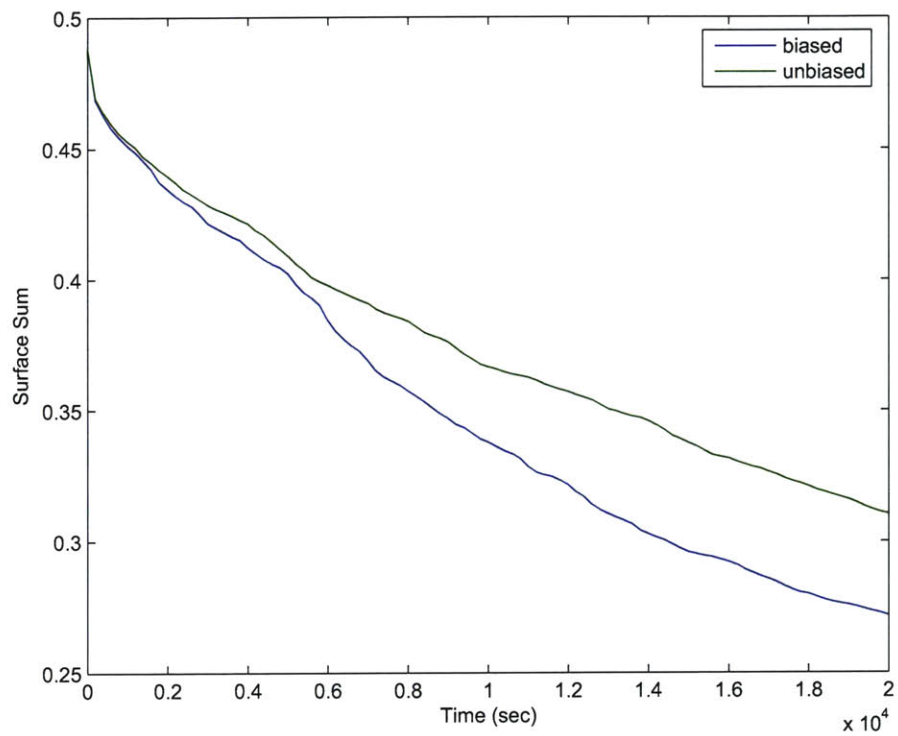
Figure 4-31: Average remaining total substance from two-dimensional surface flattening simulations, using the surface biased and unbiased versions of the random walk.

Bibliography

[1] W. Alt. Biased random walk models for chemotaxis and related diffusion approximations. *J Math Biol.*, 9(2), 1980.

[2] N. Barkai and S. Leibler. Robustness in simple biochemical networks. *Nature*, 387(6636), 1997.

[3] S.M. Block, J.E. Segall, and H.C. Berg. Impulse responses in bacterial chemotaxis. *Cell*, 31(1), 1982.

[4] D. Bray, R.B. Bourret, and M.I. Simon. Computer simulation of the phosphorylation cascade controlling bacterial chemotaxis. *Mol. Biol. Cell*, 4(5), 1993.

[5] A. Bren and M. Eisenbach. How signals are heard during bacterial chemotaxis: protein-protein interactions in sensory signal propagation. *J. Bacteriol.*, 182(24), 2000.

[6] D. Corne, M. Dorigo, and F. Glover. *New Ideas in Optimization*. McGraw-Hill, 1999.

[7] Bray D. Bacterial chemotaxis and the question of gain. *Proc. Natl. Acad. Sci. USA*, 99(1), 2002.

[8] Y.C. Eldar. *Quantum Signal Processing*. PhD thesis, MIT, 2001.

[9] A. Ishihara, J.E. Segall, S.M. Block, and H.C. Berg. Coordination of flagella on filamentous cells of *escherichia coli*. *J. Bacteriol.*, 155(1), 1983.

[10] Segall JE, Block SM, and Berg HC. Temporal comparisons in bacterial chemotaxis. *Proc. Natl. Acad. Sci. USA*, 83(23), 1986.

[11] M.S. Jurica and B.L. Stoddard. Mind your b's and r's: bacterial chemotaxis, signal transduction and protein recognition. *Structure*, 6(7), 1998.

[12] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598).

[13] J.R. Koza. Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314, Stanford University, Computer Science Department, 1990.

[14] R.M. Macnab and D.E. Koshland Jr. The gradient-sensing mechanism in bacterial chemotaxis. *Proc Natl Acad Sci U S A*, 69(9), 1972.

[15] W.G. Macready and D.H. Wolpert. Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on Evolutionary Computing*, 2(1), 1998.

[16] P. Magni, R. Bellazzi, and G. De Nicolao. Bayesian function learning using mcmc methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12), 1998.

[17] C.J. Morton-Firth. *Stochastic simulation of cell signalling pathways*. PhD thesis, University of Cambridge, 1998.

[18] C.J. Morton-Firth, T.S. Shimizu, and D. Bray. A free-energy-based stochastic simulation of the tar receptor complex. *J. Mol. Biol.*, 286(4), 1999.

[19] S. Mueller, J. Marchetto, S. Airaghi, and P. Koumoutsakos. Optimization based on bacterial chemotaxis. *IEEE Trans. on Evolutionary Computation*, 6(1), 2002.

[20] J. Pallotta and L.G. Kraft. Two dimensional function learning using cmac neural networks with optimized weight smoothing. *Proceeding of 1999 ACC*, 1999.

[21] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 1989.

[22] J.-M. Renders and S.P. Flasse. Hybrid methods using genetic algorithms for global optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 26(2), 1996.

[23] M.R. Said. *Signal Processing in Biological Cells: Proteins, Networks, and Models*. PhD thesis, MIT, 2005.

[24] M.R. Said, A.V. Oppenheim, and D.A. Lauffenburger. Modeling cellular signal processing using interacting markov chains. *Proc. Int. Conf. on Acoustics, Speech, Signal Processing (ICASSP)*, 2003.

[25] A.C. Singer, A.V. Oppenheim, and G.W. Wornell. Detection and estimation of multiplexed soliton signals. *IEEE Transactions on Signal Processing*, 47(10), 1999.

[26] P.A. Spiro, J.S. Parkinson, and H.G. Othmer. A model of excitation and adaptation in bacterial chemotaxis. *Proc. Natl. Acad. Sci. USA*, 94(14), 1997.

[27] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.

[28] G.W. Wornell. *Signal Processing with Fractals: A Wavelet-Based Approach*. Prentice Hall, 1996.

[29] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9), 1999.