
A COMPUTATIONAL ANALYSIS OF THE
GRADIENT NAVIGATION STRATEGIES OF THE
NEMATODE *Caenorhabditis elegans*

Serge Thill

Thesis submitted for the degree of
Doctor of Philosophy

Leicester, February 2008

Supervisors:

Dr Tim C. Pearce, Department of Engineering

Dr Tom Matheson, Department of Biology



**University of
Leicester**

Department of Engineering

Acknowledgments

Thanks go to **Tim Pearce** and **Tom Matheson** for supervising this thesis.

Howard Baylis and **Denise Walker** from the University of Cambridge have given me the opportunity to work with real animals and gain experience in experimental work. Discussions with Howard were very helpful and enlightening and Denise's help in setting up and conducting the experiments was invaluable. For this, I am very grateful.

Andreas Kaltenbrunner from the Computational Neuroscience Group at the Universitat Pompeu Fabra (Barcelona) has been an inspiration through many insightful and interesting discussions. In particular, his suggestions for improvements of some of the figures in the manuscript for the first paper were highly appreciated. He has also been the driving force behind the decision to move away from the *Lyx* Editor and write the present thesis in pure $\text{\LaTeX} 2_{\epsilon}$, and has provided the basis class for setting the thesis (which has since been modified and adapted to the requirements of the present work).

Manuel Sánchez-Montañés from the Escuela Politécnica Superior of the Universidad Autónoma de Madrid has provided valuable insights related to the optimisation approaches also used in the first paper.

Thanks also go to all my friends, wherever they may be in the world, for being who they are and for enriching my time in Leicester and my life in general.

And finally, a big thanks to my family who have supported me through all the years of higher

education, from Exeter over Edinburgh to Leicester, now coming to a close with the present thesis. It would not have been possible without them.

The author has been supported by the **Neuro-IT** project funded by the CEC Information Science and Technology, under the Future and Emerging Technologies programme.

© Serge Thill, 2008.

This thesis is copyright material and no quotation from it may be published without proper acknowledgment.

Peer-refereed papers

Thill, S. and Pearce, T. C. (2007) “Understanding complex behavior by analysing optimized models: *C. elegans* gradient navigation.” *HFSP Journal* 1(4), pp 263-273. (Also selected for the *Virtual Journal of Biological Physics Research*)

Sánchez-Montañés, M., **Thill, S.** and Pearce, T. C. “Optimization techniques for the analysis of behavior” *In preparation*.

Thill, S. and Pearce, T. C. “Computational considerations for isotherm tracking in *C. elegans*” *In preparation*.

Poster presentations

Thill, S. “A Simple Framework for Analysing Complex Behaviour.” *5th European Neuro-IT and Neuroengineering School, Delmenhorst, Germany, 15th-20th July 2007.*

A Computational Analysis of the Gradient Navigation Strategies of the Nematode *Caenorhabditis elegans*

SERGE THILL

In the present thesis, we apply computational methods to the study of animal behaviour. Specifically, we are interested in the gradient navigation strategies of *C. elegans*, for which we show that there are many interesting questions that have not yet been answered by existing research.

In order to study the behaviour of *C. elegans*, we first develop a range of tools to help us do so. We base a large part of our work on Markov-like models of behaviour and since these are not Markovian in the strict sense (limiting the analytical tools which can be used to study their behaviour), we first present a possible transformation from a Markov-like model with variable transition probabilities into a strictly Markovian model. We next present a framework for studying the behaviour of behavioural models which is not restricted to the work presented here but is likely to find general use in behavioural studies.

Using these tools, we then analyse the chemotactic behaviour of *C. elegans*, showing that we can adequately explain most features of this behaviour using energy-efficiency considerations. We also show that the main behavioural strategy, so-called *pirouettes* is likely to be caused by an inability to sample the environment during a turn and that the animal may not be acting upon gradient information while reversing.

Finally, we investigate the deterministic isotherm tracking strategy displayed by *C. elegans*. We develop a computational model for this behaviour which is able to reproduce all of the main features of *C. elegans* isotherm tracking and we propose a candidate neural circuit which might encode this strategy. Additionally, we briefly discuss the use of stochastic strategies by the animal when moving towards its preferred temperature.

In summary, the work presented here therefore provides contributions to two major fields: we extend the methodology available for behavioural analysis in ethology and we contribute a number of insights and advancements to the field of *C. elegans* research.

Contents

Acknowledgments	ii
Publications	iv
Abstract	v
Table of Contents	vi
List of Figures	x
List of Tables	xii
I Introduction	1
1 Aims and Motivations	2
1.1 Organisation of this thesis	4
2 Literature Review	5
2.1 The nematode <i>C. elegans</i>	5
2.1.1 Etymology	5
2.1.2 General facts	6
2.2 The neural architecture of <i>C. elegans</i>	7
2.3 Neurophysiology	9
2.3.1 <i>C. elegans</i> neurons	9
2.3.2 <i>C. elegans</i> Synapses	11
2.3.3 Implications of the lack of neurophysiological data for the present thesis . .	12
2.4 The behaviour of <i>C. elegans</i>	13
2.4.1 Locomotion	13
2.4.2 Feeding and social behaviours	18
2.4.3 Gradient navigation behaviours	19
2.4.4 Thermotaxis	22
2.4.5 Electrical fields navigation	27
2.5 Ethological considerations for modelling and analysing behaviour	28
2.5.1 The context of the present work within the field of Ethology	28
2.5.2 Modelling behaviour	29
2.5.3 Analysing models of behaviour	30
2.6 Summary of Chapter 2	30

II	Modelling and Analysing Behaviour	32
3	Translating Markov-like models of behaviour into strict models	33
3.1	A model of <i>C. elegans</i> gradient navigation	36
3.1.1	Translation into a strict Markov model	38
3.2	Moth behaviour as a dual Markov-like model	41
3.2.1	Moth behaviour	41
3.2.2	A dual Markov-like model of the moth plume navigation behaviour	42
3.2.3	Conversion into a strictly Markovian model	44
3.2.4	Goal states and optimisation in a finite grid world	47
3.3	Computational implementation	49
3.3.1	Properties of absorbing Markov chains	49
3.3.2	Considerations for the implementation in MATLAB	50
3.3.3	Optimised values	51
3.3.4	Optimisation algorithm	52
3.4	Optimised source location behaviour	53
3.4.1	Optimisation of single properties	55
3.4.2	Optimisation of combined properties	55
3.4.3	Comparison with numerical simulations	58
3.5	Summary of Chapter 3	59
3.5.1	Uses and limits of the translation into strict Markov models	60
3.5.2	Moth navigation towards the source of a plume	61
4	A framework for the systematic analysis of behavioural models	63
4.1	Using optimisation to analyse behaviour	63
4.2	Formal definition of the recursive algorithm	70
4.3	Analysing the set of optimal solutions	72
4.4	Other applications for the framework	76
4.5	Summary of Chapter 4	80
III	<i>C. elegans</i> gradient navigation	81
5	Stochastic gradient navigation strategies of <i>C. elegans</i>	82
5.1	Optimisation of the model	83
5.1.1	Parameter space of the model	83
5.1.2	Definition of the criterion for optimal performance	84
5.1.3	Model assessment	85
5.1.4	Simulation of worms	86
5.1.5	Simulated annealing	86
5.2	<i>C. elegans</i> gradient navigation strategies	87
5.2.1	Probability distributions show run shortening during down-gradient navigation	87
5.2.2	Ranking of the behavioural units	89
5.2.3	Changing the criteria for optimal performance has little impact on results	91
5.2.4	Markovian properties reveal the use of pirouettes	92

5.2.5	Pirouettes may emerge in part from an inability to sample the gradient during a turn	95
5.2.6	Performance on planar gradients	96
5.2.7	Models naturally dwell at the peak	97
5.2.8	Gradient information during reversals leads to unnatural behaviour	97
5.3	The family of solutions can be used for investigations of novel situations	99
5.3.1	Quadrant assays	101
5.3.2	Responses to step changes in concentration	102
5.3.3	Virtual mutants in quadrant assays	105
5.4	Summary of Chapter 5	107
5.4.1	Components of <i>C. elegans</i> gradient navigation	107
5.4.2	Evaluating computational models in novel tasks	109
5.4.3	On the computational analysis of behaviour	110
6	Deterministic isotherm tracking of <i>C. elegans</i>	111
6.1	Issues with the existing model of isotherm tracking	113
6.1.1	Definition of the existing model	113
6.1.2	Behaviour of the model	115
6.2	Modelling the animal	118
6.3	Available sensory information during one headsweep	122
6.4	A candidate neural circuit for isotherm tracking	129
6.4.1	Restrictions imposed by our model	129
6.4.2	Candidate neurons for computing course correction	131
6.4.3	On computing the second derivative	132
6.4.4	Plausibility of the isotherm tracking strategy	134
6.5	On the use of stochastic strategies when navigating towards the peak of gradients	134
6.6	Summary of Chapter 6	136
IV	Conclusions	138
7	Summary and discussion	139
7.1	Innovations in the analysis of models of behaviour	140
7.1.1	Ways of determining the performance of Markov-like models	140
7.1.2	Using Markov-like models to analyse behaviour	141
7.2	Contributions to the field of <i>C. elegans</i> research	143
7.2.1	Analysis and characterisation of the stochastic gradient navigation strategy	143
7.2.2	A new model for isotherm tacking	144
7.2.3	On the use of stochastic rather than deterministic strategies	146
7.3	Conclusion	148
8	Suggestions for further work	149
8.1	Extending the framework for analysing behavioural models	149
8.2	Experimental verification of predictions for <i>C. elegans</i>	150
8.2.1	Isotherm strategy	150

8.2.2	Resolution and operating range of chemosensory neurons	150
8.3	Behavioural work on <i>C. elegans</i>	150
8.3.1	Isotherm strategy	150
8.3.2	The use of stochastic strategies for navigating towards the preferred temperature	151
V	Appendix	152
A	List of neurons in <i>C. elegans</i>	153
	Bibliography	168

List of Figures

2.1	The nematode <i>C. elegans</i>	6
2.2	The layout of the nervous system in <i>C. elegans</i>	7
2.3	Tracks of 3 worms allowed to wander for 20 minutes in a uniform gradient	14
2.4	Tracks of 4 worms allowed to wander for 5 minutes in a radial gradient	14
2.5	A candidate circuit for locomotion in <i>C. elegans</i>	16
2.6	The separation of tracks into runs and pirouettes	20
2.7	Part of the chemo-sensory circuit in <i>C. elegans</i>	22
3.1	A simple Markov-like model of behaviour	34
3.2	Three state model of <i>C. elegans</i> behaviour	37
3.3	Four state model of moth behaviour	43
3.4	Visual representation of the strict moth model	45
3.5	Behaviour of the single value optimisation results	54
3.6	Behaviour of the combined value optimisation result	56
3.7	Numerical simulation of the result	58
4.1	Parameter space division examples	66
4.2	Recursive division example	67
4.3	The importance of a good division	69
4.4	Simple decision tree example	74
4.5	A random parameter space	77
5.1	Score distribution	85
5.2	Transition probabilities distribution	87
5.3	ID3 decision tree	89
5.4	Mean first passage times	92
5.5	Transition probabilities distribution	97
5.6	Example tracks	98
5.7	Quadrant assay setup	101
5.8	Upstep response	103
5.9	Downstep response	104
6.1	Behaviour of the Luo et al. (2006) model	113

6.2	Course correction of the model	116
6.3	Model head position	118
6.4	Approximating radial gradients with planar simulations	121
6.5	Amplitude information during one headsweep	123
6.6	Relationship between minima of Eqn. 6.14 and required course correction	126
6.7	Isotherm tracking performance	128
6.8	Candidate neural circuit for isotherm tracking	130

List of Tables

3.1	Single value optimisation results	53
3.2	Combined value optimisation result	57
4.1	Performance in local minima problems	78
4.2	Time requirements for equivalent performance	79
5.1	Ranking of the behavioural units	90
5.2	Transition time summary	105
6.1	Candidate neurons for CORR	131

Part I

Introduction

THE nematode *Caenorhabditis elegans* is a popular organism for behavioural and neuroscientific studies (See *e.g.* de Bono and Maricq, 2005). This is mainly due to the comparative simplicity of its behaviour, the fact that the complete connectivity of its neural circuit is known (White et al., 1986) and the fact that it is usually possible to destroy specific neurons and observe the resulting change in behaviour.

It thus has to be one of the main goals of the behavioural research on *C. elegans* to describe, quantify and understand the behaviour of the nematode as completely as possible. While reviewing the literature (Chapter 2), however, we find that several interesting questions have not yet been adequately addressed. For instance, the animal is capable of navigating chemical gradients towards the source of a chemical and it has been shown that it performs this navigation using a directed random walk strategy (Pierce-Shimomura et al., 1999) but this behaviour has not been significantly studied further since that fundamental study. Yet several questions remain; for instance, given that the locomotion of *C. elegans* can be broken down into three behavioural units (forward runs, turns and reversals, see Zariwala et al., 2003; Miller et al., 2005), how important is each behavioural

unit for performing the overall navigation effectively and what is its role and relationship to the other identified units? In what different ways could these units be combined in effective navigation strategies?

The aim of this thesis therefore is to address the above questions as well as other open ones we identify in the literature and by this further the field of behavioural studies of *C. elegans*. Specifically, we will address a range of questions regarding both chemotactic and thermotactic behaviour, but we will restrict the scope of the present work to gradient navigation strategies used by the animal.

Our main approach is computational, that is to say we will use computational models of the behaviours we are interested in and analyse them for novel predictions on these behaviours. The choice of these model is ethologically guided (Lehner, 1996; Martin and Bateson, 1993, see Chapter 2) but part of the analysis used in this thesis requires a novel approach to understanding behaviour based on computational models, as will be discussed in Chapter 4. It has therefore also become a second aim of this thesis to provide such a novel approach. While our main behavioural interest in this work is focused on *C. elegans*, we find it nonetheless desirable to formulate this approach in a general way so that it may find applications in a range of behavioural studies not necessarily restricted to the field of *C. elegans* research.

We are therefore trying to achieve two aims in this thesis. First, and most importantly, we wish to further the study of *C. elegans* behaviour using computational approaches and focusing on open questions regarding its gradient navigation strategies. Second we wish to formulate any methodology we develop in the course of this work in such a way that it may be also be useful in studies outside the field of *C. elegans* research.

1.1 Organisation of this thesis

We have divided the present thesis into four conceptually distinct parts. Parts I and IV deal with the introductory matters and the overall discussion and suggestions for further work respectively. The research carried out for this thesis is presented in 4 chapters which are spread over two parts. Part II is concerned with general computational techniques for modelling and analysing animal behaviour. It is not specific to *C. elegans* and the work presented in this part is likely to find applications in a variety of fields. Part III deals specifically with the major gradient navigation strategies of *C. elegans*.

THE aim of this chapter is to introduce the nematode *C. elegans* and review the current state of the art in *C. elegans* research. While doing so, we will simultaneously identify open question that can be addressed in the present thesis.

2.1 The nematode *C. elegans*

2.1.1 Etymology

The name *Caenorhabditis elegans* is actually a blend of three words: *Καινος* (kainos, gr.): recent, referring to the Pleistocene epoch (The word Pleistocene is derived from *Πλειστος* (pleistos, gr.) meaning 'most' and *Καινος*), *Ραβδος* (rhabdos, gr.): rod (rhabditis meaning rod-like and referring to the genus of rod-like organisms), and *elegans* (lat.): elegant, referring to the 'elegant' sinusoidal movements of the nematode.

C. elegans was initially simply named *Rhabditis elegans* by Émile Maupas (1899, 1900). In 1952, Günther Osche revised the *Rhabditis* genus, introducing *Caenorhabditis* as a subgenus (Os-



Figure 2.1: The nematode *C. elegans*. Picture adapted from (Altun and Hall, 2005).

che, 1952), changing the name to *Rhabditis (Caenorhabditis) elegans*. Finally, Ellsworth Dougherty (1953) elevated some of the sub-genera from Osche to genus status and *C. elegans* received its current name.

2.1.2 General facts

C. elegans, pictured in its adult form in Fig. 2.1, is a small free-living organism, roughly 1 mm long and with a life expectancy of 2 to 3 weeks, beginning with fertilisation. *C. elegans* moves through several larvae stages until it reaches its adult form roughly 2 days after hatching. Its body is transparent with all cells, including the neurons, visible through a microscope and it is easy to culture, making it an ideal candidate for developmental and genome studies. It reproduces sexually, although there are no female specimen but rather only males and hermaphrodites. Males are found only very rarely however and virtually all laboratory studies are on hermaphrodites.

The majority of research involving *C. elegans* is genetical (see for instance Riddle et al., 1997), since the genome of *C. elegans* is completely mapped and relatively small, with only about 100 Megabases, which are arranged on 6 chromosomes and an estimated total of around 20,000 genes (Wei et al., 1996). In the present thesis, however, we are more interested in the behavioural and

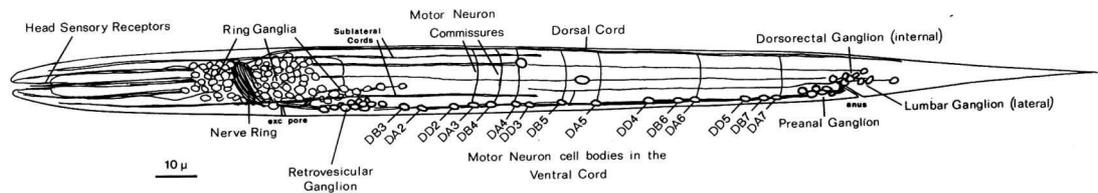


Figure 2.2: The layout of the nervous system in *C. elegans* (Durbin, 1987)

neuroscientific studies of *C. elegans* (e.g. de Bono et al., 2002; Dunn et al., 2004; Pierce-Shimomura et al., 1999; Ryu and Samuel, 2002; Mori and Ohshima, 1995). These may also make use of the research into the *C. elegans* genome (e.g. de Bono and Bargmann, 1998; Mori, 1999; Cheung et al., 2004). Indeed, since the genome is completely known, it becomes possible to breed mutants lacking specific neurons. Along with laser ablation techniques, with which it is possible to destroy selected neurons, these methods are a powerful way of investigating the behavioural function of certain neurons in live animals (e.g. Mori and Ohshima, 1995; Tsalik and Hobert, 2003; Hardaker et al., 2001).

2.2 The neural architecture of *C. elegans*

The “brain” of the nematode *C. elegans* consists of 302 neurons, a number which stays constant across individuals (White et al., 1986) (see appendix A for a list of neurons and their approximate location). Those neurons are connected via approximately 5000 chemical synapses and 2000 gap junctions (Niebur and Erdős, 1993) and the connectivity has been virtually completely mapped out (White et al., 1986), albeit only for one individual. An electronic database listing all the synaptic connections in a convenient format also exists (Oshio et al., 2003). It has also been shown that the connectivity of the *C. elegans* neural system satisfies the constraints on a small world network¹

¹A mathematical graph in which any node can be reached from any other node in a small number of steps, but in which most nodes are not actually direct neighbours.

(Nishikawa et al., 2002).

The neurons are separated into 116 morphologically different classes (Kaplan, 1996), which in turn are generally divided into three categories: sensory neurons, interneurons and motor neurons. 47 neurons are believed to be sensory neurons (Kaplan, 1996), of which 32 appear to be chemo-sensory (Lanjuin and Sengupta, 2004), as their sensory endings are generally exposed to the environment. Around 20 neurons actually belong to a smaller pharyngeal sub-circuit, which is solely concerned with body functions related to feeding and digestion and is largely independent of the main neuronal circuit (Avery, 1993; Albertson and Thomson, 1976) while most of the remaining neurons are grouped together in a nerve ring ((White et al., 1986), see Fig. 2.2).

The interneurons can be further subdivided into primary interneurons (neurons which are directly connected to sensory neurons), secondary interneurons (which are only connected to other interneurons) and command interneurons (which are connected to motor neurons) (Ferrée and Lockery, 1999). All neurons are named by letter sequences which are usually 3 to 4 letters for a class of neurons followed by a letter identifying the individual neurons inside that class.

When looking at the general function of neurons (e.g. are they sensory neurons, and if so, to what stimuli do they react?), it is usually sufficient to consider the 116 classes rather than individual neurons (e.g. Gabel et al., 2007; Gray et al., 2005; Dunn et al., 2004), although it has been shown that the difference between two neurons may be important in more detailed behaviour. For instance, Wes and Bargmann (2001) showed that the two AWC neurons, involved in chemo-sensing, are functionally distinct and a loss of this diversity results in impaired chemotaxis.

The largest part of the neural studies in the field of *C. elegans* research concerns itself with the identification of the functional role of certain neurons (for overviews, see for instance Bargmann, 1993; de Bono and Maricq, 2005), making use, for instance, of laser ablations or selective mutations to destroy or deactivate certain neurons in order to determine how this will affect behaviour. This has led for instance to the identification of neurons that take part in chemotactic (e.g. Bargmann and

Horvitz, 1991) and thermotactic behaviour (Mori and Ohshima, 1995) as well as the identification of the likely circuit for touch sensitivity (Chalfie et al., 1985). Most of these investigations aimed to discover neurons involved in the processing of a general stimulus (*e.g.* thermal or chemical). Other studies have determined neurons that are involved in generating particular locomotory output (*e.g.* Gray et al., 2005).

2.3 Neurophysiology

This section will give an overview of the neurophysiological data available on *C. elegans*. As will be evident, the available knowledge on the neurophysiology of *C. elegans* is far from complete, and while laser ablation and mutation studies have allowed us to identify the roles of different neurons, the small size of *C. elegans* has, until recently, made it impossible to obtain intracellular recordings (Nickell et al., 2002). Indeed, the cell bodies of the neurons are typically only $2\mu\text{m}$ in diameter and as an extra challenge, the worm is protected by a cuticle which explodes at attempts to dissect the animal (Goodman et al., 1998).

Thus, several key data have actually been adapted from the larger nematode *Ascaris suum*. Generally, data from *Ascaris s.* is usually used in modelling studies when equivalent data is unavailable from *C. elegans* itself (see for instance (Ferrée and Lockery, 1999)). However, this is problematic since there are indications that *Ascaris s.* differs in several aspects from *C. elegans*, both in intracellular properties (Nickell et al., 2002) as well as through the significant difference that *Ascaris s.* actually also features spiking neurons (Niebur and Erdős, 1993).

2.3.1 *C. elegans* neurons

In most animals, neurons transmit information between them using spikes (or action potentials), which are essentially electrical pulses travelling down the axons of the neurons and are

caused by the opening and closing of different types of ion channels (O'Reilly and Munakata, 2000). The detailed dynamics behind these action potentials were first described for giant squid axons by Hodgkin and Huxley (1952). However, their model is now generally used for describing the dynamics of spiking neurons in most animals, including human beings.

The most striking feature of *C. elegans* neurons then, is that they lack the Na^+ voltage dependent ion channel, which is essential for the Hodgkin-Huxley model of signalling between neurons (Nickell et al., 2002). Instead, it is assumed, based on data from *Ascaris s.* and recordings from a few *C. elegans* neurons that they use slow, graded potentials based on calcium dynamics and electrotonic effects to transmit signals (Nickell et al., 2002; Goodman et al., 1998).

Intracellular calcium dynamics have only recently been measured in detail for two sensory neurons: ASJ (Gabel et al., 2007) and AFD (Clark et al., 2006), involved in electrosensation and thermosensation respectively. These studies have used calcium imaging techniques to determine how changes in the environment might be translated into sensory signals and will be discussed in more detail later on.

One of the only detailed intracellular recordings and characterisation of individual channels has been done by Nickell et al. (2002), who looked in detail at the chemo-sensory neurons AWA and AWC. While they were unable to clearly measure the value of the resting potential due to the limitations of the equipment and techniques currently available (with values measured from different cells ranging from -16 to -65 mV), they found a region of high membrane resistance, bounded by inwardly (active at potentials lower than -50 mV) and outwardly (active at potentials higher than -20 mV) rectifying currents, between -20 and -70 mV. Outwardly rectifying channels were found to be activated by Ca^{2+} as well as, as mentioned before, by depolarisation while inwardly rectifying channels were separated into two types, both activated by hyperpolarisation, with one being difficult to characterise and the other one being most likely a non-selective cation channel.

Thus it is clear that our knowledge of the intracellular properties of *C. elegans* neurons is not

sufficient at the moment for constructing detailed computational models of these neurons. However, as will be argued in section 2.3.3, this lack of data is not a fundamental problem for the work presented in this thesis.

2.3.2 *C. elegans* Synapses

As there is virtually no physiological data on the synapses of *C. elegans* itself, data from *Ascaris s.* is generally used as it seems reasonable to assume that *C. elegans* synapses will function in a similar way (Ferrée and Lockery, 1999). Thus, post-synaptic voltage can likely be modelled as the result of a sigmoid function of the pre-synaptic voltage.

An interesting study by Schuske et al. (2004) shows that the GABA neurotransmitter in *C. elegans* actually has both an inhibitory and an excitatory function. The inhibitory effect has been observed in motor neurons controlling ventral and dorsal muscles. In order to bend the body, *C. elegans* contracts the muscles on either the ventral or the dorsal side while using GABA innervation to relax the muscles on the opposing side. The excitatory effect has been observed during defecation, where GABA releases from the AVL and DVB neurons are required to excite muscle contractions resulting in the expulsion of intestinal contents. Other known standard neurotransmitters that have been found in *C. elegans* (usually by analysing the genome for their expression) include serotonin, dopamine and acetylcholine. For additional discussion, see for instance Riddle et al. (1997).

In general however, the signs of most synapses have not been measured yet and remain thus unknown. Some studies attempt to identify synaptic signs through computational models (e.g. Majewska and Juste, 2001; Iwasaki and Gomi, 2004), however none of them are without problems. The study by Majewska and Juste (2001), violates the hypothesis (Dale, 1935) that any one neuron uses the same neurotransmitters at all its synapses, as noted by Iwasaki and Gomi (2004). For their part, Iwasaki and Gomi (2004) use spiking neurons in their simulations, which ignores one of the key facts about *C. elegans* neurons that researchers feel reasonably confident about, namely that the

animal does not use spiking neurons.

2.3.3 Implications of the lack of neurophysiological data for the present thesis

It is evident from the data presented here that neurophysiological information about *C. elegans* is rather scarce and data from *Ascaris s.* cannot be used unreservedly. For this reason it cannot be expected that detailed computational models of any neural circuitry underlying a given behaviour will be produced in this thesis. While this would not be impossible in principle and even somewhat constrained since the connectivity of the neural circuitry is known², such investigations would only have a theoretical interest at best but would remain without real predictive powers. In particular, it can be shown that even when neural and synaptic dynamics are known, any given neural circuit can nonetheless produce the same overall behaviours for a wide range of values for its parameters (Prinz et al., 2004). In the case of *C. elegans*, where not even the dynamics are known in sufficient detail, this issue would be even more severe.

However, while detailed models of the neural circuitry are not likely to be useful in the present work, the situation is different when considering functional models which focus on the computations required for performing a certain behaviour but do not actually worry about a detailed neural implementation of these computations. An interesting study in this respect has been done by Dunn et al. (2004). In their work, they train and optimise neural networks to perform chemotactic behaviour (discussed in the next section). While the merits of the architecture of the networks they find in this way are debatable for the reasons discussed above, Dunn et al. (2004) focus, in part of their discussion, on the computations performed by their networks rather than the layout itself. They thus find, for instance, that the animal is likely to compute the first time derivative of the inputs perceived by the chemosensory neurons. They have thus formulated a necessary computation that is

²This is an interesting situation in itself. *C. elegans* stands apart as being the only animal for which we know the neural connectivity but not the neural dynamics. In the case of most other, higher lifeforms, neural dynamics can be known in great detail but the exact connectivity remains unknown and can be modelled at best using general statistical guesses on its possible nature.

performed within the neural circuitry of *C. elegans* whose plausibility is independent of the actual biological neural implementation.

In this thesis, we take a similar approach, in particular in Chapter 6, in which we consider the computational requirements for certain behaviours and formulate restrictions on the computational capabilities of *C. elegans*. Any discussion of neural computations will therefore remain at the functional level.

2.4 The behaviour of *C. elegans*

It has been said before that the main interests of the present work are behavioural. In this section, we therefore give an overview of the behaviour of *C. elegans* and open questions to be addressed in the present thesis will be highlighted. Some behaviours, in particular the feeding behaviour, are mentioned only out of interest and for the sake of completeness but will not be the subject of major investigations in the remaining chapters of this thesis. The following is a quick summary of the behaviours that will be dealt with in the thesis:

- **Chemotaxis**, the ability of the animal to navigate chemical gradients using a directed random walk.
- **Thermotaxis**, the ability of the animal to navigate towards a preferred thermal region in thermal gradients, again using a directed random walk.
- **Isotherm tracking**, the ability of the animal to follow a “line” defined by a specific temperature within a thermal gradient with great accuracy.

2.4.1 Locomotion

C. elegans moves forward through sinusoidal movements of the body while lying on its side and contracting its ventral and dorsal muscles respectively (Gray et al., 2005). These sinusoidal

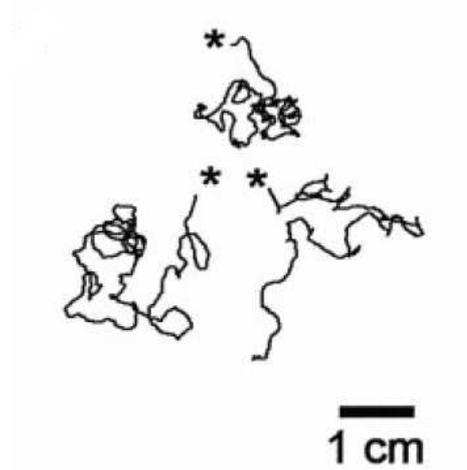


Figure 2.3: Tracks of 3 worms allowed to wander for 20 minutes in a uniform gradient (Fig. 2a from Pierce-Shimomura et al., 1999).

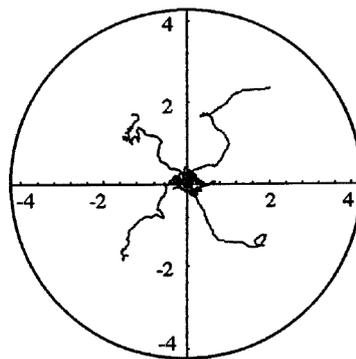


Figure 2.4: Tracks of 4 worms allowed to wander for 5 minutes in a radial gradient (Fig. 3a from Ferrée and Lockery, 1999). Peak of the gradient is at the centre of the figure.

swims, which characterise the movement of *C. elegans* can be interrupted by turns, which fall into two main categories: reversals and omega turns (Ferrée and Lockery, 1999). During reversals, the worm first moves backwards and then turns into a different direction while an omega turn is a movement during which the worm bends its head towards its tail, thus forming the shape of the Greek letter Ω before moving off into a different direction (Pierce-Shimomura et al., 1999). It is thus possible to identify three different locomotory behaviours in *C. elegans*: forward runs, reversals and turns. The overall movement of *C. elegans* generally resembles a random walk (Zariwala et al., 2003, see Fig. 2.3), but can be heavily influenced by external parameters, for example the presence of a chemical gradient in which the worm tends to move towards the peak of the gradient (Fig. 2.4).

It is known that the sinusoidal movement is produced by ventral and dorsal neurons exciting and inhibiting each other beginning in the head of the animal. The resulting bend of the head is then propagated along the length of the body, resulting in the sine-wave like movement but it is an open question whether or not these rhythmic movements are actually mediated by a Central Pattern Generator (CPG) (Suzuki et al., 2005a). Detailed models of the motor neurons and the body of *C. elegans* have also been produced (Suzuki et al., 2005b,a), which are capable of reproducing the general body shapes of *C. elegans*, but while interesting from a theoretical point of view, the biological relevance is again debatable, in particular since these studies assume the existence of a CPG even though this remains unconfirmed.

From a behavioural point of view, it is also perhaps more interesting to investigate how sensory and interneurons affect the locomotory output of *C. elegans* rather than the mechanics by which the behaviour is produced. A candidate neural circuit for the locomotion of *C. elegans* is presented in Fig. 2.5. Gray et al. (2005) show that killing the AWC or the ASK neurons (sensory neurons for detecting volatile and water soluble chemicals respectively) has no effect on the behaviour in the presence of food or on the dispersal behaviour observed when the worm has been in the absence of food for a longer time (see section 2.4.2). However, it does affect the local search behaviour

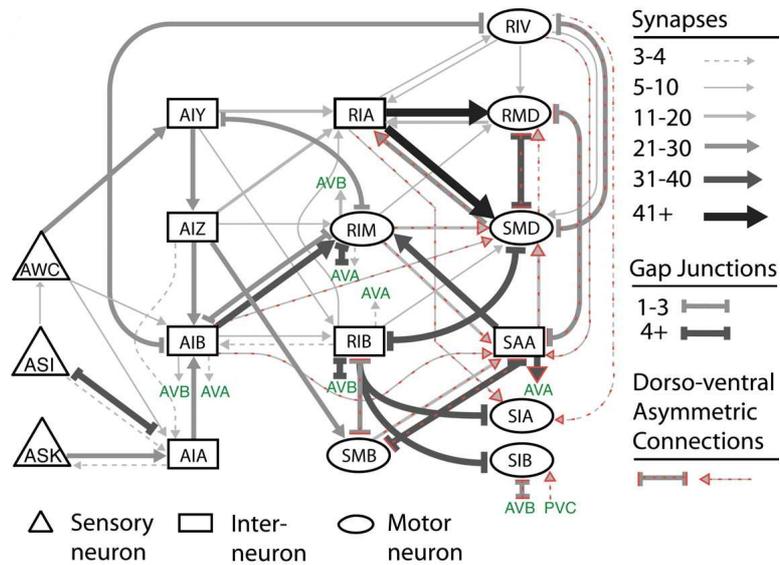


Figure 2.5: Fig. 3a from (Gray et al., 2005). A candidate circuit for locomotion in *C. elegans*. Command interneurons are shown in green, sensory-, inter- and motor-neurons are indicated by their respective shapes, arrows indicate synaptic connections and H-shaped bars gap junctions (where the style of the arrow or of the H-shape indicates the number of connections between the neurons at either end).

displayed when the worm is moved from food to an environment without food, as the killing of the AWC or the ASK neurons reduces the frequency of reversals and omega turns. In contrast, killing the ASI chemo-sensory neurons resulted in an inability to reduce the frequency of short reversals during the dispersal behaviour. Killing the thermotactic AFD neurons led to a small decrease of reversals and omega turns during local search, but this effect was much smaller than the ones previously observed (Gray et al., 2005). Killing other sensory neurons had little or no effect on the locomotion.

Tsalik and Hobert (2003) also find a decrease in reversal frequency when the AFD neurons are killed, hypothesising that it is actually caused by an increase in forward run duration, which could be explained by the fact that *C. elegans* uses the AFD neurons to suppress reversals when in aversive thermal conditions.

Gray et al. (2005) go on to identify the roles of non-sensory neurons in the locomotory be-

haviour. They show that killing the AIZ neurons reduced the frequency of short reversals on food while killing AIB or RIM neurons resulted in animals that did not exhibit a higher frequency of longer reversals and omega turns upon removal from food. Animals lacking AIY or RIM neurons were not able to initiate the dispersal behaviour after having been removed from food for a longer amount of time.

The finding on the AIY and RIM neurons is consistent with what has been observed previously by Tsalik and Hobert (2003), who noted that their removal caused hyper-reversal behaviour. They also find that the removal of AIY results in a slightly defective chemotaxis, although this defect can be overcome by increasing the chemical's concentration, thus indicating that the AIY neurons are not the only gateway through which sensory neurons can modulate locomotion.

The AVA command neurons are essential for reversals and backward movement (Chalfie et al., 1985; Niebur and Erdős, 1993) and unsurprisingly, Gray et al. (2005) show that killing these neurons results in a complete inability to generate long reversals and a much lower frequency of short reversals whilst in the presence of food. Omega turns, on the other hand, were not affected. Niebur and Erdős (1993) also show that backward movement can be completely disabled by eliminating both the AVA and the AVD neurons or the DA neurons, whereas simply eliminating AVA or DD just results in impaired backwards movement. Similarly, the animal can be rendered unable to move forward by removing both the AVB and PVC neurons or the DB neurons, whereas simply removing AVB or DD results in impaired forward movement (Niebur and Erdős, 1993). Ablating the RIM motor neurons resulted in an increase of short reversals. This effect could be reversed by also killing the AVA command neurons, suggesting that RIM might use the connections to the command neurons in order to suppress short reversals (Gray et al., 2005).

Finally, Gray et al. (2005) look at which neurons affect omega turns. It turns out that killing either the SMD or RIV neurons decreases the frequency of omega turns, with SMD being responsible for the omega turn amplitude while killing the RIV neuron removes the bias for the first head

swing after a reversal being into the ventral direction.

Perhaps the most fascinating insight to be gained from these studies is that the three main locomotory behaviours of *C. elegans* - namely forward runs, turns and reversals, appear to be directly and separately encoded in neural subcircuits. We will use this fact later when choosing a computational model of *C. elegans* behaviour.

2.4.2 Feeding and social behaviours

The movement of *C. elegans* changes dramatically depending on whether it is currently in the presence or the absence of food (Gray et al., 2005). In the presence of food, (Gray et al., 2005) have found that the worms move forward slowly, reverse frequently, although the reversals are usually quite short, and almost never exhibit omega turns. If the worms are moved off the food, however, the frequency of short reversals decreases while the frequency of long reversals increases at the same time, as does the frequency of omega turns. This high frequency of reversals and omega turns was found to decrease again after a longer time in the absence of food, resulting in longer runs in one direction.

The feeding behaviour of *C. elegans* is interesting, not only because of the feeding itself, but also because some strains of *C. elegans* exhibit a kind of social behaviour during feeding (de Bono, 2003). In fact, social strains will not reduce their speed upon encountering food until they have aggregated with other *C. elegans* individuals, whereas solitary worms will start feeding immediately (de Bono et al., 2002). This social behaviour seems to be related purely to feeding however. The aggregation behaviour of the social strains is not observed in the absence of food, or when the worms are well-fed (de Bono et al., 2002).

As far as neural mechanisms underlying this social behaviour is concerned, it appears that it is mediated by neurons responsible for detecting noxious stimuli (de Bono et al., 2002). In fact, de Bono et al. (2002) show that the two neuron groups in question are ASH and ADL, as the

aggregation behaviour is significantly disrupted if both ASH and ADL neurons are ablated. Just killing either both ASH or both ADL neurons, however, did not significantly influence the worm's behaviour. It appears that bacterial odour emanating from the food itself may induce the social aggregation behaviour, although the exact reason for this behaviour remains rather unclear (de Bono et al., 2002).

In a related study, Coates and de Bono (2002) identify more neurons which could play a role in producing the aggregation behaviour. They note that suppressing the AQR, PQR and URX neurons inhibits social feeding. These neurons are unusual, because they are the only ones to be exposed to the body fluid of *C. elegans*, which appears to be its blood analogue (Coates and de Bono, 2002), but it is unclear how exactly these neurons affect the aggregation behaviour. Indeed, it is not even known for sure if all three affect the behaviour, or merely one or two of them.

2.4.3 Gradient navigation behaviours

Most of the behavioural work on *C. elegans* has been done on its behaviour in graded environments, which is also the main interest of the present thesis. Here, we will review such behaviour in the two main environments navigated by the animal: chemical gradients and thermal ones.

Chemotaxis

Chemotaxis refers to the ability of *C. elegans* to navigate chemical gradients either towards the source of this chemical (*e.g.* if the source is food) or away from it (*e.g.* if the source is a predator). The animal is capable of identifying different chemicals and able to act upon a gradient created by one chemical even if another chemical has a strong uniform presence (Wes and Bargmann, 2001). In order to successfully perform chemotaxis, *C. elegans* must have a way of assessing the surrounding chemical gradient. Chemosensors of *C. elegans* can be found both at the head and at the tip of the tail (Ferrée and Lockery, 1999). However, mutants in which the rear sensors are blocked

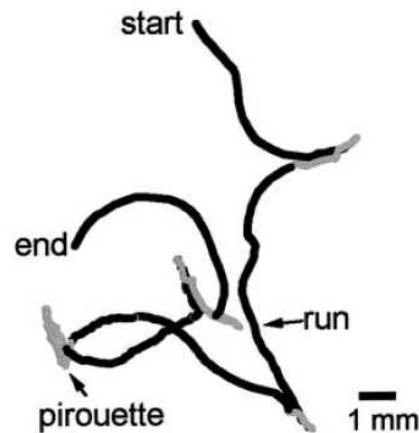


Figure 2.6: Fig. 5c from (Pierce-Shimomura et al., 1999) showing the separation of tracks into runs and pirouettes. Dark parts of the track have been classified as a run whereas lighter parts have been classified as a pirouette.

are not significantly impaired in performing chemotaxis (Ward, 1973), thus suggesting that only the front sensors play a major role. It is not known exactly how the worm assesses the chemical gradient, however (Ferrée and Lockery, 1999). Two likely mechanisms exist: either the gradient is sampled during head sweeps resulting from its typical sinusoidal motion (Ward, 1973) or simply while moving forward through the environment (Ferrée and Lockery, 1999).

One of the earliest studies into chemotaxis has been done by Ward (1973), who hypothesised that the animal performs chemotaxis by keeping its head pointed up the chemical gradient “like a weather vane pointing into the wind”. However, this strategy has since been superseded by the pirouette strategy initially proposed by Pierce-Shimomura et al. (1999). According to this strategy, a pirouette is a “series of turns interspersed with short runs” (Pierce-Shimomura et al., 1999, see Fig. 2.6). During chemotaxis, it has been shown that these pirouettes occur most frequently when the worm is currently heading down the gradient and least frequently when it is heading up the gradient (Pierce-Shimomura et al., 1999). Critically, Pierce-Shimomura et al. (1999) also show, that the start of pirouettes is not correlated with the absolute concentration of the chemical, but with a change in concentration. Further, on average the worm will be heading in a favourable direction

after a pirouette.

The directed random walk as described by the pirouette strategy remains the accepted model of *C. elegans* chemotactic behaviour to date. One subsequent behavioural study aimed at verifying that the pirouette strategy is the only behavioural strategy used in chemotaxis (Pierce-Shimomura et al., 2005) while another attempted to identify the required sensorimotor transformations necessary for the pirouette strategy (Miller et al., 2005) but in general, there have been no further significant behavioural studies of chemotaxis.

However, there are still several interesting open questions. The random walk strategy used by *C. elegans* combines forward runs, turns and reversals into the pirouette strategy, but is this strategy optimal given these available locomotory behaviours or do other strategies exist? More generally, if *C. elegans* locomotion consists of these three behaviours (which are all directly encoded in different neural circuits, as seen previously in this chapter), how should they be combined in order to perform efficient chemotaxis?

Further, does the observed behaviour give any insights in the underlying computations? It has been shown previously for instance, that *C. elegans* is likely to compute the first time derivative of the sensory input (Dunn et al., 2004), but are other insights possible as well? These questions will be addressed in Chapter 5.

Most of the sensory neurons for chemical stimuli are thought to be known. The AWC neurons are known to sense at least five odours: butanone, benzaldehyde, 2,3-pentanedione, isoamyl alcohol and 2,4,5-trimethylthiazole (Wes and Bargmann, 2001). Other sensory neurons involved in chemotaxis include AWA and AWB which also respond to volatile odorants (AWA sensing odorants to which the worm is attracted and AWB odorants by which it is repelled (Lanjuin and Sengupta, 2004)), ASE and ASK for water-soluble chemicals (Kaplan, 1996) and ADF, ASG, ASI and ASJ for water soluble compounds and pheromones (Kaplan, 1996). Fig. 2.7 shows most of the significant connections and the interneurons thought to be involved in chemotaxis.

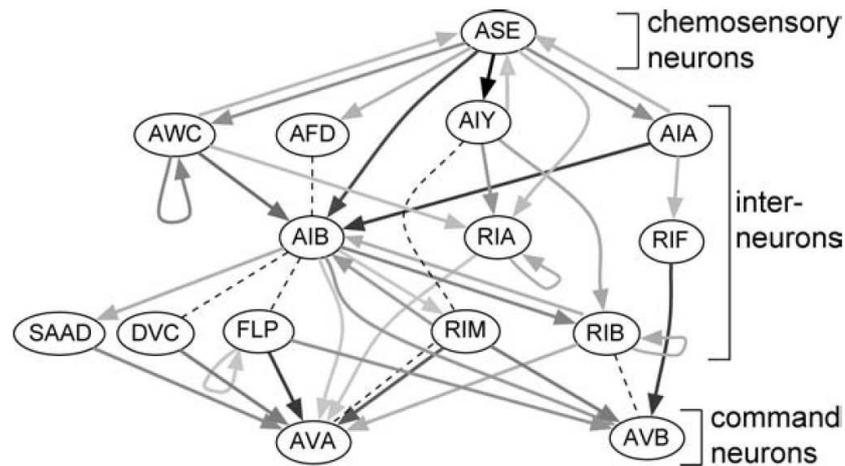


Figure 2.7: Fig. 8 from (Dunn et al., 2004). The simplified circuit shown is a part of the chemosensory circuit in *C. elegans*. Dashed lines are gap junctions, arrows are chemical synapses and self connections actually represent chemical synapses between the pair of neurons in that class. Pathways in which there were 'fewer than 2 pre-synaptic densities or fewer than 3 gap junctions' were ignored in this diagram.

Response to the sensory cues seems to be mainly mediated by the AIY neurons although a second route via AIZ and RIB also exists (Tsalik and Hobert, 2003), hinting at redundancy within the *C. elegans* neuronal circuit. However, animals whose AIY neurons have been killed will show a defective chemotactic behaviour unless higher concentrations of the chemical are present (Tsalik and Hobert, 2003).

2.4.4 Thermotaxis

Navigation towards a preferred temperature

Similarly to its behaviour in chemical gradients, *C. elegans* is able to navigate thermal gradients towards regions of preference. This "region of preference" is typically thought to be a region whose ambient temperature is roughly the cultivation temperature of the animal (Samuel et al., 2003), but it has recently been shown that the preferred temperature is actually based on recent thermal history

(Biron et al., 2006).

The history of the study of thermotaxis in *C. elegans* is perhaps more interesting than the corresponding history of chemotactic studies as our knowledge of this behaviour keeps evolving even today. One of the first studies of thermotaxis was done by Mori and Ohshima (1995), in what is essentially a study of the neural system of the nematode *C. elegans*. Mori and Ohshima (1995) distinguish between a thermophilic and a cryophilic drive in *C. elegans*, where the cryophilic drive is used for moving down a thermal gradient towards the preferred temperature and the thermophilic one for moving upwards towards the preferred temperature. The same study also identified the AFD neuron class as being the main thermosensory neuron, while hypothesising that a second thermosensory neuron also had to exist. Mori and Ohshima (1995) tried to identify more neurons involved in the thermotactic circuit by killing off neighbouring neurons of AFD. They found that killing both AIY neurons, which are post-synaptic to the AFD neurons, resulted in a significant cryophilic movement as well as the loss of the isotherm tracking ability. Interestingly, killing AIZ, a post-synaptic partner of AIY resulted in worms actually exhibiting a thermophilic behaviour.

To investigate the relationships between the AFD, AIY and AIZ neurons, Mori and Ohshima (1995) killed off pairs of them, but found that only killing AFD together with AIZ yielded any useful information. Animals to which this was done exhibited a far more extreme abnormal behaviour than those which only had AFD killed, suggesting that AIZ receives information from a further, as yet unidentified, thermosensory neuron. Further, when looking at further possible candidates for the thermotactic circuit, both the RIA (which is post-synaptic to both AIY and AIZ) and RIB (which is post-synaptic AIY and pre-synaptic to AIZ and RIA) interneurons were also found to slightly affect thermotaxis in new ways. In terms of identifying a neural circuit for thermotactic behaviour, this study by Mori and Ohshima (1995) remains the most important work.

Ryu and Samuel (2002), in the first paper investigating the behavioural strategies behind *C. elegans* thermotaxis however find no behavioural evidence for the thermophilic drive suggested by

Mori and Ohshima (1995), that is to say, they find no evidence that the animal will move towards its preferred temperature if the ambient temperature is lower. The study does find cryophilic behaviour, whose mechanism is similar to the pirouette strategy proposed for chemotaxis: run periods are extended if the worm is heading down a gradient compared to movement in an isotropic environment and shortened if it is heading up a gradient.

Interestingly, the controversy surrounding the existence of the thermophilic drive continues. Ito et al. (2006) argue for the existence of a thermophilic drive which may only become active after a longer period of time and therefore missed in studies like the one by Ryu and Samuel (2002). On the other hand, studies of the calcium dynamics within the AFD neurons, to date still the only known thermosensory neurons, show no activity at temperatures below the preferred one (Kimura et al., 2004; Clark et al., 2006).

Whether or not a second thermosensory neuron, as hypothesised by Mori and Ohshima (1995) exists also remains unknown. On one hand, the existence of such a neuron might explain the controversy around the thermophilic drive but on the other hand, it has been shown that activity of the AFD neuron is sufficient to explain all other thermotactic behaviours (Clark et al., 2007, 2006). There are thus still open research questions about the navigation of *C. elegans* in thermal environments. However, these questions are less concerned with behavioural strategies and more with the thermal range in which these strategies are active. As such, they are best answered by experimental work and do not lend themselves well to the theoretical computational work we aim to perform in the present thesis. For this reason, we will not address either the thermophilic drive or the possible existence of another thermosensory neuron in later chapters of this thesis.

A study which claims to have found a thermophilic drive however, and which deserves a little discussion has been done by Zariwala et al. (2003). In this behavioural study, Zariwala et al. (2003) find that the turning rate of the animal increases if the worm is placed in a temperature below the preferred one (similar to the observations by Ryu and Samuel (2002) for the cryophilic drive).

Further, they find that the cryophilic drive manifested by the worm in their study actually resembles an avoidance response, where the worm first reverses and then performs an omega turn.

The results from Zariwala et al. (2003) thus appear to be at odds with those from Ryu and Samuel (2002). However, it is important to note that the experimental setups in these studies differ substantially from each other. Whereas Ryu and Samuel (2002) put the worm on thermal gradients either above or below the preferred temperature and observed its movement across this gradient, Zariwala et al. (2003) either increased or decreased the ambient temperature (initially set to 20 °C, the cultivation and thus the preferred temperature of wild type worms) by 3 °C step, but did not provide a thermal gradient leading back towards the preferred temperature. This difference in setup (most notably the lack of an actual gradient in Zariwala et al. (2003)'s study) suggests that it is entirely possible for both experiments to, in fact, be looking at a different type of behaviour.

This suggestion is backed up by the fact that (Ryu and Samuel, 2002) also found, as discussed earlier, that the turn probability of the animal is different depending on whether it is in an isotropic environment or in a gradient. *C. elegans* thus appears to be exhibiting a different behaviour depending on whether it is simply placed in a warmer environment or a thermal gradient. It is thus possible that Zariwala et al. (2003) have inadvertently studied the animal's response to sudden temperature changes rather than its navigation strategies in thermal gradients. The significant difference lies in the fact that during a sudden temperature drop below the preferred temperature, the AFD neuron would switch from active to silent which in turn may trigger a behavioural response. If the animal is already in a thermal gradient below the preferred temperature, the activity of the AFD neuron would remain unchanged (*i.e.* silent) and would thus not trigger a behavioural response. The thermophilic response discovered by Zariwala et al. (2003) may thus not apply to navigation in thermal gradients.

Isothermal tracking

A second interesting behaviour observed in thermal environments is the tracking of isothermal lines in the graded environment. However, this behaviour has not yet received much attention in the literature. It was first characterised by Ryu and Samuel (2002) who noted that the behaviour only appears if the animal is within 2 to 3 °C of its preferred temperature. Interestingly, Samuel et al. (2003) have shown that the synaptic output of the AFD neurons increases in the same temperature range and may thus regulate the on- and offset of the isotherm tracking behaviour.

A more comprehensive study of isothermal tracking behaviour has been published by Luo et al. (2006). In this study, it was found that isotherm tracks are periods of prolonged forward movement, suggesting that the animal continuously performs small course corrections to stay on the track. It does so with amazing precision: while tracking isotherms it does not deviate from them by more than 0.1 °C. At the same time, however, the animal does not appear to be actively seeking isotherm alignment as isotherm tracks are separated by periods of stochastic movement and *C. elegans* does not appear to align onto them deterministically. Finally, it has been shown that the animal is not likely to keep a memory of the isotherm it is currently tracking but maintains the alignment through other means.

Luo et al. (2006) then formulate a strategy which might be employed by *C. elegans*. They propose that the animal continuously adjusts the rate of curvature of the head segment in function of the sensory input in order to balance warming and cooling phases during a head sweep and thus maintain the alignment. The strategy thus attempts to achieve and maintain a balance in thermal input which should keep the animal aligned with the isotherm. This is puzzling however, as such a balance would also exist were the animal to move directly perpendicularly to the isothermal track, that is to say directly into (or away) from the gradient. If the strategy proposed by Luo et al. (2006) could be used for such a navigation into the gradient, the animal would be able to navigate towards its preferred temperature deterministically rather than stochastically using this same strategy. Since

this would be clearly at odds with observed behaviour, it would imply that the strategy proposed by Luo et al. (2006) is not actually a candidate strategy for isothermal tracking by *C. elegans* and a new strategy would have to be formulated. We will discuss and address this issue in Chapter 6 of the present thesis.

The ability of *C. elegans* to deterministically track isotherms raises another important question that has not yet been answered: why would the animal use a stochastic strategy for navigating towards its preferred temperature at all? It is clearly able to sample the temperature at a sufficient resolution and respond with sufficiently precise movements for deterministic isothermal tracking, so what prevents it from using the same sensory and locomotory capabilities for navigating deterministically towards its preferred temperature? This question will also be addressed in Chapter 6.

2.4.5 Electrical fields navigation

It has recently become evident that electrosensory behaviour of *C. elegans* is actually also mediated by the neural circuitry (Gabel et al., 2007), rather than being, for instance, a simple physical effect. The navigation strategy used by the animal in electrical fields also appears to be deterministic: it will crawl at a specific angle to the field towards the negative pole. An interesting parallel with isotherm tracking is the fact that in both cases, the optimal direction is along the (isopotential) gradient lines rather than perpendicularly to them. Concerning electrical field navigation however, it has been found that the intracellular calcium levels in ASJ, the main sensory neuron involved in this behaviour, is directly correlated with the travel angle relative to the electrical field and is highest (lowest) when the animal is directly facing the positive (negative) pole (Gabel et al., 2007). In contrast to navigation in thermal environments thus, the output of ASJ is by itself sufficient to directly mediate the deterministic strategy and no further computational steps are needed.

Therefore, even though the optimal direction in both cases lies parallel to “lines” (isothermal

or isopotential) in the environment, the similarities between isotherm tracking and electrical field navigation end there. When navigating electrical fields, the animal is able to perceive the optimal direction immediately and as such, the behaviour is of lesser interest to us. It does however serve as another example of a navigation problem in which *C. elegans* uses a deterministic rather than a stochastic navigation strategy.

2.5 Ethological considerations for modelling and analysing behaviour

The previous section has introduced the animal we are interested in and identified several open questions relating to its behaviour. In order to successfully address these questions, we therefore need tools enabling us to both model and analyse this behaviour. We are therefore interested in methods from the field of Ethology which will allow us to achieve our goals using mainly computational means.

Perhaps the largest part of the methods available in Ethology are actually concerned with the measuring of behaviour in experimental setups (see Martin and Bateson, 1993, for an introduction). This is less interesting in the context of the present thesis since our work is largely based on existing and published behavioural measurements. Nonetheless, a few concepts from this methodology can help in guiding our work and providing its ethological context.

2.5.1 The context of the present work within the field of Ethology

First, it is generally accepted that the field of Ethology mostly aims to address 4 main types of questions (Martin and Bateson, 1993), initially proposed by Tinbergen (1963):

- How is a behaviour performed (Proximate causation)?
- How does a behaviour evolve during the lifetime of an animal (Ontogeny)?
- What is the function of a behaviour?

- How did a behaviour evolve?

The questions we have identified in the previous section are mainly related to the first and last point in this classification. We are interested in the strategies underlying chemotactic behaviour as well as the mechanisms underlying isothermal tracking, both of which related to the first class of questions. Additionally, we are interested in studying the use of stochastic rather than deterministic navigation strategies for reaching preferred temperatures in a graded environment. This investigation falls within the final class identified by Tinbergen (1963), although the question might be better formulated as “*Why* did a behaviour evolve?”. Together, these classes thus define the ethological context of our work.

2.5.2 Modelling behaviour

A second interesting concept from the methodology for measuring behaviour is the concept of *behavioural unit* (Lehner, 1996). Given a continuous observed behaviour, it is possible to break it down into functionally distinct units. Choosing appropriate units is important and can be a difficult task but at the same time, well-chosen units would provide an ideal basis for a computational model of a behaviour. In the case of *C. elegans*, we have seen previously in this chapter that locomotory patterns are generally divided into three different classes (forward runs, turns and reversals) and that these classes appear to have their own underlying neural subcircuits. Given the fact that these classes can be observed both behaviourally and within the neural circuitry, they seem ideal choices as behavioural units. There have been two previous studies which used a computational model based on these units (Zariwala et al., 2003; Miller et al., 2005), giving us additional confidence that these units are a reasonable choice. We therefore base our computational investigations on these models with three behavioural states.

2.5.3 Analysing models of behaviour

Finally, we need to consider the nature of the computational model used in this work. Zariwala et al. (2003); Miller et al. (2005) model the stochastic gradient navigation using a Markov-like model. This will be discussed in more detail in Chapter 3, but an immediate issue with such a model is that it is not strictly Markovian since its transition probabilities depend on external input. This makes it in principle impossible to directly use Markovian properties for analysing the behaviour of the model analytically, instead forcing the use of numerical simulations. Since such Markov-like models are however an attractive tool for modelling stochastic behaviours in function of external input (see also *e.g.* Sánchez-Montañés and Pearce, 2006) with potential applications outside the field of *C. elegans*, it is interesting to investigate if they can be somehow converted into a strict Markov model with the aim of making the full array of Markovian analysis techniques available. Chapter 3 is dedicated to this investigation.

2.6 Summary of Chapter 2

This chapter has introduced the nematode *C. elegans* and the current state of the art of the research on this animal. Several open questions have been identified which need to be addressed:

- Given the three locomotory behaviours of *C. elegans* (forward runs, turns and reversals), how are these best combined if the aim is efficient chemotaxis? Is the pirouette strategy employed by the animal optimal in this sense? Why are pirouettes used at all?
- Is it possible to derive novel predictions on the computational capabilities of the animal from the chemotactic behaviour?
- Is the existing proposed strategy for isotherm tracking (Luo et al., 2006) reasonable? If not, can another strategy be proposed?

- Given how well the animal tracks isotherms, why does it rely on stochastic strategies at all when navigating towards the preferred temperature? Are deterministic strategies prevented by something fundamental?

Additionally, we have identified a need for novel techniques for analysing behaviour:

- Is it possible to convert Markov-like models with variable transition probabilities into strict Markov models? If so, how can this be done and what are the restrictions, if any?
- Can we propose a framework for analysing behaviour based on computational models which is able to extract novel information and make novel predictions on this behaviour?

The remainder of this thesis is dedicated to answering these questions. The technical questions relating to the analysis of behavioural models are addressed first and the insights gained from that work is then applied to address the open issues in the study of the behaviour of *C. elegans*.

Part II

Modelling and Analysing Behaviour

Chapter 3

Translating Markov-like models of behaviour into strict models

IT is sometimes possible to model goal-oriented behaviours of animals using a Markov-like model. These are based on a Markov process, which models the sequence of states a given system can be in (Brémaud, 1999; Grinstead and Snell, 1997). A fundamental characteristic of such a process is that the next state depends only on the present one and is randomly selected from all states of the system based on *transition probabilities*, *i.e.* the likelihood for each state to be the successor of the current state. If the transition probabilities are fixed and do not change over time, the model is *strictly* Markovian.

When such a model is used in behavioural studies, the states are usually set to correspond to the different observed behavioural units while the transition probabilities are dependent on some external variable or input. A very basic example is given in Fig. 3.1. Here, we consider a fictional animal, which can only eat or sleep; the model therefore only contains two states corresponding to these behavioural units. Since the model is Markov-like, transitions between these states are probabilistic. To be useful in modelling behaviour, however, these transitions need to be a function

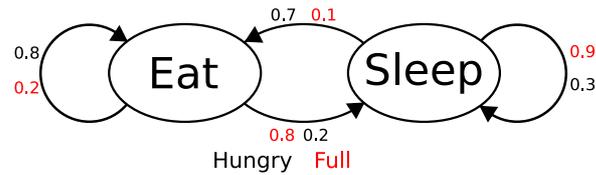


Figure 3.1: An example behavioural model of a fictional animal whose sole behaviours are eating and sleeping. The model is Markov-inspired since the transitions between the different states of the model (corresponding to behavioural units of the animal) are probabilistic, but is not strictly Markovian since the exact values of the transition probabilities are a function of the hunger level of the animal.

of some external input or variable which modulates the overall behaviour of the model, which means the model is no longer strictly Markovian (but merely Markov-like). In the example of Fig. 3.1, this variable is the hunger level of the animal, represented here as a simple binary variable; the animal is either hungry or full. If it is hungry, the transition probabilities in the model are such that the animal is very likely to eat but not likely to sleep (black transition probabilities). In other words, if the animal is currently sleeping, it is likely to stop doing so and start eating and if it is eating, it is likely to continue doing so. Once the animal is full, it becomes more likely to stop eating and start sleeping (red transition probabilities).

Such a model can then be used for different investigations. Given a certain set of transition probabilities, one can try to assess how optimal they are with respect to some goal. In the basic example of Fig. 3.1, one could for instance assume that the goal is to stop feeling hungry as quickly as possible. Alternatively, one could investigate what the optimal transition probabilities are if only the model of the behaviour and a goal are given. In terms of animal behaviour, these two investigations are very different: the first one assesses the optimality of the actual observed behaviour with respect to some goal, the second one determines what the optimal behaviour given certain behavioural units would be. It is possible that the answer to the second question turns out to be the observed set of transition probabilities used for the first question, but this does not necessarily have to be the case. Combined, the results from these investigations can then answer questions about the optimality of

the observed behaviour as well as the likely goals and motivations governing this behaviour.

Given the Markovian nature of the model, it would then be very tempting to use its Markovian properties to answer questions such as the above. In particular the *probability of absorption in a given state* (*i.e.* the likelihood that the model will end up in a given state, usually a goal state), or the *mean first passage time* (*i.e.* the mean time it takes the model to reach a certain state for the first time) seem ideally suited in principle for such investigations since they could be used to determine (1) what the likelihood that the model reaches a given goal is or (2) how long the model would take on average to reach a given goal. Unfortunately, a number of issues prevent the direct use of the Markovian properties. There are two important ones: (1) the model, due to its variable transition probabilities, is not strictly Markovian and properties like the probability of absorption or the mean first passage time are therefore impossible to compute. (2) Even though the modelled behaviour may be goal-oriented, there is no actual representation of the goal within the model itself. In other words, even though the model acts on external variables (e.g. the hunger state in our simple example), it has no representation of them and the Markovian properties could thus not inform on the performance of the model with respect to this goal.

These issues may seem critical, but they do not render Markov-inspired models of behaviour useless. It is still possible to assess the optimality of a given set of transition probabilities or to determine an optimal set of transition probabilities through numerical simulations. Depending on the model, it may even still be possible to use some of the Markovian properties for additional insights. These points will be illustrated more fully in Chapter 5 but for now, we wish to investigate whether it is possible to overcome the restrictions imposed by the variability of the transition probabilities and the absence of a goal state in order to allow an evaluation of these models based on Markovian properties. In other words, is it possible to translate a Markov-like model of behaviour, such as the one shown in Fig.3.1 into a strict Markov model?

In this chapter, we introduce a Markov-like model of *C. elegans* gradient navigation upon

which most of our investigations presented in Chapter 5 will be based. We will show that there is indeed a possible transformation into a strict Markov model and we will define this model. However, although the model exists, computational limitations prevent us from actually computing its Markovian properties. Next, we apply the same technique to a model of the behaviour of a different animal, the moth, which is sometimes studied under simplified conditions. Again, the behaviour will be modelled using a Markov-like model which we translate into a strict Markov-model. In this case, our computational resources are sufficient for computing the Markovian properties and we can investigate the behaviour of the model analytically. In particular, we will use this model to briefly investigate the optimality of the surge-cast behaviour displayed by the moth in chemotactic searches.

3.1 A model of *C. elegans* gradient navigation

C. elegans is a soil-living nematode with the ability to navigate a number of different gradients (mainly chemical and thermal) using a directed random walk strategy which decreases the frequency of random reorientation manoeuvres if the animal is moving in a desired direction with respect to the gradient and increases this frequency if it is not (Pierce-Shimomura et al., 1999; Ryu and Samuel, 2002). A key feature of the *C. elegans* strategy are so-called *pirouettes*, which are a series of small runs and turns (Pierce-Shimomura et al., 1999), interrupting long runs with increased frequency if the animal is travelling in a disadvantageous direction on the gradient. Turns produced by *C. elegans* are typically seen as being one of two types: omega turns, which are a large change in direction; and reversal turns, during which the animal reverses for a short period of time before moving forward again in a different direction.

The behaviour of *C. elegans* has been described in the literature using a three-state probabilistic model in which each of the states corresponds to one of the behavioural units identified for the

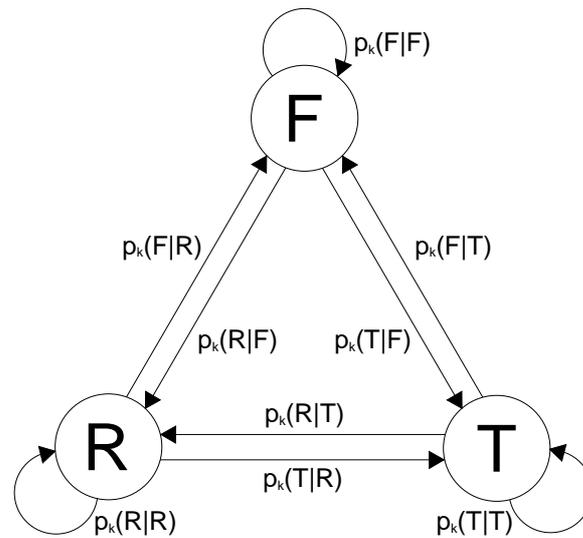


Figure 3.2: A dual Markov-like model that can be used to describe *C. elegans* gradient navigation behaviour (Pierce-Shimomura et al., 1999). The states correspond to the different locomotive behaviours shown by the animal: (F)orward runs, (T)urns and (R)eversals. Transition probabilities can have two values depending on the direction of the model relative to the gradient. They are written $p_k(Y|X)$ for the transition probability of state X into state Y with $k \in \{u, d\}$ indicating whether the value is for moving (u)p- or (d)own-gradient.

animal and the probability of moving from one state to another is dependent on the direction of travel relative to the gradient (Zariwala et al. (2003); Miller et al. (2005), see Fig. 3.2). These behavioural units correspond to runs (called the F state here, in which the animal moves forward), reversals (the R state in which it reverses) and turns (the T state in which it changes direction). In principle, it is possible to move from every state into every other state at each time step and the transition probability for moving from X to Y is written $p_k(Y|X)$, with $k \in \{u, d\}$ as the value of the probability depends on whether or not the model is moving up-gradient ($k = u$) or down-gradient ($k = d$). It has been shown that such a model is able to reproduce *C. elegans* gradient navigation in both chemical and thermal gradients given an appropriate set of transition probabilities (Zariwala et al., 2003; Miller et al., 2005).

The purpose of this section is to determine if it is possible to analyse the behaviour of this model analytically. The actual analysis of the behaviour of *C. elegans* is presented in detail in

Chapter 5. Where required in the following, we assume that the model defined in Fig. 3.2 navigates in a Cartesian space towards the peak of a gradient, located at the origin $O(0,0)$. The model moves with a fixed velocity v and the direction in which it is currently heading is given by the angle θ of the trajectory with the x -axis.

3.1.1 Translation into a strict Markov model

We are primarily interested in the navigation of *C. elegans*; therefore our goal state is represented in the physical space in which the worm moves (and of course in the corresponding virtual space for the model). Typically, this state would be located at the peak of a chemical gradient. Since a strict Markov model needs to include the goal state in order to be useful, we need to define such a model over the space the model can move in, assigning a state to every point it can occupy. It is worth pointing out here that an infinite number of states are acceptable in a strict Markov model provided that they are countably infinite. Hence, the fact that the model is potentially moving in an infinite space does not pose a fundamental problem as long the number of points the model can occupy is countable. This will be achieved through a limitation in the turn state discussed below.

Next, we note that the transition probabilities at any given time do not depend directly on the position in space but on the current state (given by the Markov-like model) and the direction in which the animal has moved with respect to the gradient in the last step. We address these dependencies by associating six states of the strict model with every position in space, three of which correspond to the up-gradient movement of the Markov-like model in Fig. 3.2 while the remaining ones correspond to the down-gradient one.

Finally, we consider changes in orientation. If the model moves forward or reverses, the heading θ remains unchanged. If the model turns, however, the orientation is increased or decreased by a value chosen from a distribution that depends on the direction relative to the gradient (Pierce-Shimomura et al., 1999). By requiring this distribution to be discrete, we can ensure that the possi-

ble new directions form a finite and countable set; we can therefore guarantee that every state in the Markov model has a countable number of successor states and thus that the total number of states in the model will be countable.

We can characterise every state in the strict Markov model we seek to obtain by five values: The position in space (x, y) , the orientation θ , the previous behavioural state $z \in \{F, T, R\}$ and the direction relative to the gradient $g \in \{0, 1\}$, where a value of 0 stands for a down-gradient and a value of 1 for an up-gradient movement. Now let $A(x, y, \theta, z, g)$ be the current active state. We next need to determine the successor states of A , *i.e.* all the states in the strict model that can be reached from A . The first two successor states are:

- $A_F(x + v \cos \theta, y + v \sin \theta, \theta, F, f(x, y, \theta))$, equivalent to a forward movement by the model
- $A_R(x - v \cos \theta, y - v \sin \theta, \theta, R, g)$, equivalent to a reversal by the model

where $f(x, y, \theta)$ is a function defined as:

$$f(x, y, \theta) = \begin{cases} 1 & \text{if } \sqrt{x^2 + y^2} - \sqrt{(x + v \cos \theta)^2 + (y + v \sin \theta)^2} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

and gives the value for g by determining if the forward movement has brought the model closer to the peak (up-gradient movement) or not (down-gradient movement). It can be noted that we only update the information related to the gradient when the model is moving forward. This is based on the observation that the animal is likely to do the same (see Chapter 5).

The remaining successor states are the result of a turn by the model. Let Θ_g be the set of all possible new orientations based on the current heading and direction relative to the gradient g . The family of successor states as a result of a turn are thus given by:

- $A_T^\alpha(x, y, \alpha, T, g), \forall \alpha \in \Theta_g$

The transition probabilities into each of these new states depend, as previously noted, on z and g as well as on the transition probabilities of the original model. Let $p_F(z, g)$, $p_R(z, g)$ and $p_T^\alpha(z, g)$ denote the transition probabilities for the transitions $A \rightarrow A_F$, $A \rightarrow A_R$ and $A \rightarrow A_T^\alpha$ respectively. We can then define all transition probabilities of the strict model as a function of those from the Markov-like model:

$$\begin{aligned}
 p_F(z, g) &= \begin{cases} p_u(F|z) & \text{if } g = 1 \\ p_d(F|z) & \text{if } g = 0 \end{cases} \\
 p_R(z, g) &= \begin{cases} p_u(R|z) & \text{if } g = 1 \\ p_d(R|z) & \text{if } g = 0 \end{cases} \\
 p_T^\alpha(z, g) &= \begin{cases} \frac{p_u(T|z)}{|\Theta_g|} & \text{if } g = 1 \\ \frac{p_d(T|z)}{|\Theta_g|} & \text{if } g = 0 \end{cases} \quad (3.2)
 \end{aligned}$$

where $|\Theta_g|$ denotes the cardinality of the set Θ_g ; we are therefore assuming that all values in Θ_g have an equal likelihood of being chosen. This completes our definition of a strict Markov model. Goal states are easily defined as they are in experiments with real worms as points that are within a certain distance d from the peak of the gradient. The set of goal states G can thus be defined as a subset of M , the set of all states in the Markov model:

$$G \subset M | \forall A(x, y, \theta, z, g) \in M, A \in G \iff \sqrt{x^2 + y^2} < d \quad (3.3)$$

If these goal states are made absorbing, it is possible in theory to calculate the mean time to absorption for any given set of transition probabilities for the original dual Markov-like model. Minimising this mean time to absorption is then equivalent to finding the set of transition probabilities

that allow efficient navigation towards the centre of the gradient by the original model. In order for this calculation to be possible, the number of states in the model must be finite.

It is easily shown, however, that any model of a useful size contains too many states to be analysed computationally. Assume for instance that the model can move only on a virtual petri dish with a diameter of 10cm. By definition, the number of states of our model in such a world would be finite. Now, for simplicity, we assume that the model can only move on a rectangular grid world whose points are separated by a distance v . We are thus severely underestimating the total number of states by setting $\Theta_g = \{0, \pi/2, \pi, 3\pi/2\}$. The number N of possible points on the virtual petri dish that can be reached by the model is thus approximately given by the area of the dish if it is expressed with v as the unit length. For a realistic value of v (Ferrée and Lockery, 1999), this gives $N = \pi \left(\frac{5}{v}\right)^2 = \pi \left(\frac{5}{0.022}\right)^2 \approx 162\,190$. Remembering that the model attaches $6|\Theta_g|$ states to each of these points, we have a total number of 3 892 560 states in this model, which means the transition probability matrix would contain $\approx 1.51 \times 10^{13}$ elements. Since we have underestimated the number of states by several orders of magnitude due to our restrictive choice of Θ_g , it is clear that it is not practical with the currently available computational resources to evaluate this model further. We therefore leave it as a theoretical model which serves as an illustration of the concept of translating Markov-like models into strict Markov models and will analyse the behaviour of *C. elegans* based on numerical simulations as discussed in Chapter 5. In the next section, however, we will show, that an analytical evaluation is not always impractical.

3.2 Moth behaviour as a dual Markov-like model

3.2.1 Moth behaviour

Similarly to *C. elegans*, the moth is also on occasion faced with a navigation problem that requires it to find the source of a chemical. Unlike *C. elegans*, however, the moth may not make use

of a clear continuous stream of information about its progress towards said source. Typically, the source, which the moth attempts to reach, emits a chemical plume consisting of individual pockets of odour which propagate with the wind (Balkovsky and Shraiman, 2002). The moth is thus not exposed to a chemical gradient but has to navigate based only on the information of whether or not it has found one of these pockets.

In nature, the moth does so by employing a so-called surge-and-cast strategy (Baker, 1986; Vickers and Baker, 1994): if it encounters a pocket, it will fly directly into the wind (surge) but if it goes without such encounters for some time, it moves backwards and forwards perpendicularly to the wind direction until it finds another pocket (cast). It is this surge-and-cast strategy and its optimality as a solution to the navigation problem that we wish to investigate here.

3.2.2 A dual Markov-like model of the moth plume navigation behaviour

Although the real moth moves in a continuous three dimensional world, simple models can be defined on a two dimensional grid world (Sánchez-Montañés and Pearce, 2006; Balkovsky and Shraiman, 2002), a Cartesian space in which the possible points the model can occupy are all $M(x,y)|x,y \in \mathbb{Z}$. In such a world with the assumption that the moth never stays at the same position two time steps in a row, the model has four possible points it can move to if currently at a point with coordinates (i, j) : (F)orward to the point $(i - 1, j)$, (U)p to the point $(i, j + 1)$, (D)own to the point $(i, j - 1)$ and finally (R)everse to the point $(i + 1, j)$. These four possible movements thus form the states of our Markov-like model of the moth (Fig. 3.3), similarly to the previous one for *C. elegans* (Fig. 3.2). Additionally, Sánchez-Montañés and Pearce (2006) have shown that it is sufficient to consider just two different strategies for understanding moth chemotaxis: one when a pocket has been encountered and another otherwise. Similarly to the *C. elegans* model, the moth model is thus dual, with one set of transition probabilities for each of these two possibilities. We can thus again write the transition probabilities for this model as $p_k(Y|X)$ for moving from state X

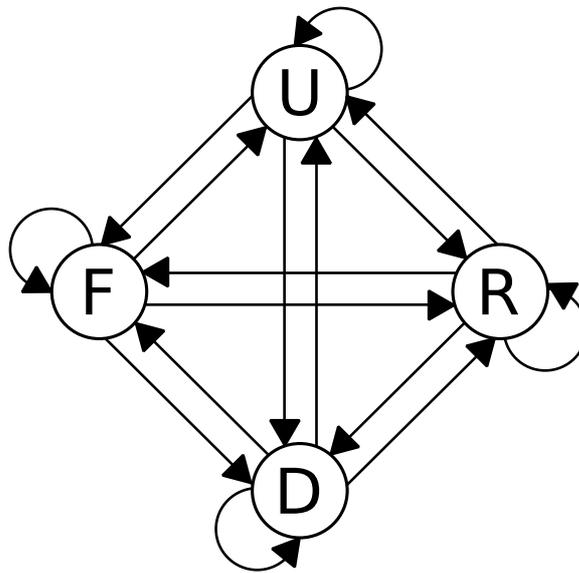


Figure 3.3: A dual Markov-like model of moth chemotactic navigation in a simple grid world. Possible state transitions are indicated by arrows but the corresponding transition probabilities are not explicitly indicated for clarity. In this model, the F and R states correspond to forward and backward movements, equivalent to a step to the left or the right respectively in the grid world. Similarly, the U and D states correspond to lateral movements, equivalent to a step up or a step down in the grid world.

to state Y with $k \in \{1, 2\}$, where a value of 1 indicates the transition probability value in the case of an encounter of a pocket and 2 the transition probability value otherwise.

It is worth underlining that there is a difference between this model and that of Sánchez-Montañés and Pearce (2006): the latter models the locomotion by randomly selecting one of the F, U, D or R states at each time step, where the probabilities of getting picked for each state are guided by the information of whether or not a pocket has been encountered at that time step. In the present model, due to its Markovian nature the probability for each state of getting picked in the next time step additionally depends on the state the model is currently in, potentially allowing more complex behaviour to emerge.

To be complete for the present purpose, the grid world needs to feature a source and emitted pockets. Here, we model this in the same way as Sánchez-Montañés and Pearce (2006) and Balkovsky and Shraiman (2002): at every time step (1) a pocket appears at the source, which is located at the origin and (2) all existing pockets currently at coordinates (i, j) move with equal probability to either $(i + 1, j + 1)$, $(i + 1, j)$ or $(i + 1, j - 1)$.

3.2.3 Conversion into a strictly Markovian model

Since the model moves in a very simple discrete grid world where the source of the plume is the goal state the model should reach, we can define a new Markov model in which each state represents a point on this grid. While this model could potentially have an infinite number of states, it remains Markovian because the number of states would be countably infinite. Similarly to the *C. elegans* model then, every point on the grid is represented by four states in the new model to incorporate the states from the original Markov-like model. Likewise, every state of the strict model can have up to four successor states (Fig. 3.4).

The transition probabilities into these states additionally depend on the fact of whether or not a pocket has been encountered in the current state. Pockets move on the (x, y) plane alone in a random

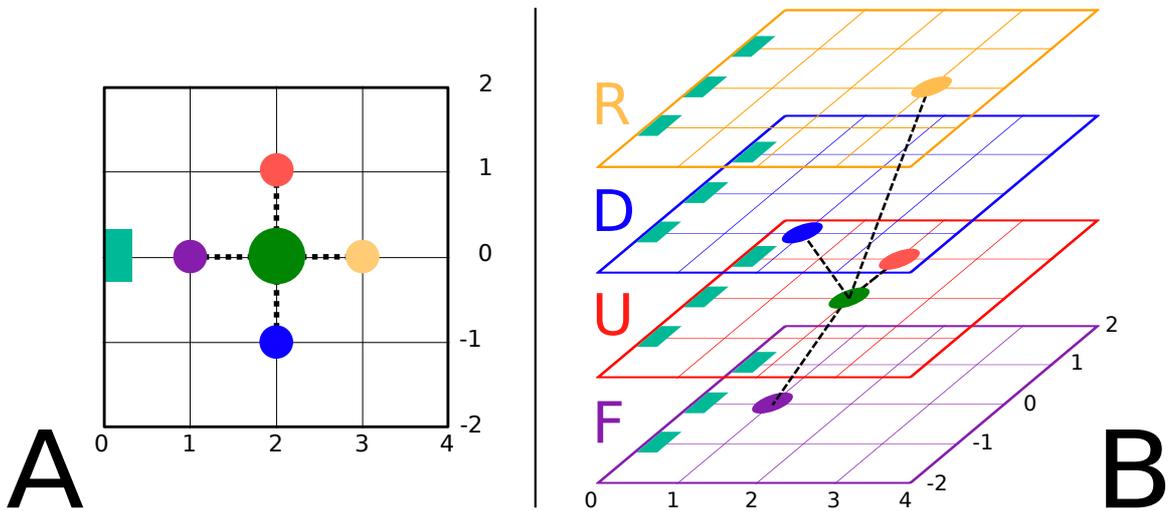


Figure 3.4:

A visual representation of a strict Markov model based on the Markov-like model of Fig. 3.3. The example is for a 5×5 grid world. **(A)** The grid world in which the model can move. Current location is given by the green dot, possible locations at the next time step by the remaining coloured dots: a forward movement places the model at the purple dot, a reversal at the orange dot, an upward movement at the red dot and a downward movement at the blue dot. Turquoise rectangle represents the plume source. **(B)** Corresponding strict Markov model. Every crossing of lines represents a state. The different actions the moth can perform are implemented as different layers in the strict model. Goal states are indicated by turquoise rectangles. An example of a current state and its the four successor states is shown based on **(A)**.

walk fashion - it is thus possible to calculate the probability of encountering a pocket in a given point on the grid at any time. Let $m(i, j)$ be this probability for the point (i, j) . Finally, let $A(x, y, z)$ be the current active state, with $z \in \{F, U, D, R\}$ corresponding to the states of the Markov-like model. Its four possible successor states are (Fig. 3.4):

- $A_F(x - 1, y, F)$, equivalent to a forward movement by the model
- $A_U(x, y + 1, U)$, equivalent to an upward movement by the model
- $A_D(x, y - 1, D)$, equivalent to a downward movement by the model
- $A_R(x + 1, y, R)$, equivalent to a backward movement by the model

The transition probabilities into each of these states depend on the value of z and $m(x, y)$ as well as the transition probabilities of the original model. Let $p_F(x, y, z)$, $p_U(x, y, z)$, $p_D(x, y, z)$ and $p_R(x, y, z)$ be the transition probabilities for the transitions $A \rightarrow A_F$, $A \rightarrow A_U$, $A \rightarrow A_D$ and $A \rightarrow A_R$ respectively. We have:

$$\begin{aligned}
 p_F(x, y, z) &= m(x, y)p_1(F|z) + (1 - m(x, y))p_2(F|z) \\
 p_U(x, y, z) &= m(x, y)p_1(U|z) + (1 - m(x, y))p_2(U|z) \\
 p_D(x, y, z) &= m(x, y)p_1(D|z) + (1 - m(x, y))p_2(D|z) \\
 p_R(x, y, z) &= m(x, y)p_1(R|z) + (1 - m(x, y))p_2(R|z)
 \end{aligned} \tag{3.4}$$

These equations are sufficient to define a strict Markov model over the grid world in which the moth moves, but we still need to give an expression for $m(x, y)$. For this, we can reformulate the problem based on the dynamics of the plume: given some y , how many different ways are there to add -1 , 0 and 1 in x steps in order to obtain y ? The answer, usually written $\binom{x}{y}_2$, is given by

the y th element in the x th row of the trinomial triangle¹ or by Andrews (1990):

$$\binom{x}{y}_2 = \sum_{j=0}^x \frac{x!}{j!(j+y)!(x-2j-y)!} \quad (3.5)$$

where not every term of the sum or even the sum itself necessarily exist. From this, we can obtain $m(x, y)$ by noting that the coefficients in the i th row in the trinomial triangle sum to 3^i :

$$\begin{aligned} m(x, y) &= \frac{\binom{x}{y}_2}{3^x} \\ &= \frac{\sum_{j=0}^x \frac{x!}{j!(j+y)!(x-2j-y)!}}{3^x} \end{aligned} \quad (3.6)$$

This completes our definition of a strict Markov model based on the grid world with a plume of pockets based on our previous dual 3-state Markov model of the moth behaviour.

3.2.4 Goal states and optimisation in a finite grid world

Since the strict Markov model has a state per point in the grid world, the source of the plume is also a state (in fact, 4 states, one for each value of z) in this model. Also, failure states (with $x = -1$, effectively meaning the model has moved past the source) are available. Based on this, we can immediately calculate the probability of absorption into one of the goal states for any starting position. It is also possible to calculate the mean time to absorption, but it is important to stress that this represents the mean time to reach either a goal or a failure state. It is not possible to single out goal states in the calculation of the mean time to absorption; to do so would require defining only the goal states as absorbing states but this would then change the behaviour of the entire Markov

¹A numerical triangle similar to Pascal's triangle, except that each value in a row is given by summing the left, middle and right values in the row above.

model since failure states would become transient. Nonetheless, we will show that the mean time to absorption can be a useful metric in this optimisation task.

At this point, we need to consider the grid world in which the model will move. The Markov model as defined previously can accommodate an infinite number of states, but in order to calculate the mean time to absorption and the probability of absorption into a particular state, the transition probability matrix of the model is needed, which in turn requires the number of states to be finite. The number of states is further limited by computational restrictions: since there is a state for every point in the grid world and there is an added third dimension to accommodate the different states of the original dual model, the number of states is given by $4XY$, where X is the number of points on the x -axis and Y the corresponding value for the y -axis. Therefore the number of elements in the transition probability matrix is given by $(4XY)^2$ and for the example of $x \in [-1, 30]$ and $y \in [-60, 60]$, this evaluates to 225 120016 elements. It is therefore clear that neither a very large world nor an attempt at deriving an analytical expression for both the probability of absorption into a goal state and the mean time to absorption are practical. We therefore derive the mean time to absorption and the probability to absorption at run time from a specific set of transition probabilities for the original dual model and use these values to optimise that model. Section 3.3 will discuss some of the techniques that can be used specifically in MATLAB to keep the computations feasible within a reasonable time scale.

Whenever one has to limit the size of the world in which a model can move, two important questions need to be answered: 1) What happens to the model if it attempts to move outside of the world and 2) How do the results generalise to larger, potentially infinite worlds?

For the present purposes, we address the first point by defining that, if the model would normally move outside of the world with a probability p , it will instead remain in the current position with the same probability. It should be noted that this will have an effect on some models that have a natural tendency to attempt to move beyond the borders of the world - for starting points close to

the border, the mean time to absorption may be reported wrongly depending on the transition probabilities of the original model. This is because the way we define the border prohibits the model from entering the state that would have taken it outside of the world (typically Up or Down) and instead forces it to remain in the current one. This is acceptable here because models that would be seriously affected would by definition not display a behaviour aimed at taking them close to the source and therefore would not be interesting. In general, the higher the probability of success for a given point, the more likely the model is to move towards the source from that point onwards and consequently the probability that it will move to the border and be affected by it drops.

The second point is easily addressed by evaluating the results from a world of a given size either in a larger or in an infinite world. To this effect, we define the size of the world in which the model is optimised as $x \in [-1, 30]$ and $y \in [-60, 60]$ whereas the result will be evaluated again based on the Markovian properties of a world with $x \in [-1, 50]$ and $y \in [-100, 100]$ as well as by running a model moth based on the optimised transition probabilities in an infinite world.

3.3 Computational implementation

In this section, we detail computational techniques that are useful for implementing calculations with the large transition probability matrices resulting from our model in MATLAB to exemplify how a careful design of a program can significantly reduce the computational cost of the operations. First, however, we recall the definitions of the mean time to absorption and the probability of absorption into a given state. Details on the optimisation process are also given.

3.3.1 Properties of absorbing Markov chains

The canonical form of the transition probability matrix of an absorbing Markov chain is:

	t_1	\dots	t_n	a_1	\dots	a_m
t_1	Q			R		
\vdots						
t_n						
a_1	0			I_m		
\vdots						
a_m						

Where $t_1 \dots t_n$ are transient and $a_1 \dots a_m$ absorbing states. I_m denotes the identity matrix of size m , 0 is a matrix of all zeros. Based on this form, the fundamental matrix N of the model can be defined as $N = (I_n - Q)^{-1}$. N always exists for absorbing Markov chains and contains the number of steps the model can be expected to spend in each state (columns) before absorption if started in a given state (rows) (for a proof, see Grinstead and Snell, 1997, p. 418). The mean time to absorption for each starting state thus follows trivially by summing across all rows in N , producing a vector T whose i th element is the mean time to absorption of the model if started in state t_i . Further, calculating $A = N \times R$ creates a n by m matrix in which each element (i, j) is the probability of absorption into state a_j if the model is started in state t_i .

3.3.2 Considerations for the implementation in MATLAB

The first important observation is that although the number of states and thus the size of the transition probability matrix is large, each row will only contain four non-zero values at most (since each state can only have a maximum of four successor states, by definition). The largest part of the transition probability matrix therefore consists of zeros and is thus useful to use a *sparse* matrix to store the transition probabilities, significantly reducing the memory load.

The second observation is that in order to compute the fundamental matrix N we first need to generate an appropriately sized identity matrix. Since we have made the transition probability matrix (and thus Q) sparse, it is sensible to generate a sparse identity matrix using the `speye` instead

of the eye function, resulting in a significant time saving.

Thirdly, we note that to compute the fundamental matrix, we need to take the inverse of $I_n - Q$, which is computationally very expensive for large matrices. It is thus worth realising that summing the rows of a square matrix with side length n , which is required to compute the mean time to absorption, is equivalent to multiplying this matrix with a column vector of length n of ones. Naming this vector C_n , we thus have $T = (I_n - Q)^{-1} \cdot C_n$ and we note that the probability of absorption is given by $A = (I_n - Q)^{-1} \cdot R$. For all values of interest, we therefore need to multiply the inverse of a matrix with another matrix. In MATLAB, this is an operation implemented as *matrix left division*, defined as $A \setminus B = A^{-1} \cdot B$, which computes the result with negligible error without actually generating the inverse. Using this operator we can therefore again save a significant amount of time.

Together, these considerations make the calculation of both the mean time to absorption and the probability of absorption into a goal state feasible at run-time in an optimisation algorithm.

3.3.3 Optimised values

All transition probabilities are optimised in a Cartesian world with coordinate ranges $x \in [-1, 30]$ and $y \in [-60, 60]$. Like Sánchez-Montañés and Pearce (2006), we consider the model to be successful if it arrives at either of the points $(0, -1)$, $(0, 0)$ or $(0, 1)$ (see Fig. 3.4) and to have failed if it arrives at a point with an x -coordinate of -1 without having passed through one of the aforementioned goal points. The total probability of absorption into a goal state p_γ , *i.e.* the probability of success, is thus defined as the sum of the probabilities for arriving in each of those states.

The mean time to absorption cannot be defined exclusively for the goal states, as previously discussed. Also, the mean time to absorption can vary with both x and y . To be able to optimise the mean time to absorption in a meaningful way for all points in the world it therefore needs to be normalised for the distance from the source. We note that the mean time to absorption into a

goal state is dependent on both x and y , but the mean time to absorption into a fail state mostly depends on x . This is because the goal states are located in a small region around the origin while the fail states form a line with equation $\mathcal{F} : x = -1$. Any normalisation of the overall mean time to absorption will thus depend on the probability of success in the sense that there is a probability p_γ that the normalisation will depend on x and y and a probability $1 - p_\gamma$ that it will only depend on x . This yields:

$$\tau_n(x, y) = \tau_{xy} \left(\frac{p_\gamma}{\sqrt{x^2 + y^2}} + \frac{1 - p_\gamma}{x + 1} \right) \quad (3.7)$$

where $\tau_n(x, y)$ is the normalised and τ_{xy} the original mean time to absorption for a starting point with coordinates (x, y) . There is a small error here caused by the fact that the points $(0, 1)$ and $(0, -1)$ are also goal states, which is not taken into account in the formula, but since their distance to the origin is small, the error is negligible for our purposes.

Since we are interested in finding results that are general for the entire world, it is best to consider p_γ for all points in the world but τ_n only for points that show a non-zero probability of success. We therefore optimise either the mean of p_γ or τ_n over all points for which they are considered. Other possible values for optimisation would be for example maximal or minimal values found in the entire world, but initial trials have shown this to be less useful than the mean, in particular because in some conditions there will always be points with either a 0 probability of success or a very large mean time to absorption. Therefore, optimising for those values would almost certainly result in a flat shape of the cost function masking otherwise interesting minima.

3.3.4 Optimisation algorithm

A standard simulated annealing approach (Kirkpatrick et al., 1983; Cerny, 1985) is used to optimise the system. The transition probability function $P = \exp((E_s - E_n)/T)$ for $E_n > E_s$ and $P = 1$

transition probabilities				transition probabilities					
model		model		model		model			
A	T	A	T	A	T	A	T		
$p_1(F F)$	1	1	$p_1(F D)$	1	$p_2(F F)$	0	1	$p_2(F D)$	0
$p_1(U F)$	0	0	$p_1(U D)$	0	$p_2(U F)$	0.5	0	$p_2(U D)$	0.5
$p_1(D F)$	0	0	$p_1(D D)$	0	$p_2(D F)$	0.5	0	$p_2(D D)$	0.5
$p_1(R F)$	0	0	$p_1(R D)$	0	$p_2(R F)$	0	0	$p_2(R D)$	0
$p_1(F U)$	1		$p_1(F R)$		$p_2(F U)$	0		$p_2(F R)$	
$p_1(U U)$	0		$p_1(U R)$		$p_2(U U)$	0.5		$p_2(U R)$	
$p_1(D U)$	0		$p_1(D R)$		$p_2(D U)$	0.5		$p_2(D R)$	
$p_1(R U)$	0		$p_1(R R)$		$p_2(R U)$	0		$p_2(R R)$	

Table 3.1: Resulting transition probabilities when the model is optimised for either mean time to absorption (T) or probability of absorption into a goal state (A) alone. Values which have no effect on model behaviour are omitted. Of interest is that the reversal state is used in neither solution. Further, the solution minimising the time to absorption relies solely on forward movement whereas the solution maximising the probability of absorption into a goal state shows a typical surge and cast behaviour.

otherwise is also standard. Here, T is the current temperature of the system and E_s and E_n are the energies of the current state and the selected neighbour respectively. The initial temperature of the system is set to a conservative $T = 20 \times 10^3$ ° and the cool-down for each time-step is defined as $T_{t+1} = \alpha T_t$ with $\alpha = 0.99$.

3.4 Optimised source location behaviour

All optimisations were done based on a model whose world size was given by $x \in [-1, 30]$ and $y \in [-60, 60]$. For the purposes of illustration however, we re-evaluate those models in a larger world ($x \in [-1, 50]$ and $y \in [-100, 100]$) to show that the results are not dependent on the size of the world. This is further underlined by an evaluation through a numerical simulation in an infinite world.

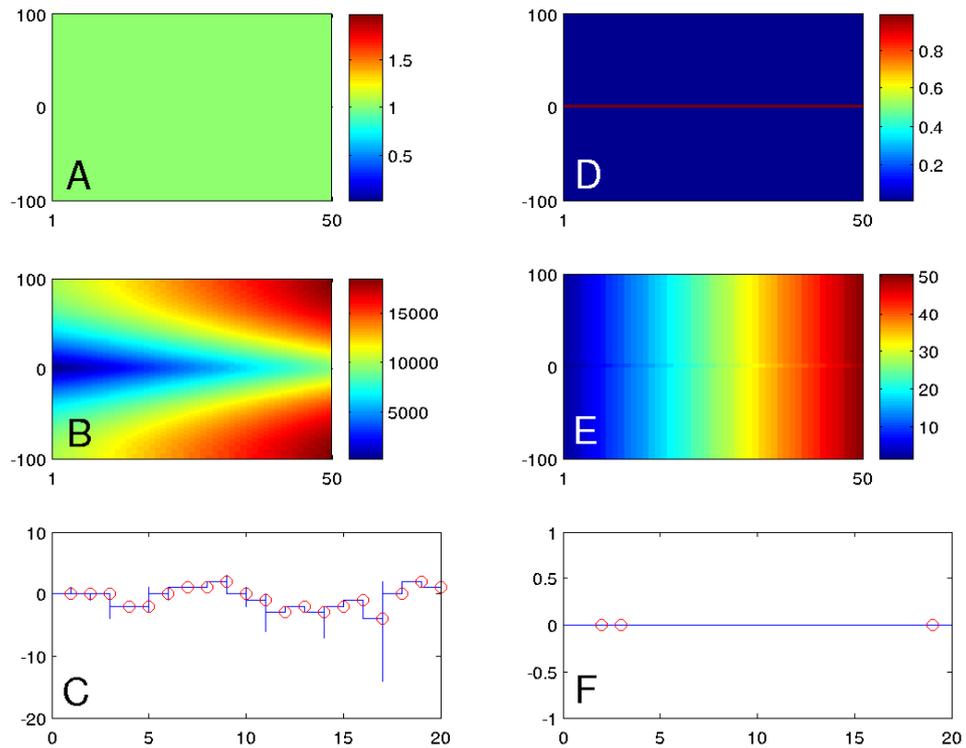


Figure 3.5: Properties and behaviour of the two solutions in a finite two dimensional grid world. **(A)** Probability of absorption in a goal state and **(B)** mean time to absorption for each possible starting point of a world with $x \in [1, 50]$ and $y \in [-100, 100]$ when the model is optimised to maximise the probability of absorption. **(C)** Example track when the model is run in an infinite world. Red circles indicate positions at which a pocket has been encountered. Source is located at coordinates $(0, 0)$ Note that vertical lines may actually be multiple overlapping up-and-down movements. **(D, E, F)** Analogue results when the model is optimised to minimise mean time to absorption.

3.4.1 Optimisation of single properties

Table 3.1 lists the two solutions found when either the mean normalised time to absorption was minimised or the probability of absorption into a goal state maximised. It is immediately obvious that the solution using a minimal normalised mean time to absorption is not realistic as it ignores all information about the plume and simply moves forward. On the other hand, the solution maximising the probability of absorption into a goal state displays behaviour, also found by Sánchez-Montañés and Pearce (2006), that is reminiscent of the strategy used by the moth: when a pocket is encountered, the model always moves forward, otherwise it moves up or down with equal probabilities. It can be pointed out that this solution for maximal probability of absorption into a goal state is not unique; other solutions which use different transition probabilities between the up and down states when no pocket is present exist. Common to all solutions however is a probability of 0 for moving forward if no pocket has been encountered.

Figure 3.5 shows the probability of absorption into a goal state and the mean time to absorption for every possible starting position in the grid world as well as example tracks of model moths driven by the respective transition probabilities. The most important observation here is that the mean times to absorption for the model optimising the probability of absorption into a goal state are generally very large. Conversely, the model optimised solely for a low mean time to absorption has a success rate of 0 for all starting positions not directly opposite the goal states, rendering it useless for all practical purposes.

3.4.2 Optimisation of combined properties

Based on these results, it is interesting to investigate if configurations exist that achieve a reasonable success rate within a reasonable time. In principle, this can be analysed by restraining one parameter while optimising the other: maximising the probability of absorption into a goal state while forcing the time to absorption to be low or minimising the time to absorption while forcing

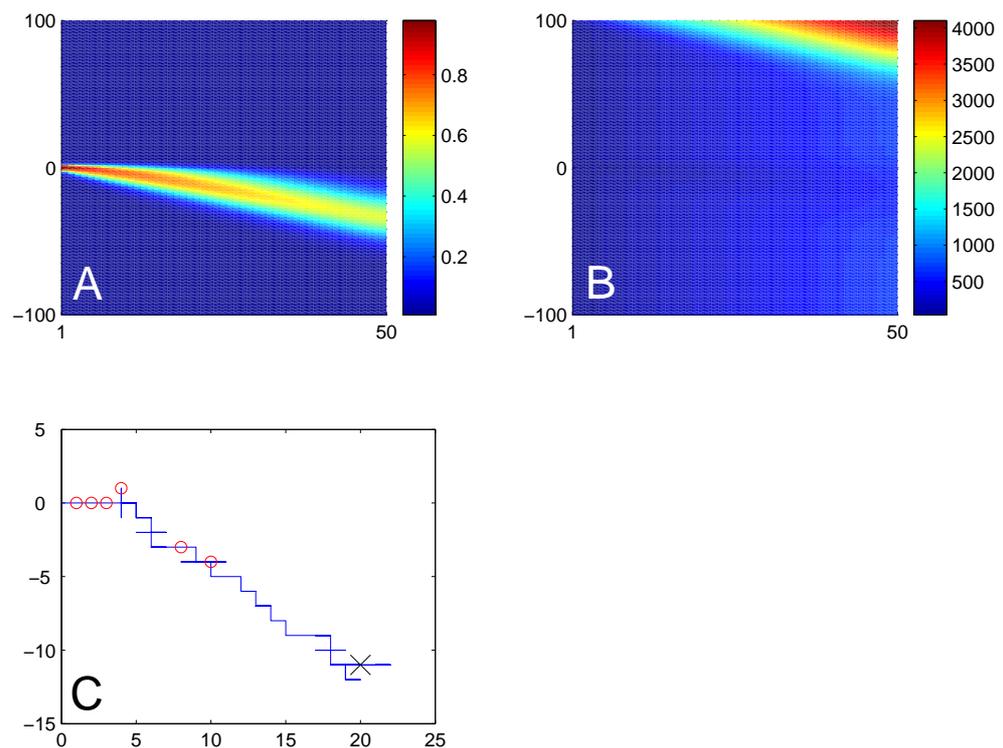


Figure 3.6: Optimal model maximising the probability of absorption while keeping the mean time to absorption low. (A) Probability of absorption and (B) mean time to absorption for each possible starting state. (C) An example successful track by a model moth based on these results. X denotes the starting position to avoid confusion based on the use of reversals in this model. Circles represent encounters with pockets of the plume. The shown trajectory may hide overlapping movements.

transition probabilities							
$p_1(F F)$	1	$p_1(F D)$	0.7	$p_2(F F)$	0.1	$p_2(F D)$	0.6
$p_1(U F)$	0	$p_1(U D)$	0.3	$p_2(U F)$	0.18	$p_2(U D)$	0.16
$p_1(D F)$	0	$p_1(D D)$	0	$p_2(D F)$	0	$p_2(D D)$	0.096
$p_1(R F)$	0	$p_1(R D)$	0	$p_2(R F)$	0.72	$p_2(R D)$	0.144
$p_1(F U)$	0.1	$p_1(F R)$	0.5	$p_2(F U)$	1	$p_2(F R)$	0.8
$p_1(U U)$	0	$p_1(U R)$	0	$p_2(U U)$	0	$p_2(U R)$	0
$p_1(D U)$	0.9	$p_1(D R)$	0.5	$p_2(D U)$	0	$p_2(D R)$	0.06
$p_1(R U)$	0	$p_1(R R)$	0	$p_2(R U)$	0	$p_2(R R)$	0.14

Table 3.2: The solution found if the probability of success is optimised while the time to absorption is forced to be low. Of interest is the high probability of alternating backwards and forward movement when no pocket is present which is most likely to be interrupted by an upward movement. The general strategy of the model is thus to make a slow progress towards the upper left, but with a mostly intact surge behaviour (different from the classic surge behaviour only because there is also a chance to move downwards when a pocket has been encountered).

the probability of absorption into a goal state to be high (*e.g.* mean over all states > 0.7). When the latter is done, the solution previously found when maximising only the probability of absorption into a goal state is found again (Fig. 3.5A and B). This is a strong indication that this is in fact the most time-efficient solution if a high probability of absorption into a goal state is required.

Fig. 3.6 and Tab. 3.2 show the result of optimising the probability of absorption while keeping the normalised mean time to absorption τ_n below 10. Although this value is chosen arbitrarily based on results from Sánchez-Montañés and Pearce (2006), the results are similar for different values. In particular the characteristic candle-flame shaped corridor with non-zero probability of absorption into a goal state (Fig. 3.6A) is always seen, although the direction may vary. The strategy employed here is thus a variation of the one used by the time-optimised model (Fig. 3.5D and E): the model moves more or less in a pre-determined direction (biased towards the upper left in this particular example) but unlike the first solution, the progress is slower and affected by pockets of the plume. In particular, it is interesting to note that the model has a high likelihood of presenting alternating forward and reverse steps, interrupted by the occasional upward movement. The strategy is thus a combination of slowly moving in a predetermined direction while frequently hovering around a

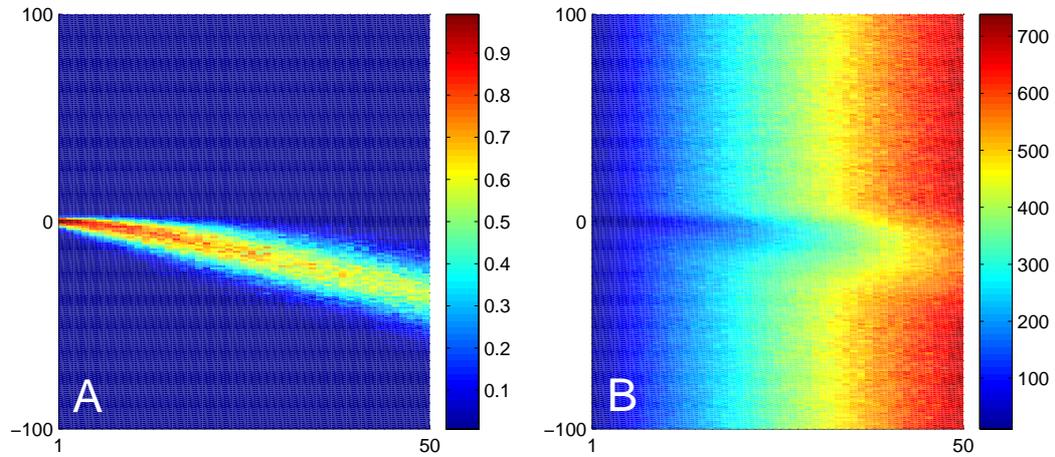


Figure 3.7: (A) Probability of absorption into a goal state and (B) mean time to absorption for the optimised model detailed in Tab. 3.2 based on numerical simulations with 100 runs per data point.

position, presumably to increase the chance of encountering a pocket if that position is within the plume. The overall failure of this solution to provide high probabilities of success for all possible points in the world is therefore a result of the need to move in a predetermined direction in order to satisfy the constraints imposed on the mean time to absorption.

Fig. 3.6B also exemplifies the previously discussed effect that defining a border can have on the results. It can be seen in the upper right corner that the mean time to absorption for points that are more likely to hit the upper border before they reach the goal or failure states increases more steeply compared to unaffected points. A comparison with Fig. 3.6A shows that this does not affect points with a non-zero probability of success. Therefore, the results also remain unaffected.

3.4.3 Comparison with numerical simulations

To investigate how the restriction of the model to a finite world for analysis affects the results, if at all, we have evaluated the set of transition probabilities detailed in Tab. 3.2 again using numerical simulations of model moths in an infinite world, starting at all points $(x, y) | x \in [1, 50]$ and $y \in [-100, 100]$. For each of these points, the success probability was calculated by starting a model

100 times and counting the number of successful runs. The mean time to absorption is calculated from the mean running time of each of these 100 simulations. Both results are shown in Fig. 3.7.

To assess the difference between the numerical results in an infinite world and those derived from Markovian properties in a finite world, we compute the RMSE between the two and find a very low value (0.067). The RMSE between the numerical and Markovian property-based normalised mean times to absorption is found to be high (13.92 steps). This is due to the previously discussed effect the border can have on the mean time to absorption for certain starting points. If only start states that have a non-zero probability of absorption into a goal state are considered, the RMSE drops to 1.20 steps.

We have thus shown that, although the restriction to a finite world can affect the mean time to absorption, it does not do so dramatically for points that are of interest. Additionally, the probability of absorption into a goal state computed with negligible error. In terms of computational requirements, the numerical simulation presented here has taken approximately two days on a Pentium IV 3.2GHz with Hyper-threading and 2GB of RAM whereas the equivalent evaluation of the Markovian properties is achieved in under 90s (and under 20s for the smaller world size used in the optimisation process itself), thus clearly underlining the advantage gained from the translation into a strict Markov model.

3.5 Summary of Chapter 3

This Chapter contains two main achievements which each merit a discussion in turn: we have shown how a dual Markov-like model of behaviour can be translated into a strict Markov model for analysis and we have analysed the behaviour of a very simple model of moth plume-navigation behaviour.

3.5.1 Uses and limits of the translation into strict Markov models

The translation into a strict Markov model has been motivated mainly through the desire to be able to optimise a behavioural Markov-like model by calculating its probability of success or the mean time required to achieve a certain goal analytically. The goal state (as well as the start states) must therefore be present in the resulting Markov model. Since we have concerned ourselves only with navigational problems here, the strict model had to be defined over the world in which the original dual state model can move, which has resulted in a rather large number of states in all examples considered here. This does not prevent the model to be used for other types of goal states (*e.g.* not being hungry) as long as the space in which the start and goal states exist are finite or countably infinite and the multi-state model of the behaviour navigates through this space.

This space also imposes the most severe limitation on the strict model - while the model itself may be countably infinite, it needs to be finite in practise so that the probability of absorption and the mean time to absorption can be computed from the transition probability matrix. The size of this finite world is then further restricted by the available computational resources. This limitation is obviously not always relevant - most laboratory behavioural experiments are carried out in finite surroundings (*e.g.* on a petri dish for *C. elegans*); models could thus be analysed in virtual versions of these experiments.

The restriction to a finite world does however require a definition for handling attempts by the model to move outside of the world, which again depends on the specific model under investigation. For the simple moth model used here, this was not defined *a priori*, so we chose to replace movements across the border with a stay in the current position. For *C. elegans* on the other hand, the abundant ethological experiments involving the observation of the animal on a petri dish (*i.e.* a finite, bounded world) allow the definition of appropriate responses, for instance via the touch avoidance response (Chalfie et al., 1985).

The moth model included a failure state, but as the *C. elegans* model shows, this is not always

necessary. The presence of failure states affects which values can be used in the analysis of the model. Without a possibility to fail, the probability of absorption into a goal state is meaningless but the mean time to absorption very relevant. If failure states are present, the probability of absorption is very informative but the mean time to absorption needs to be analysed with caution.

In general, if the restriction to a finite world in which a Markov like model can move is acceptable, it is useful to translate it into a strict Markov model so it can be evaluated without having to resort to repeated time consuming simulations, that can be subject to statistical fluctuations. The net gain depends on the problem; *C. elegans* moves in an environment which is nice in the sense that information about the environment remains similar in different runs because the gradient can be sampled reliably at any point in space. This is different for the moth: pockets of the source propagate according to a random walk and a trajectory that has encountered many plumes in one run may go without hitting a single one in a repeat attempt. It is thus much easier to numerically optimise the original dual Markov-like model of *C. elegans* (see Chapter 5) than it is to optimise the corresponding moth model. Therefore, while a translation into a strict Markov model is not necessarily required for the *C. elegans* model, it is essential for an analysis of the moth model.

3.5.2 Moth navigation towards the source of a plume

Using the strict Markov model, it has been possible to analyse the behaviour and performance of a dual Markov-like model based solely on the four directions the model can move in at any given step. It has been shown that minimising the mean time to absorption alone leads to clearly unreasonable behaviour. Maximising the probability of absorption into a goal state has resulted in a surge-and-cast behaviour similar to the ethological description for chemotaxis in the real moth, but the mean time to absorption was rather large. Attempting to optimise the probability of absorption while at the same time constraining the mean time to absorption has resulted in an interesting strategy not observed in the moth: the model will travel forward in a certain direction, usually at some

non-zero angle with respect to the x axis but will frequently hover in one position for a while which slows the overall travel down. This slow forward movement increases the probability of finding pockets and thus the probability of success for some starting points but the predetermined direction of overall movement means the starting positions that have a non-zero success probability form a corridor in the world.

Of particular interest is the result that any attempt to optimise for mean time to absorption while keeping the probability of success high also results in a surge-and-cast behaviour. While it has been speculated previously that such a strategy is indeed the best for being successful at finding the source (Balkovsky and Shraiman, 2002), this has not previously been shown explicitly. Similarly to the results by Sánchez-Montañés and Pearce (2006), it is found that 2 sets of transition probabilities are sufficient for this behaviour to emerge. The emerging casting behaviour in this case is rather primitive however and different, more efficient, casting strategies have been proposed which additionally make use of the restrictions on the source location a pocket provides (Balkovsky and Shraiman, 2002). It would thus be interesting to use the work presented here as a basis for discovering and evaluating the different types of casting strategies that exist. A question of particular interest is what type of information the moth actually uses in its navigation (Vergassola et al., 2007). Here, we have simply used the presence or absence of a pocket at any given time step, but it is also possible for instance, that the animal integrates the number of pockets it encounters in a given time window to direct its behaviour.

However, the purpose here was simply to illustrate that problems using Markov-like models which can be translated into strict Markov models while remaining computable exist and to show an example analysis. Further analysis of the moth behaviour will thus have to be deferred to a later date. For now, we will focus on ways to analyse computational models such as those introduced in this chapter.

Chapter 4

A framework for the systematic analysis of behavioural models

CHAPTER 3 has introduced a Markov-like model of *C. elegans* gradient navigation behaviour and has touched on the fact that it is interesting to consider the optimality of transition probabilities under certain conditions. In this chapter, we will present a comprehensive framework for analysing the behaviour of an animal based on a computational model. Besides being very useful for the study of behaviour, the framework has applications in other domains as well; this will be discussed at the end of the chapter.

4.1 Using optimisation to analyse behaviour

When optimisation techniques are applied to behavioural models, it can be with the aim of determining which cost parameter an observed behaviour minimises (assuming that the behaviour is optimal in some sense), as seen for instance in Optimal Foraging Theory (MacArthur and Pianka, 1966). In Action Selection studies, the aim is to identify how an animal might determine an optimal

course of action even though evaluating all possible courses is computationally prohibitive (see *e.g.* Seth, 2007). In such a scenario, finding a single optimal solution is usually sufficient. In the present work however, we are more concerned with finding at least a representative set of *different* configurations of a behavioural model as this would inform on the different possible strategies potentially available to an animal.

How does one go about finding such a representative set? The behaviour of a computational model is dictated by its parameters. In the Markov-like models used here, these are the transition probabilities but this may of course differ in other models. In general then, different strategies for reaching a given goal can be understood as being located in different regions of the parameter space of the model. Not every point in the parameter space will represent a good strategy. Some may be inefficient, others may not address the goal of the behaviour under investigation at all. It is therefore interesting to ask which regions of the parameter space of a model contain optimal strategies and which ones do not.

It is important to define what is understood by “regions” of a parameter space. Since our aim here is to apply the framework to Markov-like models, the parameter space will be finite since each parameter can only take on values from a limited range ($[0, 1]$ for transition probabilities). The entire parameter space of such a model will thus be an n -dimensional hypercube (assuming the range of each parameter is the same - if not, some sides of the parameter space may be longer than others and the shape would strictly be the n -dimensional equivalent of a rectangular prism but this has no consequences for the framework presented here) and a region is then simply an arbitrarily defined subspace contained within that hypercube. Trying to find a representative set of optimal configurations for any model then amounts to defining a set of such regions and determining for each of those regions whether or not it contains an optimal configuration.

Subspace searches have been used in previous optimisation studies as well, but with a different aim. Like most work in Optimisation Theory, these studies attempt to find methods to improve

the speed and/or the reliability of optimisation algorithms, either generally or for specific types of problems. This can be achieved, for instance, by identifying lower-dimensional subspaces of the entire search space most likely to contain the global optimum (see *e.g.* Byrd et al., 1987; Branch et al., 1999). Our interest here however is to find a set of optimal solutions by searching each subspace individually for one, a task for which these approaches are not designed for.

In a sense we are thus interested in obtaining a rough description of the shape of the cost function through searching the subspace. Other methods exist for analysing this shape, but they are mostly concerned with identifying local smoothness or shape. Philippides et al. (2005) for instance achieve this by first deriving an optimal solution and then slightly varying one or more parameters in order to assess to what extent this affects the performance of the solution. The local shape of the cost function at the point of the investigated solution can then be inferred. Again, however, we are more interested in the location of minima than the overall shape of the cost function. The main difference between the present study and other work in optimisation studies is thus the explicit search for multiple minima, which we aim to undertake by searching subspaces. Efficiency in the speed of execution is of lesser interest to us here (in contrast to many other studies), although we will show later that our approach here can in some tasks also perform well from this point of view.

If the aim is thus to derive a set of subspaces in order to find a representative set of minima, there are some restrictions on this set for it to be useful: 1) when considered together, the regions need to cover the entire parameter space and 2) no regions should overlap. The reason for the first restriction is obvious - without it some regions of the parameter space might never be explored at all. The second restriction ensures that we will not find the same optimal configurations in two different regions. With these restrictions in mind, a natural way to generate this set of regions is simply to divide the parameter space into a number of smaller spaces by cutting it along one or more dimensions as illustrated in Fig. 4.1 for a simple 2-dimensional space. There is no *a priori* restriction on the nature of these cuts; one could for instance divide one dimension into two parts

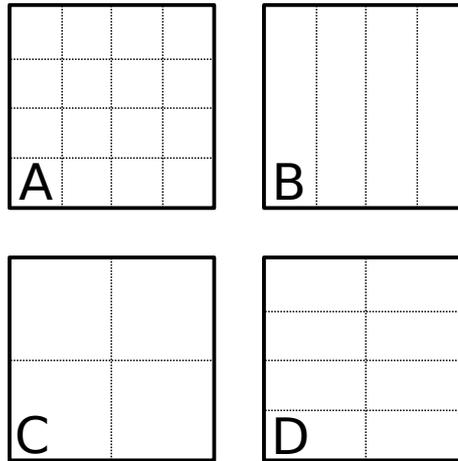


Figure 4.1: A simple illustration of how a parameter space can be divided into different regions by cutting along the dimensions. (A) 16 regions, (B) 4 regions, (C) 4 different regions and (D), 8 regions.

but another into four (Fig. 4.1D). Here, however, we only consider cuts dividing each parameter equally (e.g. Fig. 4.1A or C) - each dimension is therefore treated the same as far as the creation of subspaces is concerned.

An important side effect of treating each dimension the same is that it becomes possible to use a recursive approach for evaluating the regions of the parameter space. Consider for instance the example division of a 2-dimensional parameter space given in Fig. 4.1A. This space contains 16 regions and using a sequential approach, each of these regions would have to be evaluated in turn. It is easily seen that such an approach could be very time-consuming if the parameter space is high-dimensional. A recursive approach to finding optimal configurations within this parameter space is illustrated in a step-by-step fashion in Fig. 4.2. The parameter space is first considered in its entirety. An optimisation algorithm is run in this space and if an optimal configuration is found in it, the space is then divided into two subspaces and the optimisation algorithm rerun in each part. This subdivision of each subspace takes place until the optimal solution found in one fails to meet the performance criterion or until the division process has produced a subspace whose size meets a termination criterion. In the example here, the termination criterion is simply a side

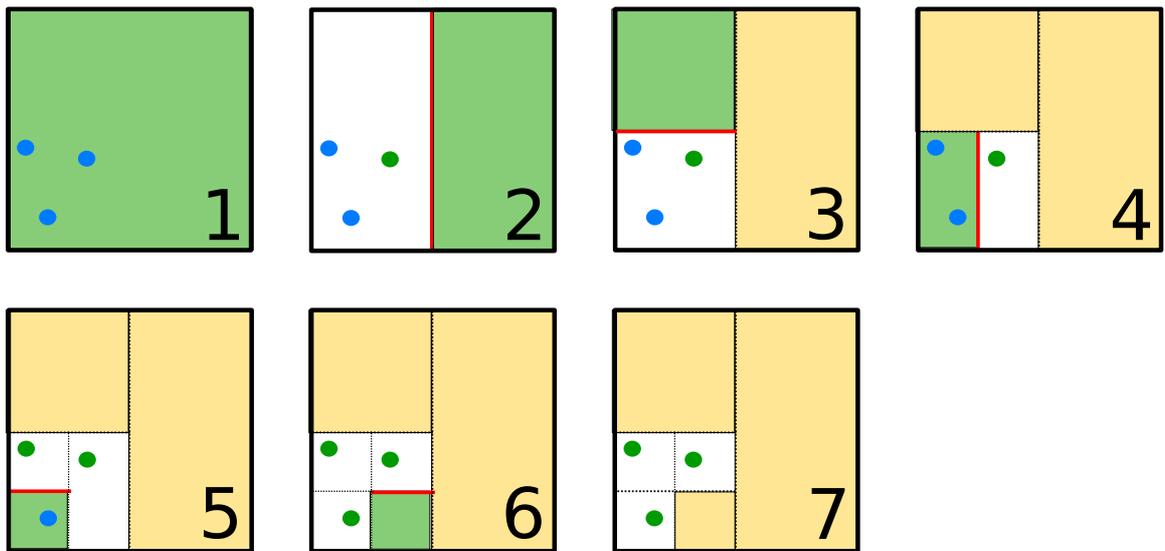


Figure 4.2: A step by step example of a recursive exploration of a parameter space, assuming a division into 16 subspaces as in Fig. 4.1A. Blue dots indicate the location of the optimal configurations the algorithm is trying to find. **Step 1:** Entire space is analysed, an optimal solution is found (green dot in subsequent steps). **Step 2:** The space is divided vertically into two equal halves (red line). An optimal solution has already been found in the left half, it does thus not need to be evaluated again. No good solution is found in the right half - this portion of the parameter space is now excluded from further analysis (coloured yellow in subsequent steps). **Steps 3-7:** repeats of step 2, dividing the remaining space alternatively along the horizontal and the vertical and evaluating each half unless an optimal solution is already known to exist in that half.

length smaller or equal to $1/4$ of the original space. In the general case of n -dimensional spaces, a good termination criterion could be based on the volume of the subspaces. Recursive analysis of the cost function has been used before, but in a different context. Koyamada et al. (2004) for instance use an optimisation algorithm which relies on a quadratic cost function in their study. Their overall cost function however could not be approximated by a quadratic function, a problem which they overcome by recursively subdividing the overall space into subspaces. However, the aim here still is to derive a single optimal configuration, while our interest remains in a representative set of them.

A final consideration which needs to be discussed concerns the definition of optimal configuration. When an optimisation algorithm is run in a subspace of the entire parameter space, the optimal solution it finds is only local to that subspace. It is therefore important to define what can be considered an important solution so that the local minimum found by the algorithm in a subspace can be correctly identified as being a global minimum or not. There are multiple possibilities for defining such a criterion of optimal performance, depending on the exact problem under consideration. One option is to base this criterion on the score of the solution found by the optimisation algorithm when it is run on the entire parameter space. In the case of stochastic models, there will always be a slight natural variation in the performance and repeated runs of an optimisation algorithm on the entire parameter space might thus generate a distribution of scores. This distribution can then be used for the definition of performance that can be considered optimal. An example of such an approach will be given in Chapter 5.

This completes the informal definition of our recursive approach for obtaining a representative family of optimal configurations for a given model of behaviour with respect to a given goal and a formal, slightly more elaborate definition will be given in the next section. It is important to realise that one of the main advantages of this approach is that it can discover a representative set of *different* optimal strategies with which the model can achieve its goal. There is less emphasis on finding all minima which are similar to each other, since the algorithm only looks for one optimal solution

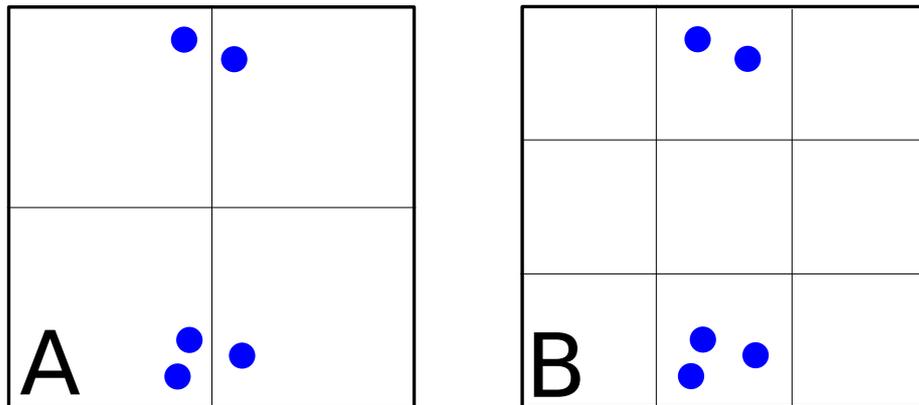


Figure 4.3: An example of how different ways of dividing the parameter space will affect the results. Dots represent all optimal configurations in the space. **(A)** A division which is not very informative since optimal solutions are located close to the dividing border. **(B)** A better division which is more useful in determining interesting regions within the parameter space.

per region. This is because two distinct but very similar strategies (represented by two points in the parameter space that are very close to each other), particularly in a stochastic model, are likely to generate the same overall behaviour. To what extent strategies are considered too similar to be different remains of course a matter of somewhat subjective judgement and is dependent on the problem under investigation. The present framework caters for this by allowing the division of the parameter space into arbitrarily small subsections. It is thus always possible to define the size of the regions one will consider in function of what one judges best for a given problem.

An issue that needs more careful consideration is that the algorithm for obvious reasons cannot place any restrictions on the location of the optimal configurations within the regions of the parameter space. This means that it is in principle possible that the optimal configurations found for two neighbouring regions are in fact located on the border between those regions and would thus encode similar behaviours. This is not necessarily a problem if it only happens a few times and it is easily verified if this is an issue by looking at the exact locations of the found optimal configurations within the parameter space. If too many are located at borders, it may be useful to rerun the algorithm so the parameter space is divided into smaller regions. This illustrates again that the best

way of dividing parameters within this algorithm is dependent on the specific problem (Fig. 4.3).

There are also a number of interesting general features to the framework: 1) it is independent of the specific optimisation technique used. This allows the investigator to choose whatever technique is best suited for the model he wishes to study. 2) It is independent of the chosen modelling technique. We have only discussed on Markov-like models here because such a model will be used in Chapter 5, but in principle any model with a finite parameter space can be used without modification to the algorithm. If the parameter space is infinite (for instance if one or more of the parameters can take any value), the division of the parameter space needs to be reconsidered since an infinite space cannot be divided into two halves. Assuming that an acceptable way of dividing the space can be found however, the algorithm remains usable.

4.2 Formal definition of the recursive algorithm

In this section, a formal definition of the recursive algorithm is given in sufficient detail to serve as a template for implementations. It is based on the previous section but extends the capabilities of the algorithm slightly beyond what has been discussed already, especially regarding the definition of the criterion of optimal performance.

Definition 1. Let S denote the finite n -dimensional parameter space and the sets $L_S = \{l_{S_i} | 1 \leq i \leq n\}$ and $U_S = \{u_{S_i} | 1 \leq i \leq n\}$ contain the lower and upper bounds for each dimension of S . We call division of the space S along dimension d and write $S \div d_2$ the creation of two spaces \mathcal{M} and \mathcal{N} with associated sets $L_{\mathcal{M}} = L_S$, $U_{\mathcal{M}} = \{u_{\mathcal{M}_i} | (\forall i \neq d, u_{\mathcal{M}_i} = u_{S_i}) \wedge (u_{\mathcal{M}_d} = u_{S_d}/2)\}$, $L_{\mathcal{N}} = \{l_{\mathcal{N}_i} | (\forall i \neq d, l_{\mathcal{N}_i} = l_{S_i}) \wedge (l_{\mathcal{N}_d} = l_{S_d}/2)\}$, $U_{\mathcal{N}} = U_S$.

Remark 1. This definition only shows the division of the parameter space into 2 halves for simplicity, but it is easily seen that the division can be defined for any $m \in \mathbf{N}$ as $S \div d_m = \{\mathcal{M}_1, \dots, \mathcal{M}_m\}$

Example 1. Let S be a parameter space such that $L_S = \{0, 0, 0, 0, 0\}$ and $U_S = \{1, 1, 1, 1, 1\}$. $S \div$

$2_2 = \{\mathcal{M}, \mathcal{N}\}$ such that $L_{\mathcal{M}} = \{0, 0, 0, 0, 0\}$, $U_{\mathcal{M}} = \{1, 1/2, 1, 1, 1\}$, $L_{\mathcal{N}} = \{0, 1/2, 0, 0, 0\}$ and $U_{\mathcal{N}} = \{1, 1, 1, 1, 1\}$.

Definition 2. Let $T = \{t_k | 1 \leq k \leq m\}$ be a set such that $t_k \in \mathbb{R}$ and $t_k < t_{k+1}$ if $k > 1$. Let s be the value of the minimum found in a parameter space S by an optimisation algorithm and S_{\min} its exact location. We call $v_S \in T | [\nexists w \in T | v_S > w > s]$ the value of S if $|T| > 1$. If $|T| = 1 \wedge s \leq T$, $v_S = s$.

Remark 2. v_S may not exist. This will be used in the algorithm below.

Definition 3. Let $V_S = (u_{S_1} - l_{S_1}) \cdot \dots \cdot (u_{S_n} - l_{S_n})$ be the volume of the initial parameter space S . We call $\tau | 0 < \tau \leq V_S$ the termination threshold of our algorithm.

Remark 3. τ may not be 0 in order to prevent infinite recursion in the algorithm below. $\tau = V_S$ is acceptable but does not improve on just running an optimisation algorithm on the entire parameter space.

Definition 4. Let X be a subdivision of the parameter space S and v_X the value of X . We call the output from X the triplet $P_X = \{X, X_{\min}, v_X\}$ and we call the output from S the set O of all triplets P generated by the algorithm on S .

Algorithm 1. Preliminaries: Let $d = 1$. Let $O = \emptyset$ be the set of results from the algorithm. Let S be the initial n -dimensional parameter space and define τ and \mathcal{T} as required.

Step 1: Obtain the location of the minimum S_{\min} and its score s in S from the optimisation algorithm and determine v_S .

Step 2: If v_S does not exist, let $v_S = +\infty$. Let $O = O \cup \{\{S, S_{\min}, v_S\}\}$ and halt.

Step 3: Calculate the volume V_S of S . If $V_S < \tau$, let $O = O \cup \{\{S, S_{\min}, v_S\}\}$ and halt.

Step 4: Perform $S \div d = \{\mathcal{M}, \mathcal{N}\}$.

Step 5: Let $d = d + 1$. If $d > n$, let $d = 1$.

Step 6: Run algorithm with $S = \mathcal{M}$, d and O .

Step 7: Run algorithm with $S = \mathcal{N}$, d and O .

Remark 4. *Time may be saved by aborting the optimisation algorithm in step 1 prematurely once a minimum $s|s < t, \forall t \in T$ is found.*

Using the present algorithm, one systematically divides the initial parameter space into smaller subspaces with a volume $V \leq \tau$ and value v_s , which allows to assess what local minima one may expect in different regions defined by τ of the parameter space. In particular, τ defines the resolution of this visualisation: as $\tau \rightarrow 0$ the resolution will increase.

The set T allows the user to define the different values for the minima that he is interested in and depends on the specific problem. This extends the notion of a criterion for optimal performance discussed previously. T may be a singleton, in which case it functions exactly like the previously discussed criterion.

4.3 Analysing the set of optimal solutions

Applying the recursive algorithm to the parameter space of a model thus generates a list of regions within the entire parameter space containing optimal configurations of the model. It follows trivially that regions which do not contain optimal configurations are now also known. The available methods for analysing this family depend in part on the nature of the model. If it is Markovian or Markov-like, one may for instance wish to look simply at the distribution of the transition probabilities across the optimal configurations found by the algorithm. It may also be possible to look at certain Markovian properties of these configurations. These model-specific types of analysis will be presented more fully in Chapter 5, in which the present framework is actually applied to a Markov-like model. Here, however, we focus on an analysis which is applicable independently of the model and very informative on the relative importance of each parameter of the model with respect to optimal performance.

It is possible to see the determination of whether or not a given region of the parameter space

contains an optimal configuration as a decision process on the values of the different parameters. As an example, consider again a simple 3-dimensional parameter space, which is divided into 8 equal regions (cut once along each dimension). Let the parameters be known as A , B and C , each varying between 0 and 1. After using the recursive algorithm, it is known for each of those regions whether or not they contain an optimal configuration. When focusing on the value range each parameter is allowed in a given region (S in the output from the algorithm as specified in section 4.2), this list will provide the following kind of information:

- If $0 \leq A < 0.5$, $0 \leq B < 0.5$ and $0 \leq C < 0.5$ a good solution is found.
- If $0 \leq A < 0.5$, $0 \leq B < 0.5$ and $0.5 \leq C \leq 0.5$ a good solution is found.
- If $0 \leq A < 0.5$, $0.5 \leq B \leq 1$ and $0 \leq C < 0.5$ no good solution is found.
- If $0 \leq A < 0.5$, $0.5 \leq B \leq 1$ and $0.5 \leq C \leq 0.5$ no good solution is found.
- If $0.5 \leq A \leq 1$, $0 \leq B < 0.5$ and $0 \leq C < 0.5$ a good solution is found.
- If $0.5 \leq A \leq 1$, $0 \leq B < 0.5$ and $0.5 \leq C \leq 0.5$ a good solution is found.
- If $0.5 \leq A \leq 1$, $0.5 \leq B \leq 1$ and $0 \leq C < 0.5$ no good solution is found.
- If $0.5 \leq A \leq 1$, $0.5 \leq B \leq 1$ and $0.5 \leq C \leq 0.5$ no good solution is found.

This information covers all the regions in the parameter space and can be represented as a decision tree, as shown in Fig 4.4. The tree shown is not the smallest possible tree. In fact, it is easily seen that the only decisive factor in determining whether or not a region contains an optimal configuration is the allowed value range of B since all good solutions are found when $B < 0.5$. While this is obvious in the present simple example, finding the smallest decision tree that accurately represents a given data set is not a trivial problem in the general case. It can be solved, however,

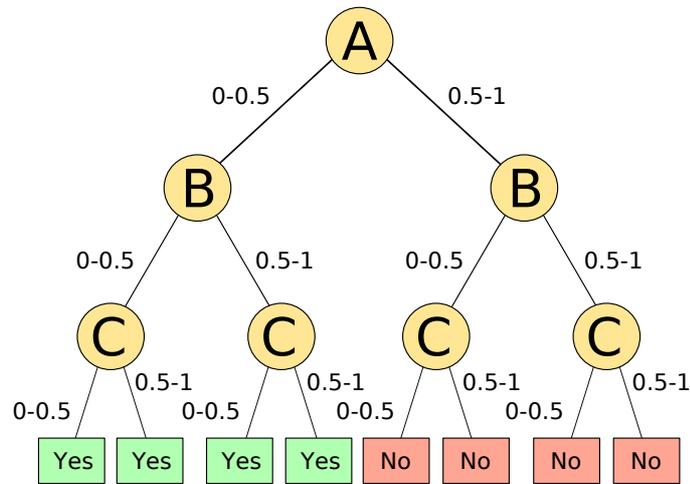


Figure 4.4: A simple decision tree based on the example from section 4.3. Yellow nodes represent parameters, edges the restrictions on the value of the parameter in the parent node and boxes indicate whether or not a particular combination of value ranges contains an optimal configuration.

using an AI technique called the ID3 algorithm (Mitchell, 1997). Briefly, the ID3 uses information theory (Shannon and Weaver, 1949) to compute the information entropy of the data set. In our example, we have 3 parameters, A , B and C and two possible conclusions - a region either does or does not contain an optimal configuration.

The entropy of the entire set S is then given by:

$$E(S) = - \sum_{i=1}^n \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \quad (4.1)$$

where $|S|$ is the cardinality of the S , $|S_i|$ the number of items with conclusion i and $n = 2$ since we only have two conclusions. A decision tree will split this entire set into a number of subsets along one of the parameters. The ID3 algorithm decides which parameter to use first for this split based on the reduction in entropy this will achieve for the resulting subsets. For a parameter Φ , the

algorithm therefore computes the gain of splitting S along this parameter:

$$G(S, \Phi) = E(S) - \sum_{v \in \Phi} \frac{|S_v|}{|S|} E(S_v) \quad (4.2)$$

where v represents each of the values that the parameter Φ can take ($0 - 0.5$ and $0.5 - 1$ for any parameter in our example), S_v is the subset of S in which Φ takes value v and $|S_v|$ is the cardinality of this set. The parameter which provides the highest gain is then used as the first node in the tree and the algorithm is rerun on each of the remaining subtrees until the entropy for a subtree is 0 or all parameters have been used in the parent tree. The resulting tree is then likely to be the smallest tree which can accurately represent the given data set.

In our example above, it is simple to see that the gain is maximal when the tree is divided along parameter B and that the resulting subtrees will all have an entropy of 0. The optimal decision tree in our example thus consists of a single node, B .

Issues can arise if the data set is not complete (*e.g.* if there were some regions in our example for which we wouldn't know whether or not they contained an optimal configuration) or if there are conflicts in the data (*e.g.* if one region were both labelled as containing and not containing an optimal configuration). However, our recursive algorithm ensures that these problems will not be encountered and it will thus always be possible to use the ID3 algorithm for the construction of a decision tree based on the output of the recursive algorithm.

The real interest here is not the tree itself, however. An example tree will be seen in Chapter 5, but in general, when the model analysed using this framework has a large number of parameters which are divided into several smaller regions, even a size-optimised tree may be too large to read. However, it is still possible to compute the average location of each parameter of the model in the tree, *i.e.* how far from the top node it can be found in the different subtrees. Since the algorithm

places parameters which are most relevant to reducing the entropy of the data set, and thus are most important in the behaviour described by the model, at the top, this effectively generates a ranking of the model parameters according to their importance in the behaviour the model is exhibiting. In our simple example, we would find that parameter *B* is the only important parameter in the model.

There are several advantages which can be gained from such a ranking. At its simplest, it provides a better understanding of the importance of the corresponding behavioural units with respect to the overall goal which in turn informs on the nature of the different strategies that exist to achieve this goal. If some behavioural units are not present in the ranking at all, as seen in our simple example above, this is also informative since those units were presumably used in the model because they were observed in the animal whose behaviour is being modelled. This analysis can thus identify behavioural units which are unlikely to contribute to achieving a certain goal and are thus more likely observed due to other, perhaps uncontrolled variables in the behavioural experiments which formed the basis of the computational model. A more detailed application of this type of analysis will be seen in Chapter 5.

4.4 Other applications for the framework

The recursive search of the parameter space is obviously useful not only for analysing computational models of behaviour but can be applied to any optimisation problem in which it is useful to understand the distribution of global minima across the parameter space. Since it is not restricted by the choice of model or optimisation technique, it is generally applicable. While it is of course always possible that faster approaches exist for a given specific problem, our approach can always be used if those are not yet known.

A second, perhaps more interesting application of this algorithm is that it can help dealing with local minima problems in heuristic optimisation tasks. Consider for instance the parameter

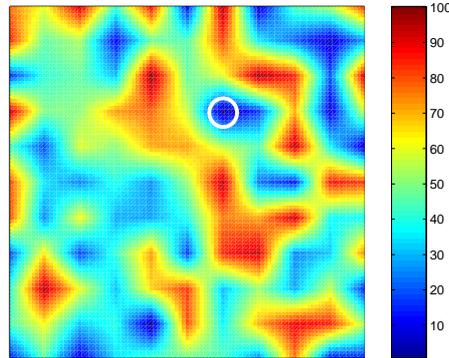


Figure 4.5: A randomly generated parameter space used here to illustrate some applications of our framework. Red indicates peaks, blue valleys. White circle shows global minimum.

space shown in Fig. 4.5. Finding the global minimum (indicated by the white circle, and with a value of 1.4233) is a hard task even for algorithms that are designed to overcome local minima, like Genetic Algorithms (Fraser and Burnell, 1970) or Simulated Annealing (Kirkpatrick et al., 1983; Cerny, 1985). Table 4.1 summarises the performance of a standard SA algorithm, our recursive approach with a variety of different divisions of the parameter space and a sequential subspace search using the same parameter space divisions. When using the recursive approach (using the same SA algorithm for the optimisation), the criterion for optimal performance is arbitrarily set to 10. It is easily seen that even when each parameter is just divided into two, the chance of locating the global minimum increases significantly from 2% to 62% as measured in 50 repeated runs of the algorithms. Similarly, the mean optima returned over those runs decreases from 6.58 ± 3.78 to 2.41 ± 1.76 indicating that even when the recursive algorithm fails to locate the global optimum, the best value found is still closer to the optimum in the recursive approach. The performance of the recursive algorithm only increases as the parameter space is divided into smaller spaces. However, the increased ability of dealing with local minima comes at an obvious cost - it requires more computational time (Table 4.1) The performance of the sequential subspace search is qualitatively similar and with a low number of subspaces almost identical to the recursive approach. As the

Technique	Prob. of finding minimum	Mean score found	Mean time (s)
Standard SA	0.02	6.58 ± 3.78	1.55
Recursive with $\tau = 1/4$	0.62	2.41 ± 1.76	8.95
Sequential with $\tau = 1/4$	0.68	2.19 ± 1.42	9.83
Recursive with $\tau = 1/16$	0.64	1.66 ± 0.78	20.97
Sequential with $\tau = 1/16$	0.70	1.65 ± 0.77	43.30
Recursive with $\tau = 1/64$	0.9	1.43 ± 0.01	48.62
Sequential with $\tau = 1/64$	0.9	1.43 ± 0.01	177.58

Table 4.1: The chance of finding the global minimum in the parameter space shown in Fig. 4.5 and the mean and standard deviation of the solutions found in 50 trials for a standard simulated annealing algorithm, a sequential subspace search and our recursive algorithm dividing the parameter space into 4, 16 and 64 subspaces respectively. The mean running time of the algorithms are also shown.

number of subspaces increases, however, the mean running time of the sequential approach becomes much larger compared to the recursive algorithm, which is expected since it always evaluates all subspaces.

To better judge the relation between the increased cost and the increased success, we ask how often each of the algorithms would need to be run in order to have returned the global minimum with a probability of 0.9 (*i.e.* the best observed success rate). To do so, we use the Pascal distribution (Feller, 1968):

$$f(t) = p(1-p)^{(t-1)} \quad (4.3)$$

which gives the probability that an event happens for the first time at time t if the probability of the event happening at any time is p . Summing from $t = 1$ to $t = n$ then gives the probability of the event having happened for any $t \leq n$. For a given p , we thus need to find the smallest n such that:

$$\sum_{t=1}^n p(1-p)^{(t-1)} \geq 0.9 \quad (4.4)$$

The results are summarised in table 4.2. It can be seen that under these circumstances, each of

Technique	Runs required	Approx. time required (s)
Standard SA	114	176
Recursive with $\tau = 1/4$	3	26.85
Sequential with $\tau = 1/4$	3	29.49
Recursive with $\tau = 1/16$	3	62.91
Sequential with $\tau = 1/16$	2	86.6
Recursive with $\tau = 1/64$	1	48.62
Sequential with $\tau = 1/64$	1	177.58

Table 4.2: The number of times an algorithm needs to be run to have found the global minimum with a probability of 0.9 and the corresponding time cost

the recursive algorithms outperforms the standard simulated annealing algorithm significantly. A similar effect has been found for the sequential approach: unless the number of subspaces is set to 64, it outperforms the standard approach significantly. Compared to the recursive approach, the performance is rather similar when the parameter space is divided into 4 regions, but this is no longer true for other divisions.

Interestingly, it can also be seen that dividing the parameter space into smaller regions does not necessarily result in a faster performance if a target probability of having found the global minimum is set. In our example, dividing the parameter space into 4 regions is a more time-efficient solution for finding the global minimum with probability 0.9 than a division into 64 regions.

While it is of course not normally known *a priori* with what probability a heuristic optimisation approach will find a global minimum in an unknown parameter space, it has thus still been shown that using our recursive approach as well as a sequential subspace search can both increase this probability in a single run at the expense of requiring more time and can achieve a target probability over multiple runs in less time than a standard approach. Compared to sequential subspace searches of the same granularity, the recursive approach mainly provides a time advantage which increases with the number of subspaces, but at least in our example above, the difference between the most time-efficient solutions for achieving a probability of 0.9 of finding the global minimum was minimal.

4.5 Summary of Chapter 4

This chapter has introduced a framework for analysing computational models of behaviour by discovering a family of configurations which allow the model to perform some behaviour optimally. This family of configurations is found by systematically searching the parameter space of the model for different strategies by which the model could achieve a certain goal. An analysis based on the ID3 algorithm has been introduced which allows the ranking of the model's parameters with respect to the goal-oriented behaviour. Additionally, it has been shown that the framework has uses outside of behavioural analysis, especially in the case of heuristic optimisation in a parameter space that is likely to trap a standard algorithm in local minima.

The strengths of the framework here are its generality and independence of specific modelling or optimisation techniques. This makes it very useful as a general tool in ethology for the analysis of goal-oriented behaviour. An example application to *C. elegans* gradient navigation behaviour will be given in Chapter 5.

Part III

***C. elegans* gradient navigation**

Chapter 5

Stochastic gradient navigation strategies of *C. elegans*

A model of *C. elegans* gradient navigation has been introduced in section 3.1. Here, we use this 3-state model to investigate this gradient navigation behaviour in detail. Initially, we answer the question: how do the behavioural units observed in *C. elegans* have to interact in order to navigate gradients towards their centre as fast as possible? Using the framework described in Chapter 4, we systematically derive a family of solutions, which represents the diversity of energy efficient solutions that could be adopted by the animal. This solution set is analysed for common properties, which allows us to determine how the behavioural units of the probabilistic model need to interact to navigate gradients efficiently and it will be shown that *C. elegans*-like strategies form the largest part of the optimal solutions. The fact that *C. elegans*-like behaviour emerges in the optimal configurations we derive allows us to be confident both in terms of the relevance of the original model as well as our new predictions that result from it, thus demonstrating the usefulness of our framework as a general approach to analysing complex behaviours.

We initially add the assumption that the model is able to act upon gradient information while it is moving forward but not during reversals, which we show later to be a reasonable assump-

tion. Also, it is worth underlining here that our results will be specific to this choice of model and therefore it is important that it has been shown to reproduce *C. elegans* behaviour for at least some specific choice of parameters (Miller et al., 2005; Zariwala et al., 2003).

5.1 Optimisation of the model

A general question when considering the optimisation of a model of animal behaviour is which cost function should be optimised as the animal's behaviour has likely evolved to address a number of constraints. Here, we choose to optimise for energy efficiency (Kooijman, 2000) and we relate the energy cost to the time taken by the model to navigate towards the centre of a radial gradient from a fixed starting position.

C. elegans is capable of navigating gradients in both directions; it will for example move towards food but away from noxious stimuli even though they both produce chemical gradients. In the context of our work, we use a radial gradient whose direction is defined towards the centre. Since the model used here only receives the sign of the change in gradient over time as an input, the exact nature of the increase in concentration as distance to the peak decreases has no effect on the results. Here, we thus simply define the gradient as increasing linearly in the radial direction. We test virtual worms whose behaviour is dictated by a specific set of transition probabilities for the model by setting a task which requires them to navigate this gradient from a fixed starting position towards the centre. The faster a virtual worm performs this navigation, the more efficient the choice of transition probabilities for the underlying model is.

5.1.1 Parameter space of the model

The model has three states ((F)orward run, (T)urn and (R)eversal), with two sets (one for up-gradient (u) and one for down-gradient (d) travel) of three probabilities attached to each state (Fig.

3.2). This gives a total of 18 transition probabilities. It is however possible to represent each set of three probabilities by just two values φ_i and ρ_i (with $\{\varphi_i, \rho_i\} \in [0, 1]^2$ and $i \in \{F, R, T\}$) as in the following example for up-gradient F state probabilities:

$$p_u(F|F) = \varphi_F \quad (5.1)$$

$$p_u(R|F) = (1 - \varphi_F)\rho_F \quad (5.2)$$

$$p_u(T|F) = (1 - \varphi_F)(1 - \rho_F) \quad (5.3)$$

In behavioural terms, the parameter φ_i represents the likelihood of remaining in the state i while ρ_i determines which of the remaining two states will be the most likely successor state if state i is left. The parameter space of the model can thus be represented in just 12 dimensions, which has no effect on the range of behaviours that the model can display but simplifies the optimisation process significantly. The multiplicative relationship between the parameters ensures that the sum of each set of probabilities will always be 1, irrespective of the values taken by φ_i and ρ_i .

5.1.2 Definition of the criterion for optimal performance

As was noted in Chapter 4, the recursive algorithm we use for analysing the model requires an *a priori* definition of a criterion for optimal performance. In this case, it can be obtained from the performance of the best solution found by the SA algorithm when run on the entire parameter space.

Due to the stochastic nature of the model, there will be some variability in the performance of this solution, but it is possible to use this variability to define a limiting value for what can be considered an optimal solution to the problem. The variability was determined by optimising models 50 times and noting the distribution of the performance scores (see Fig. 5.1). A Pearson

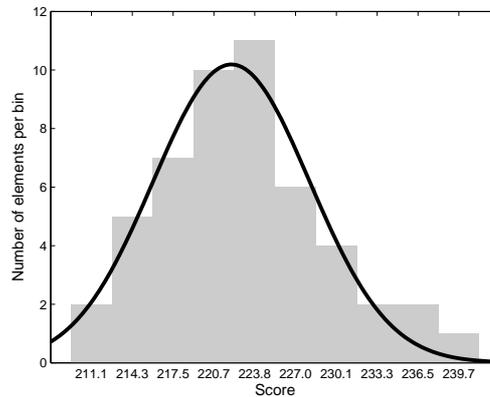


Figure 5.1: Distribution of the scores of 50 solutions found by the SA algorithm in the stochastic parameter space. Bin size is ≈ 3.1 . Mean $\mu = 222.93s$ and STD $\sigma = 6.57s$. Black line shows fitted Gaussian. The shape, mean and standard deviation of the distribution determine the definition of optimal solutions (those with a score $s \leq \mu + 3\sigma$, see text).

χ^2 test confirms that this distribution with mean $\mu = 222.93s$ and standard deviation $\sigma = 6.57s$ is normal, which (from the cumulative distribution function) indicates that 99.87% of all scores are lower than $\mu + 3\sigma$. Since the distribution represents a collection of optimal performances, we can define a model obtaining a score $s \leq \mu + 3\sigma = 242.64s$ as being optimal.

5.1.3 Model assessment

Individual simulated worms whose behaviour is determined by a specific set of transition probabilities for the model are tested in a virtual gradient. The starting position of the model is fixed at a distance of $20\sqrt{2}$ mm from the peak of the gradient and the initial orientation cycles from 0 to 360° in 5° increments for each run. A total of 72 runs are performed per assessment and the average time required to come within 0.5 mm of the peak of the gradient defines the value of the cost function at the point defined by the model's transition probabilities and thus the score of the transition probabilities.

5.1.4 Simulation of worms

All simulations of virtual worms take place in a Cartesian space with discrete time steps of 1 s. The virtual worms are only able to assess the change in gradient over time when moving forward. It is not known if this is also true for *C. elegans*, but we will show that this is likely to be a correct assumption (see *Results*). Fixed parameters have been set to realistic values where possible: travel speed has been implemented as a constant set to 0.22 mm/s (Ferrée and Lockery, 1999) and turn rates are chosen randomly from values between 0 and 50 °/s for up-gradient turns and 50 and 210 °/s otherwise, reflecting values that have been observed in real animals (Pierce-Shimomura et al., 1999). *C. elegans* also exhibits a directional bias even when the animal is moving forward (Pierce-Shimomura et al., 2005). This is implemented as a Gaussian distribution with mean 0.441 ± 2.12 °/s (Pierce-Shimomura et al., 1999). Reversals are implemented as a forward movement with a negative speed and the same directional bias.

5.1.5 Simulated annealing

A standard simulated annealing approach (Kirkpatrick et al., 1983; Cerny, 1985) is used to optimise the system. The transition probability function $P = \exp((E_s - E_n)/T)$ for $E_n > E_s$ and $P = 1$ otherwise is also standard. Here, T is the current temperature of the system and E_s and E_n are the energies of the current state and the selected neighbour respectively. The initial temperature of the system is set to $T = 15 \times 10^3$ ° and the cool-down for each time-step is defined as $T_{t+1} = \alpha T_t$ with $\alpha = 0.99$. The algorithm halts successfully before the end of the cool-down if a solution whose score is equal or better than the criterion for optimal performance is found.

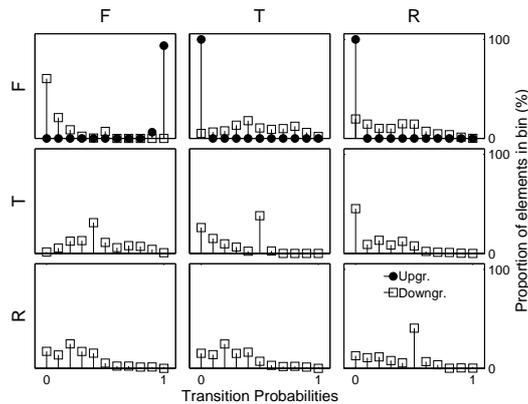


Figure 5.2: Probability distributions of solutions found by the optimisation algorithm. Each subplot indicates the distribution of the values for one transition probability across all solutions. Both up- (black circles) and down-gradient (transparent squares) distributions are shown, although up-gradient data is only meaningful for the F state (first row). Rows correspond to the active states and columns to the possible successor states. The third subplot in the first row thus displays the distribution of transition probabilities from the F state into the R state. Bin size is 0.1. Of interest are strong preferences for specific values, seen for instance for both the up-gradient and down-gradient values of $p(F|F)$ (top left).

5.2 *C. elegans* gradient navigation strategies

Dividing each dimension once using the recursive algorithm, we determined that 945 of the total 4096 hypercubes in the parameter space contained at least one optimal solution. In this section, we seek to understand what the common requirements for optimal performance are.

5.2.1 Probability distributions show run shortening during down-gradient navigation

First, we consider the distribution of values for every transition probability of the model across all identified solutions. This allows us to easily identify transition probabilities that have similar values for all solutions, thus indicating a strong preference for those values.

The distributions of the transition probabilities for the 945 solutions are shown in Fig. 5.2. The distribution for the F state is expected: 93.75% of the individuals have $p_u(F|F) = 1$, 90.26% have

$p_d(F|F) < 0.3$, thus showing the typical preference of *C. elegans* for long runs when going up-gradient and short ones when going down-gradient. However, it is interesting to note that $p_d(R|T)$ is close to 0 for most individuals (45.19% the individuals have $p_d(R|T) < 0.1$, 88.04% $p_d(R|T) < 0.5$), making a T - R sequence uncommon, which is similar to observations in real animals (Zariwala et al., 2003; Miller et al., 2005). Since $p_u(F|F) = 1$ for most solutions, all transition probabilities related to the other two states will have no effect on the behaviour when moving up-gradient and are distributed randomly across the value range.

We briefly investigate the robustness of the solutions by selecting the run-time value of a model parameter randomly from a flat distribution centred around the optimal value for that parameter and with a spread of 0.5 anew at every time step. We analyse the effect of such a fluctuation on each of the parameters in turn. Although the fluctuation is quite strong, the time the models require to navigate towards the peak when only one of the down-gradient parameters is fluctuated remains within 105% of the original performance. When all down-gradient parameters are fluctuated simultaneously, the required time increases to 112% of the original one. A fluctuation of the parameter encoding $p_u(F|F)$ however results in a complete inability of the solutions to perform well in the task. Even if the spread of the distribution is reduced to 0.1, the solutions only reach the peak of the gradient after 140% of the original time. This loss in performance however is mainly due to the fact that none of the other up-gradient parameters were actually optimised by the algorithm. If they are manually set to optimal values (*i.e.* facilitating a return to the F state), a fluctuation with spread 0.1 of $p_u(F|F)$ results in a small increase in navigation time to 113% while the original spread of 0.5 results in an increase to 150%.

We have thus shown that the model remains relatively robust even against strong fluctuations except if they affect all up-gradient parameters at the same time. If only a single parameter is fluctuated, the strongest decrease in performance is observed for $p_u(F|F)$, which is to be expected since the solutions will spend the largest amount of their travel time going up-gradient in the F state.

Rank	Node	Value Span			Mean pos.
		Low	Med	High	
1	F	14.1%	27.2%	58.7%	1 ± 0
2	$T < F > R$	55.6%	26.4%	18%	2 ± 0
3	R	2.6%	19.8%	16.4%	3.2 ± 0.44
4	$F < T > R$	2.1%	2.7%	1.2%	4.5 ± 0.84
5	T	6%	9.9%	6.8%	4.57 ± 0.78
6	$F < R > T$	0.5%	0.7%	0%	5.33 ± 1.15

Table 5.1: Summary of the ID3 tree in Fig. 5.3 emphasising the distribution of hypercubes in which no optimal solution was found by the SA algorithm over the sub-trees (low, medium and high) below each node. Mean pos. indicates the average position of a node in the tree and the standard deviation. Rank sorts nodes according to their mean position, which indicates the importance of the parameter represented by each node in gradient navigation.

fication of the parameter space: since all up-gradient chains have $p_u(F|F) \simeq 1$, it is admissible to ignore the up-gradient parameters and focus on the down-gradient ones without loss of generality. The model will simply remain in the F state when moving up-gradient and the behaviour will thus be independent of the other up-gradient parameters. Since it can be seen in Fig. 5.2 that the optimal configurations for some values are located at the division border, we re-run the recursive algorithm with every dimension divided into three, rather than two parts (with value ranges $[0, 1/3]$, $]1/3, 2/3]$ and $]2/3, 1]$ per dimension) for this analysis, as discussed in Chapter 4.

The resulting decision tree is shown in Fig. 5.3. Since it indicates all hypercubes that do not contain an optimal solution, it is possible to calculate the total percentage of the 6 dimensional space of the cost function that is occupied by such hypercubes (52.54%) and how these hypercubes are distributed over the subbranches of all nodes (squares or diamonds in Fig. 5.3) in the decision tree (see table 5.1). Table 5.1 indicates the mean position and the resulting rank of each node.

The ranking confirms that the most important feature while moving down-gradient for successful gradient navigation is the parameter determining the length of forward runs. Additionally, the distribution of hypercubes containing no optimal solution under the $p_d(F|F)$ node increases as the value increases, with over half being located in the high $]2/3, 1]$ range. This implies that longer

runs when going down-gradient are worse than shorter ones, which is expected and corresponds to observed behaviour in the real animal (Pierce-Shimomura et al., 1999).

The second important feature is the behavioural unit following the forward run, where a preference for reversals is clearly seen as 55% of all hypercubes with no optimal solution lie in a region of the parameter space which favours turns over reversals as a successor state to forward runs. This is similar to observations in *C. elegans*, which often precedes turns with a reversal (Gray et al., 2005). The length of reversals is also important and there is a clear preference for short reversals as only 2.6% of the hypercubes with no optimal solution are located in the low $[0, 1/3]$ range, in agreement with observations in the real animal (Zhao et al., 2003).

Beyond the third level, the standard deviation of the mean position increases, indicating that these nodes are scattered throughout levels 3 to 6. This makes it more difficult to assess their significance as interdependent effects are quite likely.

5.2.3 Changing the criteria for optimal performance has little impact on results

We have shown in the previous section that only the down-gradient parameters are of interest. Here, we briefly investigate how the choice of the criteria for optimal performance affects the results.

We originally defined the criterion for optimal performance as the mean plus three standard deviations of the distribution of scores obtained by running the optimisation algorithm on the entire parameter space 50 times. Here we analyse if the distributions of the probability values for the down-gradient parameters (depicted in Fig. 5.2) change substantially for different definitions of the criterion for optimal performance.

To do so, we collect three subsets from the original family of solutions: individuals whose score is less or equal than the mean (μ) plus zero, one or two standard deviations (σ) of respectively. We compute the same histograms as depicted in Fig. 5.2 for each subset and normalise for the number of individuals in each set. We then compare these histograms with the original (also normalised) one by

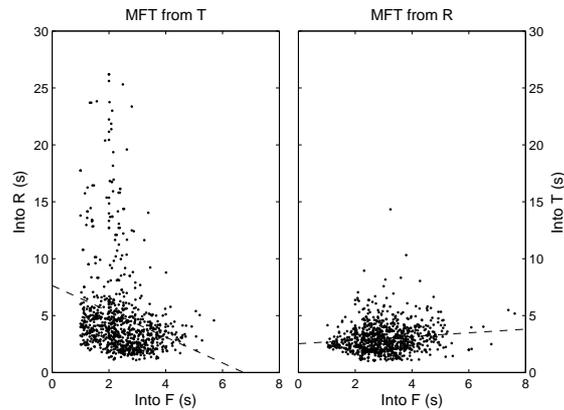


Figure 5.4: Mean first passage times from the T state into the F and R states (left) and from the R state into the F and T states (right). The preference for short MFTs into F (typically < 6 s) can easily be seen. Dashed line visualises correlation.

computing the RMSE between the original and each new one. We find a RMSE of 0.0189 between the original histogram and that of the subset with a criterion of $\mu + 2\sigma$. This RMSE increases to 0.0534 for the subset with a criterion of $\mu + \sigma$ and to 0.1029 for the one with a criterion of μ .

Since we normalised the data, the RMSE indicates the difference in change of proportion between the data in the different histograms. It can thus be seen that the difference between the histograms is less than 2% if one standard deviation is subtracted from the criterion for optimal performance, which increases to 5% and 10% if two or three standard deviations are subtracted. Thus the change in histograms remains relatively small even when the family of solutions is halved (since a criterion defined simply as the mean would remove the entire right-hand side of the curve in Fig. 5.1) and we can therefore feel confident that the results presented here remain valid for other reasonable choices of a criterion for optimal performance.

5.2.4 Markovian properties reveal the use of pirouettes

Finally, because the model used here is Markov-like, we consider the Markovian properties of all solutions found by the SA algorithm. These properties are generally useful for understanding

the behaviour of a Markov model and, as for the probability distribution analysis, strong trends and preferences for specific values across all solutions indicate requirements for optimal behaviour. It has been noted before that the type of probabilistic model used here is not Markovian in the strict sense due to the non-stationary transition probabilities (Miller et al., 2005). Nonetheless, it is possible to consider the transition probabilities for up- and down-gradient movements separately as strict Markov chains in a meaningful way as the only transition point from one chain into the other is located in the F state (the only state in which the model can receive information about the gradient which will determine whether up- or down-gradient probabilities are going to be used in the next step). Here, we call an *up-gradient chain* a sequence of states determined exclusively by up-gradient transition probabilities and *down-gradient chain* a sequence of states determined exclusively by down-gradient transition probabilities.

To confirm that the optimal solutions make use of all the states of the model, we compute their ergodicity. A Markov chain is ergodic if every state in the chain can be reached within a certain time. We found that 12 of the solutions returned by the SA algorithm have non-ergodic down-gradient chains and that in all cases, the R state is excluded from the chain. An additional 51 solutions have a very low probability of entering the R state ($p_d(R|F) + p_d(R|T) < 0.1$). 6.66% of the models under consideration therefore do not rely on the R state in their strategies, preferring an *E. coli*-like run and tumble approach (Berg and Brown, 1972). The presence of these solutions exemplifies that a bacterial strategy can in some cases perform similarly to the more intricate *C. elegans* strategy. However, the low number of such solutions in comparison with those that use reversals indicates that reversals may improve the robustness of the performance as other parameters are varied.

To determine the likely sequences of behavioural units, we next consider the mean first passage times (MFTs) of all solutions found by the optimisation algorithm. This is the mean time required by the model to reach a given state for the first time if started in a given other state. It is only possible to consider the MFT for ergodic chains. Since most solutions have absorbing up-gradient chains in

the F state ($p_u(F|F) = 1$, making it impossible to leave that state again), only the down-gradient chains (with $p_d(R|F) + p_d(R|T) \geq 0.1$) are considered. Additionally, MFTs starting from the F state cannot be taken into account as this state contains a transition point into the up-gradient chain.

Fig. 5.4 shows the MFTs for all qualifying solutions from the SA algorithm and states. It can clearly be seen that the MFT from T into F is relatively low across all solutions (mean 2.45 ± 0.87 s), similar to the MFT from R into F (mean 3.01 ± 0.96 s). Periods of reversals and turns are therefore frequently interrupted by forward runs, which will be short if the model is still heading down-gradient. The models thus tend to use a pirouette strategy (*i.e.* turn events interspersed with short runs) similar to that of *C. elegans* (Pierce-Shimomura et al., 1999).

The MFT from T into R shows a much higher variability (mean 4.87 ± 4.02 s) while the MFT from R into T remains relatively low for most individuals (mean 3.01 ± 1.26 s). A Kendall- τ test reveals a noticeable correlation (value -0.22) between the MFTs for leaving the T state but less so for leaving the R state (correlation value 0.07) (see Fig. 5.4). Since the MFT from T into R is low only if the MFT from T into F is high, runs are the clearly preferred successor behavioural unit to a turn. In other words, the models are not likely to follow turns with reversals. The lack of correlation between the MFTs leaving the R state indicates the lack of a clear preference for a successor state to R in the family of solutions, which can also be seen in the analysis of the ID3 decision tree (Fig. 5.3 and table 5.1).

The MFTs into F also give a good indication of the average length of sequences composed of reversals and turns between forward runs and confirm the preference for frequent returns into the F state already observed in the transition probability distribution (Fig. 5.2). They do not, however, give an indication about the length of the resulting pirouette itself as it is not known whether the next step in the F state will be in the up-gradient or the down-gradient chain.

5.2.5 Pirouettes may emerge in part from an inability to sample the gradient during a turn

Next, we try to pin down the cause of these pirouettes. There are two aspects of the model which could in principle give rise to them: (1) intrinsic properties of the distributions from which the turn rates are sampled, or (2) assumptions on the gradient sampling ability of the model. We have tested the effect of the second consideration by modifying the model so that it is also able to sample the gradient during turns. This modified model was then optimised for the same task and a family of solutions derived as before. First, we noticed that the new family of solutions had 1393 members (compared to the previous 945), indicating that providing gradient information during turns has facilitated the task. The threshold for optimal performance however was virtually identical to that for the original family (242.42 s versus the original 242.64 s), indicating that the additional information did not provide a large performance gain. Since this new family now has two transition states between up- and down-gradient chains (in the F as well as the T state), it is no longer possible to identify pirouettes based on Markovian properties alone. We therefore simulated each optimal solution in both the new and original families 20 times and identified the proportion of down-gradient reorientation sequences that do not contain an F state, other than those located at the beginning of the sequence. We find that on average $18 \pm 6\%$ of all down-gradient reorientation sequences in tracks produced by the original family of solutions did not involve a forward movement, whereas for the new family this value increased significantly to $64 \pm 6\%$. The number of pirouettes executed by this model is thus higher if the model is unable to evaluate the gradient during a turn. Although it is difficult to control for the exact effect of the distributions from which the turn rates are sampled, we have thus shown that they are not the only cause of pirouettes.

Comparison with biological data is difficult because the proportions of pirouettes with at least one forward movement exclusively within down-gradient reorientation sequences has not yet been identified. When all reorientation sequences are considered, the proportion is estimated at about

40% (Pierce-Shimomura et al., 1999), but since the length of forward runs is much shorter while going down-gradient, turns interspersed with small runs are more likely to happen while moving down-gradient whereas single turns are more likely to happen while moving up-gradient. The proportion of turns interspersed with runs can therefore be expected to be higher than the reported 40% when considering only down-gradient movements. It is thus possible that the animal also uses pirouettes at least in part because it may not always continuously evaluate the change in gradient during a turn, but more biological data would be needed to confirm this.

5.2.6 Performance on planar gradients

It is interesting to investigate the performance of our optimal solutions in a very different type of environment: a planar gradient. In this case, gradient concentration decreases linearly with the distance to the x axis, but is independent of the specific horizontal position in the Cartesian plane. Since the peak of the gradient is now essentially a line, we expect that models will take less time to navigate towards it than in radial environments. Indeed, when a family of optimal solutions is derived in a planar environment, it contains 1881 members compared to 945 for the planar environment and their performance when started at the same distance from the peak is lower (121.79 ± 15.64 s) than that of models in a radial environment (189.44 ± 19.89 s).

We then simulated the family originally optimised for radial environments in the planar one and find that there is no significant difference ($p > 0.3$) in performance with those optimised for the planar environment (mean time to the peak 121.40 ± 11.69 s). The converse is not true, however - models optimised for planar environments perform significantly worse ($p < 0.001$) in a radial environment (mean time to peak 202.4 ± 29.59 s versus 189.44 ± 19.89 s).

We have thus shown that our models also perform close to optimal levels in planar gradients for which they were not originally optimised and that a radial environment has been a good initial choice.

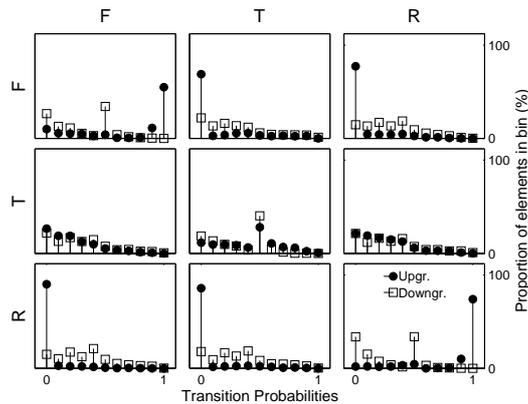


Figure 5.5: Probability distribution of the transition probabilities of optimal solutions assuming the model is able to process gradient information during a reversal. Bin size is 0.1. Compared with Fig. 5.2, a smaller proportion of the models have $p_u(F|F) \approx 1$ and more models have $p_u(R|R) \approx 1$. The functional role of reversals in these models can thus be similar to that of forward runs.

5.2.7 Models naturally dwell at the peak

Although our models have been optimised solely for efficient navigation of the gradient, we find that they naturally dwell near the peak once it is reached. After the models reached the peak, they were typically found to stay within 0.5 ± 0.39 mm of it. Although our models can not slow down and become stationary like real animals, it is interesting to observe here that the dwelling at the peak need not explicitly be represented in the cost function and may be an emergent property of the gradient navigation even in the real animal

5.2.8 Gradient information during reversals leads to unnatural behaviour

It is not known whether *C. elegans* is capable of acting upon gradient information while it is reversing, but reported state transition probabilities (Miller et al., 2005) suggest that it does not. This can be examined by enabling sampling of the gradient while reversing in the model used previously and optimising it using our recursive approach to collect a set of optimal solutions given the new conditions. If this is done, 1782 hypercubes are found to contain solutions capable of optimal

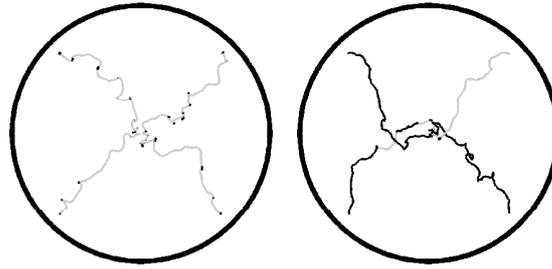


Figure 5.6: Four randomly chosen example tracks of models with reversal gradient information enabled (right) and disabled (left). Runs are indicated in grey, reversals in black. It can be easily seen that the right model's use of reversals is abnormal.

performance and the decision tree generated by the ID3 algorithm (not shown) indicates that the performance of the solutions is unsatisfactory when $p_u(F|F)$ and $p_u(R|R)$ are both less than 0.5. These results indicate that optimised models will use the F and R states as functionally equivalent units.

Fig. 5.5 shows that a large number of the solutions now found do not possess the strong preference for forward runs when moving up-gradient observed in real animals (Pierce-Shimomura et al., 1999) and our previous set of optimised models (Fig. 5.2). Similarly, a large percentage of the new solutions use $p_u(R|R) = 1$ (Fig. 5.5), effectively navigating towards the centre of the gradient while moving backwards, which has also never been observed in real animals. Example tracks of worms modeled with those probabilities (Fig. 5.6) underline the abnormal use of reversals.

When comparing the performance of 2000 runs from this new family of solutions with the original family, we found that being able to act on gradient information during reversals gives the models a small but significant ($p < 0.001$) advantage (mean time to navigate to the peak: 184.75 ± 19.65 s versus 189 ± 19.9 s). Interestingly however, when the equivalent families optimised for planar gradients are compared, no significant difference in performance is observed (122.1 ± 15.62 s versus 121.79 ± 15.64 s, $p > 0.7$).

We have thus shown that the ability to act upon gradient information during reversals can in

some cases give the models a slight advantage; it is therefore interesting to observe that the animal does not appear to do so. However, this may be due simply to mechanical reasons: since the animal reverses over its own tracks, it is possible that gradient information has been disrupted within those tracks. Alternatively, since most sensory neurons are located in the head of the animal, gradient information may simply be obscured by the animal's body and thus unavailable during a reversal. Given that the gain from such information is minimal, this is unlikely to put the animal at a severe disadvantage.

5.3 The family of solutions can be used for investigations of novel situations

An interesting test for any computational model is its evaluation in a novel task for which it was not originally designed. In this section, we first show that the family of solutions previously derived perform similarly to real animals in a situation they were not originally designed for: step changes in the concentration of the otherwise uniform chemical environment. The behaviour of *C. elegans* in such a situation has been previously studied by Miller et al. (2005) and we base our present analysis on their results.

We then use the family of solutions to investigate two interesting questions: 1) What are the effects of the responses to downsteps and upsteps in gradient concentrations displayed by *C. elegans*? This has been touched upon but not fully investigated by Miller et al. (2005). 2) The same study artificially deleted either the downstep or the upstep response in computational models of real animals and shows that models with a deleted downstep response still prefer the regions of higher concentration in the quadrant assay (Fig. 5.7). This seems at odds with the pirouette strategy (Pierce-Shimomura et al., 1999): how can a model which cannot react to entering a lower concentration (*i.e.* does not display pirouettes) outperform a model which is still able to avoid the worse

concentration (using pirouettes)? It is therefore interesting to see whether we can reproduce this behaviour with our family of solutions or find the behaviour one would expect from the pirouette strategy.

Although it would be preferable to use our family of models without modification, some adjustments are unavoidable. First, the models possess two sets of transition probabilities which regulate behaviour depending on whether the concentration of the chemical environment increases or decreases. However, since the chemical concentration in the present experiments is mostly invariant over time, a situation that has been virtually impossible in the task for which the models were optimised, they need to be extended with a set of transition probabilities for this situation. Since there is little point in optimising a model for a situation in which there is no clear goal based on the inputs available to the model, we derive this set of transition probabilities from a previous experimental study of *C. elegans* (Zariwala et al., 2003); they are thus biologically realistic.

Second, it is important to consider not just the chemical stimulus itself but also how this translates into neural activity. *C. elegans* neurons are thought to signal mainly using calcium dynamics, which have a slow timecourse. No intracellular studies of the calcium dynamics in chemosensory neurons exist so far, but they have been studied in AFD, the main thermosensory neurons in *C. elegans* (Clark et al., 2006). In those experiments, it was found that intracellular calcium increases (decreases) if a temperature upstep (downstep) is experienced and then slowly recovers to pre-stimulus levels (over a timecourse of ≈ 20 s for upsteps and > 80 s for downsteps) unless another temperature step is experienced. In the previous simulations of our optimised models, a period in which no gradient changes are experienced was impossible and therefore the behaviour of the models was consistent with the dynamics even though those were not explicitly implemented into the models. Here however, situations in which there is no change in the environment are possible and the corresponding neural dynamics therefore need to be added to the models. For simplicity, we base these on the upstep responses of the AFD neurons: information that there has been a change in

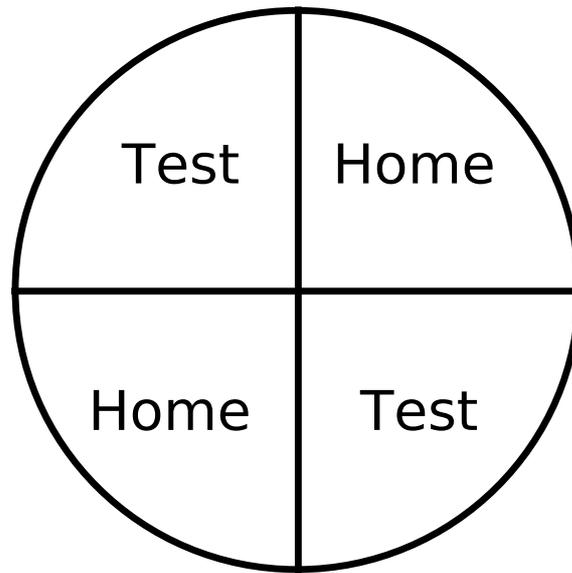


Figure 5.7: The quadrant assay setup. Quadrants denoted by 'Home' contain an environment with a reference concentration whereas the quadrants denoted 'Test' contain either an up or a downstep from these concentrations.

the chemical environment will persist in the model for 20 s unless a change in the opposite direction is noticed.

5.3.1 Quadrant assays

Miller et al. (2005) separate the plate on which the animals move into quadrants (see Fig. 5.7) with the concentration in the quadrants such that two quadrants contain environments the animal is used to (home quadrants) and the other two contain either upsteps or downsteps from the concentration in the home quadrants (test quadrants). It is shown that animals prefer to stay in the home quadrant when the other quadrants are a downstep but prefer the other quadrants if the concentration increases in them. The quantitative measure for this is simply based on the number of worms in each type of quadrant:

$$I = \frac{N_H - N_S}{N_H + N_S} \quad (5.4)$$

where N_H is the number in the home quadrants and N_S is the number in the step response test quadrants.

Similarly to Miller et al. (2005), we find that the models prefer the home quadrants if their concentration is higher than that of the test quadrant ($I = 0.65 \pm 1.3$) and the test quadrant otherwise ($I = -0.68 \pm 0.07$). The models somewhat outperform real animals in this test but this is mainly due to the fact that the models by design do not take the amplitude of the gradient step into account, so quantitative matches cannot be expected in this assay. Nonetheless, the qualitative match between the models behaviour and that of the animals is clearly seen.

5.3.2 Responses to step changes in concentration

Miller et al. (2005) expose animals to a uniform change (upstep or downstep) in gradient at a single point in time while recording their ethograms. These ethograms are converted into 4 functions of probability versus time by calculating the amount of time spent by each animal in one of four states over 10 s bins of the ethogram. The final data corresponds to the mean and standard deviation of these functions over all animals versus time. The main states correspond to forward runs, turns and reversals; a fourth state is used for behaviour that is not clearly defined but the probability of that state remains near 0 for the entire time.

When both the upstep and downstep experiments are replicated with a subset of our family of models, we observe similar responses to those reported by Miller et al. (2005) for real animals. An upstep results in a transient increase of the probability of finding models in the forward state with corresponding decreases of the probabilities for both other states (Fig. 5.8). A reduction in the standard deviation is also observed. Conversely, a downstep results in a decrease in forward probability with a corresponding increase in the standard deviation of the turn probability and an increase in the reversal probability (Fig. 5.9). Miller et al. (2005) only report forward state probability states for the downstep situation, so it is not possible to judge the realism of the changes in the reversal and

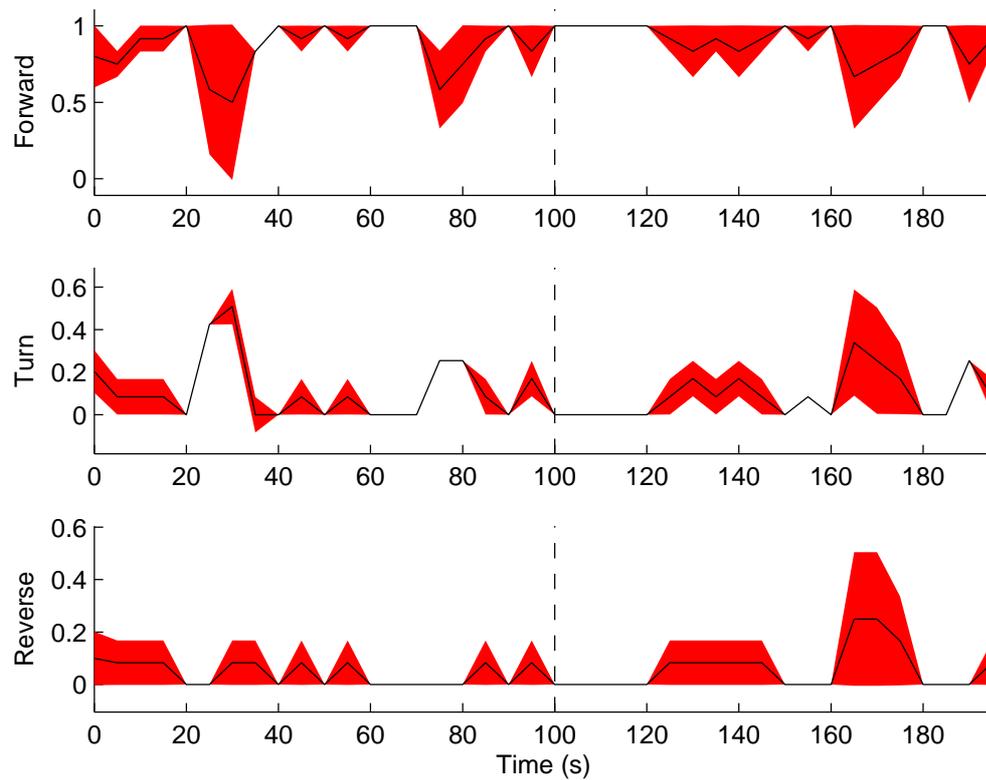


Figure 5.8: Evolution of the Probability of finding models in a given state versus time in the case of an upstep (indicated by the dashed line). Black line is mean probability for all models, red area indicates standard deviation. A clear response can be seen after the upstep as the probability of finding models in the forward state increases to 1 (with a STD of 0 for the corresponding time)

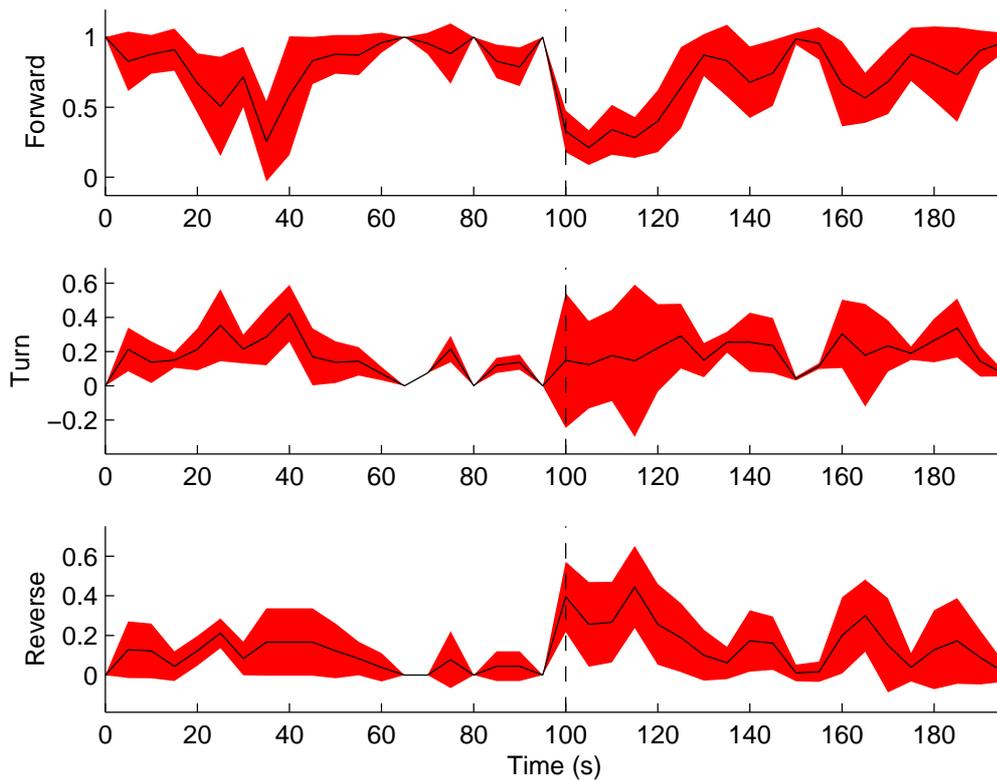


Figure 5.9: Evolution of the Probability of finding models in a given state versus time in the case of an downstep (indicated by the dashed line). Black line is mean probability for all models, red area indicates standard deviation. Again, a clear response can be seen for the downstep, as (1) a decrease in the mean probability forward runs, (2) an increase in the standard deviation of the probability of turns and (3) an increase in the mean probability of reversals.

Type of model	Transition type		Sig. of difference
	High - Low - High	Low - High - Low	
Fully functional	11.62 ± 1.54 s	34.37 ± 5.15 s	$p < 0.001$
Upstep disabled	12.90 ± 0.91 s	18.45 ± 1.8 s	$p < 0.001$
Downstep disabled	64.72 ± 35.0 s	84.44 ± 49.79 s	$p > 0.51$

Table 5.2: Summary of the times elapsed between a transition from a quadrant with the higher (lower) concentration into the other type of quadrant and the return to the higher (lower) concentration. The effects of disabling either upstep or downstep responses can be seen and compared to the performance of a fully functional model. It is immediately apparent that disabling the downstep response has a much more critical impact than disabling the upstep. Also shown is the p value from a non-parametric Kruskal-Wallis test on the difference between the two crossing times for each type of model.

turn probabilities, but the change in forward probability matches well.

5.3.3 Virtual mutants in quadrant assays

Miller et al. (2005) create virtual mutants based on the ethograms collected from the real animals and artificially remove either their ability to respond to downsteps or to upsteps in gradient concentration. When these virtual mutants are tested in the same quadrant assays, their performance is lower, indicating that both downstep and upstep responses play a role in chemotaxis.

When the downstep response is disabled in our models, a much more extreme failure of chemotaxis is seen in our models than in the models by Miller et al. (2005) ($I = -2.7 \pm 0.72$ if the test concentration is an upstep, $I = 0.03 \pm 0.57$ if it is a downstep), losing all ability to perform well in this test. When the upstep response is disabled, however, our models retain the ability to show the behaviour of the animals ($I = -2.6 \pm 0.07$ if the test concentration is a upstep, $I = 0.29 \pm 0.14$ if it is a downstep) but their performance is reduced compared to the assay with all responses enabled (as evidenced by a non parametric Kruskal-Wallis test on the chemotaxis index from each repeat of the experiment: $df = 1$, $p < 0.001$ if the step to the test concentration is a downstep, $p < 0.001$ if the step to the test concentration is an upstep).

Deleting upstep responses from our model thus replicates the effect found by Miller et al. (2005) whereas deleting downstep responses exceeds it. The finding that both responses make quantitatively detectable contributions to the performance in the quadrant assay thus holds true for our models, but we can additionally conclude from our test that the downstep response is more important than the upstep response; in other words, that pirouettes remain an important feature for our models in this test.

It is also interesting to analyse the strategy in this quadrant assay by looking at the times between the crossings of the borders of the quadrants (summarised in Tab. 5.2). If fully functional models cross from the higher concentration region into the lower concentration region, the mean time until they return to the higher concentration region is 11.62 ± 1.54 s, whereas if they cross from the lower region into the higher region, the mean time until they return to the lower region is 34.37 ± 5.15 s. There is a statistical significance between the two crossing times ($p < 0.001$).

If the response to an upstep is eliminated, the mean time spent in the lower region after crossing is 12.90 ± 0.91 s whereas the mean time spent in the higher region after crossing is 18.45 ± 1.8 . There is a strong difference between the two cross times for these mutants ($p < 0.001$) and a weaker difference between the time spent in the lower region by these mutants and that of the fully functional models ($0.001 < p < 0.01$).

Finally, if the response to a downstep is eliminated, the mean time spent in the lower region after crossing is 64.72 ± 35.0 s whereas the mean time spent in the higher region after crossing is 84.44 ± 49.79 s. There are obviously strong differences with all of the performances both by the fully functional models and those that have had an the upstep response removed ($p < 0.001$ in all cases). There is no significant difference, however, between these two crossing times ($p > 0.51$).

Therefore, the main effect of the upgradient response is to extend the time spent in the higher concentration compared to the time spent in the lower concentration. If it is removed, time in the higher concentration reduces significantly and there is a weak effect on the time spent in the lower

concentration, but overall, the existence of the downgradient response is enough to ensure more time is spent in the higher concentration than in the lower one.

If the downgradient response is removed, however, there is no longer a statistically detectable difference between the times spent in the higher and lower concentrations respectively and the time in the lower concentration increases significantly compared to both the fully functional models and those without an upgradient response. Without the downgradient response thus, it becomes impossible to differentiate between the two levels of concentration and therefore to keep the time spent in the lower concentration as low as possible, which again illustrates the importance of pirouettes in this assay.

5.4 Summary of Chapter 5

5.4.1 Components of *C. elegans* gradient navigation

Based on a simple probabilistic model connecting different behavioural units of *C. elegans*, we have been able to predict that *C. elegans* may not act on gradient information while reversing because optimised models that could do so displayed behaviour that did not agree well with that found in the real animal. We have further demonstrated that the animal might at least in part use the pirouette reorientation strategy due to an inability to sense or act upon gradient navigation while in the process of turning. The random walk strategy may thus be a consequence of an inability to use alternative, deterministic strategies due to insufficient or insufficiently accurate input at the sensory level. This is further supported by recent studies which show that *C. elegans* uses a deterministic strategy rather than a random walk for navigating electrical fields (Gabel et al., 2007) towards the negative pole. It has been suggested that the different sensory modalities may converge upon a common neural sub-circuit for navigation (Zariwala et al., 2003). If this is the case, our results and those of Gabel et al. (2007) suggest that this sub-circuit may in principle, depending on the nature

of the sensory input, be capable of more deterministic navigation strategies than the random walk usually observed. This point will be investigated in more detail in Chapter 6.

Additionally, we have shown that to be successful at gradient navigation, long forward runs when going up-gradient and short forward runs when moving down-gradient are necessary. In this, the optimised models reproduce the behaviour observed in real animals (Pierce-Shimomura et al., 1999). Our family of optimal solutions also demonstrates that reversals were not strictly required for successful gradient navigation, but in the models that did use them, a strong preference for using them at the end of a forward run rather than a turn was found, which is similar to the behaviour observed in *C. elegans* (Gray et al., 2005). Additionally, we found that $R - T$ sequences were more likely than $T - R$ ones, implying that there is a higher likelihood of following a reversal with a turn than vice versa. This feature has also been observed in real animals (Zariwala et al., 2003).

For other parameters, no strong requirements for successful gradient navigation in the conditions of our simulation were found. In particular, the lack of a strong effect of $p_d(T|T)$ indicates that the distribution from which the amplitude of a single turn is sampled (which is determined by the experimentally fixed distribution of the turn rate and the time spent in the T state) is not important, with reorientation manoeuvres consequently relying on a series of turns and runs. The optimised models thus demonstrate *pirouettes* similar to those found in *C. elegans* (Pierce-Shimomura et al., 1999).

In general, we find a close agreement between the behaviour of our models that have been optimised to minimise travel time towards the centre of a gradient and the observed behaviour of *C. elegans*. This is a strong indication that the main aim of the gradient navigation behaviour exhibited by the animal is to navigate it as efficiently as possible given the available behavioural units.

The only notable discrepancy we find is that the optimised models will not usually initiate a turn while going up-gradient, whereas real animals can (Pierce-Shimomura et al., 1999). The behaviour of the optimised models here is easily explained as they have been optimised for efficient gradient

navigation alone and in such a strategy, initiating turns while moving in a favourable direction can clearly not be optimal. *C. elegans* behaviour however may exhibit multiple strategies aimed at achieving different and sometimes conflicting goals simultaneously. The animal can for instance react to changes in oxygen levels (Cheung et al., 2005) and is sensitive to touch (Chalfie et al., 1985), which may induce behavioural patterns influenced by the surface on which it is moving. The model used here (Fig. 3.2) thus displays a pure gradient navigation strategy whereas the behaviour of the nematode, even in experimental conditions, may be more complex. This illustrates how optimisation techniques can be used to accurately describe a strategy with a single goal using the behavioural units available to the animal even though the observed behaviour might encode multiple simultaneously active strategies.

5.4.2 Evaluating computational models in novel tasks

We have shown that our models, in spite of having been neither designed for nor optimised in step response situations can reproduce the behaviour of *C. elegans* in such situations reasonably well. To achieve this, only minimal biologically realistic modifications were necessary which extended the models by giving them the ability to recognise situations in which no change in concentration is experienced.

We have also been able to demonstrate the effects of the upgradient and downgradient responses on the performance in the quadrant assay. In particular, we have shown that in our models, deleting the downstep response results in a failure to perform well in the quadrant assay, which does not reconcile well with the results by Miller et al. (2005), but fits well with the predictions of the pirouette strategy (Pierce-Shimomura et al., 1999).

5.4.3 On the computational analysis of behaviour

Overall, none of the conclusions above would be easy to arrive at from experimental observation alone, mainly due to the inaccessibility of the parameters involved. We have thus illustrated how computational techniques for optimising behavioural models and analysing the results can complement and extend the understanding of a behaviour that has been observed experimentally. Although we have focused on *C. elegans* gradient navigation here, our framework is however general and would remain useful for studying behaviours of other animals.

Chapter 6

Deterministic isotherm tracking of *C. elegans*

CHAPTER 5 has dealt exclusively with the stochastic navigation strategies employed by *C. elegans* for navigating different types of gradients towards a region of preference. In the case of chemotaxis, this is the only observed strategy. If the animal is placed in a thermal gradient however, two strikingly different strategies can be observed: if the animal is near its preferred temperature, it will sometimes track isothermal lines in the environment with remarkable precision, not deviating by more than 0.1 °C (Ryu and Samuel, 2002) and on average for about 35 s (Luo et al., 2007, initially erroneously reported as 80 s by Luo et al. (2006)). If the ambient temperature is higher than the preferred one however, the animal will move towards lower temperatures using again a directed random walk (Ryu and Samuel, 2002) similar to the one observed for chemotaxis, characterised in Chapter 5.

In the present chapter, we are interested in the isotherm tracking behaviour of the animal. While the behaviour has been known for several decades (Luo et al., 2006), it has received relatively little attention in the literature (see Chapter 2). To date, only one computational model of the strategy the animal might employ to achieve this isotherm tracking has been proposed (Luo et al., 2006). We

are not convinced by this model however, as will be shown later in the chapter, and therefore we propose a different one here, based on likely sensory information (which can be collected in a single headsweep) about the environment.

Even though we do not have detailed information about the computational capabilities of the *C. elegans* neural circuit (as discussed in Chapter 2), we are then able to formulate the general computations required for our proposed isotherm tracking strategy and, using the resulting (necessary) requirements on the connectivity between some neurons, we can search the neural circuit of *C. elegans* for subcircuits meeting these requirements.

Finally, we have some interest in the fact that the animal seems to navigate towards a temperature of preference stochastically but will track isotherms deterministically. This has been observed multiple times in the same experimental setups (*e.g.* Mori and Ohshima, 1995; Ryu and Samuel, 2002) so it is unlikely that the use of the stochastic strategy is due to differences in experimental setup. It is also unlikely to be due to noisy systems or inaccuracies at the sensory or motor levels since the animal is capable of precisely tracking isotherms in the same environment. An interesting question therefore is whether or not the animal would, at least in theory, have sufficient information about the thermal environment available to navigate towards its preferred temperature deterministically or whether the stochastic strategy is used simply because the sensory information does not permit other approaches.

We can investigate the available sensory information when navigating towards a preferred temperature in parallel to our determining the information available for isotherm tracking since both investigations are very similar. In fact, one can define the optimal travel direction for both isotherm tracking and navigation towards a preferred temperature in relation to the isotherms themselves: for isotherm tracking, it is (obviously) *parallel* to the isotherms while it is *perpendicular* to the isotherms and pointing towards colder regions¹ when the animal is travelling towards its preferred

¹This follows from the general observation that the animal only appears to move towards its preferred temperature when currently in warmer regions (Ryu and Samuel, 2002), see Chapter 2

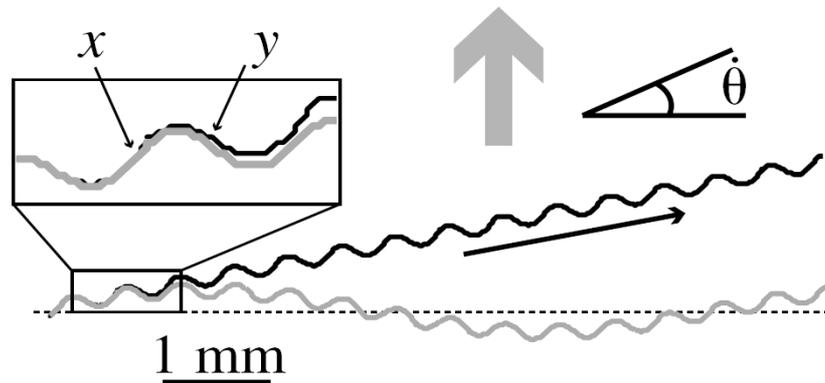


Figure 6.1: Figure 6a from Luo et al. (2006), illustrating the behaviour of their isotherm tracking model. The model is travelling at an angle θ to the isotherm. Shown are tracks when the course correction of the model is disabled (black) in which case the model keeps moving at the same angle and enabled (grey) in which case the model curves more vigorously at point x than point y (shown in magnification) since the change in gradient is higher at x . This results in a course correction eventually aligning the model onto the isotherm.

temperature. First however, this chapter will discuss the isotherm tracking model of Luo et al. (2006) in more detail.

6.1 Issues with the existing model of isotherm tracking

6.1.1 Definition of the existing model

C. elegans moves forward by producing undulating sine-like movements, alternatively contracting ventral and dorsal muscles. These contractions originate at the head segment of the animal and then travel along the remainder of the body. Luo et al. (2006) propose a model of isotherm tracking which essentially contends that the animal continuously adapts the intensity which it curves its head in function of the perceived change in temperature. The idea is that this will allow the animal to balance warming and cooling phases encountered during the lower and upper part of the headsweep since the head movements can only be identical during those parts if the animal is moving directly along an isotherm. According to their model, if the animal were to travel at a slight angle to an

isotherm, it would curve more vigorously during one part of the head sweep than the other, causing a slight turn aligning the animal more with an isotherm (Fig. 6.1).

Mathematically, Luo et al. (2006) implement their model as follows: the direction in which the head is travelling at any given time is given by its angle $\theta(t)$ relative to a fixed line so that $\theta = \pi/2$ if the animal is moving perpendicularly away from the gradient. This angle is given by an oscillator whose acceleration is dependent on the current change in gradient $T'(t)$. Thus:

$$\theta''(t) = \theta_0 \omega^2 \sin(\omega t) (1 + f(T'(t))) \quad (6.1)$$

where $\theta_0 = \pi/4$ is the amplitude of the headsweep, $\omega = \pi$ is the frequency of the headsweep and $f(\cdot)$ is a function determining in what manner $T'(t)$ affects the acceleration. Luo et al. (2006) note that the minimal requirement is that $f(x) = f(|x|)$ (which ensures that the response to a cooling phase is the same as that to a warming phase of identical amplitude) and use $f(x) = gx^2$ for most of their discussion while acknowledging that other possible forms exist. Here, g is a free parameter of the model representing a gain. The change in gradient over time, finally, is simply given by the direction of the animal's head $\theta(t)$, its current speed v and the steepness of the gradient ∇T :

$$T'(t) = |v| |\nabla T| \sin(\theta(t)) \quad (6.2)$$

It should be noted that we have given a simplified version of the gradient used by Luo et al. (2006) reduced to a spatial time-invariant gradient. Luo et al. (2006) also superimpose a time-varying pulsating gradient over the spatial one given here in Eqn. 6.2 in some of their experiments (which is included as a second term in Eqn. 6.2). They use this second gradient to show that the animal continuously determines the direction of the isotherm. To illustrate some of the issues with this model, however, it is sufficient to consider the behaviour of the model within the time-invariant gradient.

6.1.2 Behaviour of the model

Numerical simulations of the model as defined in section 6.1.1 show that it is possible to find a value for the free parameter g so that the model can correct the direction it which it travels towards a direction parallel to gradient lines (Luo et al., 2006). We are interested here in the balancing strategy the model employs in this course correction. It relies on an asymmetry between the gradient changes encountered during the upper and the lower half of the headsweep, which is indeed the case for most directions which are not parallel to the gradient lines. However, one would also expect the gradient changes during both parts of the headsweep to be symmetrical if the model were to travel directly perpendicularly to the isotherm (since the perpendicular is an axis of symmetry in both planar and radial gradients).

Here we therefore investigate the course correction of the model defined by Eqns. 6.1 and 6.2 as a function of the initial travel angle α relative to the gradient lines. We make one change, which does not affect the overall behaviour, to the model: we replace the sine-based oscillator for the headsweeps with a cosine-based one. The reason for this is simply the fact that it will simplify the specification of initial conditions for solving Eqns. 6.1 and 6.2 numerically, as the speed of the oscillator $\theta'(0)$ will be 0 regardless of the acceleration component in 6.1 if the oscillator is cosine-based. If it is sine-based, as in the original model, $\theta'(0)$ will depend on the acceleration component and the relationship is not clear. The initial condition for $\theta(0)$ is similar in both cases, it merely depends on α . For a cosine-based oscillator, this is given by $\theta(0) = \theta_0 + \alpha$, whereas it would be $\theta(0) = \alpha$ for a sine-based one. The modified Eqn.6.1 thus reads:

$$\theta''(t) = -\theta_0 \omega^2 \cos(\omega t) (1 + f(T'(t))) \quad (6.3)$$

Since the gradient as defined by Eqn. 6.2 is planar and of linear steepness, the gradient change is

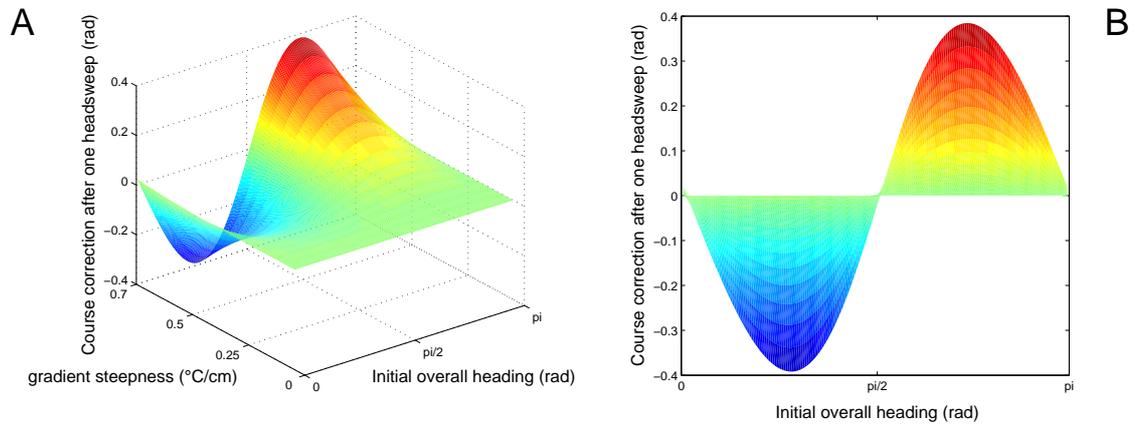


Figure 6.2: The course correction due to the gradient-dependent term in Eqn. 6.1 as a function of the initial heading and the steepness of the gradient. (A) Three dimensional view of the the function. (B) The same function viewed along the (x, z) plane. The model does not apply course correction if the initial orientation $\alpha \in \{0, \pi/2, \pi\}$.

independent of the location of the animal, it merely depends on its direction. For our purposes, we therefore investigate the course correction implemented by the model during one headsweep. If no correction is performed during the first headsweep, then the same will hold true for any following headsweeps because of the nature of the gradient and the model’s heading will never change. It is worth underlining that we are not interested here to see if the course correction helps alignment of the isotherm (which is regulated by the free parameter g), we are merely trying to determine for which initial travel directions α a course correction exists. This allows us to ignore the free parameter g (which we do by simply setting it to 200, the “realistic” value determined by Luo et al. (2006)), which will affect the amplitude of a course correction but not its existence (as long as $g \neq 0$ of course).

To compute the course correction, we first numerically derive the shape of $\theta(t)$ in function of the initial heading α and the steepness of the gradient ∇T (restricted to values between 0 and 0.7 as in (Luo et al., 2006)) over one headsweep and then calculate the difference between the two end values. The course correction thus computed is shown in Fig. 6.2. We find that, regardless of the

steepness of the gradient, there is never a course correction if the model is travelling either along the isotherm ($\alpha \in \{0, \pi\}$) or roughly perpendicular to it ($\alpha \approx \pi/2$).

If the model is not moving exactly on one of the trajectories to which no course correction is applied, the course correction will tend to reorient it towards the parallel rather than the perpendicular to the isotherm (Fig. 6.2). However, this behaviour can be changed into a reorientation towards the perpendicular simply by multiplying for instance the free parameter g by -1 . In *C. elegans* such a switch could easily be mediated by the AFD neuron, whose synaptic activity increases when the animal is in a region in which isotherm tracking occurs (Samuel et al., 2003).

This is therefore significant behaviour; when the model is travelling perpendicularly to the gradient lines it is in fact navigating the gradient in a direction that could directly take it to a region of preferred temperature. If *C. elegans* thus were to perform isotherm tracking as proposed by the model, balancing cooling and warming periods during the headsweep, it could thus use the exact same deterministic strategy for navigating gradients towards a region of interest. However, it has been shown repeatedly that navigation, when the optimal direction lies into the gradient, is stochastic even in thermal environments (Ryu and Samuel, 2002; Zariwala et al., 2003; Ito et al., 2006). We find it very unlikely that the animal will always use a different stochastic strategy for navigating towards a region of interest if the same goal could be reached more efficiently by using the strategy for navigating along gradient lines without modification. There is thus sufficient reason for asking if isotherm tracking could be performed by an alternative strategy which acknowledges the fact that isotherm tracking is the only deterministic gradient navigation strategy observed. We will see in the remainder of this chapter that such a strategy can indeed be proposed.

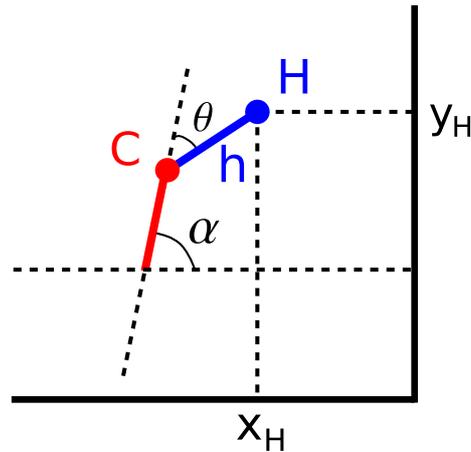


Figure 6.3: A simple diagram illustrating Eqn. 6.5. Shown are the head segment h , attached to the centroid of the model at point C and ending in the tip of the head H as well as the general travel direction α and the angle between the head and the travel direction θ . The coordinates of the head (x_H and y_H) are also given.

6.2 Modelling the animal

In the remainder of this chapter, we model the animal's movement in a slightly different way from Luo et al. (2006). Mainly, we define the direction of the head θ not with respect to a fixed line in space but with respect to the centroid of the model's body, which in turn is travelling at an angle α to the horizontal (Fig. 6.3). This addresses a shortcoming in the definition of the model by Luo et al. (2006), which does not explicitly take into account that the fixed line against which θ is defined is only fixed in a linear gradient but not in a radial one (since the fixed line itself is defined as being parallel to the gradient lines).

First, it is important to understand how a gradient may be perceived by the animal, which requires us to be able to define the location of the animal in the gradient at any point in time. We therefore model *C. elegans*, as a simple system moving in a Cartesian space. This system consists of a point C , moving at a constant speed v and a head segment of length h attached to this point C whose angle θ with respect to the travel direction α of the point C is given by a simple harmonic

oscillator following Luo et al. (2006):

$$\theta(t) = \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right) \quad (6.4)$$

In this system, the end point H of the head segment represents the tip of the head where the sensory neurons are located. It is thus the position of this point in space that we are interested in, and its coordinates can be given as a function of time t :

$$\begin{aligned} x_H(t) &= vt \cos \alpha + h \cos\left(\alpha + \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right)\right) \\ y_H(t) &= vt \sin \alpha + h \sin\left(\alpha + \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right)\right) \end{aligned} \quad (6.5)$$

Since we are interested in the different navigational strategies primarily in graded environments, we define such a gradient within the Cartesian space. For simplicity, we use a simple planar gradient with linear steepness here. This is acceptable in this case since we are only interested in how the animal will perceive the environment during a single head sweep and given its small size and slow speed, most natural environments can be approximated with a linear planar gradient within the region available to the animal during one headsweep. It should be noted that this simplification only applies to the mathematical analysis in this chapter; all numerical simulations use radial gradients. We initially define the orientation of the gradient in space so that gradient lines (*i.e.* isotherms) are parallel to the x -axis. The strength of the gradient g apparent to the model at time t is therefore simply given by the $y_H(t)$:

$$\begin{aligned} g(t) &= y_H(t) \\ &= vt \sin \alpha + h \sin\left(\alpha + \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right)\right) \end{aligned} \quad (6.6)$$

In radial gradients however, the gradient lines will not necessarily be parallel to the x axis as assumed in Eqn 6.6. Rather, they will be parallel to the tangent to the circle with centre O (the peak of the gradient) at the current location of the model. When approximating a radial gradient at a given time t , the gradient information can be derived by rotating the initial planar gradient so that the isotherm lines become parallel with the tangent (see Fig. 6.4). The gradient information thus depends on both α and the angle γ between the tangent to the gradient at the model's current location and the x -axis

The equation of the tangent to a circle centred on the origin at the point (x_0, y_0) is given by:

$$x_0x + y_0y - (x_0^2 + y_0^2) = 0 \quad (6.7)$$

A directional vector of the tangent is thus $\vec{t} (y_0/|y_0|, -x_0/|x_0|)$ and γ is then simply given in function of the scalar product between \vec{t} and a directional vector of the x -axis $\vec{i} (1, 0)$:

$$\gamma = \frac{x_0}{|x_0|} \cos^{-1} \left(\frac{y_0}{|y_0| \sqrt{1 + \left(\frac{x_0}{y_0}\right)^2}} \right) \quad (6.8)$$

where the signs of x_0 and y_0 are used to (1) ensure the correct orientation of the directional vector of the tangent and (2) to make γ directional (Fig. 6.4). For the special cases of $x_0 = 0$ or $y_0 = 0$, γ is simply $\pm\pi/2$ or 0 radians respectively and γ is obviously not defined at the origin. The gradient information g_{rad} available to the model during one headsweep in a radial gradient is thus approximated by substituting Eqn. 6.8 into Eqn. 6.6:

$$g_{rad}(t) = vt \sin(\alpha + \gamma) + h \sin\left(\alpha + \gamma + \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right)\right) \quad (6.9)$$

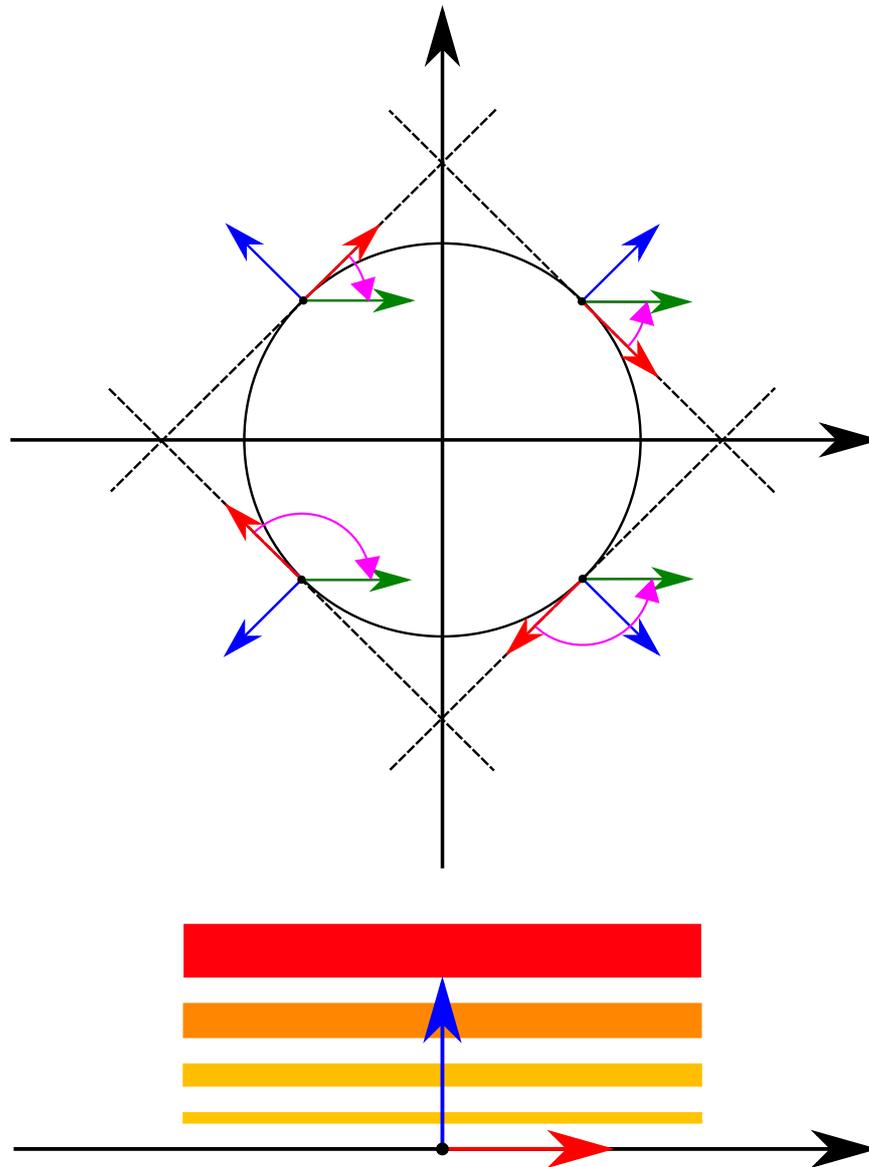


Figure 6.4: Diagram detailing how a radial gradient (top) can be approximated by planar gradients (bottom). An example is shown for each quadrant. In each case, blue, red and green arrows indicate the direction of the gradient, the directional vector of the tangent (dotted lines) and the directional vector of the x -axis respectively. Magenta arcs indicate the angle γ between the tangents and the x -axis. It is important to notice how the orientation of the tangent's directional vectors changes in each quadrant in order to keep the desired orientation relative to the gradient. Similarly, the direction of the angle between the tangents and the x -axis depends on which half of the plane the tangent is located in. The approximated shape of the gradient at any point M in the plane is then given by rotating the planar gradient (bottom) using γ so that the gradient lines become parallel to the tangent to the circle centred on the origin at M .

6.3 Available sensory information during one headsweep

Based on the model defined above, we can now investigate (1) how the animal may perceive the gradient depending on its direction of travel relative to the gradient and (2) what information the animal could thus, at least theoretically, derive about its direction relative to the gradient based on this perception. It is worthwhile to repeat at this point that there are only two principal directions of interest for us in this investigation: parallel to the isotherm (for isothermal tracking) and perpendicularly to it (for navigating towards a preferred temperature). In non-planar gradients, the perpendicular can be defined in relation to the tangent to the isotherm as opposed to the isotherm itself.

If the animal is travelling in parallel to the isotherms, we already know that its perception of the environment contains sufficient information to navigate deterministically along those lines; the question of interest is thus not whether this information exists but what form it takes and what computations are required to extract it from the perceived information about the environment. If the animal's desired direction is perpendicular to the gradient however, it is not yet fully known what kind of information it can extract from its perception of the environment. So far, it has simply been shown that it is sufficient for stochastic gradient navigation (Clark et al., 2007). Here, we take this one step further and ask if it would also in principle be sufficient for a deterministic gradient navigation strategy that does not depend on some continuous balancing of the input.

There are some previous results on the sensory capabilities of *C. elegans* that we need to take into account at this point. First, it is known that the animal can compute the change in gradient over time, dg/dt in both chemical (Dunn et al., 2004) and thermal (Clark et al., 2007) environments. During stochastic gradient navigation, the sign of dg/dt alone is sufficient for successfully reaching the region of interest in the environment (Ryu and Samuel, 2002; Miller et al., 2005, see Chapter 5). Second, it has also been shown that intracellular calcium dynamics in the thermosensory neuron

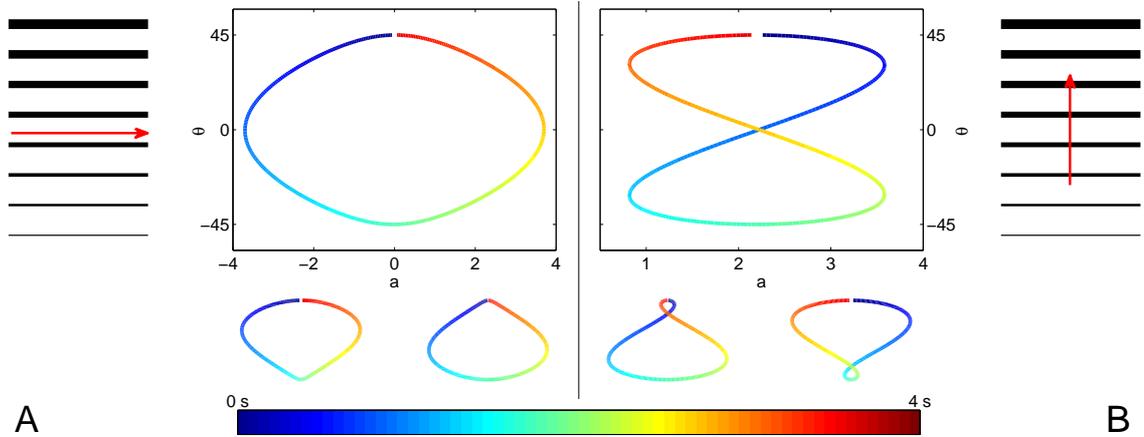


Figure 6.5: The evolution of the amplitude information a as perceived by the model during one headsweep, plotted against the angle θ between the head and the direction of travel. Colour represents time. **(A)** Large figure shows the amplitude information if the model is moving exactly parallel to the gradient (red arrow), two smaller figures show the same information when the model is travelling at an angle of $-\pi/6$ and $\pi/6$ radians respectively to the red arrow. **(B)** Same information when the model is travelling perpendicular to the gradient (large figure) or deviating from the perpendicular by $-\pi/6$ or $\pi/6$ radians (small figures).

AFD very accurately reflect changes in outside temperature (Clark et al., 2006) in thermal environments warmer than the preferred temperature. Since we base the following analysis on the observed behaviour in thermal environments, we therefore use the amplitude of dg/dt as a starting point when modelling sensory information.

We thus first give an expression for the evolution of this amplitude information a (available at the head of the modelled animal) over time. This is simply given by the time derivative of the gradient (Eqn. 6.6):

$$a(t) = v \sin \alpha - \frac{h\pi^2}{8} \cos \left(\alpha + \frac{\pi}{4} \cos \frac{\pi t}{2} \right) \sin \frac{\pi t}{2} \quad (6.10)$$

Fig. 6.5 shows how the model perceives the environment during a single head sweep, both when heading exactly into the optimal direction for a given behaviour (large figures) and when travelling

at an angle (small figures) based on Eqn. 6.10. Of particular initial interest is the fact that there appears to be a point at which the amplitude during both the downsweep and the upsweep of the head is the same if the travel direction is almost perpendicular (Fig. 6.5B). If the travel is more parallel to the gradient lines, this point disappears (Fig. 6.5A).

Does the head angle at which this crossover point is encountered convey any information about the optimal direction if the animal attempts to navigate perpendicularly to the gradient lines? Assuming that it is encountered at time t_c during the downsweep, the head will be at the same angle again at time $4 - t_c$ during the upsweep. With this consideration, we can use Eqn. 6.10 to determine t_c as a function of the travel direction α by solving:

$$v \sin \alpha - \frac{h\pi^2}{8} \cos \left(\alpha + \frac{\pi}{4} \cos \frac{\pi t_c}{2} \right) \sin \left(\frac{\pi t_c}{2} \right) = v \sin \alpha - \frac{h\pi^2}{8} \cos \left(\alpha + \frac{\pi}{4} \cos \frac{\pi(4-t_c)}{2} \right) \sin \left(\frac{\pi(4-t_c)}{2} \right) \quad (6.11)$$

where we impose $0 < t_c < 2$ to restrict the potentially infinite number of solutions to those encountered within the first downsweep. The solution is then given by:

$$t_c = \frac{2}{\pi} \cos^{-1} \left(\frac{4}{\pi} \left(\frac{\pi}{2} - \alpha \right) \right) \quad (6.12)$$

which imposes the condition $\frac{\pi}{4} \leq \alpha \leq \frac{3\pi}{4}$ on α , confirming what can be guessed from Fig. 6.5, namely that the crossover point only appears if the model is travelling with less than a $\pi/4$ radians difference to a perpendicular line to the gradient. Substituting Eqn. 6.12 in Eqn. 6.4, we see that

the angle θ_c at time t_c is given by:

$$\theta_c = \frac{\pi}{2} - \alpha \quad (6.13)$$

The model thus encounters the crossover point when the angle between its head and the travel direction is precisely the difference between the actual travel direction and a travel direction perpendicular to the gradient. It is worth noting that this result is independent of both the speed v and the length of the head segment h in this model.

It is possible to find this information in a different way, which only requires sampling of the gradient during one half of a headsweep. For this, we consider the rate of change in gradient strength during the headsweep. Fig. 6.5 suggests that head angle at which the point of fastest descent is encountered is correlated with the deviation from the optimal course. This point occurs when the derivative of Eqn. 6.10 is minimal. This derivative is given by:

$$\frac{da}{dt} = \frac{h\pi^3}{64} \left[4 \cos\left(\frac{\pi t}{2}\right) \cos\left(\alpha + \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right)\right) + \pi \sin^2\left(\frac{\pi t}{2}\right) \sin\left(\alpha + \frac{\pi}{4} \cos\left(\frac{\pi t}{2}\right)\right) \right] \quad (6.14)$$

but we have not been able to find the minima analytically. Therefore, we investigate the hypothesis numerically and compute the head angle θ_m at which Eqn. 6.14 is minimal for values of α between $\pi/4$ and $3\pi/4$ radians in $\pi/180$ increments. We find that, although there is no 1 : 1 relationship between required course correction and θ_m this time, the two are still heavily correlated and the relationship can be approximated linearly by (see also Fig. 6.6):

$$\theta_m = -\frac{\alpha - 1.8 \cdot 10^{-3}}{2.7908} \quad (6.15)$$

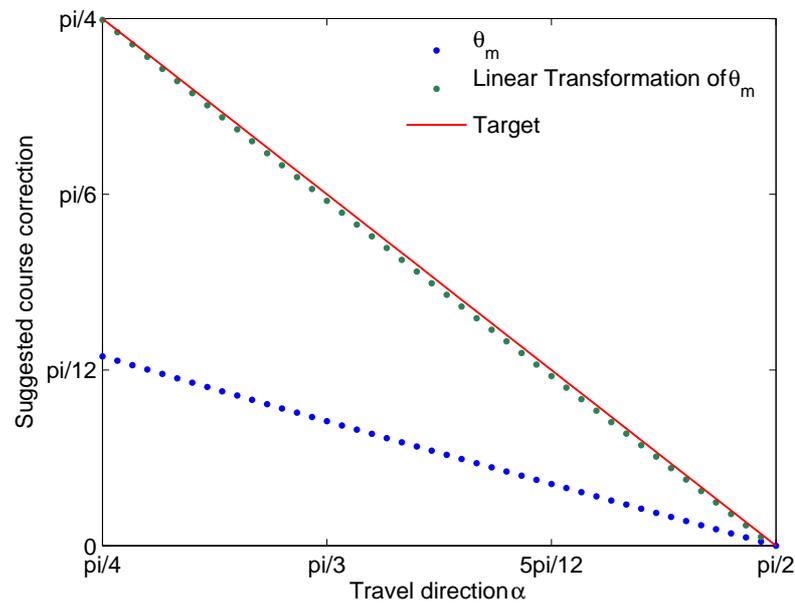


Figure 6.6: The relationship between the minima of Eqn. 6.14 (blue dots) and the required course correction for alignment onto the optimal direction (red line). It is possible to find a linear transformation approximately mapping the minima onto the required course correction (green dots, see Eqn. 6.15).

The RMSE of this fit is $\approx 2.3 \times 10^{-3}$ radians, indicating that the fit is not perfect but likely to be sufficient for the present purposes. This is thus an alternative strategy whose advantage is that it only requires information from one half of a headsweep at the cost of precision.

We have therefore shown that at least two strategies for travelling perpendicularly to the isotherms exist based on information obtained during one headsweep. Combined with the result by Clark et al. (2007), showing that the amplitude information can be used to determine whether one is generally moving up or down the gradient, it has thus been proven that the information available to *C. elegans* does contain sufficient information for a deterministic gradient navigation strategy towards a temperature of preference.

Similarly, we can consider the amplitude information in the case of isotherm tracking and ask if the amplitude information might allow a deterministic navigation strategy by providing information

on the degree of off-courseness, similar to what has just been proven for perpendicular gradient navigation. As can be seen in Fig. 6.5A, the time at which the amplitude information reaches a maximum or minimum seems again to be correlated with the travel angle; it is thus a likely candidate for providing this information. In principle, one would verify this by finding the roots of the derivative given by Eqn. 6.14, but we have not been able to find the roots analytically. Therefore, we again investigate the hypothesis numerically and compute the head angle θ_m at which Eqn. 6.14 equals 0 for values of α between $-\pi/4$ and $\pi/4$ radians in $\pi/180$ increments. We again find that the relationship between the required course correction and θ_m can be approximated linearly (with a RMSE of $\approx 3.3 \times 10^{-3}$ radians) by:

$$\theta_m = -\frac{\alpha - 5.7 \cdot 10^{-3}}{2.7109} \quad (6.16)$$

Next we investigate a model implementing this strategy that can perform isothermal tracking up to or exceeding the performance of *C. elegans*. While tracking isotherms, the animal does not deviate from them by more than 0.1°C and follows them for about 35 s on average (Luo et al., 2007). We can therefore test this model by determining at which point it will deviate from its starting gradient value by more than 0.1°C . If we use the fit as given by Eqn. 6.16, the model can only track example radial gradients for 4 s (gradient radius 1 cm) and 7 s (gradient radius 7 cm). All gradients have a linear steepness of $1^\circ\text{C}/\text{mm}$. If we optimise the constants in Eqn. 6.16 for longest isotherm tracking in a gradient with radius 7cm however, the performance improves to and close to 2 min for a radius of 1cm and 1 hour for a radius of 7cm, where 1 hour is also the time limit imposed on the simulations. The new approximation is given by:

$$\theta_m = -\frac{\alpha + 0.8 \cdot 10^{-3}}{5.1703} \quad (6.17)$$

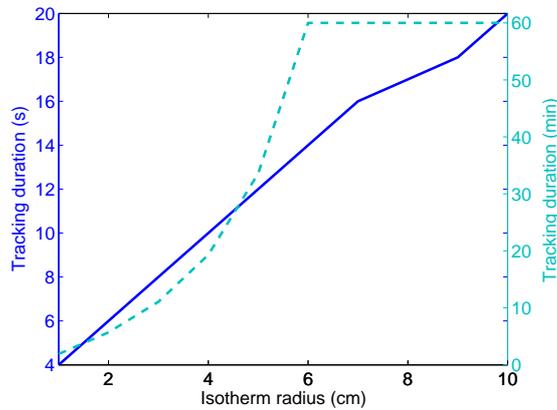


Figure 6.7: Performance of a model using the head angle θ_m at which $da/dt = 0$ as sole guidance in an isothermal tracking test for radial isotherms of different sizes. Performance is defined as the time until the model deviates by more than $0.1\text{ }^\circ\text{C}$ from the isotherm. Gradient steepness is $1\text{ }^\circ\text{C}/\text{mm}$. Solid dark blue line shows performance (left y-axis) using the initial linear approximation of the relationship between θ_m and the error in direction α given by Eqn. 6.16 while dashed light blue line shows performance (right y-axis) when the constants in Eqn. 6.16 have been optimised for longest isotherm tracking (Eqn. 6.17). Dashed line plateaus at 1 hour, the maximum duration in these simulations.

and Fig. 6.7 shows the performance of both the original and the new fit in radial gradients varying in radius from 1cm to 10cm.

Therefore, although the error caused by the initial approximation is too significant for an acceptable performance, there nonetheless exists a linear relationship between θ_m and α that can be used to match *C. elegans* in isotherm tracking performance. It is worthwhile to note that the fit presented in Eqn. 6.17 is not the only one which will produce acceptable performance and merely serves to illustrate that linear transformations between θ_m and α which enable the model to perform isotherm tracking at acceptable levels exist. Additionally, the gradients used in these experiments were extremely steep ($1\text{ }^\circ\text{C}/\text{mm}$), illustrating that the model can perform well even in extreme conditions. Its performance can be expected to improve in more realistic gradients (*e.g.* $1\text{ }^\circ\text{C}/\text{cm}$). Even though the exact relationship between θ_m and α is not clear, we have thus shown that even a

rough linear transformation of θ_m is sufficient for isotherm tracking.

It has thus been demonstrated that, based on the information available to the animal, a simple strategy relying only on determining the maximum or minimum of the amplitude information during a headsweep can be sufficient for isotherm tracking behaviour. Such a strategy is both simpler than that proposed by Luo et al. (2006) (since no continuous adaption of the motor output is required) and is more consistent with the overall behaviour of *C. elegans* (since it cannot be used for navigating perpendicularly to the gradient).

Finally, we observe that the performance of the model improves with the radius of the isotherms (Fig. 6.7) which is a novel prediction on the animal behaviour. Unfortunately, no experimental data with which this prediction could be verified exists at the moment; we therefore leave the verification for a later date.

6.4 A candidate neural circuit for isotherm tracking

We have argued in Chapter 2 that too little is known about the detailed neural dynamics of *C. elegans* to realistically expect to be able to model their computations in a meaningful manner. Nonetheless, we can consider some of the requirements from our isotherm tracking model and how they translate into requirements on the *C. elegans* neural circuitry as well as compare to what is already known about the general behaviour of some neurons.

6.4.1 Restrictions imposed by our model

Our strategy has two functionally distinct parts: (1) computing the second derivative of the sensory information and (2) correcting the current course in function of the head angle relative to the body at the time where the computed second derivative was 0. Assuming that this course correction is computed by a single neuron class CORR (see Fig. 6.8), we can then specify some

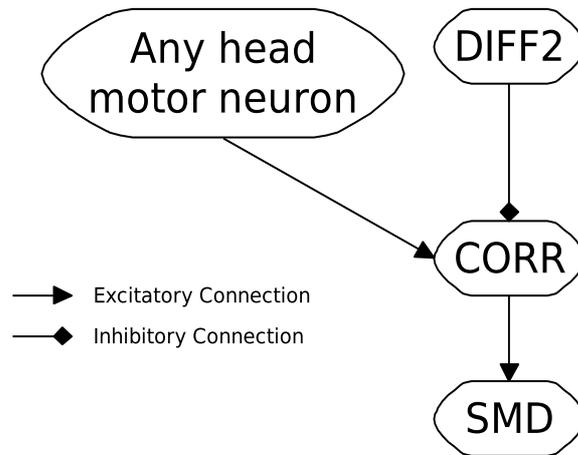


Figure 6.8: Candidate neural circuit for isotherm tracking. CORR computes course correction using input from any of the head motor neurons and the second derivative of the sensory information DIFF2 and communicates this correction to the SMD motor neuron.

requirements on the connectivity of that neuron and thus identify candidates for this function within the neural circuit of *C. elegans*.

First, CORR needs to be postsynaptic to whichever neuron computes the second derivative. Since it is hard to make any strong statements about how *C. elegans* might compute derivatives without detailed knowledge of its neurocomputational power, we will not use this from the outset to constrain the choice of candidates for CORR (but see below for some speculation).

Second, our strategy assumes that the angle of the head relative to the centroid of the body is available to that neuron. This angle could in principle be derived from the current state of the muscles and their degree of contraction/relaxation and it should therefore also be able to predict it from motor neuron activity. While it is in principle possible that any motor neuron could provide this information, we believe it is most likely to come from one of the head motor neurons, (1) because their activity is most likely to be directly correlated with the angle and (2) because the signal path transmitting this information to the interneurons in nerve ring of the animal would then be shortest. We therefore require CORR to be postsynaptic to any of the head neurons SIA, SIB, SMB, SMD,

Neuron	Known properties	Likely candidate
OLL*	Sensory neuron	No
RIA	Involved in thermotaxis	With Reservations
RIC*		Yes
PLN	Strongly associated with PLM neurons	No
SAA*	May be part of locomotory circuit	Yes
AVK*		Yes
RIB	May be involved in thermotaxis	No
ADA		Yes
AIB	May be required for executing omega turns	Unlikely
RIS		Yes

Table 6.1: List of the candidate neurons for the CORR neuron assumed to compute the course correction during isothermal tracking in our model. Neurons marked with an asterisk (*) are only candidates if no assumption on the computation of the second derivative is made.

RMD, RME, RIM or RIV (Gray et al., 2005).

Third, CORR has to be able to communicate the required course correction to a neuron which can actually implement it. Gray et al. (2005) and Gabel et al. (2007) list the SMD motor neurons as a likely candidate for controlling this kind of course corrections; in particular Gabel et al. (2007) hypothesise that it is responsible for carrying out the required course corrections during the navigation of electrical fields. We therefore require CORR to be presynaptic to the SMD neurons.

6.4.2 Candidate neurons for computing course correction

Using the restrictions above, we searched the connectivity of the *C. elegans* neural circuit (White et al., 1986; Oshio et al., 2003) for candidate neurons. The resulting list is shown in Tab. 6.1. Some of these neurons are disqualified as candidates because of prior known functions: OLL is a sensory neuron (Tsalik et al., 2003) and as such an unlikely candidate for CORR. The function of PLN is not known, but morphologically it seems to be closely associated with the PLM class (Gray et al., 2004) and since that neuron is not present in our list, we do not believe that PLN is a likely candidate for CORR. RIB may be involved in thermotaxis, but its ablation does not affect

isotherm tracking (Mori and Ohshima, 1995), it can therefore not be one of the main neurons in a theoretical isotherm tracking circuit. Finally, AIB is thought to mediate omega turns (Gray et al., 2005), which makes it unlikely as a candidate for CORR since omega turns are generally of large but stochastically chosen amplitude (Pierce-Shimomura et al., 1999).

One particularly interesting neuron in Tab. 6.1 is the RIA neuron. It has been shown to be involved in thermotaxis and its ablation leads to a complete loss of isotherm tracking abilities in the animal (Mori and Ohshima, 1995), which would make it an ideal candidate for CORR. However, its ablation also renders the animal cryophilic meaning it will no longer navigate to its preferred temperature but rather to the coldest one it can find. This behaviour is not explained through a mere loss of isotherm tracking abilities alone, yet our hypothesised neuron CORR has no other functions in thermal environments that we can predict. RIA is thus a possible candidate for CORR, but only with reservations as it is likely that it performs other functions in thermal navigation as well and these still need to be determined and their compatibility with CORR ascertained. The remaining neurons are all equally likely candidates, mainly because nothing is known about their specific function and therefore nothing contradicts the hypothesis that they represent CORR *a priori*.

6.4.3 On computing the second derivative

As said previously, it is not currently possible to know exactly how derivatives can be computed in *C. elegans*. Dunn et al. (2004) for instance assume a network of 3 neurons A B and C is required, where A is presynaptic to B and C while B is only presynaptic to C. The theory is that a signal from A to C has a fast route (the direct one) and a slow route (via B); C can therefore use these timeshifted signals to compute the derivative. However, we note that *C. elegans* neurons usually have several tens of connections between themselves (White et al., 1986), whose detailed properties are not known. If any of those have different lengths or otherwise transmit signals at a different speed, a time-shifted signal for the computation of a derivative would also be available without the

need for a third neuron. Finally, it is important to realise that when *C. elegans* neurons are discussed, they are usually referred to by class (*e.g.* AFD). A class can however contain 2 to 6 neurons (AFDL and AFDR in the case of the AFD class) and while it is usually valid to group them together because they seem to fulfil their main functional role as a class rather than individual neurons, it is entirely possible that a single class could compute a derivative using time-shifted signals between its member neurons.

It is thus clear that we cannot make strong predictions on where the second derivative is computed if we do not know exactly how it is computed. If we assume however, that one class of neurons can compute a derivative and that AFD computes the first derivative (Clark et al., 2007), then the candidate neuron DIFF2 would have to be postsynaptic to AFD and presynaptic to any of the neurons in Tab. 6.1. Searching the connectivity of the *C. elegans* neural circuit again, we find that this gives us 5 candidate neurons: AIY, ASE, AIB, AIN, and AWA. Additionally, the list of candidates for CORR is reduced, leaving only ADA, RIA and RIS of the initial neurons that we considered likely candidates.

Of these candidates for DIFF2, ASE and AWA are chemosensory neurons, which makes them unlikely to be part in the thermosensory circuit, further underlined by the fact that their ablation has little effect on thermosensation (Mori and Ohshima, 1995). The other neurons are all very general interneurons with several roles each, which makes it impossible to confirm or deny their plausibility as a difference engine. Since our isotherm strategy proposes that relevant head angle for the course correction is encountered when the second derivative of the sensory information is 0, it is possible that DIFF2 is an inhibitory neuron, only releasing CORR when its time-shifted inputs cancel each other out, but it is not known for any of those candidate DIFF2 neurons whether they are excitatory, or inhibitory.

6.4.4 Plausibility of the isotherm tracking strategy

Given the assumption that the course correction used in our isotherm tracking strategy is computed by one neuron, we have been able to formulate some necessary requirements for the connectivity of that neuron and with that, we were able to determine that only 5 of the 302 neurons in *C. elegans* fulfil these requirements and do not have other known conflicting functions. This provides a good starting point for laser ablation studies seeking to verify the existence of this course correction behaviour, but it is of course possible that the course correction is in reality computed elsewhere. In particular, we have assumed direct connections to any of the head motoneurons and the SMD motoneuron. In reality, there may be intermediate neurons within these signalling paths, but we have no way of predicting that. What we have shown, however, is that our proposed isotherm tracking strategy is plausible on a neural connectivity level even under the perhaps most restrictive requirement² of direct connections to all neurons providing or requiring information.

6.5 On the use of stochastic strategies when navigating towards the peak of gradients

We have shown previously in this chapter that it is quite likely for the animal to have sufficient information for a theoretical deterministic strategy towards its preferred temperature based on its known computations and the intracellular calcium activity of the AFD neuron (Clark et al., 2006, 2007). This indicates that it is not using the stochastic navigation simply as a result of an inability to do otherwise.

It also makes it more difficult to pin down the use of the stochastic strategy over a deterministic one on any single cause. For instance, it may fulfil an utility that is not immediately apparent through the observation of behaviour. It is also possible that a circuit for stochastic navigation is necessary

²In the sense that allowing indirect signalling paths would only increase the number of different possible paths.

due to insufficient sensory resolution for chemotaxis and that the same circuit is used (as previously suggested by *e.g.* Zariwala et al., 2003) for navigation towards the preferred temperature, perhaps simply to minimise neural requirements.

It is thus clear, that we can make no strong claims about the use of stochastic over apparently available deterministic strategies. We can however analyse the computational requirements of the the two theoretical deterministic strategies we have identified earlier and compare them to what is known about the computational capabilities of *C. elegans* to assess whether they may be fundamentally impossible to compute for the animal.

We have presented two deterministic strategies for navigating perpendicularly to the gradient lines. The first one, the “cross-over point” strategy is interesting because it can actually exactly determine any required course correction less or equal than $\pi/4$ radians in either direction. Since it relies on comparing gradient information from one half of a headsweep to that from the other half, this strategy requires at a minimum a memory system capable of functioning as a FILO stack. There is however no evidence yet to indicate that such a memory system exists in *C. elegans*. In general, the memory systems that are known in the animal integrate stimuli over time. The main example is the preference of the animal for a given temperature, initially based on the temperature at which it has been cultivated. In this case, it has been shown that this memory is encoded by the main thermosensory neuron AFD itself (Samuel et al., 2003) and that it evolves over time based on the recent thermal history (Biron et al., 2006). Such a system cannot operate as a FILO stack, however and there is no evidence supporting the existence of more complex memory systems yet.

The second, “steepest descent” strategy requires locating the minimum of the second derivative of the gradient information during one head-sweep. However, it is not immediately possible for a neural circuit to locate this minimum independently of the overall gradient strength at the current location using only the second derivative. The computation of the third derivative would thus be required as well as the location of its roots and a comparison with the sign of the second derivative

to ensure that the root indicate the location of steepest descent and not steepest ascent. It is known that *C. elegans* is capable of computing derivatives (Dunn et al., 2004; Clark et al., 2007); whether this extends to the level of third derivatives is unknown but not in principle impossible. We can thus not reject the possibility that *C. elegans* is capable of the computations required for a deterministic navigation towards its preferred temperature. We do note that it seems unlikely that the animal will compute the third derivative of the sensory information but have no definite evidence to support this.

We also note that the simplest computational strategy we could find for deterministic navigation towards a preferred temperature (computation of the third derivative of the sensory information) is more complex than our proposed isotherm tracking strategy (computation of the second derivative only), which may be another factor in the use of a stochastic strategy for navigating towards a preferred temperature and of a deterministic one when following an isotherm.

Overall, however the question of why *C. elegans* prefers to use stochastic navigation strategies over deterministic ones when navigating towards its preferred temperature remains open. Here, we have simply given some insights into required computations for theoretical deterministic strategies that may motivate future research.

6.6 Summary of Chapter 6

This chapter has investigated the deterministic isotherm tracking strategy used by *C. elegans*. We have first shown that we had sufficient doubts about the only existing computational model of this behaviour to warrant investigating whether a different model for this behaviour could be proposed. We next formulated a new model of isotherm tracking which computes the required course corrections from the second derivative of the sensory information. We verified that it performs up to the standards set by *C. elegans* and formulated a candidate neural circuit which could encode this strategy. Based on this circuit, we have tentatively identified a small list of neurons in *C. elegans*

which could be implicated in isotherm tracking.

Second, we have briefly verified that it is in theory likely for the animal to receive sufficient sensory information about the thermal environment for deterministically navigating towards a preferred temperature. We have also briefly discussed the computational requirements of two strategies we have identified to this effect. Although we are unable to draw any strong conclusions on the animal's use of stochastic strategies over deterministic ones in this case, we have shown that it is unlikely to be simply due to insufficient available information and hope that this brief analysis motivates further research.

Part IV

Conclusions

THE main objective of this thesis was to further the understanding of the behaviour of the nematode *C. elegans* using mainly computational approaches. Additionally, we aimed at keeping any methodology developed during this work sufficiently general so it may find applications in other behavioural studies not necessarily concerned with *C. elegans*.

The main contributions of the present work to existing fields of research consequently fall into two distinct categories: (1) novel insights and results into the behaviour and neural capabilities of *C. elegans* and (2) theoretical and technical innovations for analysing computational models of behaviour. Here, we briefly review and discuss the achievements of the present thesis for those two categories in turn.

7.1 Innovations in the analysis of models of behaviour

7.1.1 Ways of determining the performance of Markov-like models

We first considered the best approach to analysing the behaviour of models that are Markov-like but not strictly Markovian since their transition probabilities are variable. Since the variable transition probabilities prevent the analysis of their Markovian properties, the obvious approach is to study these models through numerical simulations. It was however interesting to investigate whether this is always necessary or whether it is possible to transform these models in such a way that the analytical Markovian tools, in particular the computation of the mean time to absorption and the probability of absorption into certain states, become applicable.

We have found that such a transformation is indeed possible and have formulated the approach, illustrated with examples for two Markov-like models. An important restriction we found was that it is necessary to include the goal states of the behaviour in the strict model resulting from this transformation. This is needed so it becomes possible to calculate for instance the mean time a model takes to reach the goal state or the probability that it will reach it analytically. Since this goal state is normally a physical location in the world (in the examples we have considered, they were either the source of a chemical plume or the peak of a gradient), it becomes necessary to transform the Markov-like model of behaviour into a strict Markov model in such a way that the states of the strict model correspond to physical locations the animal whose behaviour is represented by the Markov-like model can reach. This necessity finally restricts the computational model of the real world to one which is both finite and discrete. Additionally, since the number of states in the strict model is essentially a function of the number of locations within the modelled world that the behavioural model can occupy, the strict model may become computationally intractable if the modelled world, even if finite, is too large.

The transformation from Markov-like models into strict models we have presented here, while

theoretically possible, is thus of limited use since it may not always be reasonable to accept the necessary restrictions on the computational model of the world. In particular, we find that the transformation is too restricted to be useful in our analysis of *C. elegans* behaviour and we have therefore studied the Markov-like model of this behaviour using numerical simulations.

Nonetheless the insight that Markov-like model with variable transition probabilities can be transformed into strict Markov models if required is interesting and useful in problems where the restriction to simplified models is acceptable. We have given such an example by considering the navigation of a moth towards the source of a chemical plume. Since both the behaviour of the moth and the environment which influences this behaviour are random, analysing a corresponding model purely through numerical simulations would be very time-consuming since a significant amount of repeat measure would be necessary in order to quantify the statistical variation between trials in a useful way. Using our transformation, however, it has become possible not only to quantify the behaviour of the model analytically but to do so in a reasonable time-frame.

Finally, we note that the major obstacle for this model is due to computationally limited resources as the restrictions on the modelled world are correlated with the number of states in a Markov model that can be held in memory and processed in a reasonable time by computers. It is thus entirely possible that the attractiveness of applying the approach we have presented here to problems which are more complex than the example one considered in this thesis will only increase as technology progresses.

7.1.2 Using Markov-like models to analyse behaviour

Next, we introduced a comprehensive framework for analysing computational models of behaviour with the intention of deriving novel insights into this behaviour. The fundamental motivation was that such insights might be gained from the model when its parameters are tuned so that the overall behaviour of the model becomes optimal for some criterion other than realistic modelling of

the animal's behaviour. The behaviour of the optimised model would then, in comparison with the behaviour of the animal's behaviour, yield insights into the optimality of the strategy used by the animal. Additionally, we took into account the fact that there may well be multiple optimal strategies with which stochastic models in particular can achieve a certain behaviour. We have therefore devised an algorithm which attempts to systematically find a family of optimal configurations for these models.

We then proposed that the distribution of these configurations over the parameter space of the model can provide interesting information which is not readily available from mere observation of a behaviour itself. Indeed, we expected some parameters of the model to be more critical for achieving a certain behaviour than others, which would be reflected in the distribution of the optimal solutions. In ethological terms, it therefore becomes possible to understand the relative importance of the different behavioural units (which are derived from observation and originally inspired the computational model) in achieving a certain goal. Another consequence of the existence of multiple strategies for achieving a given aim is that different individual animals might rely on different strategies. Determining the different strategies as made possible using the present framework therefore gives a more general understanding of the behaviour that is able to account for individual differences.

In order to analyse the distributions of these configurations in a meaningful way, we draw upon decision solving techniques from AI which are in turn inspired by information theory. This analysis completes the definition of the framework which is generally applicable to the study of most models with multiple optimal solutions assuming the analysis remains computationally tractable. We have thus provided a powerful tool for the analysis of behaviour which represents a significant contribution to the available methods in ethological studies.

Additionally to its value for the field of ethology, we have also demonstrated as a side note that the subdivision of the parameter space of a model can be generally useful in optimisation tasks

when the corresponding cost function is highly uneven and featuring many local minima. In such cases, the approach can both increase the probability of finding the global minimum (at the cost of requiring more time) or decrease the time required to reach a target probability that the global minimum has been found. We have not considered the applications of this approach further since it falls outside the scope of our intended work. It seems reasonable to assume, however, that it will be of value in some investigations.

7.2 Contributions to the field of *C. elegans* research

7.2.1 Analysis and characterisation of the stochastic gradient navigation strategy

Using our framework for analysing behaviour, we have been able to quantify, for the first time, the necessary and sufficient interactions of the different behavioural units identified in *C. elegans* for efficient navigation towards the peak of gradients. Some of these were rather expected - we found for instance that runs should be long when moving in a favourable direction and short otherwise. Other findings were less obvious, for instance that reversals should not be too long, that the exact amplitude of a turn is not very important or that it is more preferable to follow reversals with turns than the converse. Overall though, the behaviour of our optimised models captured the actual behaviour of *C. elegans* rather well and we have therefore been able to explain subtle behavioural sequences in the animal behaviour using considerations of efficiency.

Additionally, we have been able to show that pirouettes as used by *C. elegans* may specifically emerge from an inability to sample the gradient during a turn and that the animal may also be unable to act upon gradient information while reversing. These insights are especially interesting since they illustrate nicely how computational models can be used to predict the sensory capabilities of animals.

In summary, we have thus extended the knowledge about the stochastic navigation strategy

from a mere observation of behavioural units to a detailed understanding of the relative importance of these units and why they appear in the sequences that are observed. As such, the work presented here is likely to be the most comprehensive study of *C. elegans* gradient navigation since the fundamental study of Pierce-Shimomura et al. (1999). While therefore of interest to the *C. elegans* research community, these results also illustrate the usefulness of our computational framework for analysing behaviour and have been published in a high-level, interdisciplinary peer-refereed journal (Thill and Pearce, 2007).

It is perhaps interesting to relate our work to other studies of stochastic navigation behaviours. Random-walk based gradient navigation, particularly as exhibited in the case of bacterial chemotaxis employed for instance by *E. coli*, has been analysed in great detail. The intrinsic mechanics have been simulated as far down as the molecular level (*e.g.* Bray et al., 2007; Likow et al., 2005) and analytical treatments inspired by Brownian motion also exist (*e.g.* Schnitzer, 1993). However all the methods used in these cases are specific to bacterial chemotaxis (or molecular mechanics thereof), which is different from *C. elegans* chemotaxis in that it does not typically use reversals, and some simulations rely on detailed data which has been collected over decades (Bray et al., 2007). In our work, however, we were less interested in modelling the detailed intrinsic mechanics, perhaps down to the molecular level, of a given behaviour. Rather, we wanted to study the necessary relationships between given behavioural units in the production of a complex behaviour and as such, the present work addresses different issues than previous studies of direct random walk strategies, even though the studied behaviour is the same.

7.2.2 A new model for isotherm tacking

Isothermal tracking behaviour in *C. elegans* has received relative little attention in the literature so far, with only one previously published mathematical model aimed at defining the strategy used by the animal (Luo et al., 2006). We were however unconvinced by this model; most notably we

have shown that the same strategy can be used for deterministic navigation towards the peak of a gradient with only one additional toggle. Since such a toggle has been known to exist in the neural circuitry of *C. elegans* for some time now (Samuel et al., 2003) we found it unlikely that the strategy proposed by Luo et al. (2006) would indeed represent the strategy used by the animal.

It has thus become necessary to investigate whether isothermal tracking could also be performed using a different strategy. We have shown that this was indeed possible using only information from the world for which we felt confident that it was actually available to the animal. We were able to show that our proposed strategy can also perform isotherm tracking with a precision that is sufficient to account for the actual behaviour of *C. elegans*. We do note that the strategy is not perfect, making use of only an approximate linear transformation; however the fact that we can still significantly exceed the performance of *C. elegans* increases our confidence that this is not a fundamental problem. We therefore believe that this model by itself is a significant contribution to the study of *C. elegans* behaviour in thermal environments as it addresses an important flaw of the only previously proposed model.

We tentatively proposed a neural circuit encoding this strategy and identified a number of neurons that could be involved in this circuit. In doing so, we have made use of the major tool available for neuroscientific exploration in *C. elegans*: the availability of a wiring diagram for its entire neural circuit (White et al., 1986). Even with only two restrictions on our proposed neural circuit, we have thus been able to reduce the list of candidate neurons for computing the course correction in our model down to 5. This makes an initial experimental verification of our strategy through laser ablation studies possible but at the same time it illustrates the limitations on behavioural neuroscientific studies when starting from a behavioural angle (as opposed to using laser ablation studies to determine the role of some neurons for instance): since the detailed neural dynamics in *C. elegans* remain unknown, it is only realistically possible to formulate a theoretical high-level computational strategy for the behaviour one is interested in and perhaps make some general predictions on the

required neural architecture, which can then be verified experimentally at a later time. It cannot at present be known how computations are executed in detail, however.

On a final note, we have proposed that this strategy is performed by the computation of the second derivative of the gradient information over time, aimed essentially at locating the time of maximal change in gradient information during one headsweep. We have done so because this seems an obvious reasonable strategy to determine such a time - it is independent of the actual gradient strength (which is significant since the animal is known to track isotherms in a thermal bracket of approximately 6°) and it can be easily implemented by an inhibitory neuron falling silent when its time-shifted inputs cancel each other out. We do however acknowledge that we cannot be absolutely certain that the second derivative is computed. It is possible that there may be simpler strategies based for instance on simply the first derivative in ways that we were not able to determine. However, since this would not change any of our claims besides the exact computation (it would still be necessary to locate a certain time point in the first derivative and the strategy would still require simpler computations than a deterministic navigation towards a region of preference), we do not consider this further.

7.2.3 On the use of stochastic rather than deterministic strategies

One of the most fundamental questions regarding *C. elegans* behaviour which has not yet been adequately addressed in the literature is the use of a directed random walk strategy to navigate towards regions of interest. It would be tempting and at first glance reasonable to attribute this choice to insufficient sensory resolutions or perhaps an inability to perform sufficiently precise movements in response to the sensory input. This remains a valid theory in the case of chemotaxis, for which the precision at which the animal may sample the gradients remains unknown. In the case of thermotaxis, however, it is more difficult to make such an argument; indeed the exceptional performance of the animal at tracking isotherms appears to suggest that it is both able to sample the

environment at sufficient detail and execute sufficiently precise movements to render deterministic strategies possible.

We therefore thought it important to briefly investigate whether the animal could at least in theory use its available information about the gradient to navigate towards regions of interest deterministically and if so, how. We first found, importantly, that the information which appears to be available to the animal is indeed sufficient for such a deterministic strategy. In fact, we have been able to identify two such strategies, one which could provide perfect reorientation manoeuvres at the expense of requiring a memory capable of operating as a FILO stack (for which there is no evidence yet in the neural circuitry of *C. elegans*) and another which could operate using a similar linear approximation than the one used in our proposed isothermal tracking model, at the expense of at a minimum requiring the computation of the third derivative.

It thus appears that the animal is likely to be able to sample sufficient information for deterministic navigation towards its temperature of preference from the environment and it is therefore interesting that it nonetheless prefers a stochastic approach. We can offer a few speculations why this may be so. Most likely, it is possible that chemosensory neurons may not actually be able to sample the gradient at a sufficient resolution. This would make a stochastic strategy necessary for chemotaxis and since the same strategy would remain usable for navigation towards the temperature of preference (Ryu and Samuel, 2002), there may not be a sufficient advantage to warrant the evolution of a circuit for deterministic navigation only applicable in thermal environments. Alternatively, it may be possible that the physical distance the signal from the AFD neuron has to travel through the neural circuitry is too large to compute the third derivative with sufficient precision. We note that both reasons are still speculative given the current knowledge of *C. elegans*; we therefore hope that the animal's use of stochastic strategies over deterministic ones when navigating towards temperatures of preference will receive further attention now that we have shown it not to be simply due to limitations in the available information.

7.3 Conclusion

In Chapter 2, we identified four major open questions about the behaviour of *C. elegans*:

- Given the three locomotory behaviours of *C. elegans* (forward runs, turns and reversals), how are these best combined if the aim is efficient chemotaxis? Is the pirouette strategy employed by the animal optimal in this sense? Why are pirouettes used at all?
- Is it possible to derive novel predictions on the computational capabilities of the animal from the chemotactic behaviour?
- Is the existing proposed strategy for isotherm tracking (Luo et al., 2006) reasonable? If not, can another strategy be proposed?
- Given how well the animal tracks isotherms, why does it rely on stochastic strategies at all when navigating towards the preferred temperature? Are deterministic strategies prevented by something fundamental?

As has become evident in the present chapter, we have been able to adequately address all of these points. To address some of them, we have formulated a novel framework for analysing behaviour based on optimised models. The utility of this framework is not restricted to *C. elegans* research and we expect that it will find applications in a large variety of behavioural studies. The work presented in this thesis has thus contributed not only to *C. elegans* research specifically but has also provided a methodology that we expect to have applications in the general field of ethology.

Even though the present thesis has addressed the major open questions we initially identified in Chapter 2, there are always possibilities for further research. Here, we will briefly discuss some possible future work based on the present thesis.

8.1 Extending the framework for analysing behavioural models

The framework for analysing behavioural models, as defined in Chapter 4 can only strictly be applied to models with a finite parameter space. This has suited our needs and it is likely that it will remain useful for many other models, but for the sake of completeness it would be desirable to extend the framework so it can also deal with infinite parameter spaces.

This is not necessarily a trivial task. The framework as defined relies on the division of the entire parameter space into subspaces of equal volume. In infinite parameter spaces, this will no longer be possible and it therefore becomes necessary to define other meaningful ways of division into subspaces.

8.2 Experimental verification of predictions for *C. elegans*

We have made a number of novel predictions on the computational and sensory capabilities of *C. elegans*. It would be very interesting to verify these in the real animal.

8.2.1 Isotherm strategy

We have identified some candidate neurons that may be involved in the isotherm tracking strategy of *C. elegans*. An important follow-up to this identification would be verifying whether isotherm tracking is disturbed when any of these neurons are destroyed.

8.2.2 Resolution and operating range of chemosensory neurons

We hypothesised that *C. elegans* may be using stochastic rather than deterministic strategies for navigating towards the peak of chemical gradients due to an insufficient resolution of the chemosensory neurons. Additionally, we predicted that the animal may only be able to act upon gradient information while moving forward. These predictions could be verified by measuring the activity of chemosensory neurons using calcium imaging techniques in a similar fashion to the work by (Clark et al., 2006) on the thermosensory neuron. This may be less trivial for chemosensation than it was for thermosensation though. In particular the fact that the animal has several chemosensory neurons may make it more difficult to determine their sensitivity.

8.3 Behavioural work on *C. elegans*

8.3.1 Isotherm strategy

One particularly interesting neuron that we have identified as possibly being involved in the isotherm tracking strategy of *C. elegans* was the RIA neuron. It is interesting because its ablation is in fact known to disrupt isotherm tracking behaviour (Mori and Ohshima, 1995), which makes

it relevant to our model. However, its ablation also causes cryophilic behaviour, which suggests that RIA also plays a role in navigating towards the preferred temperature. These two roles are not in principle contradictory, however their relation would need to be clarified. Specifically, could its putative role in our isotherm tracking strategy also have implications for navigating towards the preferred temperature? Would it be possible to propose a unified model of both navigation strategies which makes use of RIA in a way that is consistent with both what is already known about those strategies and our predictions?

8.3.2 The use of stochastic strategies for navigating towards the preferred temperature

We have shown that the animal does not simply use stochastic strategies for navigating towards the preferred temperature as a result of being unable to extract the required information for deterministic strategies from the available sensory information. Additionally, we have argued that both the sensory resolution and the precision of the motor output should be sufficient to correctly determine and execute such deterministic strategies.

It would now be interesting to examine why the animal makes use of these stochastic strategies over deterministic ones in more detail. Our outline of the necessary computations for two theoretical deterministic strategies for navigating towards the preferred temperature may assist in such investigations.

Part V

Appendix

Appendix A

List of neurons in *C. elegans*

THE following is a list of neurons found in the hermaphrodite *C. elegans*. It is adapted from Altun and Hall (2006), where detailed diagrams of each neuron's location can also be found.

Neuron name	Brief description
ADAL	Ring interneuron
ADAR	Ring interneuron
ADEL	Anterior deirid, sensory neuron
ADER	Anterior deirid, sensory neuron
ADFL	Amphid neuron
ADFR	Amphid neuron
ADLL	Amphid neuron
ADLR	Amphid neuron
AFDL	Amphid finger cell
AFDR	Amphid finger cell

Neuron name	Brief description
AIAL	Amphid interneuron
AIAR	Amphid interneuron
AIBL	Amphid interneuron
AIBR	Amphid interneuron
AIML	Ring interneuron
AIMR	Ring interneuron
AINL	Ring interneuron
AINR	Ring interneuron
AIYL	Amphid interneuron
AIYR	Amphid interneuron
AIZL	Amphid interneuron
AIZR	Amphid interneuron
ALA	Neuron, sends processes laterally and along dorsal cord
ALML	Anterior lateral microtubule cell
ALMR	Anterior lateral microtubule cell
ALNL	Neuron associated with ALM
ALNR	Neuron associated with ALM
AQR	Neuron, basal body. not part of a sensillum, projects into ring
AS1	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS2	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS3	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS4	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS5	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart

Neuron name	Brief description
AS6	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS7	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS8	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS9	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS10	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
AS11	Ventral cord motor neuron, innervates dorsal muscles, no ventral counterpart
ASEL	Amphid neuron, single ciliated endings
ASER	Amphid neuron, single ciliated endings
ASGL	Amphid neuron, single ciliated endings
ASGR	Amphid neuron, single ciliated endings
ASHL	Amphid neuron, single ciliated endings
ASHR	Amphid neuron, single ciliated endings
ASIL	Amphid neuron, single ciliated endings
ASIR	Amphid neuron, single ciliated endings
ASJL	Amphid neuron, single ciliated endings
ASJR	Amphid neuron, single ciliated endings
ASKL	Amphid neuron, single ciliated endings
ASKR	Amphid neuron, single ciliated endings
AUAL	Neuron, process runs with amphid processes but lacks ciliated ending
AUAR	Neuron, process runs with amphid processes but lacks ciliated ending
AVAL	Ventral cord interneuron
AVAR	Ventral cord interneuron
AVBL	Ventral cord interneuron

Neuron name	Brief description
AVBR	Ventral cord interneuron
AVDL	Ventral cord interneuron
AVDR	Ventral cord interneuron
AVEL	Ventral cord interneuron, like AVD but outputs restricted to anterior cord
AVER	Ventral cord interneuron, like AVD but outputs restricted to anterior cord
AVFL	Interneuron
AVFR	Interneuron
AVG	Ventral cord interneuron
AVHL	Neuron, mainly postsynaptic in ventral cord and presynaptic in the ring
AVHR	Neuron, mainly postsynaptic in ventral cord and presynaptic in the ring
AVJL	Neuron, synapses like AVHL/R
AVJR	Neuron, synapses like AVHL/R
AVKL	Ring and ventral cord interneuron
AVKR	Ring and ventral cord interneuron
AVL	Ring and ventral cord interneuron and an excitatory GABAergic motor neuron for rectal muscles. Few synapses
AVM	Anterior ventral microtubule cell, touch receptor
AWAL	Amphid wing cells, neuron having ciliated sheet-like sensory endings closely associated with amphid sheath
AWAR	Amphid wing cells, neuron having ciliated sheet-like sensory endings closely associated with amphid sheath
AWBL	Amphid wing cells, neuron having ciliated sheet-like sensory endings closely associated with amphid sheath

Neuron name	Brief description
AWBR	Amphid wing cells, neuron having ciliated sheet-like sensory endings closely associated with amphid sheath
AWCL	Amphid wing cells, neuron having ciliated sheet-like sensory endings closely associated with amphid sheath
AWCR	Amphid wing cells, neuron having ciliated sheet-like sensory endings closely associated with amphid sheath
BAGL	Neuron, ciliated ending in head, no supporting cells, associated with ILso
BAGR	Neuron, ciliated ending in head, no supporting cells, associated with ILso
BDUL	Neuron, process runs along excretory canal and into ring, unique darkly staining synaptic vesicles
BDUR	Neuron, process runs along excretory canal and into ring, unique darkly staining synaptic vesicles
CANL	Process runs along excretory canal, no synapses, essential for survival
CANR	Process runs along excretory canal, no synapses, essential for survival
CEPDL	Cephalic neuron, contain dopamine
CEPDR	Cephalic neuron, contain dopamine
CEPVL	Cephalic neuron, contain dopamine
CEPVR	Cephalic neuron, contain dopamine
DA1	Ventral cord motor neuron, innervate dorsal muscles
DA2	Ventral cord motor neuron, innervate dorsal muscles
DA3	Ventral cord motor neuron, innervate dorsal muscles
DA4	Ventral cord motor neuron, innervate dorsal muscles
DA5	Ventral cord motor neuron, innervate dorsal muscles

Neuron name	Brief description
DA6	Ventral cord motor neuron, innervate dorsal muscles
DA7	Ventral cord motor neuron, innervate dorsal muscles
DA8	Ventral cord motor neuron, innervate dorsal muscles
DA9	Ventral cord motor neuron, innervate dorsal muscles
DB1/3	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DB2	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DB3/1	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DB4	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DB5	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DB6	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DB7	Ventral cord motor neuron, innervate dorsal muscles, reciprocal inhibitor
DD1	Ventral cord motor neuron, reciprocal inhibitors
DD2	Ventral cord motor neuron, reciprocal inhibitors
DD3	Ventral cord motor neuron, reciprocal inhibitors
DD4	Ventral cord motor neuron, reciprocal inhibitors
DD5	Ventral cord motor neuron, reciprocal inhibitors
DD6	Ventral cord motor neuron, reciprocal inhibitors
DVA	Ring interneuron, cell bodies in dorsorectal ganglion
DVB	An excitatory GABAergic motor neuron/interneuron located in dorso-rectal ganglion. Innervates rectal muscles.
DVC	Ring interneuron, cell bodies in dorsorectal ganglion
FLPL	Neuron, ciliated ending in head, no supporting cells, associated with ILso
FLPR	Neuron, ciliated ending in head, no supporting cells, associated with ILso

Neuron name	Brief description
HSNL	Herm. specific motor neuron (die in male embryo), innervate vulval muscles, serotonergic
HSNR	Herm. specific motor neuron (die in male embryo), innervate vulval muscles, serotonergic
I1L	Pharyngeal interneuron: ant sensory, input from RIP
I1R	Pharyngeal interneuron: ant sensory, input from RIP
I2L	Pharyngeal interneuron, ant sensory.
I2R	Pharyngeal interneuron, ant sensory.
I3	Pharyngeal interneuron, ant sensory.
I4	Pharyngeal interneuron.
I5	Pharyngeal interneuron, post sensory.
I6	Pharyngeal interneuron, post sensory.
IL1DL	Inner labial neuron
IL1DR	Inner labial neuron
IL1L	Inner labial neuron
IL1R	Inner labial neuron
IL1VL	Inner labial neuron
IL1VR	Inner labial neuron
IL2DL	Inner labial neuron
IL2DR	Inner labial neuron
IL2L	Inner labial neuron
IL2R	Inner labial neuron
IL2VL	Inner labial neuron

Neuron name	Brief description
IL2VR	Inner labial neuron
LUAL	Interneuron, short process in post ventral cord
LUAR	Interneuron, short process in post ventral cord
M1	Pharyngeal motorneuron
M2L	Pharyngeal motorneuron
M2R	Pharyngeal motorneuron
M3L	Pharyngeal sensory-motorneuron
M3R	Pharyngeal sensory-motorneuron
M4	Pharyngeal motorneuron
M5	Pharyngeal motorneuron
MCL	Pharyngeal neuron that synapse onto marginal cells
MCR	Pharyngeal neuron that synapse onto marginal cells
MI	Pharyngeal motor neuron/interneuron
NSML	Pharyngeal neurosecretory motorneuron, contain serotonin
NSMR	Pharyngeal neurosecretory motorneuron, contain serotonin
OLLL	Lateral outer labial neuron
OLLR	Lateral outer labial neuron
OLQDL	Quadrant outer labial neuron
OLQDR	Quadrant outer labial neuron
OLQVL	Quadrant outer labial neuron
OLQVR	Quadrant outer labial neuron
PDA	Motor neuron, process in dorsal cord
PDB	Motor neuron, process in dorsal cord, cell body in pre-anal ganglion

Neuron name	Brief description
PDEL	Neuron, dopaminergic of postderid sensillum
PDER	Neuron, dopaminergic of postderid sensillum
PHAL	Phasmid neuron, chemosensory
PHAR	Phasmid neuron, chemosensory
PHBL	Phasmid neuron, chemosensory
PHBR	Phasmid neuron, chemosensory
PHCL	Neuron, striated rootlet in male, possibly sensory in tail spike
PHCR	Neuron, striated rootlet in male, possibly sensory in tail spike
PLML	Posterior lateral microtubule cells, touch receptor neuron
PLMR	Posterior lateral microtubule cells, touch receptor neuron
PLNL	Interneuron, associated with PLM
PLNR	Interneuron, associated with PLM
PQR	Neuron, basal body, not part of a sensillum, projects into preanal ganglion
PVCL	Ventral cord interneuron, cell body in lumbar ganglion, synapses onto VB and DB motor neuron, formerly called delta.
PVCR	Ventral cord interneuron, cell body in lumbar ganglion, synapses onto VB and DB motor neuron, formerly called delta.
PVDL	Neuron, lateral process adjacent to excretory canal
PVDR	Neuron, lateral process adjacent to excretory canal
PVM	Posterior ventral microtubule cell, touch receptor
PVNL	Interneuron/motor neuron, post. vent. cord, few synapses
PVNR	Interneuron/motor neuron, post. vent. cord, few synapses
PVPL	Interneuron, cell body in preanal ganglion, projects along v. cord to nerve ring

Neuron name	Brief description
PVPR	Interneuron, cell body in preanal ganglion, projects along v. cord to nerve ring
PVQL	Interneuron, projects along ventral cord to ring
PVQR	Interneuron, projects along ventral cord to ring
PVR	Interneuron, projects along ventral cord to ring
PVS	PVPR interneuron of male, cell body in preanal ganglion, sexually dimorphic connectivity
PVT	Interneuron, projects along ventral cord to ring
PVWL	Interneuron, posterior ventral cord, few synapses
PVWR	Interneuron, posterior ventral cord, few synapses
RIAL	Ring interneuron, many synapses
RIAR	Ring interneuron, many synapses
RIBL	Ring interneuron
RIBR	Ring interneuron
RICL	Ring interneuron
RICR	Ring interneuron
RID	Ring interneuron, projects along dorsal cord
RIFL	Ring interneuron
RIFR	Ring interneuron
RIGL	Ring interneuron
RIGR	Ring interneuron
RIH	Ring interneuron
RIML	Ring motor neuron
RIMR	Ring motor neuron

Neuron name	Brief description
RIPL	Ring/pharynx interneuron, only direct connection between pharynx and ring
RIPR	Ring/pharynx interneuron, only direct connection between pharynx and ring
RIR	Ring interneuron
RIS	Ring interneuron
RIVL	Ring interneuron
RIVR	Ring interneuron
RMDDL	Ring motor neuron/interneuron, many synapses
RMDDR	Ring motor neuron/interneuron, many synapses
RMDL	Ring motor neuron/interneuron, many synapses
RMDR	Ring motor neuron/interneuron, many synapses
RMDVL	Ring motor neuron/interneuron, many synapses
RMDVR	Ring motor neuron/interneuron, many synapses
RMED	Ring motor neuron
RMEL	Ring motor neuron
RMER	Ring motor neuron
RMEV	Ring motor neuron
RMFL	Ring motor neuron/interneuron
RMFR	Ring motor neuron/interneuron
RMGL	Ring interneuron
RMGR	Ring interneuron
RMHL	Ring motor neuron/interneuron
RMHR	Ring motor neuron/interneuron
SAADL	Ring interneuron, anteriorly projecting process that runs sublaterally

Neuron name	Brief description
SAADR	Ring interneuron, anteriorly projecting process that runs sublaterally
SAAVL	Ring interneuron, anteriorly projecting process that runs sublaterally
SAAVR	Ring interneuron, anteriorly projecting process that runs sublaterally
SABD	Ring interneuron, anteriorly projecting process that runs sublaterally, synapses to anterior body muscles in L1
SABVL	Ring interneuron, anteriorly projecting process that runs sublaterally, synapses to anterior body muscles in L1
SABVR	Ring interneuron, anteriorly projecting process that runs sublaterally, synapses to anterior body muscles in L1
SDQL	Post. lateral interneuron, process projects into ring
SDQR	Ant. lateral interneuron, process projects into ring
SIADL	Receive a few synapses in the ring, have sublateral posteriorly directed processes
SIADR	Receive a few synapses in the ring, have sublateral posteriorly directed processes
SIAVL	Receive a few synapses in the ring, have sublateral posteriorly directed processes
SIAVR	Receive a few synapses in the ring, have sublateral posteriorly directed processes
SIBDL	Similar to SIA
SIBDR	Similar to SIA
SIBVL	Similar to SIA
SIBVR	Similar to SIA

Neuron name	Brief description
SMBDL	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMBDR	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMBVL	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMBVR	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMDDL	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMDDR	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMDVL	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
SMDVR	Ring motor neuron/interneuron, has sublateral posteriorly directed processes
URADL	Ring motor neuron
URADR	Ring motor neuron
URAVL	Ring motor neuron
URAVR	Ring motor neuron
URBL	Neuron, presynaptic in ring, ending in head
URBR	Neuron, presynaptic in ring, ending in head
URXL	Ring interneuron
URXR	Ring interneuron
URYDL	Neuron, presynaptic in ring, ending in head
URYDR	Neuron, presynaptic in ring, ending in head
URYVL	Neuron, presynaptic in ring, ending in head
URYVR	Neuron, presynaptic in ring, ending in head
VA1	Vent. cord motor neuron, innervates vent. body muscles
VA2	Vent. cord motor neuron, innervates vent. body muscles
VA3	Vent. cord motor neuron, innervates vent. body muscles

Neuron name	Brief description
VA4	Vent. cord motor neuron, innervates vent. body muscles
VA5	Vent. cord motor neuron, innervates vent. body muscles
VA6	Vent. cord motor neuron, innervates vent. body muscles
VA7	Vent. cord motor neuron, innervates vent. body muscles
VA8	Vent. cord motor neuron, innervates vent. body muscles
VA9	Vent. cord motor neuron, innervates vent. body muscles
VA10	Vent. cord motor neuron, innervates vent. body muscles
VA11	Vent. cord motor neuron, innervates vent. body muscles
VA12	Vent. cord motor neuron, innervates vent. body muscles, but also interneuron in preanal ganglion
VB1	Vent. cord motor neuron, innervates vent. body muscles, also interneuron in ring
VB2	Vent. cord motor neuron, innervates vent. body muscles
VB3	Vent. cord motor neuron, innervates vent. body muscles
VB4	Vent. cord motor neuron, innervates vent. body muscles
VB5	Vent. cord motor neuron, innervates vent. body muscles
VB6	Vent. cord motor neuron, innervates vent. body muscles
VB7	Vent. cord motor neuron, innervates vent. body muscles
VB8	Vent. cord motor neuron, innervates vent. body muscles
VB9	Vent. cord motor neuron, innervates vent. body muscles
VB10	Vent. cord motor neuron, innervates vent. body muscles
VB11	Vent. cord motor neuron, innervates vent. body muscles
VC1	Vent cord motor neuron innervates vulval muscles and vent body muscles

Neuron name	Brief description
VC2	Vent cord motor neuron innervates vulval muscles and vent body muscles
VC3	Vent cord motor neuron innervates vulval muscles and vent body muscles
VC4	Vent cord motor neuron innervates vulval muscles and vent body muscles
VC5	Vent cord motor neuron innervates vulval muscles and vent body muscles
VC6	Vent cord motor neuron innervates vulval muscles and vent body muscles
VD1	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD2	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD3	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD4	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD5	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD6	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD7	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD8	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD9	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD10	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD11	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD12	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor
VD13	Vent cord motor neuron, innervates vent body muscles, reciprocal inhibitor

Bibliography

- Albertson, D. G. and Thomson, J. N. (1976). The pharynx of *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society of London Series B*, 275:299–325.
- Altun, Z. and Hall, D., editors (2002-2005). *Atlas of C. elegans anatomy - an illustrated handbook*, chapter Hermaphrodite Anatomy. www.wormatlas.org.
- Altun, Z. F. and Hall, D. H., editors (2002-2006). *Worm Atlas*. www.wormatlas.org.
- Andrews, G. (1990). Euler's "exemplum memorabile inductionis fallacis" and q -trinomial coefficients. *Journal of the American Mathematical Society*, 3:653–669.
- Avery, L. (1993). Motor neuron m3 controls pharyngeal muscle relaxation timing in *Caenorhabditis elegans*. *Journal of Experimental Biology*, 175:283–297.
- Baker, T. C. (1986). *Mechanisms in Insect Olfaction*, chapter Pheromone-modulated movements of flying moths, pages 39–48. Oxford University Press, Oxford, UK.

- Balkovsky, E. and Shraiman, B. I. (2002). Olfactory search at high Reynolds number. *Proceedings of the National Academy of Sciences of the United States of America*, 99(20):12589–12593.
- Bargmann, C. I. (1993). Genetic and cellular analysis of behavior in *C. elegans*. *Annual Reviews Neuroscience*, 16:47–71.
- Bargmann, C. I. and Horvitz, H. R. (1991). Chemosensory neurons with overlapping functions direct chemotaxis to multiple chemicals in *C. elegans*. *Neuron*, 7:729–742.
- Berg, H. C. and Brown, D. A. (1972). Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. *Nature*, 239:500–504.
- Biron, D., Shibuya, M., Gabel, C., Wasserman, S. M., Clark, D. A., Brown, A., Sengupta, P., and Samuel, A. D. T. (2006). A diacylglycerol kinase modulates long-term thermotactic behavioral plasticity in *C. elegans*. *Nature Neuroscience*, 9(12):1499–1505.
- Branch, M. A., Coleman, T. F., and Li, Y. (1999). A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21:1–23.
- Bray, D., Levin, M. D., and Lipkow, K. (2007). The chemotactic behavior of computer-based surrogate bacteria. *Current Biology*, 17:12–19.
- Brémaud, P. (1999). *Markov Chains, Gibbs Field, Monte Carlo Simulation and Queues*. Springer.
- Byrd, R. H., Schnabel, R. B., and Shultz, G. A. (1987). Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical Programming*, 40:247–263.
- Cerny, V. (1985). A thermodynamical approach to the travelling salesman problem: an efficient solution algorithm. *Journal of Optimisation Theory and Applications*, 45:41–51.

- Chalfie, M., Sulston, J., White, J., Southgate, E., Thomson, J., and Brenner, S. (1985). The neural circuit for touch sensitivity in *Caenorhabditis elegans*. *Journal of Neuroscience*, 5(4):956–964.
- Cheung, B. H. H., Arellano-Carbajal, F., Rybicki, I., and de Bono, M. (2004). Soluble guanylate cyclases act in neurons exposed to the body fluid to promote *C. elegans* aggregation behavior. *Current Biology*, 14:1105–1111.
- Cheung, B. H. H., Cohen, M., Rogers, C., Albayram, O., and de Bono, M. (2005). Experience-dependent modulation of *C. elegans* Behaviour by ambient oxygen. *Current Biology*, 15:905–917.
- Clark, D. A., Biron, D., Sengupta, P., and Samuel, A. D. T. (2006). The *afd* sensory neurons encode multiple functions underlying thermotactic behavior in *Caenorhabditis elegans*. *The Journal of Neuroscience*, 26(28):7444–7451.
- Clark, D. A., Gabel, C. V., Lee, T. M., and Samuel, A. D. T. (2007). Short-term adaptation and temporal processing in the cryophilic response of *Caenorhabditis elegans*. *Journal of Neurophysiology*, 97:1903–1910.
- Coates, J. C. and de Bono, M. (2002). Antagonistic pathways in neurons exposed to body fluid regulate social feeding in *Caenorhabditis elegans*. *Nature*, 419:925–929.
- Dale, H. H. (1935). Pharmacology and nerve endings. *Proceedings of the Royal Society of Medicine*, 28:319–332.
- de Bono, M. (2003). Molecular approaches to aggregation behavior and social attachment. *Journal of Neurobiology*, 54:78–92.
- de Bono, M. and Bargmann, C. I. (1998). Natural variation in a neuropeptide y receptor homolog modifies social behavior and food response in *C. elegans*. *Cell*, 94:679–689.

- de Bono, M. and Maricq, A. V. (2005). Neuronal substrates of complex behaviors in *C. elegans*. *Annual Reviews Neuroscience*, 28:451–501.
- de Bono, M., Tobin, D. M., Davis, M. W., Avery, L., and Bargmann, C. I. (2002). Social feeding in *Caenorhabditis elegans* is induced by neurons that detect aversive stimuli. *Nature*, 419:899–903.
- Dougherty, E. C. (1953). *Thaper Commemoration Volume*, chapter The genera of the subfamily Rhabditinae Micoletzky, 1922 (Nematoda), page 69.76. Lucknow.
- Dunn, N. A., Lockery, S. R., Pierce-Shimomura, J. T., and Conery, J. S. (2004). A neural network model of chemotaxis predicts functions of synaptic connections in the nematode *Caenorhabditis elegans*. *Journal of Computational Neuroscience*, 17:137–147.
- Durbin, R. M. (1987). *Studies on the Development and Organisation of the Nervous System of Caenorhabditis Elegans*. PhD thesis, University of Cambridge.
- Feller, W. (1968). *An introduction to probability theory and its applications*. Wiley.
- Ferrée, T. C. and Lockery, S. R. (1999). Computational rules for chemotaxis in the nematode *C. elegans*. *Journal of Computational Neuroscience*, 6:263–277.
- Fraser, A. and Burnell, D. (1970). *Computer Models in Genetics*. McGraw-Hill, New York.
- Gabel, C. V., Gabel, H., Pavlichin, D., Kao, A., Clark, D. A., and Samuel, A. D. T. (2007). Neural circuits mediate electrosensory behavior in *Caenorhabditis elegans*. *The Journal of Neuroscience*, 27(28):7586–7596.
- Goodman, M. B., Hall, D. H., Avery, L., and Lockery, S. R. (1998). Active currents regulate sensitivity and dynamic range in *C. elegans* neurons. *Neuron*, 20:763–772.
- Gray, J. M., Hill, J. J., and Bargmann, C. I. (2005). A circuit for navigation in *Caenorhabditis*

- elegans*. *Proceedings of The National Academy Of Sciences Of The United States Of America*, 102(9):3184–3191.
- Gray, J. M., Karow, D. S., Lu, H., Chang, A. J., Chang, J. S., Ellis, R. E., Marletta, M. A., and Bargmann, C. I. (2004). Oxygen sensation and social feeding mediated by a *C. elegans* guanylate cyclase homologue. *Nature*, 430:317–322.
- Grinstead, C. M. and Snell, J. L. (1997). *Introduction to Probability*, chapter Markov Chains, pages 405–452. American Mathematical Society.
- Hardaker, L. A., Singer, E., Kerr, R., Zhou, G., and Shafer, W. R. (2001). Serotonin modulates locomotory behavior and coordinates egg-laying and movement in *Caenorhabditis elegans*. *Journal of Neurobiology*, 49:303–313.
- Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544.
- Ito, H., Inada, H., and Mori, I. (2006). Quantitative analysis of thermotaxis in the nematode *Caenorhabditis elegans*. *Journal of Neuroscience Methods*, 154:45–52.
- Iwasaki, Y. and Gomi, S. (2004). Stochastic formulation for a partial neural circuit of *C. elegans*. *Bulletin of Mathematical Biology*, 66:727–743.
- Kaplan, J. M. (1996). Sensory signaling in *Caenorhabditis elegans*. *Current Opinion In Neurobiology*, 6:494–499.
- Kimura, K. D., Miyawaki, A., Matsumoto, K., and Mori, I. (2004). The *C. elegans* thermosensory neuron and responds to warming. *Current Biology*, 14:1291–1295.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.

- Kooijman, S. A. (2000). *Dynamic Energy and Mass Budgets in Biological Systems*. Cambridge University Press, Cambridge.
- Koyamada, K., Sakai, K., and Itoh, T. (2004). Parameter optimization technique using the response surface methodology. In *Conference Proceedings IEEE Engineering in Medicine and Biology Society*.
- Lanjuin, A. and Sengupta, P. (2004). Specification of chemosensory neuron subtype identities in *Caenorhabditis elegans*. *Current Opinion In Neurobiology*, 14:22–30.
- Lehner, P. N. (1996). *Handbook of Ethological Methods*. Cambridge University Press: UK.
- Likow, K., Andrews, S. S., and Bray, D. (2005). Simulated diffusion of phosphorylated cheY through the cytoplasm of *Escherichia coli*. *Journal of Bacteriology*, 187:45–53.
- Luo, L., Clark, D. A., Biron, D., Mahadevan, L., and Samuel, A. D. T. (2006). Sensorimotor control during isothermal tracking in *Caenorhabditis elegans*. *Journal of Experimental Biology*, 209:4652–4662.
- Luo, L., Clark, D. A., Biron, D., Mahadevan, L., and Samuel, A. D. T. (2007). Sensorimotor control during isothermal tracking in *Caenorhabditis elegans*. *Journal of Experimental Biology*, (corrigendum).
- MacArthur, R. H. and Pianka, E. R. (1966). On optimal use of a patchy environment. *American Naturalist*, 100:603 – 609.
- Majewska, A. and Juste, R. (2001). Topology of gap junction networks in *C. elegans*. *Journal of Theoretical Biology*, 212:155–167.
- Martin, P. and Bateson, P. (1993). *Measuring Behaviour*. Cambridge University Press, Cambridge, Massachusetts, 2nd edition.

- Maupas, E. (1899). La mue et l'enkystement chez les nématodes. *Archives de Zoologie Expérimentale et Générale*, 7:563-628.
- Maupas, E. (1900). Modes et formes de reproduction des nématodes. *Archives de Zoologie Expérimentale et Générale*, 8:463-624.
- Miller, A. C., Thiele, T. R., Faumont, S., Moravec, M. L., and Lockery, S. R. (2005). Step-response analysis of chemotaxis in *Caenorhabditis elegans*. *Journal of Neuroscience*, 25(13):336-3378.
- Mitchell, T. (1997). *Machine Learning*. McGraw Hill.
- Mori, I. (1999). Genetics of chemotaxis and thermotaxis in the nematode *Caenorhabditis elegans*. *Annual Review of Genetics*, 33:399-422.
- Mori, I. and Ohshima, Y. (1995). Neural regulation of thermotaxis in *Caenorhabditis elegans*. *Nature*, 376(6538):344-348.
- Nickell, W. T., Pun, R. Y. K., Bargmann, C. I., and Kleene, S. J. (2002). Single ionic channels of two *Caenorhabditis elegans* chemosensory neurons in native membrane. *Journal of Membrane Biology*, 189:55-66.
- Niebur, E. and Erdős, P. (1993). Theory of the locomotion of nematodes: Control of the somatic motor neurons by interneurons. *Mathematical Biosciences*, 118:51-82.
- Nishikawa, T., Motter, A. E., Lai, Y., and Hoppensteadt, F. C. (2002). Smallest small-world network. *Physical Review E*, 66(046139):1-5.
- O'Reilly, R. C. and Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience*. MIT Press, Cambridge, Massachusetts.
- Osche, G. (1952). Systematik und phylogenie der gattung rhabditis (nematoda). *Zoologische Jahrbücher (Abteilung Systematik)*, 81:190-280.

- Oshio, K., Iwasaki, Y., Morita, S., Osana, Y., Gomi, S., Akiyama, E., Omata, K., Oka, K., and Kawamura, K. (2003). Database of synaptic connectivity of *C. elegans* for computation. Technical Report of CCeP, Keio Future 3, Keio University.
- Philippides, A., Husbands, P., Smith, T., and O'shea, M. (2005). Flexible couplings: Diffusing neuromodulators and adaptive robotics. *Artificial Life*, 11:139–160.
- Pierce-Shimomura, J. T., Dores, M., and Lockery, S. R. (2005). Analysis of the effects of turning bias on chemotaxis in *C. elegans*. *Journal of Experimental Biology*, 208:4727–4733.
- Pierce-Shimomura, J. T., Morse, T. M., and Lockery, S. R. (1999). The fundamental role of pirouettes in *Caenorhabditis elegans* chemotaxis. *Journal of Neuroscience*, 19(21):9557–9569.
- Prinz, A. A., Bucher, D., and Marder, E. (2004). Similar network activity from disparate circuit parameters. *Nature Neuroscience*, 7(12):1345–1352.
- Riddle, D., Blumenthal, T., Meyer, B., and Priess, J. R., editors (1997). *C. elegans II*. Cold Spring Harbor Laboratory Press, Plainview (NY).
- Ryu, W. S. and Samuel, A. D. T. (2002). Thermotaxis in *Caenorhabditis elegans* analyzed by measuring responses to defined thermal stimuli. *Journal of Neuroscience*, 22:5727–5733.
- Samuel, A. D. T., Silva, R. A., and Murthy, V. N. (2003). Synaptic activity of the *afd* neuron in *Caenorhabditis elegans* correlates with thermotactic memory. *Journal of Neuroscience*, 23(2):373–376.
- Sánchez-Montañés, M. A. and Pearce, T. C. (2006). Prediction of the behavioural strategy in a chemotaxis search task. *Unpublished*.
- Schnitzer, M. J. (1993). Theory of continuum random walks and application to chemotaxis. *Physical Review E*, 48:2553–2568.

- Schuske, K., Beg, A. A., and Jorgensen, E. M. (2004). The gaba nervous system in *C. Elegans*. *Trends In Neurosciences*, 27(7):407–414.
- Seth, A. K. (2007). The ecology of action selection: Insights from artificial life. *Philosophical Transactions of the Royal Society of London Series B*, 362(1485):1545–1558.
- Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press.
- Suzuki, M., Goto, T., Tsuji, T., and Ohtake, H. (2005a). A dynamic body model of the nematode *C. elegans* with neural oscillators. *Journal of Robotics and Mechatronics*, 17(3):318–326.
- Suzuki, M., Tsuji, T., and Ohtake, H. (2005b). A model of motor control of the nematode *C. elegans* with neuronal circuits. *Artificial Intelligence In Medicine*, 35:75–86.
- Thill, S. and Pearce, T. C. (2007). Understanding complex behaviors by analyzing optimized models: *C. elegans* gradient navigation. *HFSP Journal*, 1(4):263–273.
- Tinbergen, N. (1963). On aims and methods of ethology. *Zeitschrift für Tierpsychologie*, 20:410–433.
- Tsalik, E. L., Niacaris, T., Wenick, A. S., Pau, K., Avery, L., and Hobert, O. (2003). Lim homeobox gene-dependent expression of biogenic amine receptors in restricted regions of the *C. elegans* nervous system. *Journal of Developmental Biology*, 263:81–102.
- Tsalik, L. E. and Hobert, O. (2003). Functional mapping of neurons that control locomotory behavior in *Caenorhabditis elegans*. *Journal of Neurobiology*, 56:178–197.
- Vergassola, M., Villermaux, E., and Shraiman, B. I. (2007). 'infotaxis' as a strategy for searching without gradients. *Nature*, 445:406–409.

- Vickers, N. J. and Baker, T. C. (1994). Reiterative responses to single strands of odor promote sustained upwind flight and odor source location by moths. *Proceedings of the National Academy of Sciences of the United States of America*, 9:5756–5760.
- Ward, S. (1973). Chemotaxis by the nematode *Caenorhabditis elegans*: Identification of attractants and analysis of the response by use of mutants. *Proceedings of The National Academy Of Sciences Of The United States Of America*, 70:817–821.
- Wei, A., Jegla, T., and Salkoff, L. (1996). Eight potassium channel families revealed by the *C. elegans* genome project. *Neuropharmacology*, 35(7):805–829.
- Wes, P. and Bargmann, C. I. (2001). *C. elegans* odour discrimination requires asymmetric diversity in olfactory neurons. *Nature*, 410(6829):698–701.
- White, J. G., Southgate, E., Thomson, J. N., and Brenner, S. (1986). The structure of the nervous system of the neamtode *C. elegans*. *Philosophical Transactions of the Royal Society of London Series B*, 314(1165):1–340.
- Zariwala, H. A., Miller, A. C., Faumont, S., and Lockery, S. R. (2003). Step response analysis of thermotaxis in *Caenorhabditis elegans*. *Journal of Neuroscience*, 23(10):4369–4377.
- Zhao, B., Khare, P., Feldman, L., and Dent, J. A. (2003). Reversal frequency in *Caenorhabditis elegans* represents an integrated response to the state of the animal and its environment. *Journal of Neuroscience*, 23(12):5319–5328.