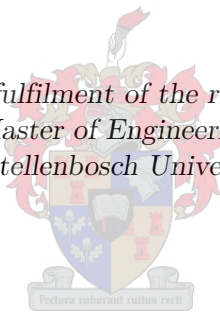


Biologically Inspired Flight:
The autonomous control of a quadcopter using spiking neural
networks

by

Durham Valdon Smith

*Thesis presented in partial fulfilment of the requirements for the degree of
Master of Engineering
at Stellenbosch University*



Supervisors:

Prof T. Jones

Department Electrical and Electronic Engineering

December 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2015

Abstract

With the recent increase in the availability and popularity of micro aerial vehicles new methods for their autonomous control should be investigated. In this project the autonomous control of a flying drone using biologically inspired models is investigated. The models that are investigated for this project are those of the various neurological structures which form the insect's visual system. The models for the processes which take place in an insect's lamina, medulla and lobula and are responsible for contrast enhancement, motion detection and collision detection are used to build a system which allows the drone to navigate an enclosed testing environment without collision. These models are implemented and tested in iqr, a spiking neural network simulator able to interface with external hardware. Additional models for colour extraction and tracking using spiking neural networks are also developed and tested using iqr. The models are combined to form two systems. The first of these systems uses the models from the insect's visual system to create a system that allows the drone to navigate a constrained testing environment without collision. The second system uses the models developed for colour tracking and following to allow the drone to follow a colour object that appears in its visual environment. The results show that the system developed for autonomous navigation works successfully every time and that the system developed for tracking a coloured object worked as long as the coloured object is present in the system's field of view. The results show promise for the use of biologically inspired models for autonomous control of vehicles, especially now with the advances being made in parallel computing and neuromorphic computing. Suggestions for topics of investigation that would improve the systems created during this project as well as in this field in general are also given.

Uittreksel

Die huidige toename in die gewildheid en beskikbaarheid van mikrolugvaartuie noop die ondersoek na nuwe metodes van outonome beheer. Hierdie projek ondersoek die outonome beheer van 'n onbemande vliegtuig wat biologies geïnspireerde modelle gebruik. Die modelle wat ondersoek word is die verskeie neurologiese strukture wat deel uitmaak van 'n insek se visuele of sig-sisteem. Modelle om die prosesse wat in die lamina, medulla en lobula van 'n insek plaasvind en verantwoordelik is vir kontrasverhoging, beweging en botsing word gebruik om 'n sisteem te bou wat die vliegtuig toelaat om 'n geslote eksperimentele area te navigeer sonder dat botsings plaasvind. Die modelle is geïmplementeer en getoets in iqr, 'n "spiking neural network"-nabootser met koppelvlakke aan eksterne hardeware. Addisionele modelle vir kleurekstraksie en volging met die gebruik van "spiking neural networks" is ook ontwikkel en met behulp van iqr getoets. Die modelle is gekombineer om twee sisteme te vorm. Die eerste sisteem gebruik modelle van die insek se visuele sisteem om die vliegtuig toe te laat om in 'n beperkte eksperimentele omgewing te beweeg sonder om botsings toe te laat. Die tweede sisteem maak gebruik van die modelle ontwikkel vir kleuridentifikasie en volging om die vliegtuig toe te laat om 'n voorwerp te volg wat in sy sigveld verskyn. Die resultate toon aan dat die sisteem ontwikkel vir outonome navigering keer op keer suksesvol is. Die opsporing van 'n gekleurde voorwerp is suksesvol solank die gekleurde voorwerp in die sisteem se sigveld is. Die resultate toon die potensiaal vir die gebruik van biologies geïnspireerde modelle vir outonome beheer van voertuie, veral nou met parallelle vooruitgang in rekenaar en neuromorfiese rekenaar ontwikkeling. Voorstelle vir onderwerpe vir verdere ondersoek en wat ook die sisteem sal verbeter word gemaak.

Contents

Abstract	iii
Uittreksel	iv
List of Figures	viii
Nomenclature	xi
Acknowledgements	xiii
1 Introduction	1
1.1 Motivation and Topicality of This Work	1
1.2 Objectives of This Study	2
1.3 Contributions	3
1.4 Overview of This Work	4
2 Literature Review	10
2.1 Insect Behaviour and Vision	10
2.1.1 Introduction	10
2.1.2 The Insect Visual System	10
2.1.2.1 The Retina	10
2.1.2.2 The Lamina	11
2.1.2.3 The Medulla	11
2.1.2.4 The Lobula	11
2.2 Neural Networks	12
2.2.1 Introduction to Neural Networks	12
2.2.2 Biological Neural Networks	14
2.2.2.1 Learning in Biological Neural Networks: Spike Timing De- pendent Plasticity	14
2.2.2.2 Biological Neuron Models	18
2.2.3 Artificial Neural Networks	19
2.2.3.1 Network Topology	19
2.2.3.2 Learning Algorithm	20
2.2.3.3 Activation Function	20
3 Methods and Materials	23
3.1 iqr Neural Simulator	23
3.1.1 Introduction	23
3.1.2 Multi-Level Simulation Environment	23
3.1.2.1 The System Level	23
3.1.2.2 The Process Level	24
3.1.2.3 The Group Level	24

3.1.2.4	The Neuron Level	24
3.1.2.5	Synapses	25
3.1.3	Interface With External Devices	25
3.1.4	User Customisability	26
3.1.5	Real Time Processing	27
3.2	AR Drone 1.0	27
3.2.1	Introduction	27
3.2.2	Specification	27
3.2.3	Control of the AR.Drone 1.0	28
3.2.4	Receiving Video Input From the AR.Drone 1.0	28
3.2.5	Motivation for Choice	28
3.3	Ground Station	29
3.3.1	Introduction	29
3.3.2	Hardware	29
3.3.3	Software	29
3.3.4	Operation	29
3.4	Testing Area	30
3.4.1	Introduction	30
3.4.2	Physical Structure	30
3.4.3	Object Tracking	30
4	Biologically Inspired Models	34
4.1	Introduction	34
4.2	Image Preprocessing: The Lamina Model	34
4.2.1	Edge Detection	35
4.2.1.1	Introduction	35
4.2.1.2	Method	35
4.2.1.3	Results	38
4.2.2	Contrast Enhancement	40
4.2.2.1	Introduction	40
4.2.2.2	Method	40
4.2.2.3	Results	40
4.3	Course Stabilisation	42
4.3.1	Introduction	42
4.3.2	Elementary Motion Detectors	42
4.3.2.1	Introduction	42
4.3.2.2	Method	43
4.3.2.3	Results	45
4.3.3	HS/VS cells	45
4.3.3.1	Introduction	45
4.3.3.2	Method	45
4.3.3.3	Results	49
4.3.4	Speed Interpolation	54
4.3.4.1	Introduction	54
4.3.4.2	Method	55
4.3.4.3	Results	59
4.3.5	Winner Takes All neural model	62
4.3.5.1	Introduction	62
4.3.5.2	Method	62
4.3.5.3	Results	65
4.4	Collision Detection: The Lobula Giant Movement Detector	67
4.4.1	Introduction	67
4.4.2	Method	67

4.4.3	Results	68
4.5	Colour Extraction and Tracking	77
4.5.1	Introduction	77
4.5.2	Colour Extraction	78
4.5.2.1	Introduction	78
4.5.2.2	Method	78
4.5.2.3	Results	79
4.5.3	Colour Tracking	80
4.5.3.1	Introduction	80
4.5.3.2	Method	80
4.5.3.3	Results	81
5	Experimental Investigation	86
5.1	Autonomous Flight in a Constrained Environment	86
5.1.1	Introduction	86
5.1.2	Motivation for Undertaking	87
5.1.3	Experimental Setup	87
5.1.4	Results	90
5.1.5	Interpretation of Results	92
5.2	Colour Following	92
5.2.1	Introduction	92
5.2.2	Motivation for Undertaking	92
5.2.3	Experimental Setup	92
5.2.4	Results	95
5.2.5	Interpretation of Results	95
6	Conclusion and Recommendations	96
6.0.1	Introduction	96
6.0.2	Hardware	96
6.0.3	Autonomous Flight in a Constrained Environment Using Biological Models	97
6.0.4	Colour Tracking	97
A	Colour Tracking Experiment Results	99
	Bibliography	104

List of Figures

2.1	Hierarchical layout of the insects visual system taken from [1]	11
2.2	Basic ANN model	13
2.3	Biological neuron	14
2.4	(a) Excitatory excitation (b) Inhibitory excitation	15
2.5	Spike creation process.	15
2.6	Types of summation: (a) No summation, (b) Temporal summation, (c) Spatial summation	16
2.7	Typical STDP learning window	17
2.8	Types of weight bounding.	17
2.9	Integrate-and-fire neuron model	18
2.10	Leaky integrate-and-fire neuron model	19
2.11	Process of computing activation function value $f(net)$	21
2.12	(a) Activation function of type 1 ANN (b) Activation of type 2 ANN	21
2.13	Activation function of a type 3 ANN	22
3.1	The structure of the multi-level simulation environment of the <code>iqr</code> neural simulator	24
3.2	a) Using modules to receive data from an external device to a neuron in <code>iqr</code> . b) Using modules to have a neuron send commands to an external device	26
3.3	The distributed architecture which allows <code>iqr</code> to be run in real time	27
3.4	Parrot AR.Drone 1.0	28
3.5	The sequence diagram of the manner in which the AR.Drone 1.0, the ground station and <code>iqr</code> communicate during the startup and during every working cycle of the simulation	31
3.6	Object tracking performed by the method of sequential images applied to a red ball that is thrown upwards against a white background.	32
3.7	Method of sequential images applied to test video	32
4.1	The hierarchical structure of retina and lamina models and their connection topology implemented in the neural simulator <code>iqr</code>	35
4.2	Difference of Gaussian kernel in red with the two Gaussians it is comprised of, $f(\mu, \sigma_1, x)$ and $f(\mu, \sigma_2, x)$ in blue and green respectively	36
4.3	Centre inhibition, surrounding excitation edge detection topology mapping	37
4.4	Centre inhibition, surrounding excitation edge detection kernel	37
4.5	<code>iqr</code> implementation of Edge Detection Model	38
4.6	Environments that the various biological models were tested	39
4.7	Output of edge detection model when subjected to different environments	39
4.8	Effect of the threshold potential, v_i , on the contrast enhancement model.	41
4.9	Output of contrast enhancement model when subjected to different environments.	41
4.10	A visual representation of the Reichardt Correlator, also know as the Elementary Motion Detector (EMD), with both the preferred and null branches indicated	44
4.11	<code>iqr</code> implementation of the system used to test the elementary motion detectors	46

4.12	Conceptual illustration of the HS/VS cells of the fly. The original image has been shifted to the left making the array of EMDs sensitive to motion from right to left. The response of the array of EMDs is then integrated by the HS/VS cell to give the whole field motion.	47
4.13	HS/VS Cell model that has been modified to give a normalised output based on the amount of optic flow that has been detected	50
4.14	iqr implementation of the HS/VS cells. The output from the lamina model, the three arrays of EMDs tuned to different speeds and the HS/VS cell model are shown. Red connections indicate excitatory synapses, blue connections indicated inhibitory synapses and green connections indicate modulatory synapses	51
4.15	Membrane potential of the different populations of HS/VS cells when rotated at different speeds.	51
4.16	Membrane potential of HS/VS cells tuned to be selective to different speeds plotted against angular speed	52
4.17	Different environments in which the HS/Vs cells was tested to determine the environment's effect on its output	53
4.18	The effect of the environment on the HS/VS cell's performance	53
4.19	Membrane potential of the HS/VS cell when neurons in the EMD array have failed	54
4.20	Membrane potential of the HS/VS cell when neurons in the preferred branch of the EMDs in the EMD array have failed	55
4.21	Membrane potential of the HS/VS cell when neurons in the null branch of the EMDs in the EMD array have failed	55
4.22	Interpolation mechanism which encodes the angular speed as the angle of a vector	57
4.23	Interpolation mechanism using projection vectors representing a known angular speeds.	58
4.24	iqr System to test interpolation model using user defined inputs	60
4.25	(a) Connection topology of the speed interpolation model. (b) Synaptic weights of speed interpolation model.	60
4.26	Inputs and outputs of the speed interpolation model implemented in the iqr neural network simulator.	61
4.27	Response of the speed interpolation model over different speeds	62
4.28	Conceptual representation of the working of the Winner Take All Model with conversion of the output to be represented by a neuron which emits a spike	64
4.29	iqr implementation of the WTA model with modifications for representing the maximum and minimum values as a neuron that spikes and for implementation on an AR.Drone 1.0.	65
4.30	The iqr system used to test the Winner Takes All model. Red and blue connections indicate excitatory and inhibitory synapses respectively	66
4.31	Results of testing the WTA model under various inputs	67
4.32	Structure of the model describing the LGMD	69
4.33	Output of the LGMD model for different membrane persistence values of the HS/VS cell while on a collision course at $0.25m/s$	70
4.34	Output of the LGMD model for different membrane persistence values of the HS/VS cell while on a collision course at $0.5m/s$	71
4.35	Output of the LGMD model for different membrane persistence values of the HS/VS cell while on a collision course at $0.75m/s$	71
4.36	Output of the LGMD model for different membrane persistence values of the HS/VS cell while on a collision course at $1m/s$	72
4.37	Membrane voltage of the HS/VS cell of the LGMD model for different membrane persistence values while on a collision course at $0.25m/s$	72
4.38	Membrane voltage of the HS/VS cell of the LGMD model for different membrane persistence values while on a collision course at $0.5m/s$	73

4.39	Membrane voltage of the HS/VS cell of the LGMD model for different membrane persistence values while on a collision course at $0.75m/s$	73
4.40	Membrane voltage of the HS/VS cell of the LGMD model for different membrane persistence values while on a collision course at $1m/s$	74
4.41	Effect sensitivity of the EMDs on the membrane potential of the HS/VS cell of the LGMD	75
4.42	Effect of V_t on the LGMD model's performance	76
4.43	Effect of neural failure in the preferred branch of the EMDs in the EMD array of the LGMD	76
4.44	Effect of neural failure in the null branch of the EMDs in the EMD array of the LGMD	77
4.45	Effect of total failure of EMDs within the EMD array on the membrane potential of the HS/VS cell	78
4.46	Example of the RGB colour model	78
4.47	Effect of the <i>GainRatio</i> on the colour extraction model's performance	79
4.48	Effect of the threshold voltage on the colour extraction model's performance	80
4.49	Colour tracking model with no input to colour sensitive sections. Top: Neural activity. Bottom: Model output neuron voltages	82
4.50	Colour tracking model with 50% of input to colour sensitive section at the top of the image. Top: Neural activity. Bottom: Model output neuron voltage	83
4.51	Colour tracking model with 100% of input to colour sensitive section at the top of the image. Top: Neural activity. Bottom: Model output neuron voltage	84
4.52	Colour tracking model with a square 40 neurons in length and located in the centre of the frame given as the input. Top: Neural activity. Bottom: Model output neuron voltage	85
5.1	View angle represented by each neuron in the drone's field of view	88
5.2	System diagram for autonomous flight in a controlled environment experiment	91
5.3	Drone's flight paths during the experiments performed to investigate autonomous flight in a constrained environment	93
5.4	The full colour following system	94
5.5	Colour tracking experiment 1 results	95
A.1	Colour tracking experiment 2 results	99
A.2	Colour tracking experiment 3 results	100
A.3	Colour tracking experiment 4 results	100
A.4	Colour tracking experiment 5 results	101
A.5	Colour tracking experiment 6 results	101
A.6	Colour tracking experiment 7 results	102
A.7	Colour tracking experiment 8 results	102
A.8	Colour tracking experiment 9 results	103
A.9	Colour tracking experiment 10 results	103

Nomenclature

Abbreviations and Acronyms

ANN	Artificial Neural Network
BNN	Biological Neural Network
EMD	Elementary Motion Detector
GPU	Graphics processing unit
HS/VS	Horizontal/Vertical System
I&F	Integrate and fire (neuron)
LI&F	Leaky integrate and fire (neuron)
LPTCs	Lobula Plate Tangential Cells
LTP	Long Term Potentiation
MAV	Micro aerial vehicle
SDK	Software Development Kit
STDP	Spike Timing Dependent Plasticity
UDP	User Datagram Protocol

Equation Constants and Variables

D	EMD spatial separation
δ	EMD time delay
$ExcitatoryGain$	The edge detection model's excitatory synapse's connection strength
$ExcitatoryWindow_{inner}$	The side length of the inner square group of presynaptic neurons which form the excitatory inputs to the edge detection model
$ExcitatoryWindow_{outer}$	The side length of the inner square group of presynaptic neurons which form the excitatory inputs to the edge detection model
$GainRatio$	The ratio between $ExcitatoryGain$ and $InhibitoryGain$
$InhibitoryGain$	The edge detection model's inhibitory synapse's connection strength

<i>InhibitoryWindow</i>	The side length of the square group of presynaptic neurons which form the inhibitory inputs to the edge detection model
<i>IncomingExcitation</i>	The total excitation coming from presynaptic inputs to a neuron
<i>IncomingInhibition</i>	The total inhibition coming from presynaptic inputs to a neuron
M_p	Membrane persistence of the neuron
$n(v_m(t))$	Neuron activity
v_m	Neuron membrane potential
v_t	Threshold voltage

Syntax and Style

\mathbf{x}	The vector \mathbf{x} (usually lowercase)
\mathbf{A}	The matrix \mathbf{A} (usually uppercase)

Acknowledgements

My parents, Marianna and Valdon Smith, for their love and support.

My supervisor, Thomas Jones, for his guidance.

Chapter 1

Introduction

1.1 Motivation and Topicality of This Work

Micro aerial vehicles (MAVs) have traditionally been used in military applications but technological advances and reduced production costs are making them increasingly accessible to the public. MAVs have many applications, such as aerial sports, photography, wildlife tracking, exploration of hazardous environments, search and rescue missions and recreation. MAVs have evolved from blimps, helicopters and aeroplanes to more advanced vehicles such as quad-, hexa- and octocopters (with 4, 6 and 8 propellers respectively) and more biologically-inspired vehicles resembling insects, called ornithopters.

The vehicles themselves are not the only hardware that has advanced; there has also been an explosion in sensor and wireless communication technology and in processing power. The improvement in sensor technology enables MAVs to take more precise, more accurate and more meaningful measurements of the world around them, resulting in greater functionality and applicability for a range of tasks. Improvements in wireless communication have enabled MAVs to transmit data over long distances. The increase in processing power allows more complex algorithms to be used, which significantly enhances the MAV's functionality.

In addition to these improvements in MAV hardware, there have been significant advancements in the field of machine learning. Amongst the most exciting of these advances concern the field of neural networks, a class of machine learning inspired by the workings of the brain. A neural network uses simple processing elements, called neurons, which are interconnected to allow the network to perform high level functions such as pattern recognition and classification. Neural networks operate in a parallel way, i.e. numerous basic calculations are performed simultaneously. Currently the limitation of implementing neural networks is that existing processors perform calculations sequentially rather than in parallel. However, there have been significant advances in implementing neural networks on graphics processing units (GPU) [2], neural processing chips such as IBM's TrueNorth, [3], as well as 3D FPGA technology [4]. These advancements to neural network-based systems have the potential to drive innovation in a variety of fields and encourage us to look at for ways of solving new and existing problems using this newly available technology. An additional advantage of neural network based-systems is that they are considerably more robust than traditional algorithmic solutions [5].

Technological advancements have also allowed for much more efficient and in-depth investigations of the biological world [6], [7]. Scientists have inserted probes into insects' brains and observed the activity patterns of neurons when subjected to different stimuli [8], [9], [10]. This has allowed models to be constructed that describe the functioning of different areas of the brain. Some of these models are extremely interesting when considering the control of an aerial vehicle. Flies, for example, are able to perform complex flight manoeuvres despite

the relatively small number of neurons in their brains. Millennia of evolution have refined the insect brain, resulting in computationally-efficient operations responsible for their behaviour. The technology that allows us to investigate the mechanisms behind these operations is now available. The thesis presented here addresses the operations and mechanisms behind an insect's flight behaviour.

1.2 Objectives of This Study

The primary aim of the research project undertaken for this thesis was to investigate the feasibility of using biologically-inspired models to control, in real time, the flight behaviour of a parrot AR.Drone 1.0. The system was constrained to use only the drone's front camera as the input to a series of processes that enable it to navigate inside the testing environment. The system was required to perform the following functions using biologically inspired processes:

- Navigate a constrained 6m x 6m testing area
- Detect and avoid collisions
- Fly in a straight line when not performing collision avoidance
- Operate in real time
- Use only the drone's front camera to control flight behaviour

In the course of the research, additional components were developed that were not in the scope of the original project. For instance, the knowledge and intuition gained while working on the biologically-inspired models of flight control was used to develop a system, employing a spiking neural network, able to track a coloured object. Although the specific structures within the network are not specifically described by particular biological structures per se, they do draw inspiration from them.

The system that tracks the coloured object was required to perform the following functions:

- Track a red object which appears within the AR.Drone 1.0s field of view over an S-shaped path
- Maintain a minimum following distance from the object (based on the objects size in the drone's field of view)
- Operate in real time
- Use only the drone's front camera to control flight behaviour

It must be emphasised that the research project aimed only at controlling the flight *behaviour* of the drone, i.e. colour tracking, flying in a straight line, collision detection and avoidance, not the control system responsible for the roll, pitch, yaw and altitude of the drone.

1.3 Contributions

Throughout this research various contributions to the relevant academic field were made. These contributions can be separated into three classes, hardware, software and implementation.

The hardware contributions consist of a 6 x 6m testing area that was developed and can be seen in Section 3.4. The testing area allows experiments to be performed in a controlled environment and gives a way to track the drone during these experiments.

For the software contributions an iqr module was written which allows the Parrot AR.Drone 1.0 to interface with the iqr neural network simulator. Additionally user defined neurons were written for iqr which allow the division of the neurons membrane potential by the incoming signal from modulatory synapses.

The first of implementation contributions of this research is the implementation of the various neural models in the iqr neural simulator. The implementation of these neural models can be seen in Chapter 4. Furthermore these models were used to build a system which allows the drone to navigate the testing environment without collision using only video input from the front camera, which can be seen in Section 5.1.

Throughout this research various novel contributions were also made. These novel contributions can be separated into three classes. The first class of novel contributions are various improvements and simplifications that were made to the existing models and neural network connection topologies that describe the various functions of the insect brain. The improvements/simplifications are:

- Simplification of the difference of Gaussians algorithm for implementation in a spiking neural network, section 4.2.1.2
- Modification of the model of contrast enhancement performed in the lamina to simplify computation and make the model better suited to robotics applications, section 4.2.2
- A normalisation mechanism for the HS/VS cell model making it more suited for robotics applications, section 4.3.3.2
- Conversion of the output of the neural model of the winner takes all operation to be represented by a neuron firing 4.3.5.2.

The second class of the novel contributions that were made are the development of new neural models. The first novel contribution in this class is the colour-tracking mode, a neural structure designed to identify objects of a specific colour (in this case red) from the incoming video stream and to convert the colour's presence to neural activity. This model can be seen in section 4.5.2.

The second contribution made in the colour tracking system is a neural structure able to detect the percentage of the coloured object present in different segments of the visual field, and can be seen in section 4.5.3. The model draws inspiration from the HS/VS cells of an insect's brain and is responsible for keeping the object that is being tracked in the centre of the drone's field of vision, while maintain a following distance that is dependent on the size of the tracked object.

The last class of novel contributions is the development of a colour-following system which makes use of neural models from the second class of contributions. This system allows the drone to track a colour object which appears in its visual field while maintaining a constant following distance, which based on the perceived size of the object in the visual field. This system can be seen in section 5.2.

1.4 Overview of This Work

The main objective of this study was to investigate whether the biological models developed to describe various functions which take place within specific regions of an insect's brain, particularly those responsible for the insect's movement and flight behaviour, can be used to control a drone. The drone controlled by these biological models must be able fly around in a constrained environment for a fixed amount of time, maintaining a straight flight path where applicable and without crashing into the walls of the system. Using the knowledge gained in this investigation, a method for colour-tracking using purpose-specific neural structures was developed. A literature review on the topics addressed in this study is provided in Chapter 2. The review expands on the two main foci of the study; (1) insect vision and flight behaviour and (2) the role of neural networks in vision and flight behaviour.

The section on insect vision and behaviour, section 2.1, comprises four subsections, each dealing with a particular hierarchical structure in the insect's vision system. The first subsection, 2.1.2.1 concerns the retina, responsible for converting light into electrical signals in the brain. The next, 2.1.2.2, concerns the lamina, a structure thought to be responsible for contrast enhancement and logarithmic compression of the signal from the retina. Subsection 2.1.2.3 deals with the medulla, the brain structure responsible for key stages of local motion detection, and lastly subsection 2.1.2.4 deals with the lobula, a structure responsible for higher level motion functions. The functions of interest performed in the lobula concern course stabilisation in the horizontal and vertical planes as well as collision avoidance. Two structures in the lobula are thought to be responsible for these functions. The horizontal and vertical cells (HS/VS cells) are thought to be responsible for course stabilization, by integrating the outputs from the neurons in the medulla that detect local motion, causing sensitivity to motion in the whole visual field. The Lobula Giant Motion Detector (LGMD) is responsible for detecting an impending collision.

The next section of the literature review, section 2.2, discusses neural networks. Neural networks are information-processing structures, inspired by the manner in which the brain functions. This section is also divided into two subsections, the first dealing with biological neural networks and the second with artificial neural networks.

The subsection on biological neural networks, section 2.2.2, provides an overview of the basic principles of the manner in which the brain, and the neurons that comprise it, operate. The subsection on biological neural networks is divided into two parts. First part 2.2.2.1, explains spike timing dependent plasticity, the method by which many interconnected neurons are able to learn. The second part examines the various models that describe the behaviour of a biological neuron. The models described are (1) the integrate-and-fire neuron, which considers the neuron as a capacitor that unleashes a current spike when the neuron's membrane potential reaches a threshold value, (2) the leaky integrate-and-fire neuron, which models the neuron as a capacitor and resistor in parallel that also releases a current spike when the voltage of the neuron reaches a threshold value, but allows and corrects for the subsequent decay of the neuron's membrane voltage.

In section 2.2.3 a review of artificial neural networks is given. The section starts with a short overview of the history and basic operation of ANNs, and is followed by three subsections, dealing with different aspects of the artificial network. The first, section 2.2.3.1, describes network topology, the layout and interconnection of neurons in the network. The types of problems a network can be taught to solve, based on its topology and some basic guidelines that must be followed when choosing the network structure, are mentioned. The next subsection, section 2.2.3.2, deals with the learning algorithm of the network. The learning algorithm of an ANN defines how the synaptic weights between neurons are modified based on the incoming signal and, in some cases, also the desired output of the network. Specific learning algorithms are not described but the overarching principles behind the learning algorithms for supervised, unsupervised and reinforcement learning are mentioned. In the

final subsection, section 2.2.3.3, the activation function of the ANNs are discussed. The activation function determines how a neuron within the network responds to an incoming stimulus. Binary, continuous and time dependent activation functions and their role in classifying the type of network are discussed.

Chapter 3 details the software, hardware and testing area used in this study. The chapter comprises four sections. The first, section 3.1, outlines iqr, the neural simulator used to run the biological models investigated in the study and to communicate with the drone. The iqr's multi-level simulation structure is explained, including each level's function and how they interact with one another. The way the iqr neural simulator interfaces with external devices, such as robots, through the process level is then briefly described. Finally an explanation is given on the manner in which iqr's distributed architecture allows it to run simultaneously on multiple computers giving it the capability to perform real time processing of complex neural systems.

Chapter 3, section 3.2, introduces the AR.Drone 1.0 used in this investigation to determine whether biological models for flight control can be used for autonomous control of a drone. The section starts off by detailing the drone's hardware, software and physical specifications. Wireless control by sending UDP strings with control instructions to a specific port on the drone, and processes by which the drone's video stream is received and decoded, are described. Lastly, motivation for choosing the AR.Drone 1.0 for this investigation is given.

The third section of chapter 3, section 3.3, details the hardware used as a ground station in the investigation. The ground station's purpose was to run the iqr neural simulator, receive video input from the AR.Drone 1.0 and send UDP control strings to the drone based on the output of the iqr neural simulator. The hardware and software, which were a mid 2012 Apple Macbook Pro running version 14.04 of Ubuntu, a Linux based operating system, are detailed. The manner in which the drone and the base station operate together through an ad hoc Wi-Fi network, and the way in which the signals are converted between data string and neural activity, is described.

The last section of chapter 3, section 3.4, describes the size of, and visual surroundings of, the testing area used in the study. It is explained how the drone's flight path can be tracked within the testing area by applying a tracking algorithm to the video information received from a camera mounted on the ceiling of the testing area.

In chapter 4 the biological models that were investigated for the use of flight control of the drone, are described. Additional models for colour extraction and tracking developed through the course of the investigation are also explained. This chapter is divided into 4 sections.

The first section describes the preprocessing in the lamina of the visual input received from the retina. The section starts by discussing how edge detection is performed using a traditional algorithm called the difference of Gaussians, and then describes modifications made in order to make it more computationally efficient and to allow it to be implemented via a neural network. The results can be seen in section 4.2.1.

Next, a model for contrast enhancement is described. The model allows the neuron's threshold potential to trigger only when a specified level of neural activity is attained. The implementation of this model can be seen in section 4.2.2. It was seen that this model is able to perform the necessary contrast enhancements as well as simplifying the system to allow for easier implementation of the subsequent models used for flight control and the implementation of these models for controlling robots

The next section in chapter 4, section 4.3, deals with the models used for course stabilisation, which is the ability of an insect to stay on course in the horizontal and vertical planes. The section starts with a brief discussion on the motivation behind the various models used for course stabilisation, as well as the manner in which these models interact

with one another. Each of these models used for course stabilisation is contained in its own subsection. The first subsection, subsection 4.3.2, discusses the elementary motion detector (EMD) model used to achieve course stabilization. The EMD is a structure able to detect motion on a neural level through the correlation between the signals from two neurons representing spatially separated photoreceptors. One of the signals is time-delayed, causing the EMD to be sensitive to motion at specific speeds. It is also shown how this model is implemented in the iqr neural simulator.

In the next subsection, subsection 4.3.3, the model of the horizontal and vertical system cells (HS/VIS cells) of the fly is described. The HS/VIS cells are responsible for detecting whole field motion in both the horizontal and vertical directions, similar to optic flow, in order to keep the insect flying in a straight line. The section starts with an existing model of the HS/VIS cell, which relies on the integration of an array of EMDs and is deficient in that its output depends on the size of the EMD array feeding input to the model. A modified model that draws inspiration from the locusts' lobula giant motion detector and uses a normalisation signal in an attempt to overcome this deficiency is introduced. However, since the modified model is also not wholly indifferent to the size of the EMD array, a further modification is described in which the normalized signal is partly a function of the activity of the neurons supplying input to the EMD array.

The model's sensitivity to different speeds, based on the parameters of the EMDs input to the model, was first investigated. It was shown that even though the EMDs are sensitive to specific speeds, the HS/VIS cells, which integrate the responses of an array of EMDs, are able to detect motion at a wide range of speeds. The HS/VIS cells response is symmetric around the speed to which the EMDs are sensitive to, becoming lower the further from that speed. The result of this can be seen in figure 4.16. Next the effect of the visual environment on the performance of the HS/VIS cells is investigated. The conclusion from this is that the HS/VIS cells are able to detect motion regardless of the environment. The only visual factor influencing the output of the HS/VIS cells is the number of edges present in the plane which the cell are sensitive. This is illustrated by figure 4.18, with the environments that the HS/VIS cells were subjected to, being seen in figure 4.17.

Next, the effect of neuron failure in the array of EMDs which the HS/VIS cells integrate, is investigated. It was found that the output of the HS/VIS cells is directly proportional to the number of EMDs that have failed, as seen in figure 4.19. Lastly, the effect of neuron failure in the preferred and null branches of the EMD is investigated. It was found that, when neurons fail in a specific branch of the EMD, the output of the HS/VIS cells detecting motion in that branch is lower proportional to the amount of neurons that failed, as can be seen in figures 4.20 and 4.21.

The third subsection, subsection 4.3.4, deals with the speed interpolation model used to determine the actual speed of objects moving in the visual field based on the responses of the HS/VIS cells. The speed interpolation model works by encoding the speed into angles of polar co-ordinates. The output of the HS/VIS cells tuned for different speeds are projected onto projection vectors representing fixed angular speeds. The projection with the greatest magnitude represents the projection which is closest to the actual speed. Figure 4.26 shows the response of the model when it has been implemented in the iqr neural simulator and subjected to input stimuli representing different speeds. Figure 4.27 shows the implemented model response closely resembles the ideal model response.

The last subsection concerning course stabilisation, subsection 4.3.5, details the model used to perform a winner takes all (WTA) operation in order to select the input with the maximum value. The model used to perform the WTA operation makes use of a comparison matrix, a matrix in which each entry is one of the model's input subtracted from another. Using this technique the WTA operation is able to compute both the maximum and minimum input by looking at the columns and rows of this matrix. The mathematics behind the WTA

operation is described, as are the model's limitations and the modifications implemented to overcome those limitations. The WTA operation is implemented through the use of a group of neurons containing excitatory and inhibitory connections. This, however, poses a problem since the model selects the inputs with the minimum and maximum values through a zero entry in either a column or row, or in the case of a neural implementation, a neuron with no membrane activity. To combat this problem, a modification to convert the model output to a neuron with activity is proposed; the modification can be seen in section 4.3.5.2. The modified model is able to perform the WTA operation as desired.

Chapter 4, section 4.4, deals with the model of the Lobula Giant Motion Detector (LGMD), used for collision avoidance. First the existing model for the LGMD is explained. The LGMD model functions in a similar way to the HS/VS cells except that the array of EMDs that provide input to it are configured to detect expansion as opposed to sideways motion. A similar modification to that made on the model of the HS/VS, is also made to the LGMD to make its response invariant to the size of the EMD group supplying the model input, allowing it to be easily implementable and upgradeable when used for tasks such as control of robotics. The performance of the model and the factors that influence it (membrane persistence, spatial separation of the EMDs preferred and null branches, threshold potential of the integrate-and-fire neuron, neural failure) are investigated.

The effect of membrane persistence is investigated by measuring the LGMD model's output neurons membrane potential and its neural activity, by approaching a collision at different speeds. It is seen that the higher the membrane persistence, the earlier collisions are detected by the model. The results can be seen in figure 4.33- 4.40.

Next, the effect of spatial separation between the preferred and null branches of the EMDs which supply input to the LGMD is investigated. It is seen that the LGMD model's output depends on how fast an impending collision is being approached. Furthermore, it is seen that the speed to which the model is sensitive is dependent on the spatial separation between the branches of the EMDs. These results can be seen in figure 4.41.

The effect of the threshold potential of the integrate and fire neuron on the output of the LGMD model is investigated. This is done by approaching a collision with one of the walls of the testing area at a speed of $0.25m/s$. The membrane persistence of the HS/VS cell in the LGMD is set equal to 0 and can be seen in figure 4.37a. The threshold voltage of the integrate and fire neuron responsible for the models output varied from 0.1V to 0.9V in increments of 0.1V. From this test it was seen that the threshold voltage is another parameter influencing how early a collision is detected, as can be seen in figure 4.42. The ability to detect collisions earlier at low membrane potentials was seen to be poor and influenced by external factors.

Next, the effect of neuron failure in the preferred and null branches of the array of EMDs providing input to the LGMD model, is investigated. The results of this test show that failure in the EMDs causes the membrane voltage of the model's output neuron to retain the same shape but be offset by a constant. From figure 4.43 it is seen that failure in the preferred branch of the EMDs causes the membrane voltage of the models output neuron to be lower than expected, with the amount being lower in proportion to the amount of neurons that failed. In figure 4.44 it is seen that neuron failure in the null branch of the EMDs has the same effect, except that the it results in a higher membrane potential.

The last section of chapter 4, section 4.5, deals with the colour extraction and tracking model developed in this study. There are two subsections. First, in subsection 4.5 the colour tracking model is introduced. The model works by splitting the incoming RGB values from the camera into three groups of neurons. The colour to be tracked is then extracted by connecting the neural group representing it via excitatory synapses, and the neural groups representing the other two colours via inhibitory synapses, to the neural group representing the output of the model. It is shown that this model is able to perform colour extraction

effectively. The factors influencing colour extraction are investigated. First, the ratio of the strengths of the excitatory and inhibitory synapses is investigated. A lower ratio was found better able to extract the colour, as can be seen in figure 4.47. Next, the effect of the threshold voltage of the neural group representing the image on which colour extraction has been performed was investigated. The effect of the threshold voltage was to affect which shades of red were allowed through to the extracted image. Higher threshold voltages allow darker shades of red through, as can be seen in figure 4.48.

Subsection 4.5.3, describes the colour tracking model that works by dividing the visual field into thirds, both vertically and horizontally. The percentage of the coloured object within each of the sections is calculated using a normalisation technique. The output of the model is a group of neurons whose activity represents the percentage of the image in a particular section. The model is then tested using various different inputs.

Chapter 5 describes the experiments carried out to test the ability of the aforementioned models to enable the drone to fly autonomously. The chapter is divided into two subsections. The first, subsection 5.1, describes a test of the drone's ability to fly for five minutes in a constrained environment. First the motivation for undertaking such an experiment is discussed, then the experimental setup and the manner in which the system is constructed from the models introduced in sections 4.2, 4.3 and 4.4, are described, including how those models interact with one another to control flight behaviour. The results of the experiment are then given. The experiment was repeated 10 times and the results can be seen in figure 5.3. The results show that the use of biological models is a viable method for controlling the drone in the constrained environment.

Section 5.2 examines the colour tracking experiments. The drone was required to follow a red object over an S-shaped path in the tracking environment while the object was moved up and down sinusoidally. Motivation why such an experiment is of interest is given and it is described how the models introduced in chapter 4.5 were assembled to achieve a system able to perform colour tracking. The results of the experiments can be seen in Appendix A. They show that the models are able to perform colour tracking as long as the colour object remains within view of the camera.

Chapter 6 presents the main conclusions drawn from the results of the study, some recommendations for improvements on the systems that were built and suggestions for further research in the field. The chapter starts with section 6.0.2 giving some recommendations for improving the hardware used in the study. These include a better camera, implementing the processing of the neural system that controls the drone onto the drone itself, and the testing of different types of drones

Next in section 6.0.3, recommendations are made on the ways in which autonomous flight in a constrained environment system might be improved. The first suggestion is to implement a method that optimises the synaptic weights between the connections in the system so that it operates at optimal efficiency. It is suggested that the system can be made more robust by building redundancy into the models most sensitive to neuron failure, or by finding more robust models for functions that suffer most due to neuron failure. It is also suggested that a model for optimally searching through an environment, such as the Lévy Flight Pattern, be implemented.

In the last section of chapter 6, section 6.0.4, suggestions for improving the colour following system are made. The first recommendation is the implementation of a model that detects the coloured object, into the system. The next recommendation is to augment the system with a model that allows it to find the object just after it has moved out of view. A suggestion as to how this could be achieved is made. The third recommendation is given is for modifying the system so that it can track things other than colour, for example faces, and a recommendation on the way in which this could be achieved is also given. The last

recommendation is that a method be implemented to optimise the synaptic weights of the various models.

Chapter 2

Literature Review

2.1 Insect Behaviour and Vision

2.1.1 Introduction

Insects are capable of robust flight navigation, ranging from reflexive manoeuvres such as collision avoidance [11], [12], [13] and flight control [14], [15], [16], to navigational tasks such as path integration and landmark-based navigation [17], [18] [19]. What makes these manoeuvres remarkable is that they occur despite the very limited processing capability of the insect brain. For instance, the typical brain of a fly is only 1 mm^3 in size and consists of approximately 200000 neurons [20], two thirds of which is dedicated to processing visual information [21]. Furthermore, the flight manoeuvres performed by the insect are largely initiated by the input of only one sensor, its compound eye. Despite the limitations imposed on flying insects by their small sized brains and lack of multiple sensors, they have evolved extremely efficient methods of flight control. The physiological structures used by insects in their flight behaviour are discussed in the next sections.

2.1.2 The Insect Visual System

The insect visual system is a hierarchical structure consisting of four components [22]. The first is the retina, responsible for the conversion of light into an electrical signal in the nervous system. Next there are three neural structures (the lamina, the medulla and the lobula complex) connected to one another by a feedforward connection topology. The lobula complex is further divided into two sections, the lobula and the lobula plate. The insect visual system and brain are depicted in Figure 2.1.

2.1.2.1 The Retina

The insect eye is unlike that of a mammal, bird or reptile. It is unable to move or focus on particular objects. Instead, the insect's compound eye is made of many ommatidia, arranged to give the insect a wide field of vision, with poor resolution but an ability to detect fast movements.

Each ommatidium contains retinula cells, also known as photoreceptors. A retinula cell is extremely sensitive to light, which it converts into an electrical signal. This conversion occurs in a structure within the retinula cell, the rhabdom, a small tube-like structure containing thousands of microvilli. When light strikes these microvilli an electrical impulse is created which is transmitted via axons connected to the base of rhabdoms and the lamina [23], [24].

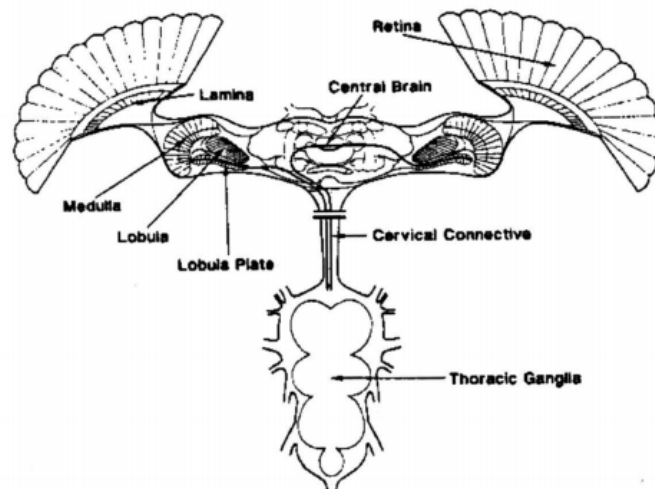


Figure 2.1 – Hierarchical layout of the insects visual system taken from [1]

2.1.2.2 The Lamina

The connection topology between the retina and the lamina is such that every ommatidium within the retina corresponds to one neuron in the lamina [25]. The lamina is where the first stage of processing of the electrical signal from the retina takes place. This processing in the lamina has been shown to involve logarithmic compression and contrast enhancement [26].

2.1.2.3 The Medulla

Once the electrical signal representing the visual information has been processed in the lamina, it is transmitted via long visual fibres to the medulla. The medulla is a complex structure for which there is little physiological information. The finding that neurons within the medulla are responsible for key stages of motion detection suggests that the medulla is the first neural structure showing sensitivity to visual motion [27], [28], [29].

2.1.2.4 The Lobula

The lobula is connected to the medulla and is responsible for higher level functions, such as collision detection and whole field motion sensitivity [27]. A structure within the lobula, the lobula giant motion detector (LGMD), has been shown to be responsible for collision detection. Whole field motion detection is performed by cells in the Horizontal System and Vertical System (HS/Vs).

The Lobula Giant Motion Detector The LGMD was first identified in the locust eye as the structure sensitive to looming objects, i.e. objects on a collision course with the insect [12], [30], [31]. The LGMD exhibits a non-linear response to looming stimuli, once thought to be caused by neurons firing when an object is a certain distance from the insect. This supposition was due to the correlation between the LGMD's response and the approach of the object [11], [32], [33]. However, further research on the LGMD has shown that its non-linear response can be modelled as the integration of on/off neurons [12]. It is now known that the LGMD's response is an emergent property due to the integration of many motion sensitive cells [34].

The nature of the LGMD response under different conditions has been extensively studied. The conditions include the shape and texture of the approaching object and the speed and angle of its approach. The response of the LGMD was shown to be independent of all these factors. A mathematical model has been developed to describe the output of the LGMD as the product of the object's angular velocity, $\dot{\theta}$ and its angular size, θ [35], [36]. Output of the model gives the firing rate, f , of the LGMD's output neuron and can be seen in equation 2.1.1 with α given in Equation 2.1.2 and $\theta_{threshold}$ being a species dependent parameter.

$$f^{-1} = e^{\log(\dot{\theta}) - \alpha\theta} \quad (2.1.1)$$

$$\alpha = 2 \cdot \tan\left(\frac{2}{\theta_{threshold}}\right) \quad (2.1.2)$$

The Horizontal and Vertical System Within the lobula plate are cells called lobula plate tangential cells (LPTCs), responsible for integrating the response of the neurons in the medulla to give whole field movement sensitivity [37], [16]. The input neurons whose responses are integrated by the LPTCs are arranged in a retinotopic fashion. The exact mapping and preferred direction of motion allows the LPTCs to be sensitive to motion in different directions and speeds. Vertical System (VS) cells are the LPTCs responsible for detecting upwards and downwards motion. Horizontal System (HS) cells are the LPTCs responsible for detecting sideways motion. Within each hemisphere of an insect's brain there are three HS cells and 11 VS cells, responsible for detecting motion of different speeds. While each HS/VS cell has its own receptive field, which covers different and often overlapping regions of the insect's visual field, the combination of these HS and VS cells covers the entire field of vision [37], [38], [39].

The output of the HS/VS cells conveys information about the motion of the visual environment of the insect and is fed to structures which control the insect's motor neurons responsible for the required flight manoeuvres [29].

2.2 Neural Networks

2.2.1 Introduction to Neural Networks

A neural network is a distributed computing paradigm which aims to mimic the decision making processes which take place in the brain. The network itself consists of two main components, namely neurons, which form the basic information processors of the network and synapses, which are the connections between neurons. The network can be trained to perform different functions by the adjustment of the strength of the synapses through a process called synaptic plasticity. This type of training corresponds to learning new information about specific data based on one's experiences or learning how perform a specific task.

An example of an artificial neural network can be seen in figure 2.2. This example is of a network with a feed-forward topology which consists of an input layer of neurons, N^{in} , a hidden layer of neurons, N^h , and an output layer of neurons, N^o . Making a biological analogy, the input layer can be viewed as afferent neurons (i.e. senses) that send information to the integrating centre (central nervous system) of the network. The input layer supplies the network with information from the outside world which can then be further processed by the network. This processing is done by the hidden layer of neurons which in the biological analogy would be the central nervous system. This layer allows for the projection of data into higher dimensions allowing non-linearly separable data to be separated based on features of

the data. The output layer takes the information produced by the hidden layer and further processes it into a form which can be used to perform the desired function of the network. Continuing with the biological analogy, the output layer is analogous to motor neuron output signals that control muscle movement based on incoming sensory data compared to past experience of similar data.

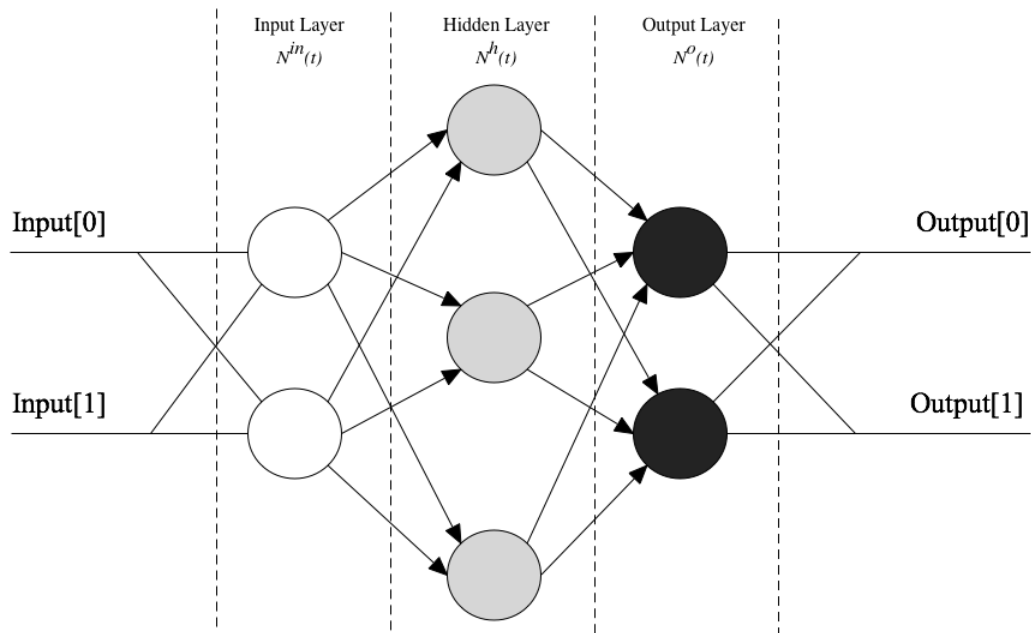


Figure 2.2 – Basic ANN model

There are many different neural network models but they can be divided into two main categories: biological neural networks (BNNs) and artificial neural networks (ANNs). Biological neural networks are those found within biological organisms with a brain. The models describing BNNs are often complex taking into account many physiological factors such as ion concentration in the synaptic fluid. This type of neural network is discussed in greater depth in Section 2.2.2.

Artificial neural networks can be found in a few different forms with the main categorisation between them being the learning algorithm. There are three types of learning algorithms that define ANNs, they are supervised, unsupervised and reinforcement learning. Models describing ANNs are more abstracted and simplified than their biological counterparts and will be discussed in more detail in Section 2.2.3.

In the following sections first the history of the spiking neural networks (SNN) will be examined, discussing the history of its development, the most common neuron models used to describe the network's operation and the learning within these networks. Next, artificial neural networks are discussed, highlighting the development and refinement of the artificial neural models, the network topology, the training (learning algorithms) of these networks and the various types of activation functions as well as the classification of the type of neural network based upon these activation functions.

2.2.2 Biological Neural Networks

Biological neural networks exist within the brains of animals and form their core information processing structures. A biological neuron is made up of four main components: the soma, the dendrites, the axon and the synapses. These components can be seen in figure 2.3. The functioning of the neuron was first described in the Nobel Prize winning work of Hodgkin and Huxley [40]. While the model described by Hodgkin and Huxley is rather complicated and takes into account many biophysical phenomena such as nerve currents produced due to the sodium and potassium conductances at the synapse of the neuron, a short overview of the underlying principle will be given here.

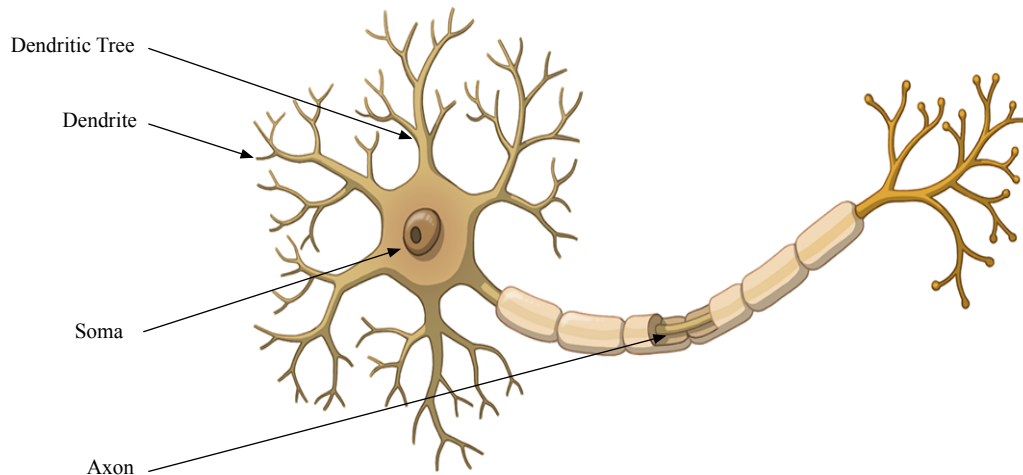


Figure 2.3 – Biological neuron

Incoming current spikes are transmitted from the axon of a presynaptic neuron across a synapse to a dendrite of a postsynaptic neuron. There are two types of synapses that exist within BNNs. These are excitatory and inhibitory synapses. The incoming current spike then causes the postsynaptic neuron's cell body to become potentiated or depressed depending on the type of synapse. Excitatory synapses potentiate the membrane potential whereas inhibitory synapses depress the membrane potential. Examples of excitatory and inhibitory responses can be seen in figure 2.4a and 2.4b respectively. The extent of the potentiation or depression is based on the strength of the synaptic connection between the pre- and post-synaptic neurons. When the membrane potential reaches a certain threshold the soma of the postsynaptic neuron produces a current spike which is propagated along its axon, across a synapse and to the dendrites of another neuron or an effector (e.g a muscle). The membrane potential always decays over time (only short current pulses cause it to be potentiated). However, once the neuron has emitted a spike there is a period afterwards where it cannot fire regardless of incoming current spikes. This period void of the ability to fire is called the refractory period. The entire process is can be seen in figure 2.5.

2.2.2.1 Learning in Biological Neural Networks: Spike Timing Dependent Plasticity

Spike Timing Dependent Plasticity (STDP) was discovered in 1998 by Bi and Poo [41] and is used to explain how synapse are strengthened or weakened depending on the relative timing between the incoming synaptic spike and the resulting postsynaptic spike. The use of spike

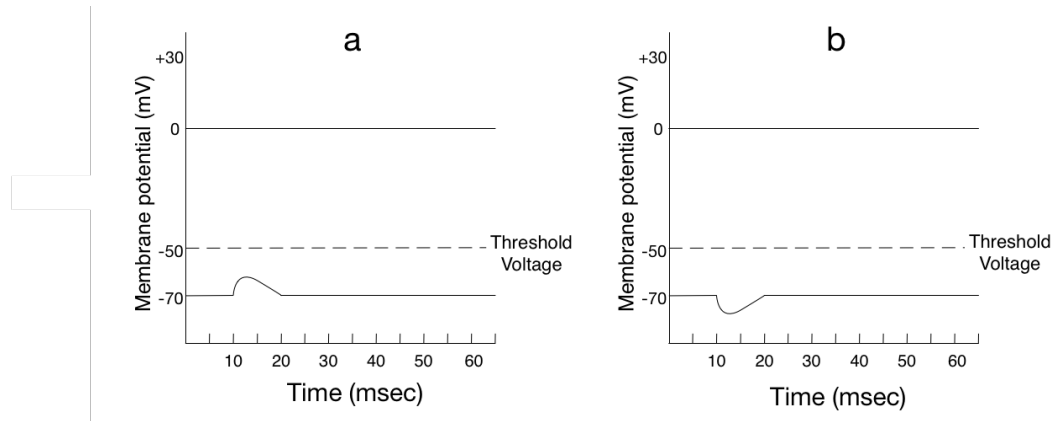


Figure 2.4 – (a) Excitatory excitation (b) Inhibitory excitation

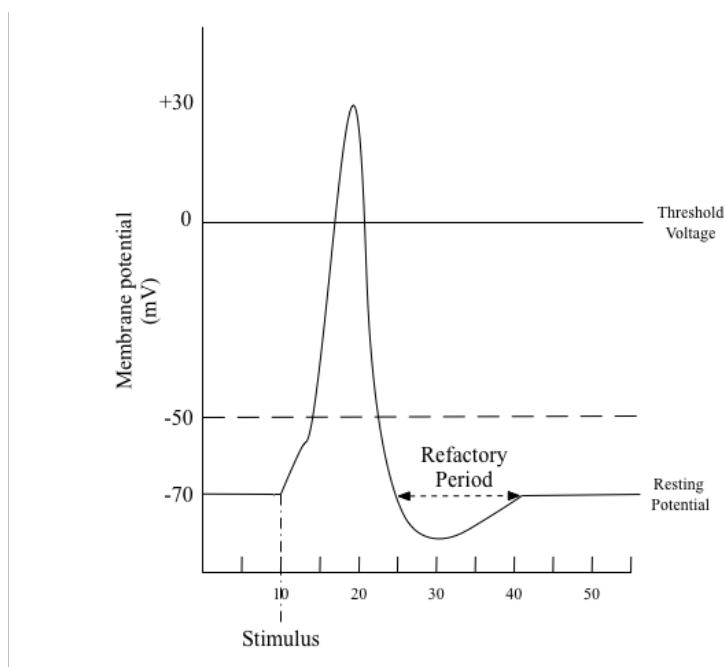


Figure 2.5 – Spike creation process.

encoding allows for much higher information content in the signals as both temporal and spatial summation can be used to encode information. Temporal summation refers to one neuron firing many times in quick succession to potentiate the membrane and cause the neuron to produce an output spike. Spatial summation refers to presynaptic spikes coming from neurons in different regions of the brain arriving and potentiating at the postsynaptic neuron at the same time, thus causing the neuron to fire. An example of temporal and spatial summation can be seen in figure 2.6.

Temporal summation allows the network to gain predictive power even though individual neurons have no knowledge of the network structure [42]. This is done via time correlation of the pre and postsynaptic spikes. If a presynaptic neuron repeatedly fires before the postsynaptic neuron, it can be assumed that there is a causal relationship between

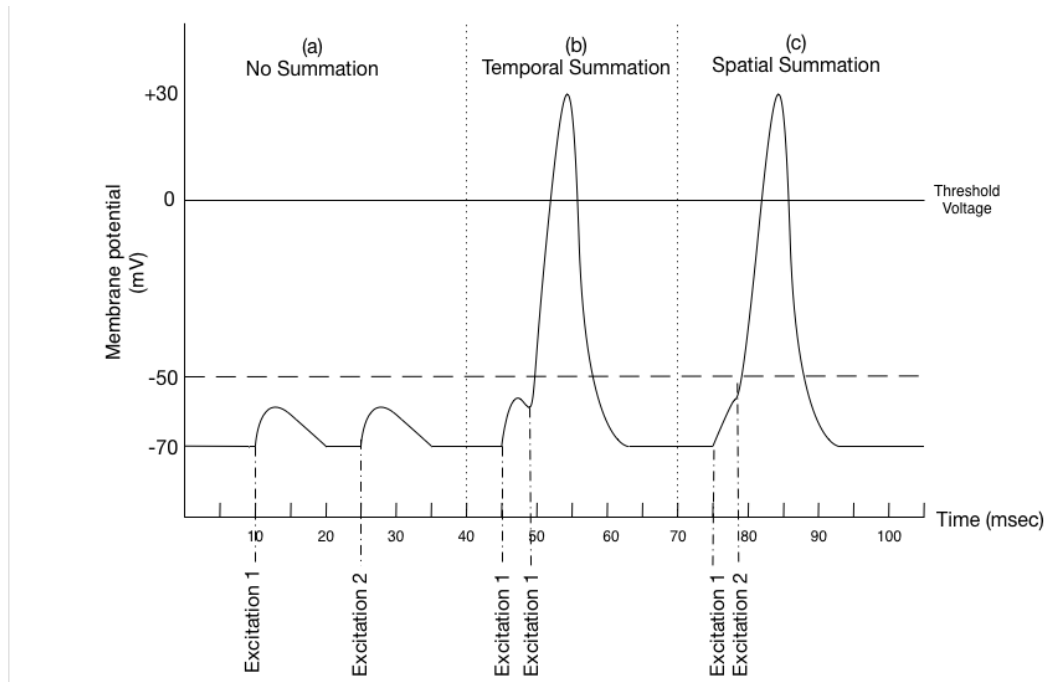


Figure 2.6 – Types of summation: (a) No summation, (b) Temporal summation, (c) Spatial summation

the firing of the two neurons. This leads to the synapse between the two neurons being strengthened. This strengthening of the synapse is called long term potentiation (LTP) and is a form of Hebbian learning. The converse is also true, i.e. if a postsynaptic neuron fires before one of its presynaptic input neurons, the synapse between them is weakened. This process is called long term depression (LTD) and is a form of Anti-Hebbian learning. The core of this mechanism has been well summed up by Norman Doidge in the phrase “*Cells that fire together, wire together*” [43]. The form of the learning window, i.e. how the synaptic strengths will be adjusted according to time, can be seen in figure 2.7 where Δt is defined as the time between post and presynaptic spikes, i.e. $\Delta t = t_{presynaptic} - t_{postsynaptic}$. It can be seen that the learning window is typically asymmetric around the time axis.

It is thought that STDP is responsible for many tasks within biological neural networks. It is thought, first of all, that STDP is responsible for the structural development of the connections and function regions of the brain [44]. Furthermore, it is thought that STDP causes latency reduction between temporally correlated neurons, i.e. the time between a presynaptic spike and postsynaptic spike is reduced if neurons routinely fire together [45]. STDP is also responsible for the learning of new information allowing for tasks such as binding features to objects and the ability to differentiate objects.

There have been some criticisms of the STDP model. The most common of these criticisms are:

- Only In-Vitro experiments have been performed
- Wide range of diversity within STDP
 - Some cases of the learning window being swapped around the time axis
 - STDP forms can differ between neurons

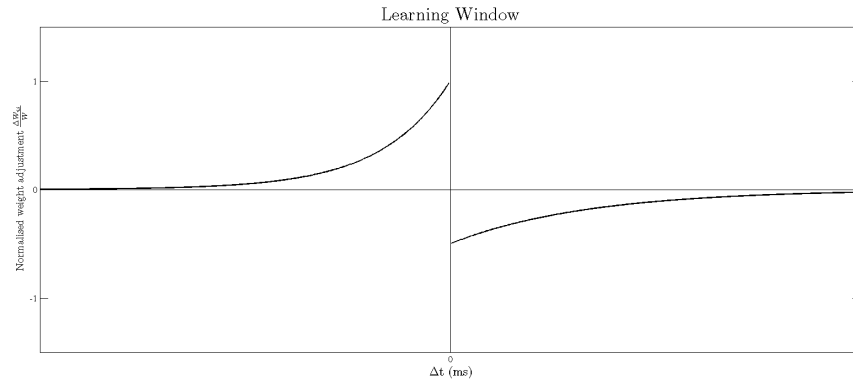


Figure 2.7 – Typical STDP learning window

- STDP can differ between dendrites of the same neuron (sometimes dependent on synaptic location)
- STDP may have different activation requirements (e.g at least two presynaptic neurons must have fired)

There are also many factors which regulate the STDP process. The first of these factors is the many biological neuromodulators which are released in the brain. These neuromodulators have been investigated and it has been found that they are able to influence the temporal characteristics of the STDP process [46]. Furthermore, the strength of the synapse only exists within certain biological bounds. An example of this is shown in figure 2.8 where it can be seen that the synaptic strength (synaptic weight) is limited between w_{min} and w_{max} . Finally it can be seen that STDP is regulated by the postsynaptic membrane voltage through an effect called Voltage Dependent Synaptic Plasticity.

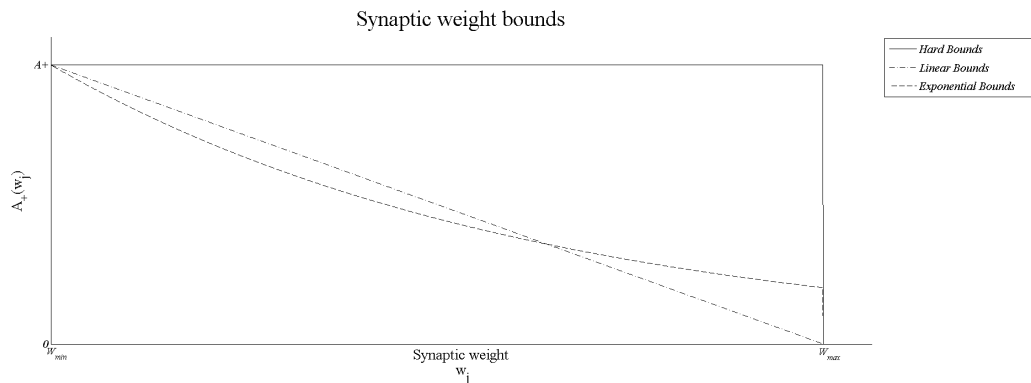


Figure 2.8 – Types of weight bounding.

Additionally, there is some debate on whether an actual biological neuron's synaptic weights work in continuous or discrete manner. Most of the existing models model neurons as continuous, however, there have been some experiments performed on the CA1 pyramidal cells in the hippocampus that suggest that synaptic weights take on discrete values [47], [48]. An advantage of the discrete weights of the synapses between neurons is that it guarantees long term stability of the synapses as show by Lisman [49]. However, there is also evidence that

indicates that synaptic weight changes may be continuous [50]. Additionally, the situation may not have a well defined boundary between continuous and discrete synaptic weights. It may be possible that presynaptic neurons are of one type whereas postsynaptic neurons are of another. Furthermore, the synaptic distance may play an important role as the distance determines the amount of dendritic filtering [51] that the incoming signal undergoes making it difficult to tell if the synaptic weights are continuous or discrete.

2.2.2.2 Biological Neuron Models

The Integrate-and-Fire Neuron One of the first models developed to describe the behaviour of a biological neuron was developed by Louis Lapicque in 1907 [52]. The integrate-and-fire model models the neuron as a capacitor, as seen in figure 2.9. The current flowing into the capacitor is equal to the current spikes received from other neurons through the neuron's dendrites. The membrane voltage of the neuron can then be calculated by integrating this current. This can be seen in equation 2.2.1, where V_m is the membrane potential, $I(t)$ is the current entering the neuron and C_m is the capacitance of the neuron.

$$I(t) = C_m \frac{dV_m(t)}{dt} \quad (2.2.1)$$

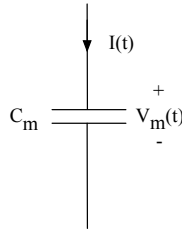


Figure 2.9 – Integrate-and-fire neuron model

Once the membrane voltage, V_m , reaches a threshold, V_t , the neuron emits a current spike defined by the Dirac-delta function. Once this spike is emitted the membrane potential of the neuron is reset.

It can be seen that the firing frequency of the neuron only depends on the current coming into the neuron. This behaviour, however, is not exhibited in real neurons as there is often a refractory period before the neuron is able to emit another spike. In order to combat this, Hill, in 1936, proposed a constraint on the firing rate, f , of the neuron [53]. This constraint on the firing rate of the neuron as a function of the incoming current can be seen in equation 2.2.2, with t_{ref} being the time delay of the refractory period.

$$f(I) = \frac{I}{C_m V_{th} + t_{ref} I} \quad (2.2.2)$$

The Leaky Integrate-and-Fire Neuron While the integrate-and-fire neuron is able to describe how the neuron's membrane voltage increases as a function of the incoming currents from presynaptic neurons that have fired, it is unable to describe how the neuron's membrane voltage decays over time. To combat this problem a new model, the leaky integrate-and-fire neuron (LI&F neuron), was developed. In this model a resistor was added in parallel to the capacitor in the traditional integrate-and-fire model, as seen in figure 2.10. This resistor allows the membrane voltage, which is modelled as the voltage over the capacitor,

to discharge through the resistor and decay over time. The mathematical formula that describes the leaky integrate-and-fire neuron in equation 2.2.3, with R_m being membrane resistance.

$$I(t) = C_m \frac{dV_m(t)}{dt} + \frac{V_m(t)}{R_m} \quad (2.2.3)$$

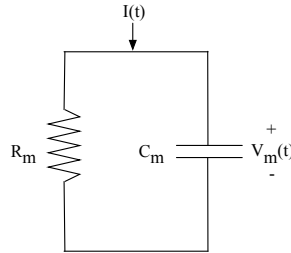


Figure 2.10 – Leaky integrate-and-fire neuron model

2.2.3 Artificial Neural Networks

Artificial neural networks are computing structures that attempt to make use of computing techniques similar to those of biological neural networks. The first computational model of neural networks was developed in 1943 by Warren McCulloch and Walter Pitt [54]. Artificial neural networks have similar structures to biological neurons, in that they contain a neuron which is responsible for the processing connections which either link neurons together or provides the output of the network (biologically these would be the dendrites and axon) as well as synaptic weights which model the connection strength between neurons. Learning is done by adapting the synaptic weights based on a chosen learning algorithm. This allows the network to develop an internal statistical model representing the input/output relationship of the data and training algorithm that has been used. This statistical model allows the network to gain a generalisation ability which allows it to handle inputs which have not yet been seen in a manner that is consistent with previous data. Artificial neural networks have three main properties that define them and give them their performance characteristics. These properties are the topology of the network, the learning algorithm used to train the network and the activation function of the neurons within the network.

2.2.3.1 Network Topology

Topology refers to the manner in which the network is structured, namely the way in which individual neurons are connected to one another and the size of the network layers. The connection topology of the network has a huge effect on the network's performance. It can be seen that the topology of the network plays a large role in the type of problems that can be able solved, for example a hidden layer of neurons is needed if a linearly non-separable problem is to be solved. It has been shown that it is best to choose the smallest possible number of neurons that allows the network to still perform its desired function [55]. The motivation for this choice is that it gives the network the best possible generalising abilities as well as reduces the time necessary to train the network. Furthermore ANNs with connections across layers have been shown to be easier to train as well as more powerful than their counterparts. Some of the common network topologies are:

- Feed forward networks
- Recurrent networks

2.2.3.2 Learning Algorithm

The learning algorithm is a set of rules which describes the manner in which synaptic weights are adjusted. There are three main forms of learning that are found in ANNs namely supervised, unsupervised learning and reinforcement learning.

Supervised learning refers to the method of applying a specific training signal to the network and comparing its output to the desired output for that specific training signal. An error function which compares the actual output to the desired output is then minimised by the adjustment of the synaptic weights within the network.

Unsupervised learning refers to the creating of a cost function that models the desired function of the network. Input data is then fed into the network and the cost function is then minimised and the synaptic weights are adjusted accordingly. This causes input data to form statistical clusters within the network relating to the specific input output relationships as defined by the cost function governing the synaptic weight adjustments.

Reinforcement learning is done by setting up a long term cost function. The system is then made to perform an action. An observation is made based on this action and the resulting output of the network is judged against the cost function. The cost function is then minimised by the adjustment of synaptic weights causing the network to learn what actions to take when faced with similar input data in future scenarios.

2.2.3.3 Activation Function

The activation function refers to the processing done by the neuron on the incoming signals. The form of the activation can be used to classify the network according to type. There are three types of networks, type 1, type 2 and type 3.

For type 1 and 2 networks the inputs to the neuron are multiplied by their corresponding synaptic weights w_{ij} and are then summed to give the net input to the network. This can be seen in equation 2.2.4.

$$net = \sum_{j=0}^n w_{ij}x_j \quad (2.2.4)$$

The result of this sum, net , is then put through an activation function which maps the summed input to the output of the neuron. The output of the activation which is limited to a finite range which is normally between -1 and 1, or between 0 and 1 for bipolar and unipolar activation functions respectively. This process can be seen in figure 2.11.

The type 1 networks refers to neurons with binary activation functions. Binary activation functions can only take on one of two discrete values, either 0 or 1 for unipolar binary activation functions or -1 and 1 for bipolar activation functions. An example of a bipolar binary activation function, with its equation described by equation 2.2.5, can be seen in figure 2.12a.

$$f(net) = \begin{cases} -1 & \text{for } net < 0 \\ 0 & \text{for } net = 0 \\ 1 & \text{for } net > 0 \end{cases} \quad (2.2.5)$$

Type 2 networks have continuous activation functions which are bound between the region [0,1] for unipolar activation functions and [-1,1] for bipolar activation functions. The most

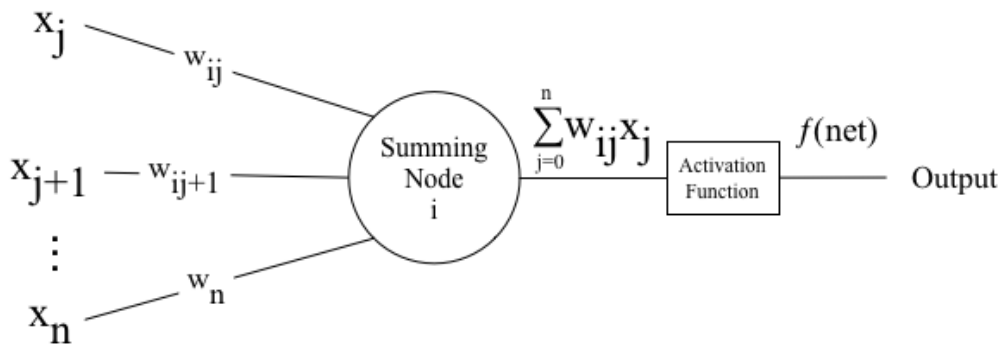


Figure 2.11 – Process of computing activation function value $f(net)$

common activation function for type 2 networks is the bipolar sigmoid function which is defined by equation 2.2.6. The bipolar sigmoid activation function with $\lambda=1$ can be seen in figure 2.12b.

$$f(net) = \frac{2}{1 + \exp(-\lambda net)} - 1 \tag{2.2.6}$$

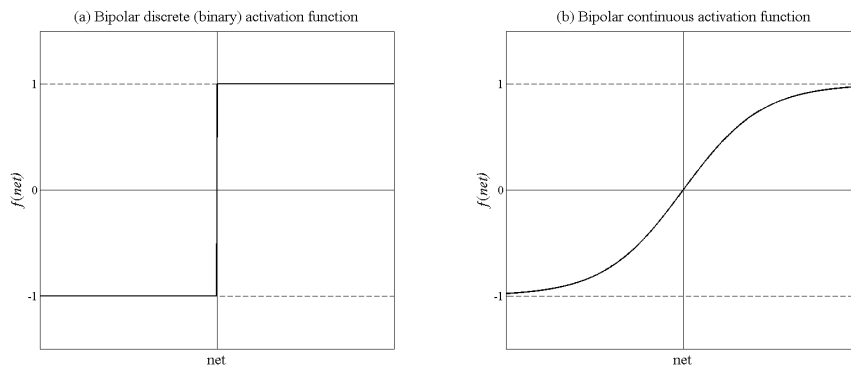


Figure 2.12 – (a) Activation function of type 1 ANN (b) Activation of type 2 ANN

Activation functions for type 3 networks are a little different than the activation functions for type 1 and 2 networks. Type 3 ANNs most closely resemble biological neural networks. For type 3 networks incoming spikes from presynaptic neurons cause the cell membrane to be potentiated by the amount defined by the synaptic weight between the pre and postsynaptic neurons. If enough presynaptic spikes arrive at the postsynaptic neuron causing the membrane potential to exceed the membrane threshold, an output spike is produced by the neuron. The membrane potential is also in a constant state of decay. This allows the network to make use of both temporal and spatial summation as a time dependence between presynaptic and postsynaptic neuron spikes now exists. This process is described in more detail in section 2.2.2.1. An example of this type of activation function can be seen in figure 2.13.

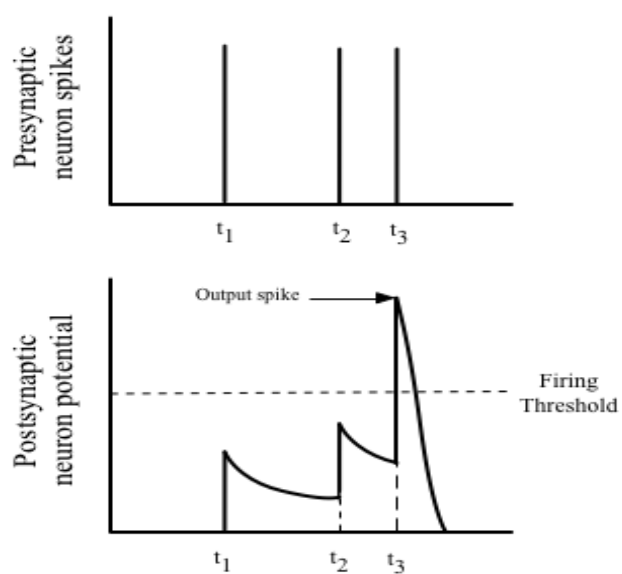


Figure 2.13 – Activation function of a type 3 ANN

Chapter 3

Methods and Materials

3.1 iqr Neural Simulator

3.1.1 Introduction

The brain is an extremely complex organ responsible for controlling the behaviour of the body based on external factors (e.g. the current environment) and internal factors (e.g. past experience). The brain's ability to do so arises from the structural organisation of the neurons and synapses contained within it. Understanding the brain's structural organisation will give us a much better understanding of the emergent properties of the structures contained within it. These emergent properties are, collectively, brain function. Using physiological data and/or phenomenological observations, models have been developed that describe brain functioning and organism behaviour. Simulating the brain at a neuronal level allows us to test the validity of these models.

For this project the `iqr` neural simulator for large scale neural systems [56] was used. The `iqr` neural simulator provides a platform that allows for the fast creation of neural systems that can interface with external devices, such as robots, through its graphical user interface and high level of user customisability.

The `iqr` neural network simulator was developed at the Institute of Neuroinformatics in Zurich. It is written in C++ and uses the Qt widget-set for the implementation of its GUI. Its development was done on the Linux operating system and is open source under the GNU public licence. The `iqr` simulator allows for the implementation of neural models describing behaviour, in real time and within a multi-level neuronal simulation environment. The simulations of a neural model can be interfaced with robots so that its ability to describe behaviour in the real world can be evaluated.

3.1.2 Multi-Level Simulation Environment

One of the main attractions of the `iqr` neural simulator is its multi-level simulation environment, comprising four levels. The levels are the system, the process, the group and the neuron. Each level is responsible for a different level of structural complexity in the brain. The structure of this multi-level simulation environment can be seen in figure 3.1.

3.1.2.1 The System Level

The system level of the `iqr` neural simulator is analogous to the brain. Each instance of `iqr` can only contain one system, which itself may contain many different processes. The system is responsible for the speed at which the simulation runs, which is dependent on the number of times a neuron can be updated per second.

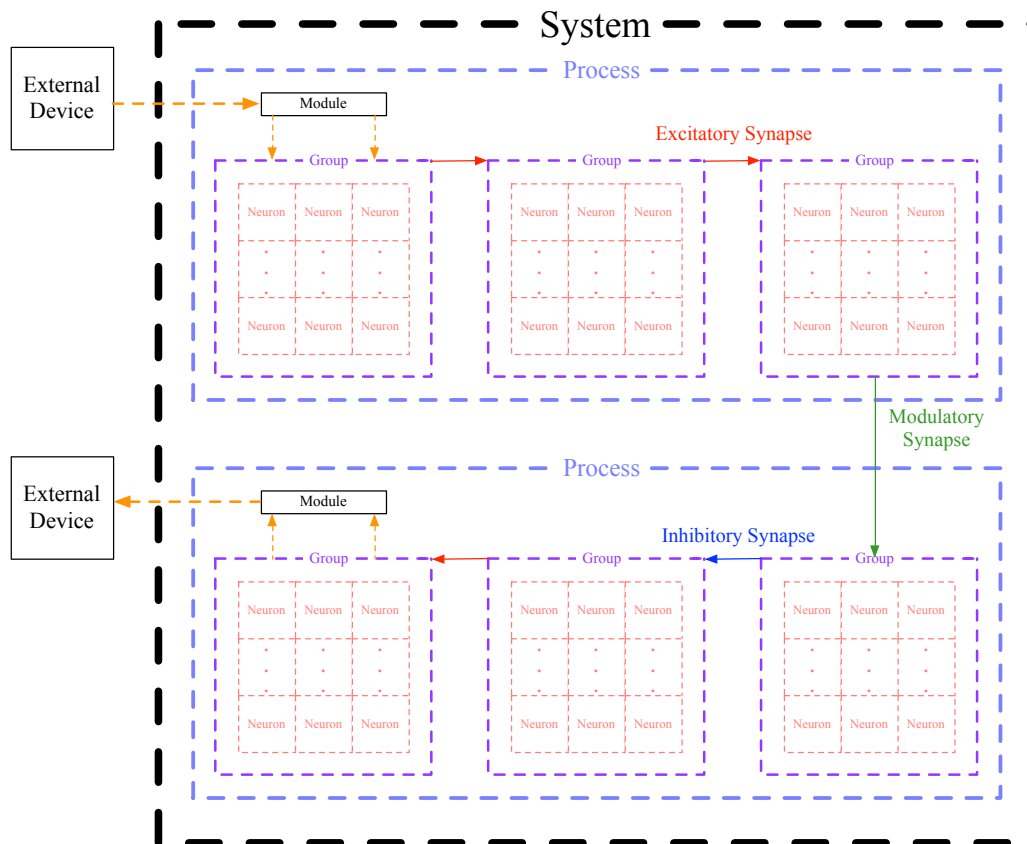


Figure 3.1 – The structure of the multi-level simulation environment of theiqr neural simulator

3.1.2.2 The Process Level

The process level of theiqr neural simulator is analogous to a specific brain structure responsible for a particular action. Many different processes may be contained in each system level of theiqr simulator.

3.1.2.3 The Group Level

Within each process neurons of the same type and with the same parameters are collected into groups. Neuron groups may either receive input from other neuron groups via synapses or they can receive and transmit data to and from external devices. This data transmission is done through the use of modules which will be discussed in section 3.1.3.

3.1.2.4 The Neuron Level

The neuron level is the most basic level within theiqr neural simulator. Neurons are contained in groups, which are large arrays of neurons of the same type. Neurons in different groups may be connected to one another through synapses. Neuron behaviour when receiving input from different types of synapses can be defined by the user.

There are two main neuron models which have been implemented iniqr and are of interest for this research; the linear threshold neuron model and the leaky integrate-and-fire neuron

model.

The Linear Threshold Neuron model in*iq*r The neural activity, $n(v_m(t))$, of a linear threshold is its membrane potential, v_m , if $v_m \geq v_t$ otherwise it is 0. In this equation v_t is the threshold value and v_m is the membrane persistence of the neuron. The equation defining v_m can be seen in equation 3.1.1, where *IncomingExcitation* and *IncomingInhibition* is the total excitation and inhibition from the neurons presynaptic inputs respectively, and M_p is the membrane persistence, which is a constant set by the user. It is worth noting that the time here, t , is discrete simulation time and thus $v_m(t-1)$ indicates the value of v_m one simulation cycle ago.

$$v_m(t) = \text{IncomingExcitation} - \text{IncomingInhibition} + M_p * v_m(t-1) \quad (3.1.1)$$

The Leaky Integrate-and-Fire Neuron model in*iq*r The neural activity, $n(v_m, t)$, of the leaky integrate-and-fire neuron implemented in*iq*r is given by equation 3.1.2, where v_m is the membrane potential defined in equation 3.1.1.

$$n(v_m(t)) = \begin{cases} 1 & \text{if } v_m(t) \geq v_t \\ 0 & \text{otherwise} \end{cases} \quad (3.1.2)$$

3.1.2.5 Synapses

Synapses in the*iq*r neural simulator are analogous to synapses in the brain. They connect different neuron groups, allowing the connection topology between the neurons of two different groups to be defined, as well as the signal propagation delay, in number of simulator cycles, between the pre and post synaptic neurons.

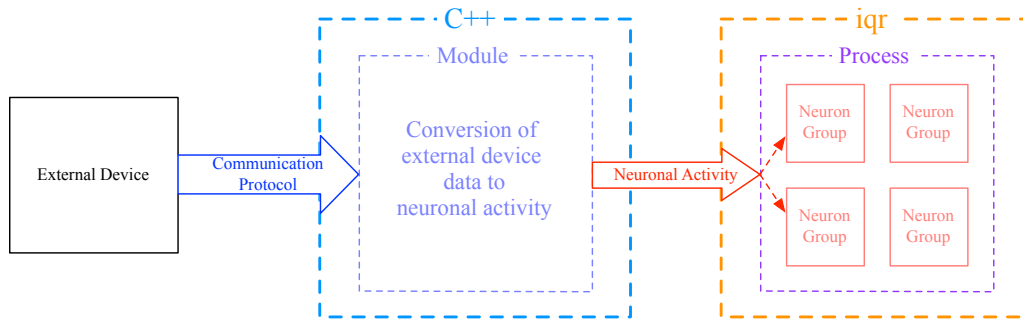
There are three types of synapses in the*iq*r neural simulator; excitatory, inhibitory and modulatory synapses. Excitatory synapses are indicated as red connections in*iq*r and for the default neuron types provided in the*iq*r neural simulator and cause the membrane potential of the input neuron to be multiplied by a synaptic weight and then added to the membrane potential of the target neuron. Inhibitory synapses are indicated by blue connections and perform the same function as excitatory synapses with the exception that the membrane activity of the presynaptic neuron is subtracted from, rather than added to, the membrane potential of the postsynaptic neuron. Modulatory synapses are indicated as green connections in*iq*r and do not have a predefined function. They are used to implement user-defined behaviour, for example multiplication and division.

3.1.3 Interface With External Devices

The*iq*r neural simulator allows interfacing with external devices such as cameras and robots. The interfacing is done through modules, which are plug-ins that allow data exchange between an external entity and a group or groups of neurons. Only one module is associated with a particular process, as described in section 3.1.2.2. There may be many different modules contained in the system, as defined in section 3.1.2.1, but each module will be associated with its own process.

Modules pass data to the neurons within the*iq*r neural simulator by reading the data from an external entity in the format transmitted by that entity. This data is then converted into neuron activity by the user-defined method which defines the module. This neuron activity is then passed to the group or groups of neurons within the process connected to the module. An example of this is reading in the red, green and blue values from a camera and then using a module to convert each pixel's red, green and blue pixel values into a membrane

a) Receiving Data From an External Device



b) Transmitting Data to an External Device

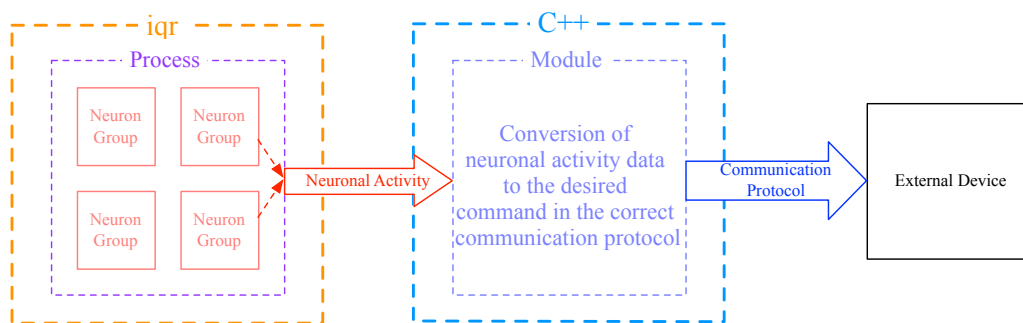


Figure 3.2 – a) Using modules to receive data from an external device to a neuron in *iqr*. b) Using modules to have a neuron send commands to an external device

potential value for the red, green and blue neuron groups within the processes connected via the modules. The process of receiving of data from an external device is shown in figure 3.2a.

In a similar manner, modules can be used to interface the neuron groups within a process with an external device. This is done by converting the activity of the neurons within a certain group into commands which can be sent and understood by the interfacing device. The interfacing process takes place within the module. The module first reads in the neuronal activity value of the connected neuron group or groups and then converts it into the desired command to be transmitted to the external device. The commands are sent through the correct transmission protocol (specified by the user when writing the C++ code defining the module) to the external device. An example of this would be using the module to read in the membrane potential value of the neuron group connected to it, converting that value to an angle, and then sending that angle via the appropriate communication protocol to the actuator. This process is shown in figure 3.2b.

3.1.4 User Customisability

An important feature of the *iqr* neural simulator is that users can write their own code for their intended project, including the definition of both neurons and modules. The definitions of the neurons used in this project are explained in section 3.1.2.4.

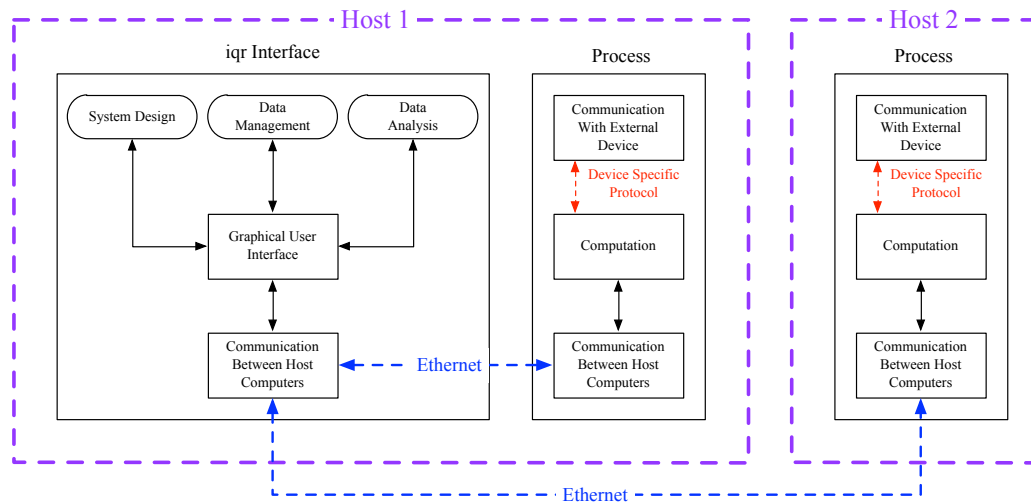


Figure 3.3 – The distributed architecture which allows iqr to be run in real time

3.1.5 Real Time Processing

An important feature of the iqr neural simulator is its ability to perform real time processing. This enables the models implemented in iqr to be used to control a quadrotor with fast dynamics. To perform this real time processing, a distributed architecture is used [57].

In this distributed architecture, each of the models is divided into functions that are implemented in processes explained in section 3.1.2.2. Each process can be assigned to run on different host computers, depending on their requirements or computational complexity. Once the calculations for a simulation cycle are completed, the results are transmitted between host computers through the use of Ethernet. This process can be seen in figure 3.3.

3.2 AR Drone 1.0

3.2.1 Introduction

The Parrot AR.Drone 1.0 was used [58] to test the applicability of some models describing the structures for autonomous flight in insect brains.. The drone 3.4 is a commercially available quadrotor released in 2010 by the French company Parrot.

3.2.2 Specification

The drone is powered by an ARM9 468MHz processor with 128Mbytes of DDR memory operating at 200MHz [58]. It runs the Beebox version of the Linux operating system. Communication between the drone and the device controlling it is through an on-board wireless Wi-Fi controller. The drone sets up an *ad-hoc* Wi-Fi network that can be connected to by an external device. Data (e.g. the video stream from the drone and movement commands) can be sent between the drone and the controlling device via User Datagram Protocol (UDP) strings.

The drone measured diagonally spans 57cm and, with its indoor hull, weighs 420 g. It is powered by 4 brushless DC motors which run at 35,000 rpm. It draws power from a 3 cell, 11.1V, 1000mAh lithium polymer battery with a discharge capacity of 10c.



Figure 3.4 – Parrot AR.Drone 1.0

The drone is equipped with various sensors: 3-axis accelerometer, a 2-axis gyrometer, a 1-axis yaw precision gyroscope and an ultrasound altimeter with a range of 6m and an emission frequency of 40kHz. The drone also has two cameras, one facing downward and the other frontward. The downward camera operates at 60FPS and has a viewing angle of 63° . It is used to control the position of the drone by computing the optical flow seen in the horizontal plane and sending the appropriate motor controls to compensate for any drift in the drone's position in the plane parallel to the ground. The frontward camera operates at 30 frames per second with a viewing angle of 93° and a resolution of 640x480.

3.2.3 Control of the AR.Drone 1.0

Port 5556 of the drone receives movement commands as User Datagram Protocol (UDP) strings. The UDP strings represent input values of between -1 and 1 which are mapped to the 4 degrees of freedom i.e roll, pitch, yaw and altitude.

3.2.4 Receiving Video Input From the AR.Drone 1.0

The drone sends an encoded video stream to the computer controlling it through UDP port 5555. The video stream is then decoded by the computer and is able to be displayed.

3.2.5 Motivation for Choice

The motivation behind the choice of the AR.Drone 1.0 as the platform for testing the models responsible for biological flight, was its low cost and commercial availability.

A significant advantage of the drone is that Parrot has released a software development kit, SDK, to the public. The SDK allows the user to create applications for controlling the drone. This enables the creation of an interface between the drone and the *iqr* neural simulator, section 3.1. The interface allows the *iqr* simulator to access the information coming from the drone's camera and to convert that information to neural activity within the simulator. The interface also allows control of the drone, using the activity of specific neurons implemented

within the `iqr` neural simulator, by sending a UDP string encoding the necessary roll, pitch and yaw angles as well as the altitude required for the necessary flight manoeuvres.

A further advantage of the drone is that, being a commercial product, the SDK is regularly updated and it has a large support community of programmers and hobbyists.

The fact that the drone is equipped with sensors allowing for internal control of its yaw, pitch, roll and altitude was another important factor in choosing it, as the robotics platform for testing the applicability of biologically inspired models to autonomous flight. This allowed the focus of the project to be on investigating the ability of biologically inspired models to be used to control the flight behaviour of a quadrotor, as opposed to developing a control system to control the quadrotor itself. The existence of such systems has biological analogue in small wing-like structures, called halteres, which function as a gyroscope and inform the insect about its body position during flight. Halteres fall outside the range of this project but more information on them may be found in [59].

3.3 Ground Station

3.3.1 Introduction

In order to test the models implemented in the `iqr` neural simulator, a computer capable of running them is needed. The computer is required to run the `iqr` neural network simulator as well as communicate with the drone via Wi-Fi. It needs to be fast enough to receive and decode the video stream from the drone and pass the information through all the models in the `iqr` neural simulator, encode the output of the neurons responsible for the control of the drone into a suitable form and then transmit the encoded data via the Wi-Fi network set up by the drone so that the appropriate motor command is received. All of this needs to happen within 1 simulation cycle of the `iqr` neural simulator.

3.3.2 Hardware

The computer chosen to function as the ground station is a mid-2012 Macbook Pro [60] with a 2.5GHz dual-core Intel Core i5 processor and 4GB of 1600MHz DDR3 memory. It has a 500GB hard drive that operates at 5400rpm and an IEEE 802.11a/b/g compatible 802.11n Wi-Fi wireless networking adapter.

3.3.3 Software

The default operating system of the Macbook is OSX Mavericks. Since there is no current version of `iqr` for OSX, a Linux-based operating system was installed to run the neural network simulator. Ubuntu 14.04 (Trusty Tahr) [61] was the chosen Linux operating system. The development of neurons and modules in `iqr`, as outlined in section 3.1.2.4 and 3.1.2.2, respectively, required the installation of QT 5.3.2 [62] to manage the graphical user interface. Furthermore, OpenCV [63] was installed to handle the decoding of the video stream received from the drone.

3.3.4 Operation

The ground station is the platform where the biological models, described in Chapter 4, are implemented and is used to control the flight of the drone. The ground station must thus be able to receive the video stream data from the drone via the Wi-Fi network created by the drone. The video stream then needs to be decoded and converted into neural activity

for neurons implemented in the `theiqr` neural simulator. Based on this neural activity, the outputs of the models describing the biological processes that regulate flight then need to be computed. The output of the neurons that represent the necessary flight actions then need to be encoded as UDP strings and sent back to the drone over Wi-Fi.

The first step in this process is the connection of the the ground station to the drone. On powering up, the drone sets up an *ad-hoc* Wi-Fi network with an address of 192.168.1.1. The ground station then connects to this network. This process only occurs once during the operation of the system.

The next step is to receive camera data from the drone. This is done by sending a wakeup bit to the drone, facilitated through a module `iniqr`. The drone then sends the current frame of its video stream back to the ground station.

The received video stream is decoded and converted into neural activity that is passed to a neuron group in the `theiqr` neural network simulator. This step is controlled through a module `withiniqr`. The neural activity is then fed through to the various biological models implemented in the system.

The outputs of these models are used by an `aniqr` module to convert the neural activity of the neuron group (based on the incoming visual information) into an appropriate movement command sent to the drone as a UDP string via Wi-Fi. This process happens once during every simulation cycle of the `theiqr` neural network simulator. The sequence diagram of the entire process can be seen in figure 3.5.

3.4 Testing Area

3.4.1 Introduction

To test the ability of the models describing the different parts of an insect's brain responsible for flight behaviour (outlined in Chapter 4), as well of models developed in this project to control of the drone during autonomous flight, an indoor testing area was built.

3.4.2 Physical Structure

The testing area is constructed with PVC pipes and is 2m high, 6m long and 6m wide. It is covered in white material so that the visual environment can be controlled. To give the biologically inspired control system visual cues of the surrounding environment that the models can use in processing, 25 sheets of black A4 paper were stuck to each wall of the testing area. The floor is a grey carpet with no distinguishing features.

3.4.3 Object Tracking

A tracking system was developed to follow the drone during its autonomous flight. It consists of a GoPro Hero3 camera suspended in the middle of the testing area and provides input to tracking software developed for this project using OpenCV [63]. The camera has a 170° field of view and is able to capture 1920x1080 video at 30 frames per second [64]. To track the drone during its autonomous flight the method of sequential images was employed. This method compares two sequential images from the video feed and detecting what has changed. The centre of what has changed is then calculated and assumed to be the centre of the moving object, as the drone is the only thing that moves in the test environment.

The first step in this method of sequential images takes two sequential images from the video feed of the web cam and calculates the absolute difference between the values of each pixel in the image. This result of this is a difference image which shows what has changed in the image and the intensity by which it has changed. This step can be seen in figure 3.6a.

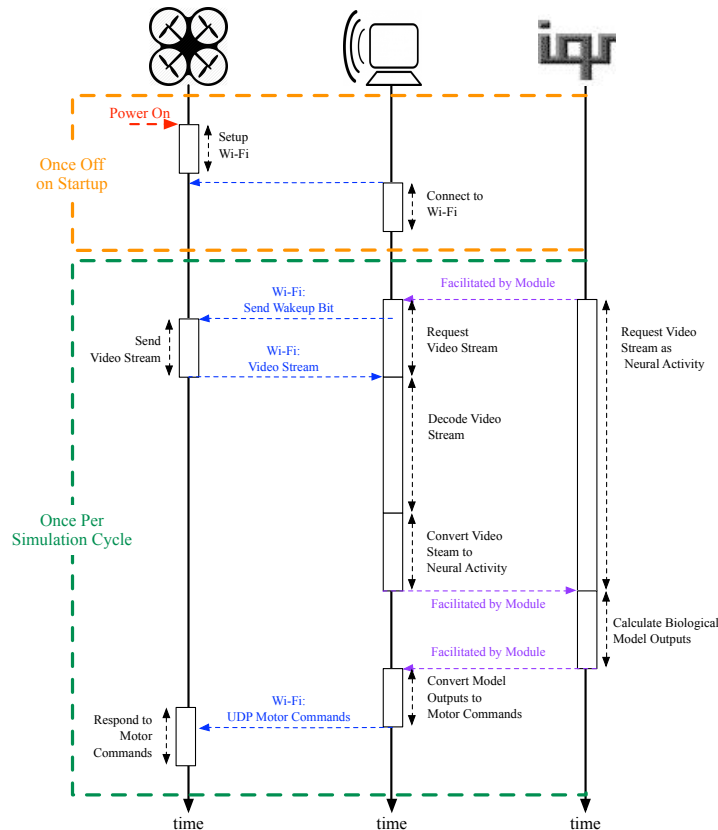


Figure 3.5 – The sequence diagram of the manner in which the AR.Drone 1.0, the ground station and iqr communicate during the startup and during every working cycle of the simulation

The next step is to convert the difference image, that represents the intensity of which parts of the image have changed on a pixel level, to an image which contains only black and white. This is done by passing the difference image, figure 3.6a, through a thresholding function. The thresholding function converts the pixel to white if its intensity is above a certain level, or to black if the intensity is below that level. The result of passing the difference image in figure 3.6 through the threshold function can be seen in figure 3.6b.

Because the two sequential images have some overlap, there will be an area in the difference and thresholded image where nothing has apparently changed (no movement has occurred). This occurs between the differences detected between the images and can be seen in figures 3.6a and b. To reduce the space between the two sequential images, the thresholded image is passed through a blur filter to increase the size of the detected change between the two images. The result of this is seen in figure 3.6c.

The contours of this new blurred threshold image are then detected and a bounding rectangle drawn around them, as shown in figure 3.6c. The centroid of this bounding rectangle, corresponding to the centre of the tracked object is then calculated. A cross-hair is drawn at this point on the original image to check if the object is being tracked accurately. The result can be seen in figure 3.6c.

To test that this method works correctly it was applied on a test video with a resolution of 720x480 pixels. The test video is of a ball being thrown against a wall three times in an

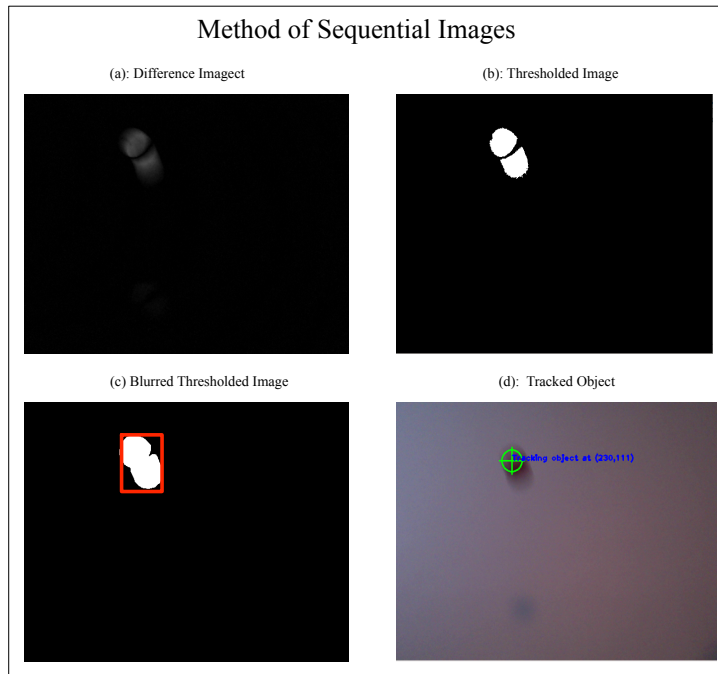


Figure 3.6 – Object tracking performed by the method of sequential images applied to a red ball that is thrown upwards against a white background.

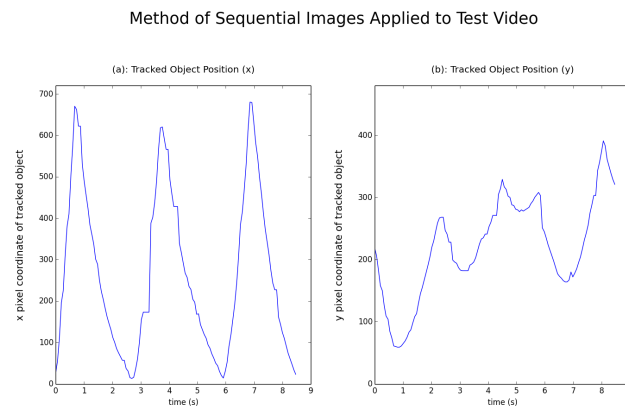


Figure 3.7 – Method of sequential images applied to test video

otherwise still room. This video was chosen as there is only one moving object present in the room at the time, the ball, which is similar to the experimental setup where the drone is the only moving object present in the testing area. The results of this can be seen in figure 3.7 with position 0,0 indicating the top left of the video.

From figure 3.7a it is seen that the x-position coordinate of the tracked object moves across the screen and returns to its starting position three times. This corresponds to the ball being thrown against the wall and bouncing back. Similarly in figure 3.7b it can be seen that the ball quickly moves upwards until it hits the wall and bounces back, decreasing in height until approximately 3 seconds into the video where the ball is caught and thrown

back to the wall. This process repeats itself three times, with the ball bouncing against the floor before being caught at around the 5 second mark which can also be seen when investigating figure 3.7b.

From this it is seen that the method of sequential images is an acceptable choice for the object tracking system of the testing area.

Chapter 4

Biologically Inspired Models

4.1 Introduction

Various neuronal models have been used to describe the functioning of the compound eye of an insect, and of the brain structures responsible for processing the information received from the eye and initiating a response based on that information. Here, these models are described, including the derivations of their mathematical formulations. The modifications made to the models, in order to make them more applicable in the research reported on here, are highlighted. Two classes of models, namely physiological models and phenomenological models, are used in this research.

Physiological models refer to models developed by measuring and mapping neuronal activity in various regions of an insect brain when the insect is subjected to specific stimuli. The measurements and mapping enable an evaluation of the relationships between input stimuli, neuronal activity and motor output (behaviour). Models are then developed to describe the dynamics of the physiological processes that occurred.

Phenomenological models are models that are developed by subjecting the insect to certain stimuli and observing or measuring the characteristic or characteristics to be modelled. A model that describes what is taking place on a phenomenological level based on the various input conditions is then built to describe the insect's behaviour. These models do not take into account any of the physiological processes underlying the observed behaviour.

4.2 Image Preprocessing: The Lamina Model

Before the signal coming from the retina of an insect's compound eye reaches the neural structures responsible for determining the necessary flight behaviour, it is preprocessed. This preprocessing is done in a sequential manner as the signal passes through the various hierarchical layers of the eye described in section 2.1.

The layer of the eye where most of the preprocessing is performed is the lamina. The image from the retina is passed to the lamina where contrast enhancement is performed on the image [26]. The model that describes the contrast enhancement in the lamina comprises of two submodels, the output of one forming the input for the other. This hierarchical structure of the retina and lamina models implemented in the `iqr` neural network simulator is shown in figure 4.1.

The first submodel detects the edges of the objects in the image coming from the neural group representing the retina, (labelled RGB in figure 4.1). This is done using a centre inhibition, surrounding excitation connection topology. Edge detection is performed as it allows the different objects in the environment to be distinguished from one another while removing unnecessary membrane excitation. Once the different objects in the environment

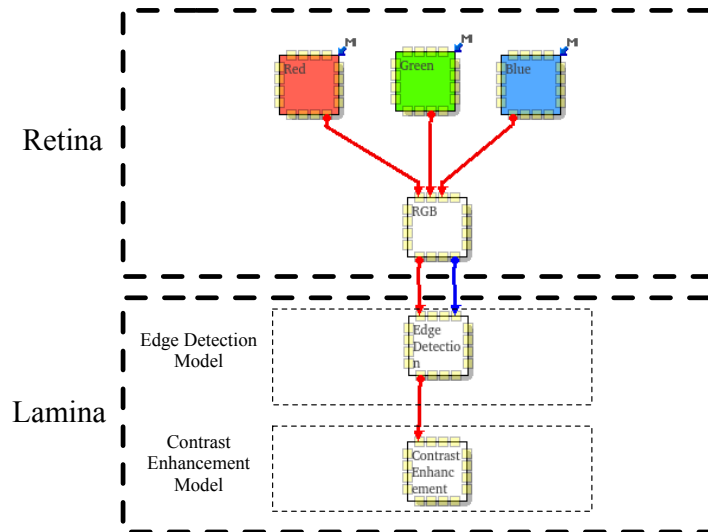


Figure 4.1 – The hierarchical structure of retina and lamina models and their connection topology implemented in the neural simulator `qr`

have been distinguished, their movement can be detected, as discussed in greater detail in sections 4.3.2, 4.3.3 and 4.4.

The second submodel performs contrast enhancement on the edge-detected image. The output of the contrast enhancement model causes the neurons to spike with an amplitude of 1 if an activity above a certain level is detected in a neuron in the neural group representing the edge detected image. This spiking represent the fact that an edge was detected at the position in the visual field represented by this neuron. This is done so that less information needs to be processed in the later models describing the processes that take place in the insect's brain. It is also vital to the model described in section 4.3.3, which shows potential to be applicable to many fields of robotics where motion sensitivity is needed.

4.2.1 Edge Detection

4.2.1.1 Introduction

Edge detection is the first step in the contrast enhancement used in the model of the pre-processing that occurs in the lamina. The methods described here have drawn inspiration from [65] and [66]. Modifications have been made to the models described in [65] and [66]. This makes the model presented here less computationally-intensive, allowing for better real time performance.

4.2.1.2 Method

Difference of Gaussians The first step in the contrast enhancement performed in the lamina is to extract the edges of the image. First, a method called difference of Gaussian based zero-crossing edge extraction [65] is investigated. In this method the incoming image is spatially convolved with a kernel which is described as the difference between two Gaussians. The equation describing the kernel can be seen in equation 4.2.1. In equation 4.2.1 f is defined in equation 4.2.2 with x being equal to distance from the neuron, whose position is $(0, 0)$, μ the mean value, σ the standard deviation with the condition $\sigma_1 > \sigma_2$. An example of this kernel is given in figure 4.2.

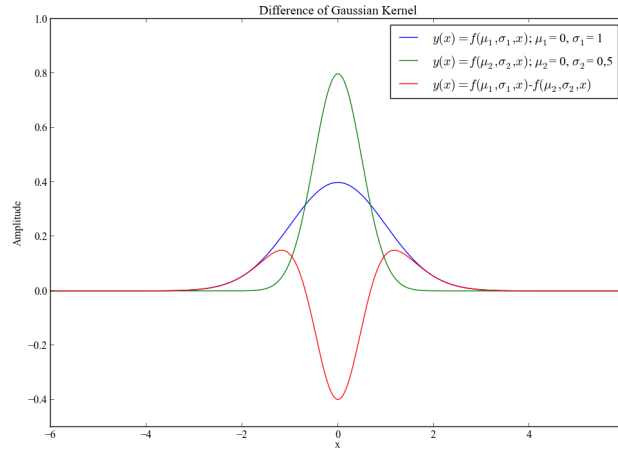


Figure 4.2 – Difference of Gaussian kernel in red with the two Gaussians it is comprised of, $f(\mu, \sigma_1, x)$ and $f(\mu, \sigma_2, x)$ in blue and green respectively

$$Difference_{Kernel}(x) = f(\mu, \sigma_1, x) - f(\mu, \sigma_2, x) \quad (4.2.1)$$

$$f(\mu, \sigma, x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (4.2.2)$$

The difference of Gaussian kernel is then convolved with the incoming image from the retina. The mathematical description of this process can be seen in equation 4.2.3.

$$Edges = Difference_{Kernel} * Image \quad (4.2.3)$$

Altered Edge Detection Model

Modifications were made to the difference of Gaussian based zero-crossing edge extraction to make it less computationally-intensive, meeting the requirement for real time operation of the system.

The modifications limit the number of inhibitory and excitatory synapses between the input and output neurons as well as the synaptic strength between these connections. The neurons that make up this model are linear threshold neurons, section 3.1.2.4, with a membrane persistence of 0. The inhibitory synapse topology is limited to a square of side length of $InhibitoryWindow$ and the synaptic weights of these connections are limited to $InhibitoryGain$. The topology of the excitatory synapses is limited only to the square window with inner and outer side lengths equal to $ExcitatoryWindow_{inner}$ and $ExcitatoryWindow_{outer}$ respectively and the synaptic gain limited to $ExcitatoryGain$. This topology mapping is shown in figure 4.3 and the kernel describing it in figure 4.4.

These modifications make the model less computationally-intensive since it is no longer necessary to work out the value of the Gaussian function for each neuron, which would be represented by the synaptic gain. The complexity of the model is further reduced by limiting the amount of neurons connected to the input (the neuron group representing the image coming from the retina) of the model to only the surrounding square of neurons of side length $ExcitatoryWindow_{outer}$ as seen in figure 4.3.

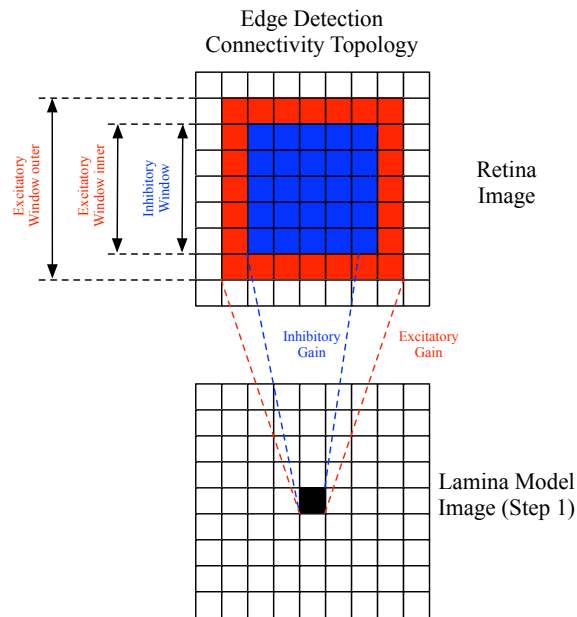


Figure 4.3 – Centre inhibition, surrounding excitation edge detection topology mapping

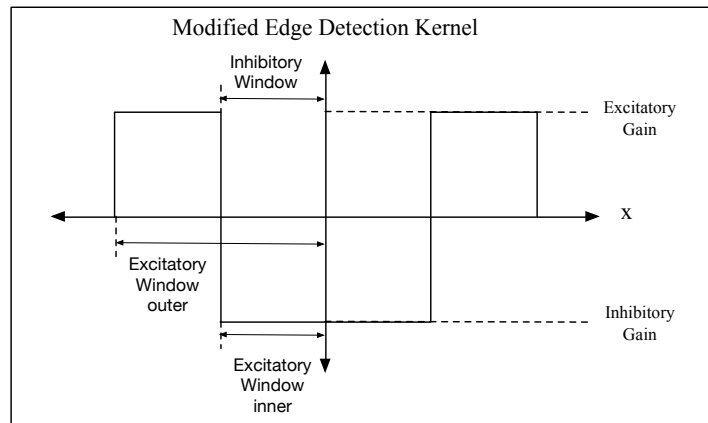


Figure 4.4 – Centre inhibition, surrounding excitation edge detection kernel

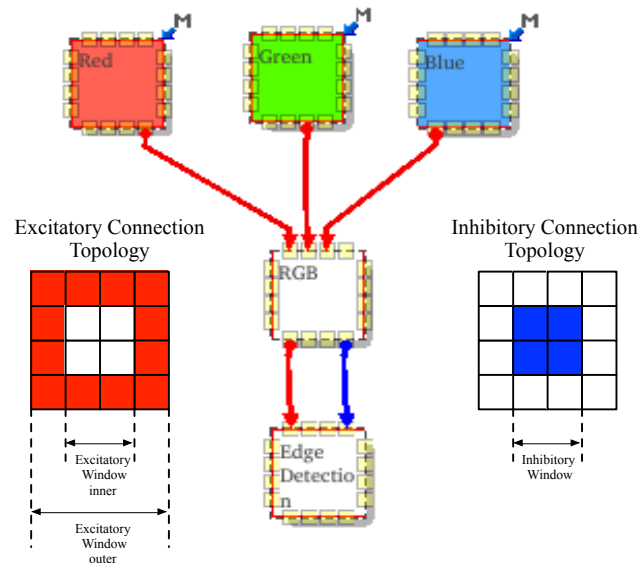


Figure 4.5 – iqr implementation of Edge Detection Model

4.2.1.3 Results

To test the altered edge detection model it was implemented in the *iqr* neural simulator. The altered edge detection model receives visual input from the front camera of the AR.Drone 1.0. The drone's front camera has a resolution of 640x480 pixels and the RGB values of the camera are projected to a neuron group of 64x48 neurons in the *iqr* neural simulator. This input group of neurons represents the image produced by the retina in the compound eye of an insect.

A centre inhibition, surrounding excitation connection topology is used to connect the neuron group representing the image coming from the drone's camera to the neuron group representing the detected edges. The connection topology of the network is shown in 4.3 with $InhibitoryWindow = 5$, $ExcitatoryWindow_{inner} = 5$ and $ExcitatoryWindow_{outer} = 7$. These values were chosen so that the total number of inhibitory and excitatory synapses were roughly equal. The synaptic gains, $ExcitatoryGain$ and $InhibitoryGain$ were set equal to 1 and -1 respectively.

The full system implemented in the *iqr* simulator is shown in figure 4.5. The input to the edge detection model is the neuron group labelled "RGB" and the output of the model is the neuron group labelled "edges" in the figure. The system was tested by placing the drone in different visual environments.

The first of these environments consisted of vertical black and white stripes each with a width of 3cm. The representation of this in terms of neural activity can be seen in figure 4.6a. The intensity of red within the image indicates a higher membrane voltage for the neuron representing this image. The second environment consisted of black and white squares (9cmx9cm) arranged in a chequerboard pattern, as shown in 4.6b. The third environment consisted of black A4 size papers stuck in a random pattern on a white background, as seen in figure 4.6c. This was also the environment used to test the final system and is described in detail in section 3.4. Two other environments that correspond to real world environments,

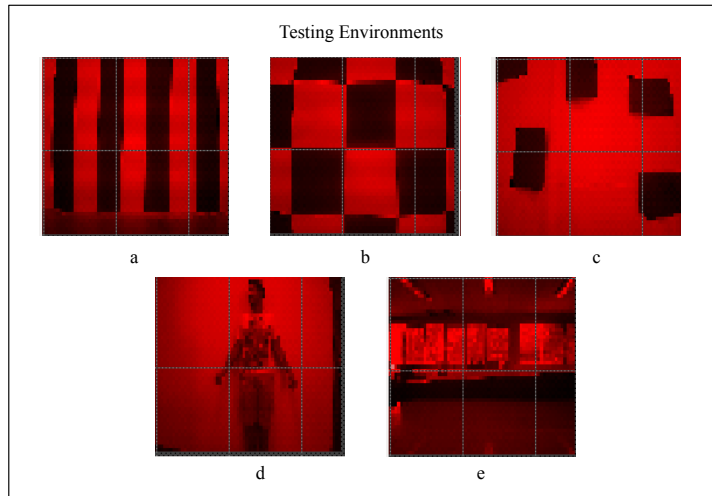


Figure 4.6 – Environments that the various biological models were tested

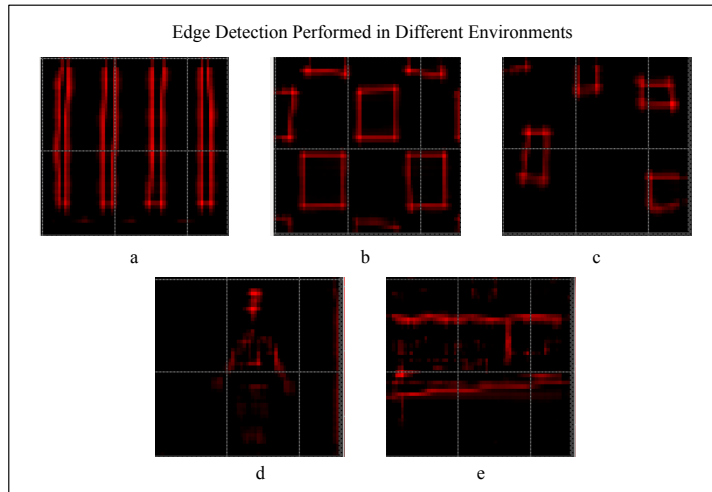


Figure 4.7 – Output of edge detection model when subjected to different environments

with no limitation on the colour or shape of the contained objects, were also used in order to test the system. One consisted of a person standing against a wall, figure 4.6d. The other comprised of desks and windows in the laboratory, figure 4.6e.

Edge detection performance in these environments is shown in figure 4.7. It was seen that the altered edge detection model worked as expected in all the environments.

That the edge detection model performed well when confronted with the various visual environments, suggests that it is robust to varying levels of contrast, textures and colours, as seen in figure 4.7a-d. For the image in figure 4.7e it was seen that some edges were detected but that the specific details of the image were difficult to discern. This was due to the quality of the camera and the amount of neurons used to represent this image. If the resolution of the camera and the amount of neurons in the neural group representing this image were higher, the contrast enhancement model should perform better. However,

the edge detection model fulfils the requirement of being able to detect edges effectively and is suitable to use as a model describing a portion of the preprocessing that occurs in the lamina.

4.2.2 Contrast Enhancement

4.2.2.1 Introduction

The model for the contrast enhancement that takes place in the lamina is completed by feeding the output of the edge detection image, described in section 4.2.1, to a neuron group which performs a binary contrast enhancement. This contrast enhancement is performed by having the neurons in the neuron group which make up the edge detected image, fire with an amplitude of 1 when an edge is detected. This model is of particular importance, since it is required for the modifications to make the model described in section 4.3.3 better suited to robotics.

4.2.2.2 Method

The output of the edge detected image as described in section 4.2.1 is passed to a neuronal group with the same size as that of the edge detected image. Each of the neurons in this group represents the signal from one of the photoreceptors in the insect's compound eye which has undergone the edge detection preprocessing described in section 4.2.1. Each individual neuron in this neuron group is a leaky integrate-and-fire neuron (LI&F) as described in section 3.1.2.4.

Neurons in the neuron group performing the contrast enhancement will only fire if their membrane potential reaches a certain threshold, v_t . The time constant describing the LI&F neuron (the membrane persistence) is 0, so that the previous image does not effect that of the current image. Thus a neuron in the output neuron group will only fire if activity above a certain level is detected.

4.2.2.3 Results

Contrast enhancement is performed by feeding the neural group representing the image that has had edge detection performed on it (described in section 4.2.1), to the contrast enhancement model. This was implemented in the `iniqr` simulator by adding a group of LI&F neurons to the system built to test the edge detection in section 4.2.1. The system that was implemented in `iniqr` can be seen in figure 4.1. The output of the edge detection model implemented in `iniqr` (the group labelled "edges") was supplied as the input to a group of LI&F neurons of size 64x48 (labelled Contrast Enhancement in figure 4.1). The effect of both the threshold voltage, v_t of the LI&F neurons and the robustness of the contrast enhancement model to different visual environments, was investigated.

First, the effect of the threshold voltage, v_t , of the LI&F neurons was investigated to determine its effect on the model's ability to perform contrast enhancement. This was done by varying the threshold voltage when performing contrast enhancement on a visual environment that consisted of a chequerboard of black and white squares with a side length of 9 cm and had been passed through the edge detection model. The input to the model can be seen in figure 4.7b. The results of these tests can be seen in figure 4.8.

From figure 4.8 it can be seen that the lower the threshold voltage the less effective the model is at contrast enhancement. This can be explained due to less neural activity being needed for the LI&F neuron to emit a spike at low threshold voltages. A lower threshold potential leads to greater sensitivity to parameters such as light intensity and colour of the object. When a higher membrane threshold potential is used, the effects of these parameters become negligible and the model behaves as expected.

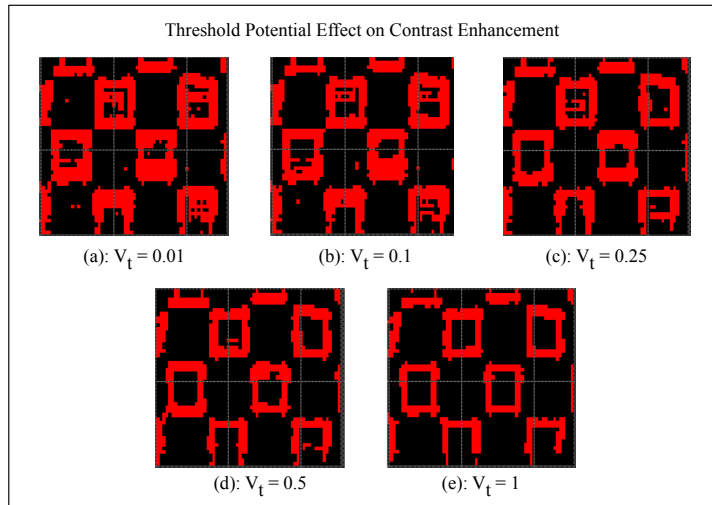


Figure 4.8 – Effect of the threshold potential, v_t , on the contrast enhancement model.

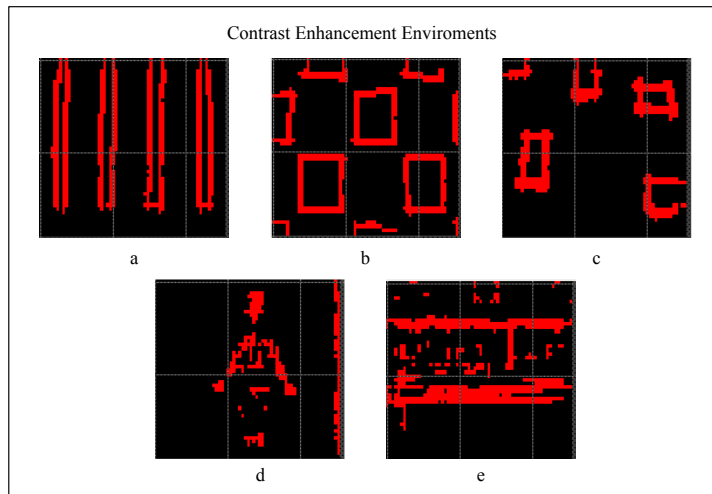


Figure 4.9 – Output of contrast enhancement model when subjected to different environments.

Next, the ability of the model to cope under different environmental conditions was tested. The environments that the model was tested in were the same as those used in the testing of the edge detection model, described in section 4.2.1. These environments are shown in figure 4.6. The inputs to the model were the outputs of the edge detection model as shown in figure 4.7. The membrane persistence of the neurons making up the output neuron group was made equal to 0, and the threshold voltage, v_t , set to 1. The output of the model subjected to these tests can be seen in figure 4.9.

From the results it is seen that the contrast enhancement, figure 4.9, that the model used to describe the contrast enhancement performed in the lamina works as intended. Contrast enhancement is performed correctly for images a-e in figure 4.9. For the image in figure 4.9e it is seen that contrast enhancement is performed but specific details of the image are difficult to distinguish from one another. This is due to the quality of the camera and the number

of neurons used to represent this image. If the resolution of the camera and the number of neurons are increased the contrast enhancement model for this image would perform better. The loss of these details is not of importance as models which receive input from the contrast enhancement model need only high level features, e.g. edges, to be present in order to perform correctly.

Thus it is seen that the contrast enhancement model is acceptable for use in the final system.

4.3 Course Stabilisation

4.3.1 Introduction

In this section the basic models used for the course stabilisation are developed. Course stabilisation is done by reacting to any perturbations in the optic flow patterns of the image that has undergone preprocessing in the insect's lamina [14]. The perturbations detected in the optical flow are fed to structures that determine the flight manoeuvres that need to be taken [29]. Although there is not much research into which structures in the insect brain are responsible for determining the necessary flight manoeuvres, it has been suggested that the Medulla and the Lobula play an important role [27]. The course stabilisation system is broken into four models, the elementary motion detector (EMD) model, the HS/VS cells model, a speed interpolation model and a winner takes all model.

The elementary motion detector is a neural structure sensitive to both speed and direction. Directional sensitivity is expressed as the change in the output neuron's membrane potential in response to the visually detected optical flow. The membrane potential of the EMD's output neuron increases when the detected optic flow is in the direction of preferred motion and decreases when the detected optic flow is opposite to the direction of preferred motion. Speed sensitivity is expressed as the amount of change in the membrane potential when optic flow is detected.

The horizontal and vertical system (HS/VS) cells are the neurons responsible for detecting whole field motion [67], [68]. Whole field motion is detected by integrating the responses of an array of EMDs that span the visual field. Integration of the responses of the EMDs gives the total optic flow of the image as seen by the insect's compound eye. Since the response of the HS/VS cells arises from the integration of an array of EMDs, HS/VS cells are sensitive to particular speeds, determined by the parameters of the EMDs.

A speed interpolation model that interpolates the outputs of HS/VS cells to give a more accurate estimation of the actual speed was developed.

The winner takes all model allows the maximum and minimum values of the inputs from the presynaptic connections to be identified. The first stage of the WTA network selects the maximum and minimum values of the inputs [66]. The second stage of the network enables the output from the network, described in [66], to be used to drive motor output for robotics applications.

4.3.2 Elementary Motion Detectors

4.3.2.1 Introduction

The elementary motion detector, or EMD, forms the core of the course stabilisation model and the collision detection model. The elementary motion detector is also known as the Reichardt correlator since it was proposed by Werner Reichardt in 1961 [69]. It relies on the correlation of the outputs of two spatially separated input neurons. The result is a neuron able to detect motion in a preferred and a null direction at different speeds. Detection of motion in the EMD's preferred direction causes the neuron's membrane potential to increase whereas detection of motion that is opposite to that of the preferred direction (the

null direction) causes the potential to decrease. The output of the EMD is maximal when the speed of the detected optic flow is equal to of the speed to which the EMD is sensitive. The factors that influence the speed sensitivity of the EMD are the spatial separation, D and the time delay, δ of the signals coming from the pre-synaptic neurons that provide input to the model.

4.3.2.2 Method

The EMD model is a phenomenological model that describes the motion of the fly accurately [70] without having insight into the underlying physiological processes. The model makes use of the cross-correlation between two neurons. The membrane potential of each neuron represents the signals coming from two photoreceptors in the retina separated by distance, D , which have undergone preprocessing performed in the lamina. The model is comprised of two branches, the preferred direction branch and the null direction branch. Each branch is responsible for detecting motion in a particular direction. The directions to which the preferred branch and null branch of the EMDs are sensitive, are dependent on the topology of the presynaptic connections of the preferred and null branches. However, the two directions are always opposite to one another. One of the signals from the input neurons of the EMD also experiences a time delay, δ . Which signal experiences the time delay depends on whether it is from the preferred branch or the null branch, as described in equations 4.3.2 and 4.3.3. A visual representation of the EMD model, showing both of these branches, can be seen in figure 4.10. The branches are computed independently and in parallel. After the output of each branch has been computed the two outputs are combined, as seen in equation 4.3.1, to form the model's output.

$$Out_{EMD} = Out_{Preferred} - Out_{Null} \quad (4.3.1)$$

The preferred and null directions of the EMD are determined by the spatial separation between the photoreceptors that stimulate the neurons providing the signals that undergo preprocessing in the lamina. For example, if the input neuron to the time delayed connection of the preferred branch, as shown in figure 4.10, represents a photoreceptor that is located spatially above the photoreceptor represented by the input neuron which has experienced a time delayed input to the null branch, the preferred branch will detect upwards motion. This is due to a high level of cross-correlation being detected between the actual image and the upward-shifted, time delayed image. This results in the directional sensitivity of the EMDs.

$$Out_{Preferred}(P_1, P_2) = P_1(t - \delta)P_2(t) \quad (4.3.2)$$

$$Out_{Null}(P_1, P_2) = P_1(t)P_2(t - \delta) \quad (4.3.3)$$

For a given speed, v , and given photoreceptor separation, D , neuron P_2 can be defined in terms of neuron P_1 as can be seen in equation 4.3.4.

$$P_2(t) = P_1\left(t - \frac{D}{v}\right) \quad (4.3.4)$$

Equations 4.3.2 and 4.3.3 can now be rewritten in terms of P_1 and P_2 respectively, as can be seen in equations 4.3.5 and 4.3.6 respectively.

$$Out_{Preferred}\left(P_1, P_1\left(t - \frac{D}{v}\right)\right) = P_1\left(t - \delta\right)P_1\left(t - \frac{D}{v}\right) \quad (4.3.5)$$

$$Out_{null}\left(P_2\left(t + \frac{D}{v}\right), P_2\right) = P_2\left(t + \frac{D}{v}\right)P_2(t - \delta) \quad (4.3.6)$$

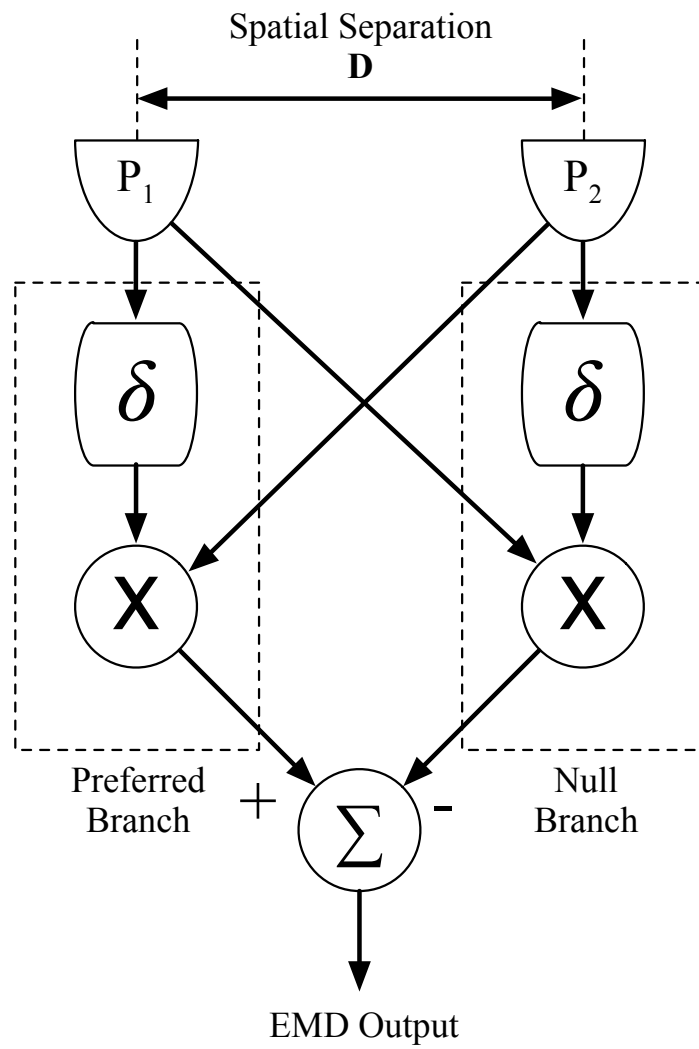


Figure 4.10 – A visual representation of the Reichardt Correlator, also known as the Elementary Motion Detector (EMD), with both the preferred and null branches indicated

The maximum outputs of the preferred and null branch of the EMDs can then be found by taking the derivatives of the equations that define the branches' output with respect to δ and setting the result equal to zero. The result can be seen in equations 4.3.7 and 4.3.8 for the preferred and null branches respectively. From these equations it is seen that the EMD's output is a maximum for $\delta = \frac{D}{v}$ and a minimum for $-\delta = \frac{D}{v}$.

$$\frac{dOut_{Preferred}}{d\delta} = 0 \text{ When } \delta = \frac{D}{v} \quad (4.3.7)$$

$$\frac{dOut_{Null}}{d\delta} = 0 \text{ When } \delta = -\frac{D}{v} \quad (4.3.8)$$

Looking at the equation that defines the output of the EMD, equation 4.3.1, it is seen that the closer $\frac{D}{v}$ is to δ the higher will be the membrane potential of the EMD. Consequently, EMDs exhibit sensitivity to a specific speed, with that speed being $\frac{D}{\delta}$.

4.3.2.3 Results

To test the EMD model, it was implemented in the `iqr` neural simulator. The model accepts user generated patterns, created using `iqr`'s state manipulation panel as the input, and passes them to the EMD structure. The model's implementation in the `iqr` neural simulator can be seen in figure 4.11. Integrate and fire neurons with a membrane persistence of 0 and a firing threshold of $0.1v$ were used for this model.

The model was tested by tuning the parameters of the EMD for a certain speed and then passing inputs of either 1 or 0 to each of the model's input neurons. The reasoning for using inputs of either 1 or 0 was to imitate the signal expected from the contrast enhancement model of the lamina (described in section 4.2.2). The combination of inputs used for testing either do or do not correspond to the speed to which the EMD is tuned. The outputs of EMD's positive and negative branches are then measured.

It was found that the EMD model does behave as expected, producing an output of 1 when the optic flow occurs at the speed for which the EMD is sensitive and is moving in the preferred direction and an output of -1 when optic flow in the null direction is detected at the speed the EMD is sensitive to. When optic flow is detected at a speed for which the EMD is not sensitive, the model's output is zero. The results of this testing have not been included due to their simplicity and lack of meaning in context. They are more understandable if the outputs of many EMDs are integrated and analysed; this is done in section 4.3.3.

4.3.3 HS/VS cells

4.3.3.1 Introduction

The next model used in the neural structure responsible for course stabilisation is the horizontal and vertical system (HS/VS) cells of the fly. The HS/VS cells are neurons which integrate the responses of an array of EMDs, described in section 4.3.2, to give the optic flow in the entire visual field. The model developed in this section draws inspiration from [66] with modifications made to the model to make it more suitable for robotics applications.

4.3.3.2 Method

The responses of an array of EMDs, all of which are sensitive to the same speed, are connected to a single neuron via excitatory synapses. This neuron represents the HS/VS cell and is a leaky integrate-and-fire neuron, described in section 3.1.2.4. The summation of all the responses of the EMDs, each representing the optic flow experienced by a particular photoreceptor, yields the optic flow of the whole visual field.

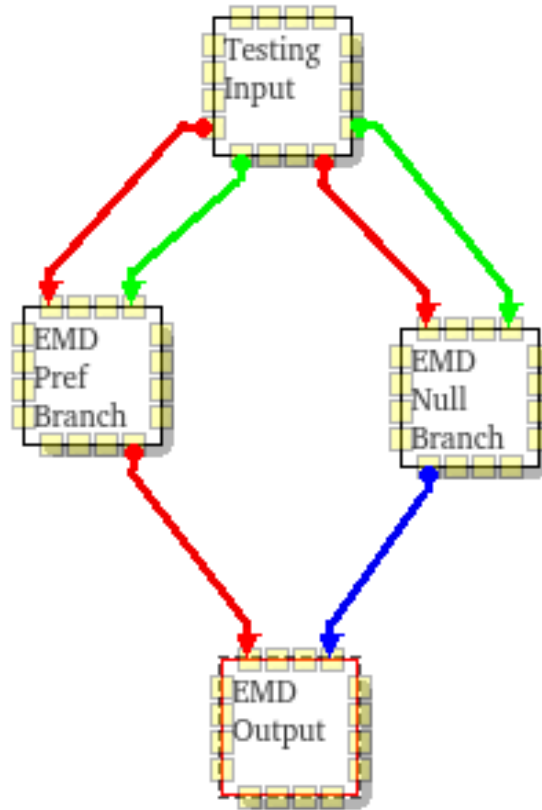


Figure 4.11 – iqr implementation of the system used to test the elementary motion detectors

The equation defining the output of the HS/VS cell can be seen in equation 4.3.9, where M_p is the membrane persistence, the constant term in equation 3.1.1. This system, as well as the processing stages leading up to it, can be seen in figure 4.12.

$$Out_{HS/VS}(t) = (M_p \cdot Out_{HS/VS}(t-1)) \cdot \sum_{i,j=1, i \neq j}^{EMDs} Out_{EMD}(P_i(t), P_j(t)) \quad (4.3.9)$$

Modifications From equation 4.3.9 it is obvious that the output of the HS/VS cell is proportional on the number of EMDs detecting optic flow in the visual field. This is undesirable as it makes the output dependent on the amount of neural activity in the original image. For example, if the neural group representing the visual input contained 100 neurons that have spiked after being passed through the contrast enhancement model, section 4.2.2, and, after being passed through the array of EMDs, correlation was only detected for 10 photoreceptors, it would be safe to assume that there is very little movement at the speed for which the EMDs are sensitive. However, if there were only 10 spiking neurons in the original image, and the EMD array detected optic flow for 10 photoreceptors, it can safely be assumed that there is movement at the speed to which the EMDs are sensitive. The problem that arises is that without a normalisation factor there is no way of telling the difference between these two cases since the membrane potential of the HS/VS cells would be the same.

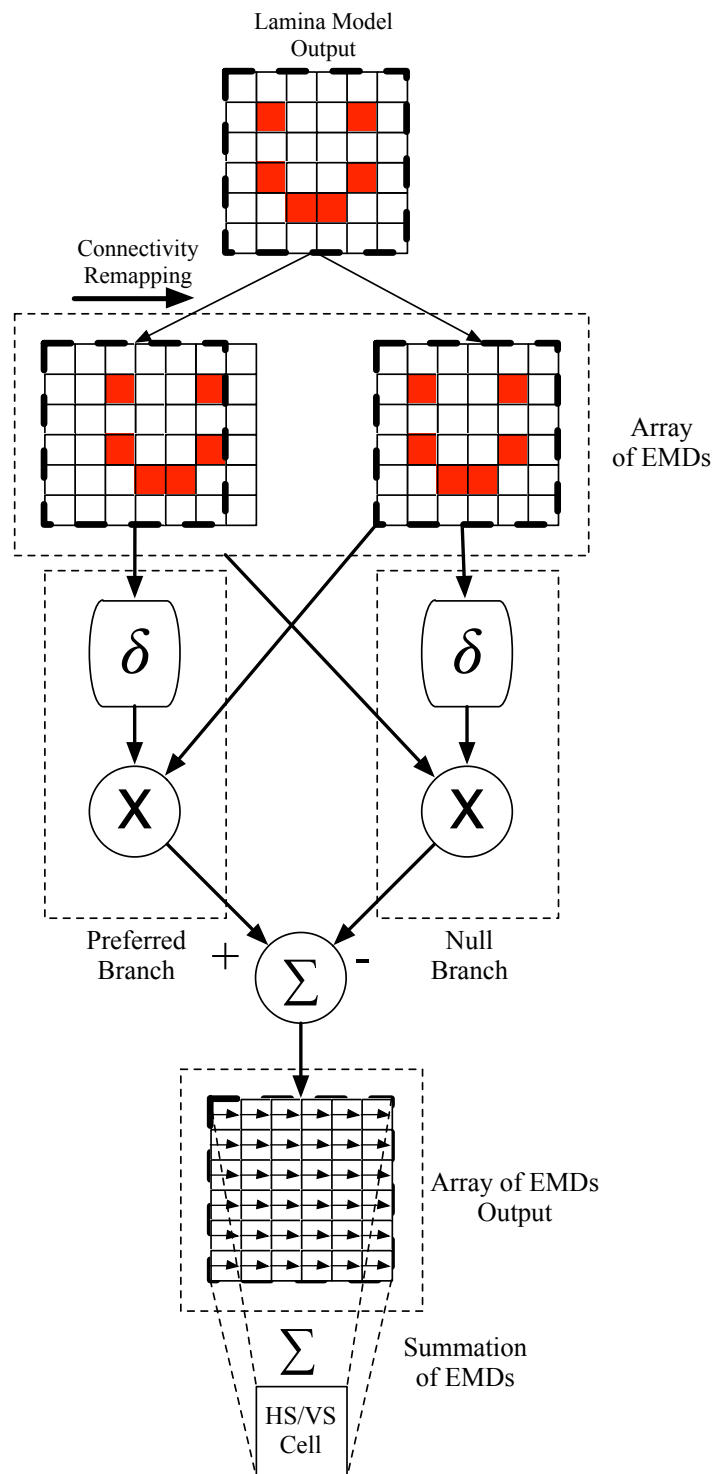


Figure 4.12 – Conceptual illustration of the HS/V5 cells of the fly. The original image has been shifted to the left making the array of EMDs sensitive to motion from right to left. The response of the array of EMDs is then integrated by the HS/V5 cell to give the whole field motion.

In [66] a feed-forward inhibition mechanism was used to combat a similar problem in the implementation of a model of the locusts lobula giant motion detector (LGMD). A feed forward inhibition mechanism was implemented by creating inhibitory synapses between the image passed to the array of EMDs and the neuron representing the HS/VIS cell. The output of the HS/VIS cell can then be described by equation 4.3.10.

$$Out_{HS/VIS} = \sum_{i=1}^{EMDs} Out_{EMD}(P_i(t), P_i(t - \frac{D}{v})) - \sum_{i=1}^{Lamina_{Neurons}} Inhibitory_{gain} \cdot Lamina_{neuron}(i) \quad (4.3.10)$$

While this approach is effective in making the output of the HS/VIS cells more robust at the level of activity in the original image, there are some problems when applying it to robotics. The first problem is that, although it does suppress the output based on the activity of the neurons representing the input image, the output is not normalised. For example, if the neural group representing the image output of the lamina model had 20 neurons that have fired, and 15 of the EMDs detect motion, arbitrarily choosing the gains in equation 4.3.10 to be equal to 1, the membrane potential of the HS/VIS cell would be $-5v$. If there were 200 neurons firing in the input image and 150 EMDs detect motion, the membrane potential of the HS/VIS cell would be $-50v$, even though in both of the examples, 75% of the EMDs in the EMD array detected motion.

This introduces two further problems. The first of these is that the offset in membrane potential can interfere with the HS/VIS cell's ability to detect motion correctly. This is because the output of the model for contrast enhancement (performed in the lamina and described in section 4.2) is always greater than the output of the EMD array. This is, in turn, due to the output of the array of EMDs being the cross-correlation between both the current signal and a time-delayed signal coming from the lamina model. This type of compensation results in a preference for the null direction due to the connection to the neuron representing the HS/VIS cells output being an inhibitory synapse.

Another problem of the feed-forward inhibition approach is that the output of the HS/VIS cells is not normalised. As seen in the example given above, the output of the HS/VIS cell is proportional to the amount of activity in the original image and not to the amount of optic flow detected. This is a major issue that needs to be considered when dealing with applications for robotics. The output needs to be normalised so that it becomes a function of only the amount of detected optic flow. By having a normalised output the interface with the control unit can also be more easily implemented. For example, a robot need only know what the appropriate motor command to send is, in response to the amount of optic flow from a specific HS/VIS cell, as opposed to having to calculate the appropriate command based on the level of activity of the image. A normalisation approach also allows for the whole system to be more easily upgraded, for example, by increasing the resolution of the camera, without having to compensate for the different size of the neural group representing the visual input.

To combat the problems described above, the model was augmented to make it more suitable to the needs of the system as a whole and also to provide a better base model for future work on biologically-inspired robotics systems. The augmentations make the output of the HS/VIS cells dependent only on the amount of optic flow detected, rather than on the level of activity in the neural group representing the output from the lamina model.

The modification made is that instead of subtracting the amount of activity of the neurons passed to the EMDs from the lamina model, the total activity of the EMDs, described by equation 4.3.9, is divided by the total amount of neuron group activity. The equation describing the new model for the HS/VIS cells can be seen in equation 4.3.11.

$$Out_{HS/VS} = (M_p \cdot Out_{HS/VS}(t-1)) \cdot \frac{\sum_{i=1}^{EMDs} Out_{EMD}(P_i(t), P_i(t - \frac{D}{v}))}{\sum_{i=1}^{LaminaNeurons} Inhibitory_{gain} \cdot Lamina_{neuron}(i)} \quad (4.3.11)$$

4.3.3.3 Results

The model for the HS/VS cells was tested by creating a system `iniqr` that employs the systems used to test the contrast enhancement (described in section 4.2.2) and the EMDs (described in section 4.3.2) to supply input to the HS/VS model. The output of the contrast enhancement model is given by a 64x48 group of integrate-and-fire neurons that either have a value of 1 or 0, depending on whether or not an edge was detected in the image. The output of the contrast enhancement is passed to three different arrays of 64x48 EMDs, each sensitive to different speeds ($3^\circ/s$, $6^\circ/s$ and $9^\circ/s$ respectively). This system can be seen in figure 4.14, with the HS/VS cells indicated as slow, medium and fast depending on which array of EMDs they are connected to. The green connections to the HS/VS are modulatory synapses responsible for the normalisation of the HS/VS cells, as detailed by equation 4.3.11.

The tests that were performed investigated three different characteristics of the HS/VS cells. The first test investigated the ability of the HS/VS cells to detect motion at various speeds. The second test investigated the effect that the environment has on the output of the HS/VS cell. Lastly, the ability of the HS/VS cells to cope with neuron failure in both the image input to the array of EMDs as well as with failure of the neurons in the preferred and null branches, was investigated.

To test the performance of the HS/VS cells at different angular speeds, the AR.Drone 1.0 was placed in an environment consisting of 3cm thick vertical black and white stripes. This environment can be seen in figure 4.6a. The drone was then rotated at different angular speeds and the value of the membrane voltage of three HS/VS cells was measured. The three HS/VS cells receive input from the EMD arrays labelled 'slow', 'medium' and 'fast'. The EMDs in these arrays are sensitive to speeds of $3\frac{\circ}{s}$, $6\frac{\circ}{s}$ and $9\frac{\circ}{s}$, respectively. The membrane potential of the three HS/VS cells measured during this test can be seen in figure 4.15. The average of the absolute value of the membrane potential of the HS/VS cells during these rotations is plotted against the angular speed in figure 4.16.

From this it is seen that the HS/VS cells are sensitive to different speeds. The speed that a HS/VS cell is sensitive to corresponds to the speed which the EMDs that supply input to the HS/VS cell are sensitive to. There is an anomaly when looking at the measurements of HS/VS cells that are sensitive to slow and fast motion. This arises due to the platform that was used to rotate the AR.Drone 1.0, which feeds the video stream to the input of the neural models, not being able to operate at the exact speed at which the EMDs are sensitive to. This causes the sensitivity profile of the HS/VS cells in figure 4.16 to deviate somewhat from what is expected. The expected result is that there would be a peak of the slow EMD at $3^\circ/s$ and a peak of the fast EMD at $9^\circ/s$. If the platform was able to be rotated at these speeds it is hypothesised that this behaviour would be seen.

This test reveals two important properties of the HS/VS cell model. The HS/VS cells are shown to be broadly tuned, i.e. they do not peak only at the speeds to which they are sensitive, and show a zero output at all other speeds. The test also shows that membrane potential response of the HS/VS cell is symmetrical around the angular speed to which it is sensitive.

The influence of the visual environment, particularly contrast, texture and shape, on the output of the EMDs was tested using a HS/VS cell sensitive to a speed of $2.7^\circ/s$. The

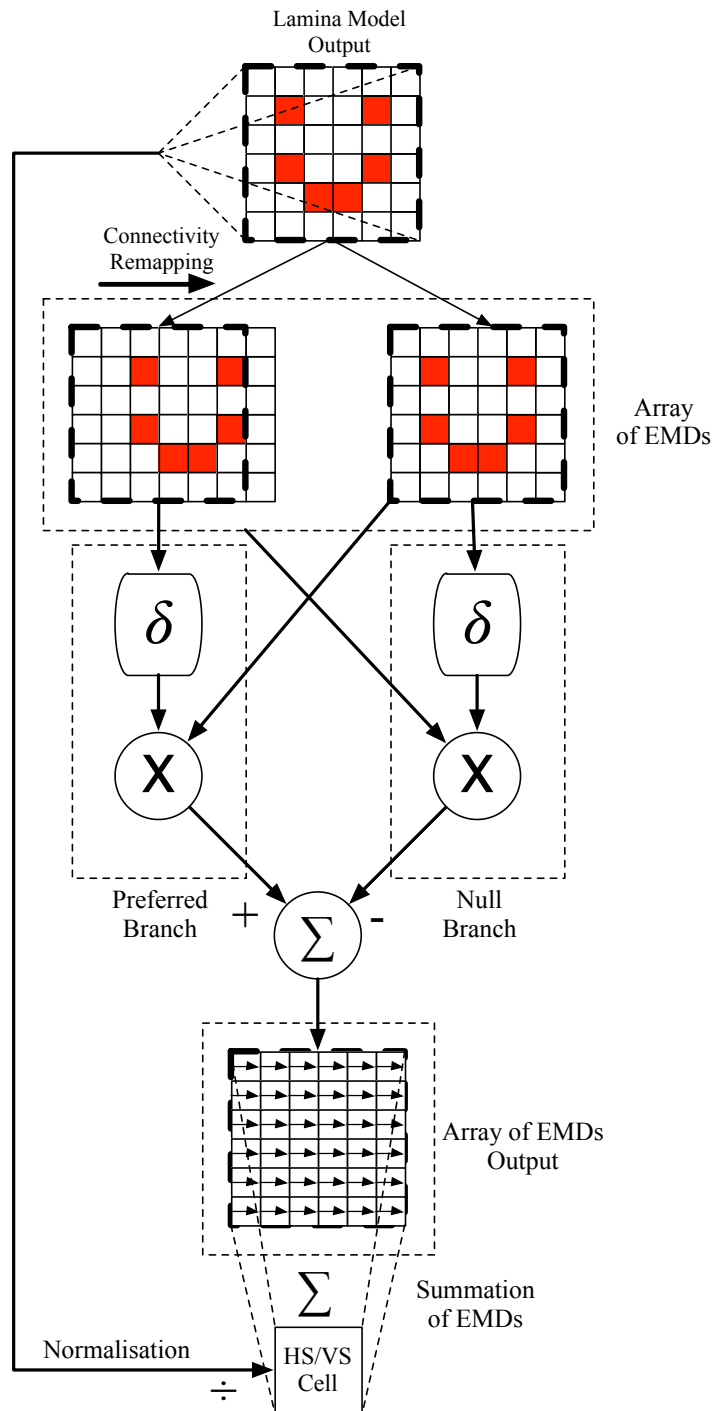


Figure 4.13 – HS/VS Cell model that has been modified to give a normalised output based on the amount of optic flow that has been detected

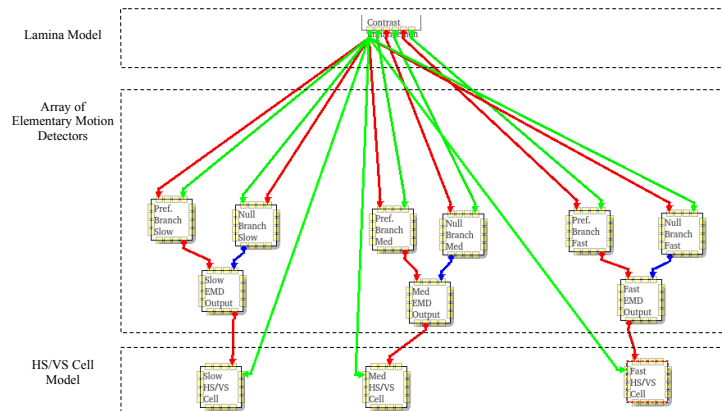


Figure 4.14 – iqr implementation of the HS/Vs cells. The output from the lamina model, the three arrays of EMDs tuned to different speeds and the HS/Vs cell model are shown. Red connections indicate excitatory synapses, blue connections indicated inhibitory synapses and green connections indicate modulatory synapses

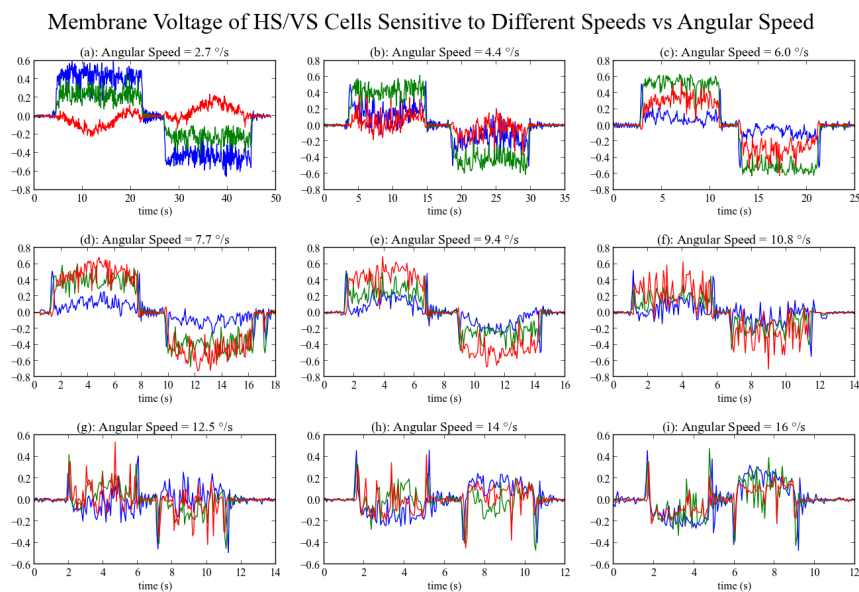


Figure 4.15 – Membrane potential of the different populations of HS/Vs cells when rotated at different speeds.

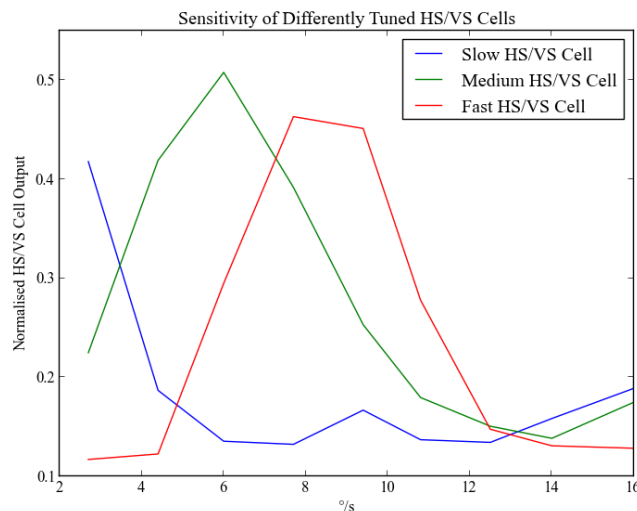


Figure 4.16 – Membrane potential of HS/VS cells tuned to be selective to different speeds plotted against angular speed

AR.Drone 1.0 was rotated at this speed in five different environments and the membrane potential measured. The results are shown in figure 4.17.

The first of these environments was vertical black and white stripes, with a width of 3cm as seen in figure 4.17a. The second environment consists of black and white squares arranged with a side length of 9cm in a chequerboard pattern, as seen in 4.17b. The third environment consists of black A4 size papers pasted randomly on a white background, as seen in figure 4.17c, this is also the environment that the final system was tested in and is described in depth in 3.4. The final two environments represented real world environments with no limitation on the colour or shape of the contained objects. They are shown in figure 4.17d and figure 4.17e.

The results show that the HS/VS cell is able to detect motion in both the preferred and the null direction regardless of the visual environment. The output of the HS/VS cell is highest when the visual environment contains only vertical stripes, as in figure 4.18a. Since the speed was kept constant and the output of the HS/VS cell normalised, the only factor capable of lowering the HS/VS cells output is the output of the null branch. Hence, the fact that the highest output of the HS/VS cells is for a visual environment containing only vertical stripes is the result of the fact that there is less chance of accidental correlation in branches of the EMDs.

Next, the effect of failure of the neurons within the array of EMDs was determined, by placing the system in an environment of vertical black and white stripes each with a thickness of 3 cm. The EMDs in the EMD array that supply input to the HS/VS cells were sensitive to motion at $2.7^\circ/s$. A percentage of neurons representing the output of the array of EMDs were disabled. The AR.Drone 1.0, which provides the visual input to the system through its front camera, was then rotated at $2.7^\circ/s$. and the membrane potential of the HS/VS cell recorded. The results of this test can be seen in figure 4.19.

The results show that the HS/VS cell is able to detect motion even under extreme conditions of neuron failure within the array of EMDs. This can be attributed to the fact that the output of the HS/VS cell is due to the integration of the responses of many EMDs in an array. Even if only a fraction of the EMDs in the array is still active, motion will be detected.

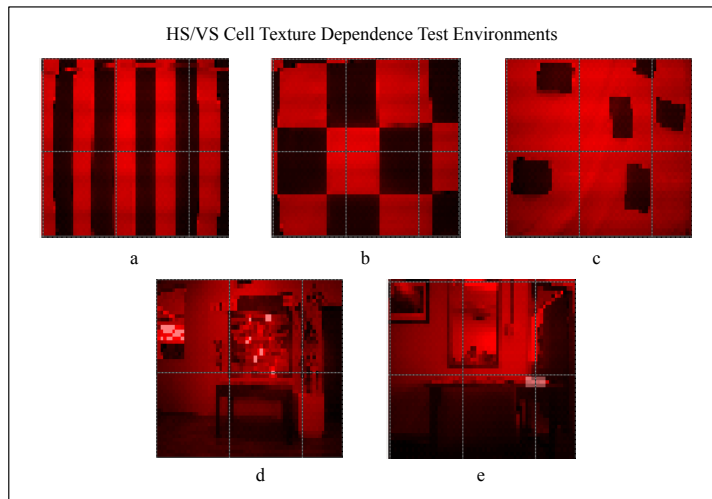


Figure 4.17 – Different environments in which the HS/Vs cells was tested to determine the environment's effect on its output

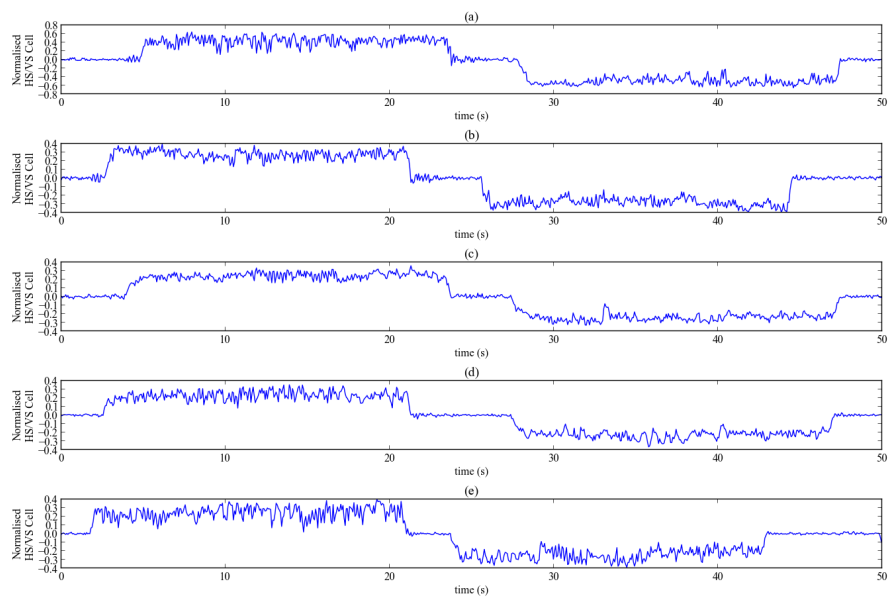


Figure 4.18 – The effect of the environment on the HS/VS cell's performance

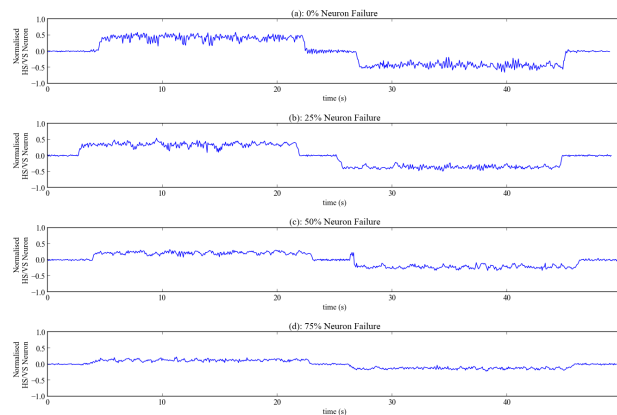


Figure 4.19 – Membrane potential of the HS/Vs cell when neurons in the EMD array have failed

The higher the level of neuron failure, the lower the output of the HS/Vs cell. This is due to the normalisation of the output of the model. The normalisation technique divides the membrane potential of the HS/Vs cell by the amount of activity in the input neurons. Neuron failure reduces the total amount of activity in the EMD array. However, if no neurons in the input image fail the amount of normalisation will still be the same, leading to a lower output of the HS/Vs cells in proportion to the amount of neurons that have failed. The exact amount that the HS/Vs cells membrane potential will be lower, cannot be determined as there are no constraints posed on which neurons fail. A neuron that detected correlation, or one that did not do so, could fail. Hence the exact effect on the amount of neuron activity in the EMD array cannot be determined.

The next test determined the effect of failure of the neurons representing the preferred branch of the EMDs, by placing the system in an environment of vertical black and white stripes with a thickness of 3cm. The EMDs in the EMD array forming the input of the HS/Vs cells were tuned to be sensitive to motion at $2.7^\circ/s$ and a percentage of neurons representing the preferred branch of EMDs disabled. The AR.Drone 1.0, which provides the visual input to the system through its front camera, was then rotated at $2.7^\circ/s$ and the membrane potential of the HS/Vs cell recorded. The results of this test can be seen in figure 4.20.

From these tests it was seen that failure in the preferred branch of the EMDs caused the output of the HS/Vs cells to be lower than expected. This due to the positive branch of the EMDs contributing towards the positive membrane potential of the HS/Vs cells. If the neuron in the positive branch of an EMD fails, motion can no longer be detected in the preferred direction. The result is that only the null branch's output contributes to the EMD array's total output. This creates a bias towards the null direction, represented by a negative membrane potential. Similar results were seen when using the same experimental setup but disabling the neurons in the null branch (figure 4.21). However, in this case the bias was toward the preferred direction.

4.3.4 Speed Interpolation

4.3.4.1 Introduction

So far, the models that have been described allow optic flow on a neural level to be detected by the EMD model discussed in section 4.3.2. The optic flow in the whole visual field can also be detected by the model of the HS/Vs cells by combining the responses of an array of

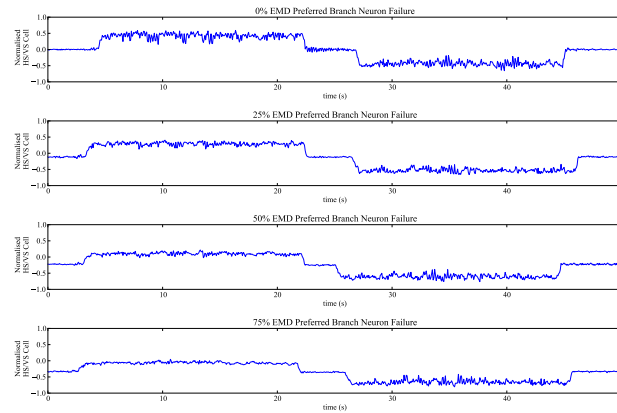


Figure 4.20 – Membrane potential of the HS/VIS cell when neurons in the preferred branch of the EMDs in the EMD array have failed

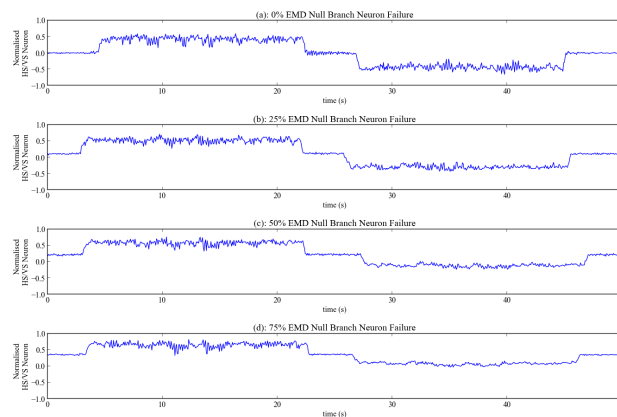


Figure 4.21 – Membrane potential of the HS/VIS cell when neurons in the null branch of the EMDs in the EMD array have failed

EMDs. While these models are able to provide a good measurement of the amount of optic flow detected in the visual field, they are limited by the fact that they are only sensitive to a specific speed. This introduces the limitation that, when the surroundings are moving at a speed faster or slower than the speed to which the EMDs are sensitive, the output of the HS/VIS cell may not detect motion. To overcome this requires many arrays of EMDs, each sensitive to a different speed, but this entails implementing a considerable number of additional neurons. For example, to detect seven different angular speeds would require seven EMD arrays and seven HS/VIS cells. Each of the EMD arrays would need to be as large as the neuron group representing the output of the lamina model. The result is a considerable increase in the amount of neurons needed. A model was thus implemented, based on the work done in [75] that interpolates the angular speed using only 3 HS/VIS cells, each sensitive to a different speed.

4.3.4.2 Method

Consider that there are three HS/VIS cells representing sensitivity to motion at either slow, medium or fast speeds. If, when being subjected to objects moving at a speed which the

HS/VS cell that is sensitive to motion at slow speeds has a membrane potential of 1, HS/VS cells maximum, and the HS/VS cells tuned for medium and fast speeds membrane potentials are near to 0, it would be safe to assume that the object was moving at the same speed as the speed which the HS/VS sensitive to slow speeds is sensitive to. If, however both the HS/VS cell sensitive to slow speed and medium speed have a membrane potential of 0.5 it can be assumed that the actual speed lies between the speed to which the two cells are sensitive to.

A method to interpolate between the responses of the HS/VS cells tuned for different speeds was developed, in order to give a reliable measurement of the actual speed. This was done by mapping each of the speeds that three HS/VS cells, sensitive to slow, medium and fast speeds, to a vector whose angle encodes the speed which the cell is sensitive to in a polar co-ordinate system. The mapping can be seen in figure 4.22.

Three conditions must be met for the interpolation mechanism to work effectively. The first of these conditions requires that the speeds which the HS/VS cells are sensitive to, be equally and evenly spaced from one another in terms of speed sensitivity. The second condition is that the responses of the HS/VS cells be broadly tuned, as described in section 4.3.3. The last condition is that the response of the HS/VS cells be symmetric around their peak. The response profile of HS/VS cells that meet these condition can be seen in figure 4.16.

Since speed is encoded as the vector's angle, not its magnitude, the actual angular speed can be extracted by finding the responses of each HS/VS cell relative to one another. To do this, the angular orientation, α , of the vector sum as expressed in equation 4.3.12 needs to be found. In equation 4.3.12 x_{sum} and y_{sum} are the Cartesian coordinates of the vector sum as expressed in equations 4.3.13 and 4.3.14 respectively.

$$\alpha = \arcsin\left(\frac{y_{sum}}{\sqrt{y_{sum}^2 + x_{sum}^2}}\right) \quad (4.3.12)$$

$$x_{sum} = r_{sum} \cdot \cos(\alpha) \quad (4.3.13)$$

$$y_{sum} = r_{sum} \cdot \sin(\alpha) \quad (4.3.14)$$

The coordinates of the vector sum, x_{sum} and y_{sum} , can then be understood as the sum of the contributions of the three HS/VS cells. Thus x_{sum} and y_{sum} can be defined in terms of the responses of the HS/VS cells as shown in equations 4.3.15 and 4.3.16 where the angles α_{slow} , α_{medium} and α_{fast} and magnitudes \mathbf{r}_s , \mathbf{r}_m and \mathbf{r}_f can be seen in figure 4.22.

$$x_{sum} = \mathbf{r}_s \cos(\alpha_{slow}) + \mathbf{r}_m \cos(\alpha_{medium}) + \mathbf{r}_f \cos(\alpha_{fast}) \quad (4.3.15)$$

$$y_{sum} = \mathbf{r}_s \sin(\alpha_{slow}) + \mathbf{r}_m \sin(\alpha_{medium}) + \mathbf{r}_f \sin(\alpha_{fast}) \quad (4.3.16)$$

Although this seems simple, there are two problems with this method. The first is the fact that, although the computations are relatively simple, they require trigonometric operations which are unlikely to be implementable by actual neurons [66]. The second problem is that, even if these trigonometric operations were possible, the actual angular orientation, α , would need to be known. In other words the actual speed would need to be known, which would make the speed interpolation model redundant.

The solution to these problems is to construct equally spaced projection vectors throughout the polar co-ordinate system. The angle of each of these projection vectors has a fixed angular orientation and thus represents a known speed in the co-ordinate system. A representation of this solution can be seen in figure 4.23.

The value of the projection of the vectors representing the output of HS/VS cells, \mathbf{r}_s , \mathbf{r}_m and \mathbf{r}_f onto the projection vectors can then be calculated using equation 4.3.17, with the angles θ , ϕ and β shown in 4.23.

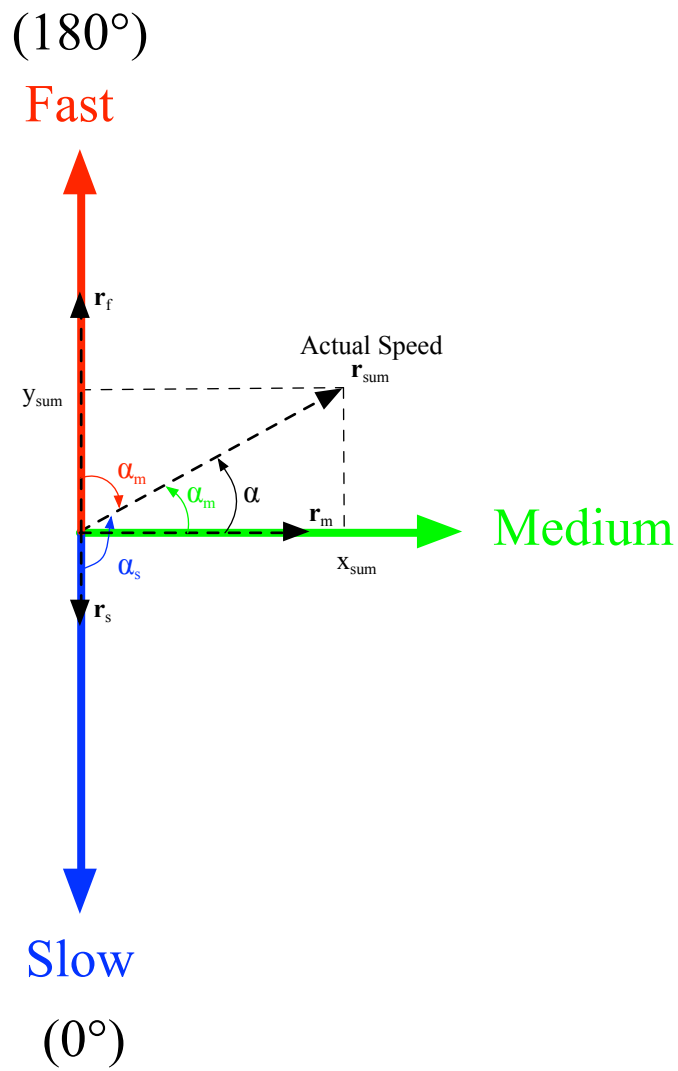


Figure 4.22 – Interpolation mechanism which encodes the angular speed as the angle of a vector

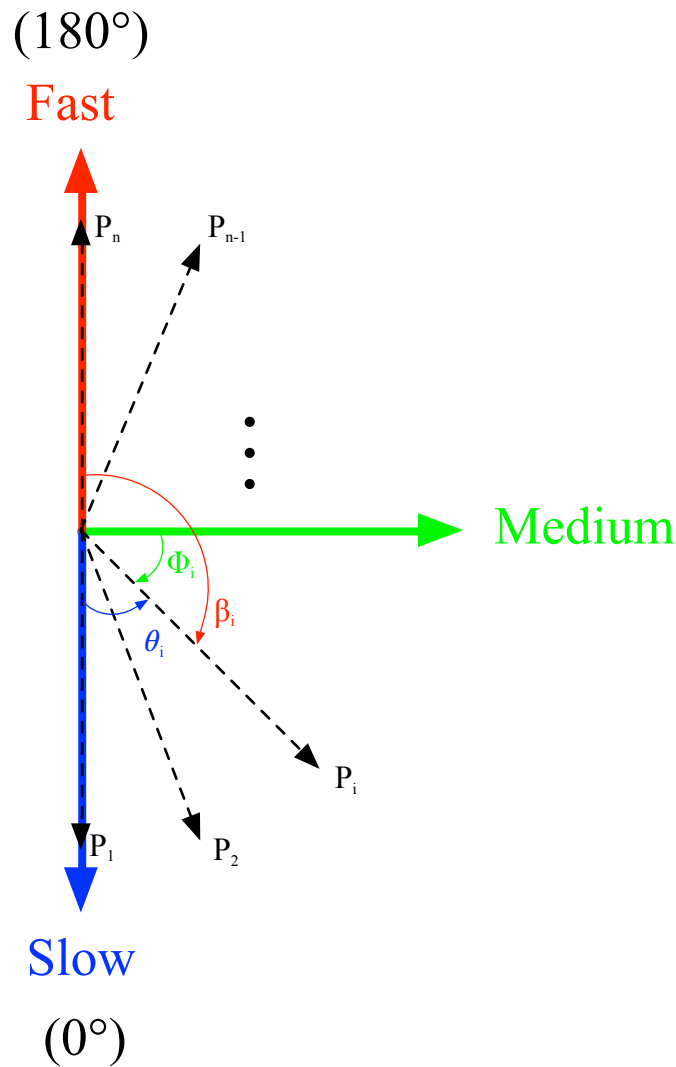


Figure 4.23 – Interpolation mechanism using projection vectors representing a known angular speeds.

$$Projection_i = \mathbf{r}_s \cos(\theta_i) + \mathbf{r}_m \cos(\phi_i) + \mathbf{r}_f \cos(\beta_i) \quad (4.3.17)$$

This use of projections representing a known speed solves the problem of not knowing the actual speed. The angle of the projection vector with the largest magnitude gives the closest approximation to the actual speed.

To solve the problem of the implausibility of a biological system having to perform trigonometric calculations, the necessary values are 'hard coded' into synaptic weights between the HS/VS cells and the neurons that represent the projection vectors (projections neurons). In other words, the values of $\cos(\theta_i)$, $\cos(\phi_i)$ and $\cos(\beta_i)$ are encoded directly as synaptic weights; they are known values since the angular orientation of the projection vector is fixed.

With three equally spaced HS/VS cells sensitive to different speeds and tuned broadly to be symmetric around their peak, a better estimation of the actual speed can be obtained using a speed interpolation system. The quantisation error introduced by this process is maximal when the actual speed lies equidistantly between two adjacent projections, as is given by equation 4.3.18.

$$QuantisationError_{max} = \frac{|Projection(i)_{AngularSpeed} - Projection(i+1)_{AngularSpeed}|}{2} \quad (4.3.18)$$

4.3.4.3 Results

In this section the performance of the speed interpolation module is investigated by implementing the model in the `iqr` neural simulator. First the ability of the speed interpolation model to correctly project the inputs to the model onto the correct projection vectors is investigated. Next the ability of the speed interpolation model to correctly detect the projection vector which corresponds to the speed that is closest to that of the actual speed is tested.

To test the ability of the speed interpolation model to correctly project the inputs onto the projection vectors, three inputs, representing the slow, medium and fast HS/VS cells, were implemented in the model in the `iqr` neural simulator. The system can be seen in figure 4.24. The state manipulation panel, which allows the user to define which neurons fire at what times as well as the magnitude of the spikes, is used to manipulate the HS/VS cells. The model makes use of seven projection vectors, each spaced 30° apart. The connection topology between the HS/VS cells, as well as the synaptic weights of the connections, can be seen in figure 4.25a and b respectively. The model was tested by subjecting it to different combinations of inputs, that represent the outputs of the HS/VS cells, and measuring the membrane voltage of each projection neuron. The results of these tests can be seen in figure 4.26.

These tests show that the model used for speed interpolation works correctly. If all the inputs to the model have the same value, figure 4.26f, no single projection neuron responds with a maximum value. It is assumed that this will not influence the model's performance since the HS/VS cells are sensitive to different speeds and the probability of two or more HS/VS cells having the same output is extremely low.

The next test assessed whether the speed interpolation model is able to identify the speed closest to the actual speed. This was done by connecting the speed interpolation model to 3 HS/VS cells that are sensitive to speeds of $3^\circ/s$, $6^\circ/s$ and $9^\circ/s$ respectively. A modification was made to the output neuron of the HS/VS cell by changing it to a linear threshold neuron, 3.1.2.4. This was done so the input to the speed interpolation model could be the membrane activity of the HS/VS cell, as opposed to the 1 or 0 if the neuron was a leaky integrate-and-fire neuron. The video input to the system comes from the AR.Drone 1.0 and is represented by a group of 64x48 neurons. The AR.Drone 1.0 was then rotated at varying speeds and outputs of the speed interpolation model recorded. The projection neuron with the maximal response for the given speed is shown in figure 4.27. The results show that the speed interpolation model works as expected for speeds above the speed to which the slowest HS/VS cell is sensitive. At speeds lower than that, the slow HS/VS cell still exhibits the highest response of the three HS/VS cells. This problem can be overcome by using a HS/VS cell sensitive to slower speeds as input to the speed interpolation model.

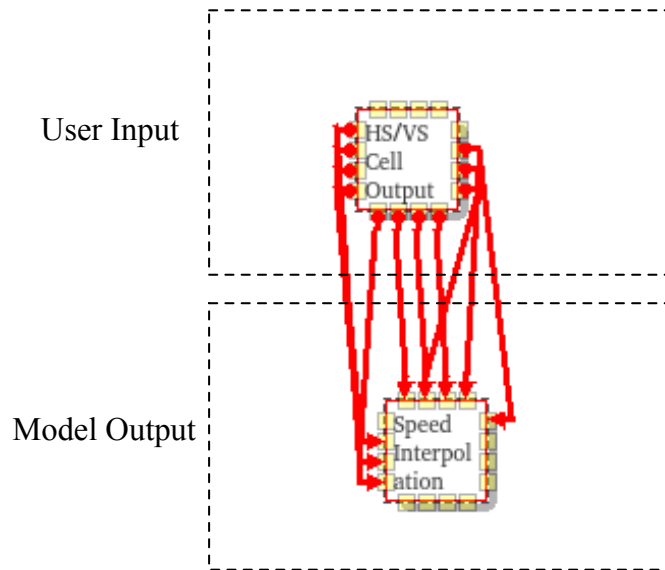


Figure 4.24 – iqr System to test interpolation model using user defined inputs

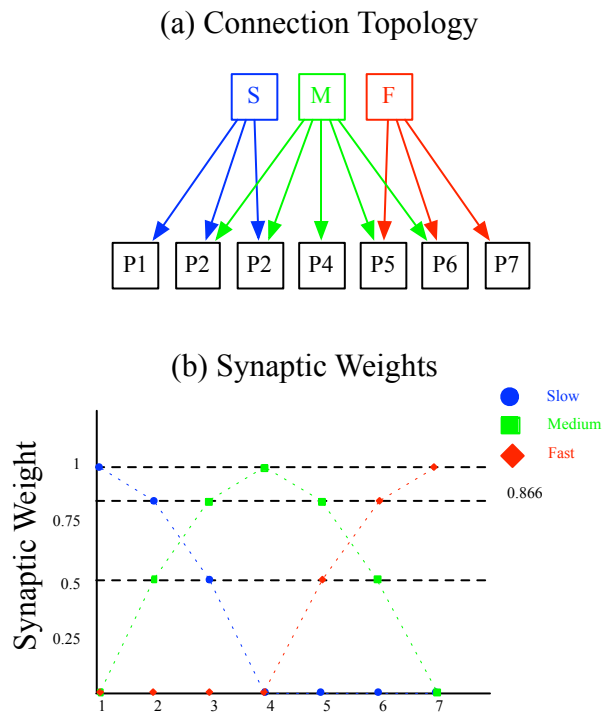


Figure 4.25 – (a) Connection topology of the speed interpolation model. (b) Synaptic weights of speed interpolation model.

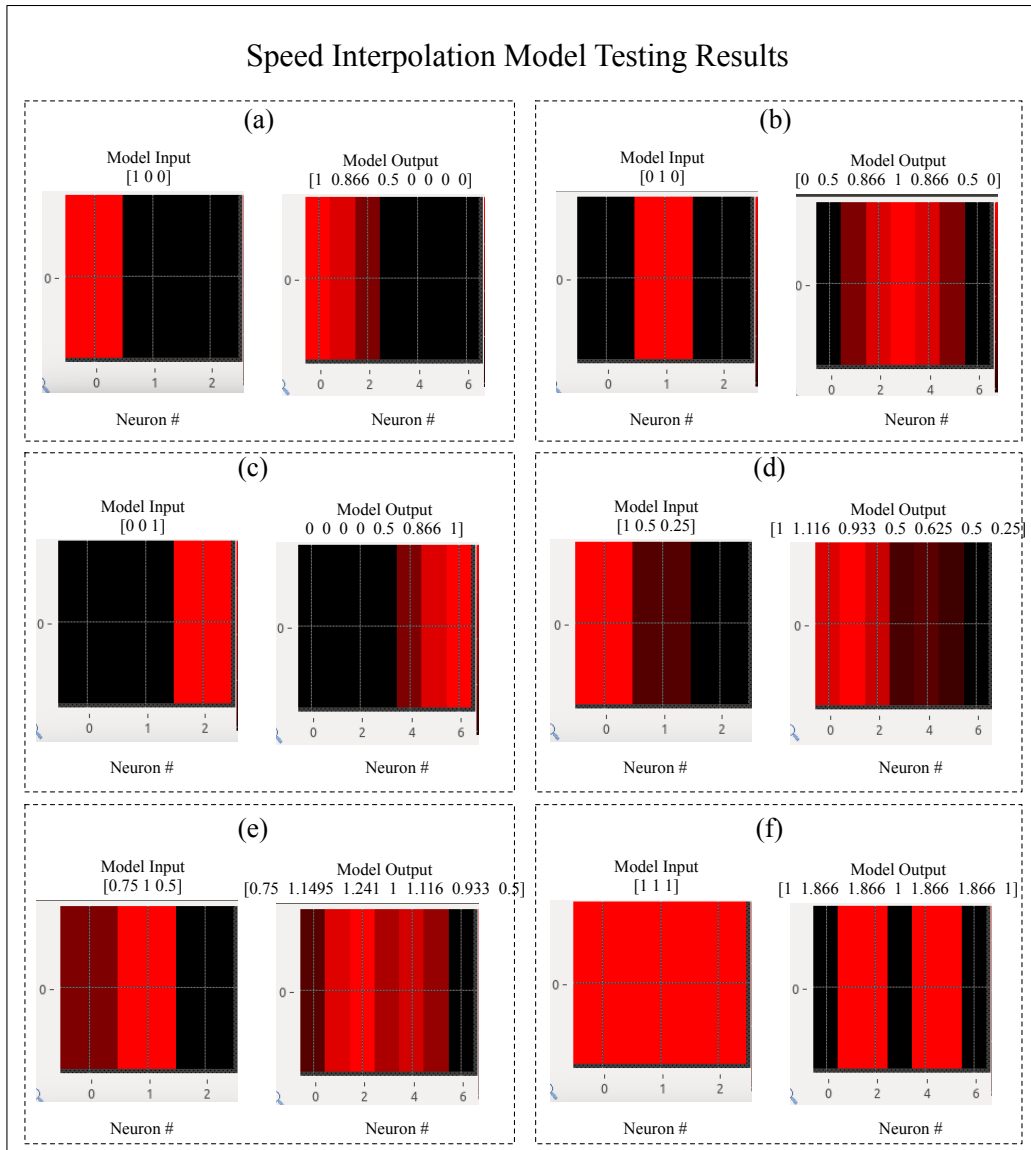


Figure 4.26 – Inputs and outputs of the speed interpolation model implemented in theiq neural network simulator.

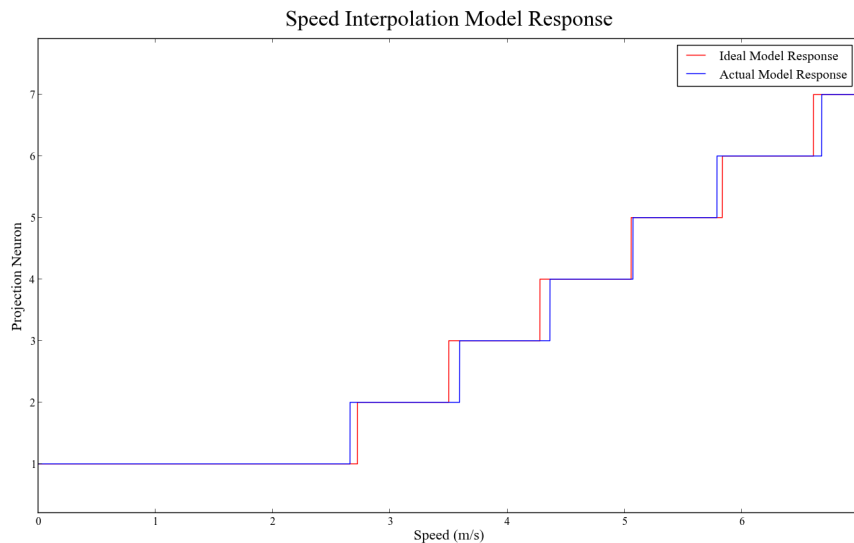


Figure 4.27 – Response of the speed interpolation model over different speeds

4.3.5 Winner Takes All neural model

4.3.5.1 Introduction

In Section 4.3.4 a model for interpolating between the responses of three HS/VS cells, each sensitive to a different speed, to give a more accurate prediction of the actual speed, is introduced. The three HS/VS cells form a new polar co-ordinate system in which speed is represented by the angle of the vector. The projections of the responses of the three HS/VS cells onto projection vectors of known speed (represented by a fixed angle) are then calculated. The actual speed thus corresponds to the speed represented by the projection neuron with the highest response. A method is thus needed to select the maximally responding projection neuron. The reason for selecting the projection closest to the actual angular speed allows the appropriate movement command to be sent to the system when optic flow is detected.

To accomplish the task of selecting the maximum responding neuron, a model of the Winner Take All (WTA) operation was created. The WTA operation selects the maximum input passed to the model; its mathematical description can be seen in equation 4.3.19. The WTA operation is widely used in many different computational brain models and there are many biologically plausible models that perform the WTA operation [71], [72], [73], [74]. In this section a model that was proposed by Sergi Bermúdez [75] is used. A modification to this model is also proposed that allows the output of the WTA model developed in [75] to be better suited for robotics applications.

$$WTA(i_1 \dots i_n) = \begin{cases} 1 & \text{if } i_j \geq i_k \text{ for all } j \neq k \\ 0 & \text{if } i_j < i_k \text{ for some } j \neq k \end{cases} \quad (4.3.19)$$

4.3.5.2 Method

The biological model for the WTA operation makes use of a structure called a WTA matrix. The WTA matrix entries are created by computing both the maximum and minimum of its n inputs through the use of $n^2 + 2n$ neurons arranged in a square matrix structure. Taking

the n inputs, as defined in equation 4.3.20, a parallel comparison between the components of its inputs is performed, as seen in equation 4.3.21.

$$i = [i_1 \quad i_2 \quad \cdots \quad i_n] \quad (4.3.20)$$

$$Comparison_{Matrix} = \begin{bmatrix} i_1 - i_1 & \cdots & i_n - i_1 \\ \vdots & \ddots & \vdots \\ i_1 - i_n & \cdots & i_n - i_n \end{bmatrix} \quad (4.3.21)$$

Next, a thresholding operation is performed on the $Comparison_{Matrix}$. The thresholding operation's output is a one (for an input value >0) or a zero (for an input value ≤ 0). The thresholding operation is performed by using an integrate-and-fire neuron with an extremely small threshold voltage, 10^{-5} , that fires with an amplitude of 1 when the threshold is exceeded. The mathematical formulation of this thresholding operation is shown in equation 4.3.22. The rationale behind passing the comparison matrix through the thresholding operation is that the winning input (the input with the maximum value) will inhibit all other neurons while the losing input (the input with the lowest value) will be inhibited by all the other inputs.

$$Threshold(i) = \begin{cases} 1 & \text{if } i > 0 \\ 0 & \text{if } i \leq 0 \end{cases} \quad (4.3.22)$$

The maximum and minimum inputs can now be extracted from the matrix that resulted in passing the $Comparison_{Matrix}$ through the thresholding operation. This is done by creating a vector that represents the sum of the columns of this matrix and a vector that represents the sum of the rows of this matrix. The mathematical formulation of the minimum and maximum vector can be seen in equations 4.3.23 and 4.3.24 respectively.

$$\mathbf{V}_{min} = \left[\sum_j^n Threshold(i_1 - i_j) \quad \cdots \quad \sum_j^n Threshold(i_n - i_j) \right] \quad (4.3.23)$$

$$\mathbf{V}_{max} = \begin{bmatrix} \sum_j^n Threshold(i_j - i_1) \\ \vdots \\ \sum_j^n Threshold(i_j - i_n) \end{bmatrix} \quad (4.3.24)$$

Model Modifications Currently, the output of the model is two neuron groups of size n representing the vectors, \mathbf{V}_{max} and \mathbf{V}_{min} . The neurons in these neuron groups that have a zero membrane potential correspond to the neurons representing the maximum and minimum input to the WTA model. While the maximum and minimum inputs are known, the lack of activity does not allow the output of the model to excite or inhibit other neurons. This means that the output of the WTA model cannot be used to influence the behaviour of the system. In this section a method for transforming the output neuron groups representing two new neuron groups, whose contents are equal to the vectors \mathbf{V}_{max} and \mathbf{V}_{min} which has undergone a transformation as defined in equation 4.3.25.

$$Threshold(i) = \begin{cases} 0 & \text{if } V_m > 0 \\ 1 & \text{if } V_m = 0 \end{cases} \quad (4.3.25)$$

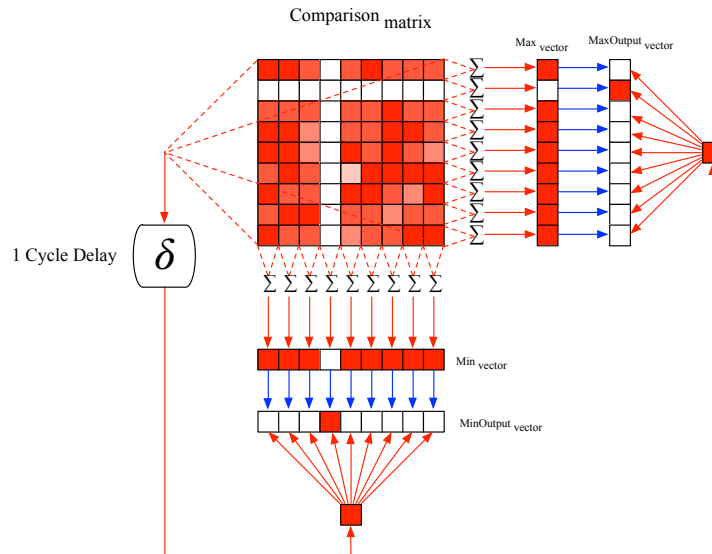


Figure 4.28 – Conceptual representation of the working of the Winner Take All Model with conversion of the output to be represented by a neuron which emits a spike

To perform the transformation, two groups of integrate-and-fire neurons, both with a size equal to that of the number of neurons in the neuron groups that represent the vectors \mathbf{V}_{min} and \mathbf{V}_{max} , are created. These new LI&F neuron groups both have a threshold voltage, v_t , of 1 volt and form the model's new output. These new neural groups are connected to the neuron groups representing \mathbf{V}_{min} and \mathbf{V}_{max} by a one to one connection topology using inhibitory synapses. This means that each neuron in the neuron group representing the vectors \mathbf{V}_{min} and \mathbf{V}_{max} is connected only to its corresponding neuron in the new neuron group via one inhibitory synapse. A constant offset membrane voltage of 1 is supplied to the newly created neural groups by connecting all the neurons in the comparison matrix to a LI&F neuron with a threshold of $10^{-5}v$ via excitatory connections. This new neuron which supplies the offset is then connected to all the neurons in the two new neural groups using excitatory connections. This connection topology allows any activity in the neurons representing \mathbf{V}_{min} and \mathbf{V}_{min} to inhibit activity in the corresponding output neuron of the two new neuron groups. Hence, only the neurons with zero neural activity, which in this case would represent either the minimum or maximum input to the WTA model, cause neurons in the output neuron groups to fire. This system can be seen in figure 4.28

Implementation Issues While this system works in steady state, a modification is needed to allow it to start up correctly. Since the activity of neurons in the `iqr` neuronal simulator is only propagated once per cycle, it takes a few cycles on startup before any input is supplied to the winner take all matrix. This results in a zero membrane potential of all the neurons of the neural groups representing the vectors \mathbf{V}_{min} and \mathbf{V}_{min} , which in turn causes the output (membrane potential) of the neural groups to be equal to one. The result is that the simulator sends motor commands to the AR.Drone 1.0 while it is still taking off, causing it to crash.

To overcome this, a structure similar to that used in the modification to the model was created. An integrate-and-fire neuron with a threshold voltage of $10^{-5}v$ forms the post synaptic neuron of a connection (via excitatory synapses) to all the neurons that form the winner take all matrix. These excitatory synapses have a delay of 1 cycle, to compensate

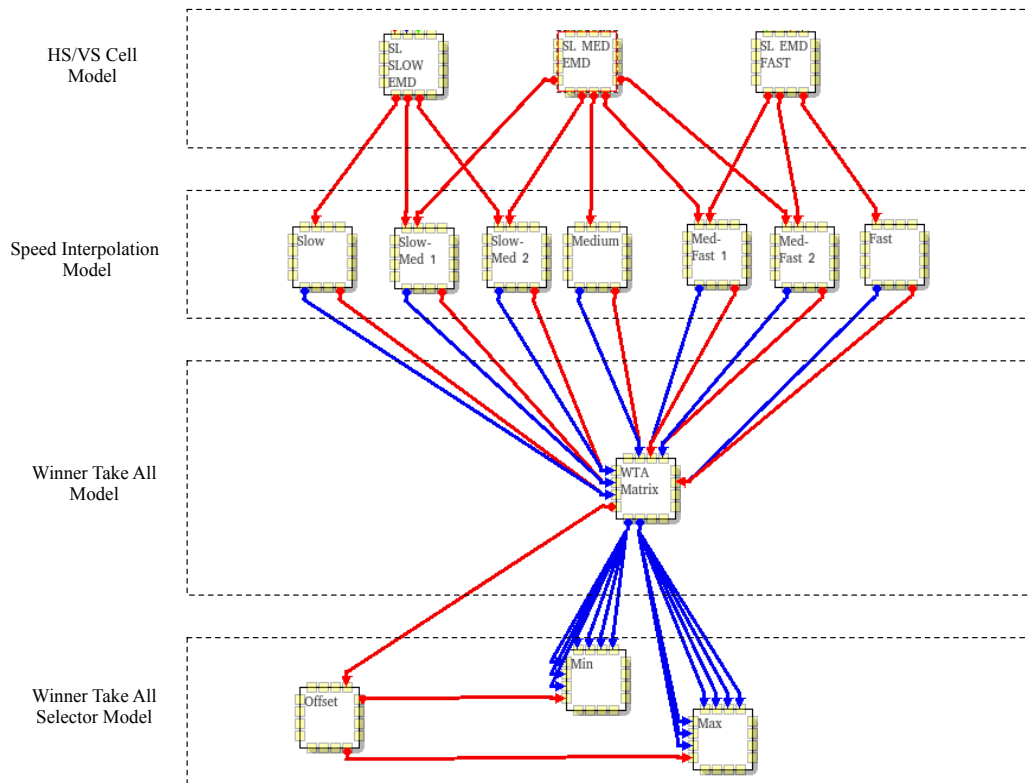


Figure 4.29 – *iqr* implementation of the WTA model with modifications for representing the maximum and minimum values as a neuron that spikes and for implementation on an AR.Drone 1.0.

for the delay of the signals propagating through the WTA model. This implementation in *theiqr* neural simulator can be seen in figure 4.29 with the modification being labelled as *Offset*.

4.3.5.3 Results

To test the model of the winner take all operation the model was implemented in *theiqr* neural simulator. The implementation can be seen in figure 4.30

The input to the comparison matrix, which is of size 7×7 , is 7 neurons. The values of each of these neurons can be manipulated by the user using the state manipulation panel in *theiqr* neural simulator. The neurons are connected to the comparison matrix through excitatory and inhibitory synapses. The connection topology between the user input neurons and the comparison matrix is defined such that each neuron within the comparison matrix represents one of the entries of the mathematical formulation of the comparison matrix, as seen in equation 4.3.21.

All the neurons in a row or column of the comparison matrix are connected via excitatory synapses to the neural groups of size 1×7 and 7×1 . These neuron groups represent the minimum and maximum vectors \mathbf{V}_{min} and \mathbf{V}_{max} .

All the elements of the WTA matrix are connected to the startup compensation neuron, which can be seen as the neuron labelled *Offset* in figure 4.30, via excitatory connections. The start-up compensation neuron is an integrate-and-fire neuron with a threshold voltage

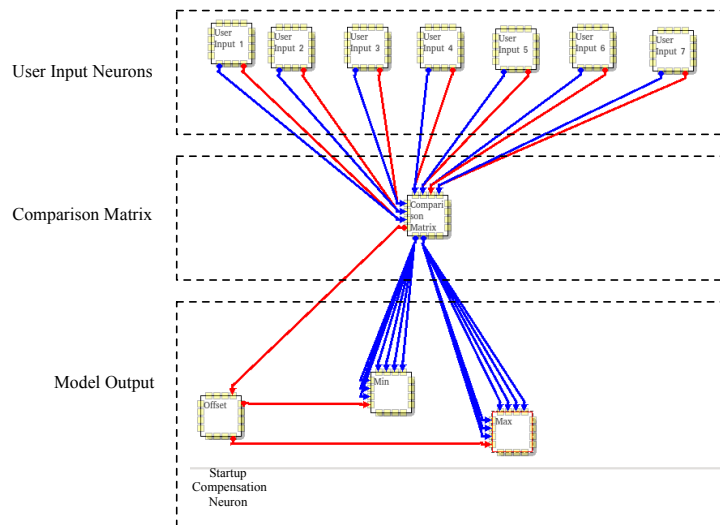


Figure 4.30 – Theiqr system used to test the Winner Takes All model. Red and blue connections indicate excitatory and inhibitory synapses respectively

of 10^{-5} v and a spike amplitude of 1v. This means that any activity in the comparison matrix will cause the neuron to fire. This start-up compensation neuron is then connected to each of the neurons in the neuron groups that represent the outputs of the model. The model was tested by subjecting the model to a variety of input conditions using theiqr simulator's state manipulation panel. The values of the neurons within the neuron group representing the entries in the comparison matrix were checked to see that the comparison matrix does indeed correctly perform the comparison between inputs. Next the neural groups representing the outputs of the model were checked to see if the model is giving the correct outputs. All of these factors were checked both during the start-up and the steady state operation of the system.

The first test was run under ideal conditions, meaning that there was only one minimum and one maximum input to the model. In this test the first user input neuron had a membrane potential value of 0.1v and each subsequent neuron's value was increased by 0.1v until the maximum value of 0.7v was reached. The result of this test can be seen in figure 4.31b.

The second test that was run investigated what happens when two minimum values are present in the input to the system. In this case the same values as the first test were used except the input neuron which had a membrane potential of 0.7v was made equal to 0.1v. The results of this test can be seen in figure 4.31c.

The third test measures the response of the system when two maximum values are input to the system. In this case the same values as the first test were used except the input neuron which had a membrane potential of 0.1v was made equal to 0.7v. The results of this test can be seen in figure 4.31d.

The start-up response was seen to be the same for all of the tests that were run and can be seen in figure 4.31a.

These tests showed that the model was able to perform the winner take all operation. A limitation of the model is that neuron failure in any of its parts will strongly effect its output. For example, if a neuron representing an input, or a neuron in the comparison matrix, were to fail, then the entries in the comparison matrix would no longer be a true representation of what is actually happening. If a neuron in the neuron group representing the vectors

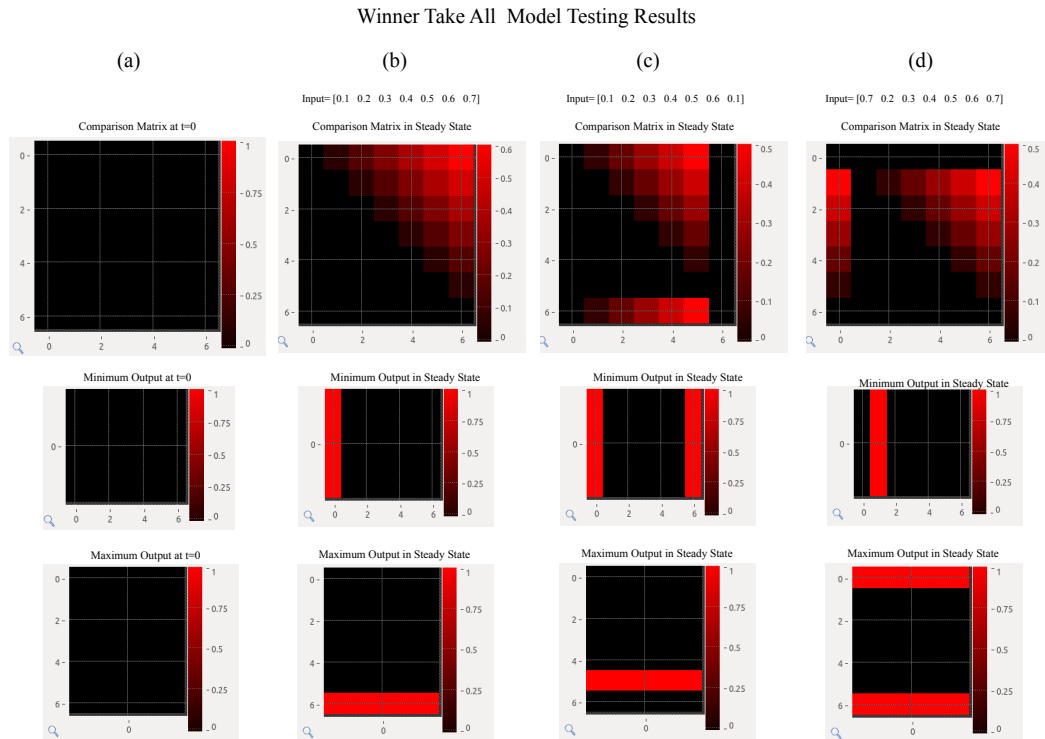


Figure 4.31 – Results of testing the WTA model under various inputs

V_{min} and V_{max} were to fail (i.e. have no membrane potential), this would cause the model to incorrectly identify the maximum and minimum inputs to the model.

4.4 Collision Detection: The Lobula Giant Movement Detector

4.4.1 Introduction

In this section a model for a biologically-inspired collision detection system is developed. The model is based on the Lobula Giant Movement Detector (LGMD) of the locust (for more information see section 2.1.2.4). The model draws much of its inspiration from [66] where a model of the LGMD was used on a blimp to detect collisions. The model receives visual input from the output of the lamina model described in section 4.2.2, and detects the expansion and contraction of objects in consecutive frames of the video stream.

Expansion and contraction are detected using a structure similar to the HS/VS cells of the fly, section 4.3.3, but with a different connectivity mapping of the array of EMDs. The outputs of an array of EMDs sensitive to expansion and contraction are then passed to a leaky integrate-and-fire neuron which integrates the contraction and expansion experienced over time to detect collisions.

4.4.2 Method

The output of the model developed to describe the behaviour of the LGMD is in the form of spikes emitted by a LI&F neuron. The frequency of these spikes is determined by the speed of approach to the impending collision. This is done by passing the signal from the lamina model through an array of EMDs with a remapped connection topology. The

outputs of these EMDs are then integrated in a similar manner to the HS/VIS cells described in section 4.3.3 to give a measurement of the whole field expansion present in the image. The modification to the HS/VIS cells that was proposed in section 4.3.3.2 to make the output of the HS/VIS cell applicable to robotics was also implemented. The output of this HS/VIS cell is then fed to a LI&F neuron which spikes when an approaching collision is detected. This structure of this model can be seen in figure 4.32.

The first step in the model describing the LGMD is the detection of expansion and contraction of objects in the image. This is done through the use of an array of EMDs, as can be seen in section 4.3.2, whose connection topology has been remapped so that the preferred direction is sensitive to expansion, and the null direction sensitive to contraction, of objects in the visual field.

To achieve this sensitivity to expansion and contraction of objects with regards to the center of the image, the input to the preferred branch of the EMDs is an expanded version of the neural group representing image output by the lamina. The amount of expansion in the input to the preferred branch of the EMDs is indicated by the spatial separation, D , as seen in figure 4.32. The input to the null branch is an unmodified version of the neural group representing image output by the lamina. The rationale for this setup is that when an object is approaching it will appear increasingly larger in the visual field, leading to correlation of the current image at time t with the expanded image at time $t - \delta$.

The output of the EMDs are then integrated by the HS/VIS cells. A modification is added to make the HS/VIS cells' output independent of the amount of activity present in the neurons representing the image output by the lamina. The result is a neuron whose membrane potential increases when expansion is detected.

The output of the HS/VIS cell in the LGMD is then connected to a LI&F neuron via an excitatory connection. The LI&F neuron integrates the response of the HS/VIS cell and releases a spike once a certain threshold of expansion in the image over time is exceeded.

4.4.3 Results

In this section, the ability of the LGMD model to detect impending collisions is investigated. First, the effect of the threshold potential of the LI&F neuron, which forms the model's output, is investigated. Next the effect of the approach speed towards the impending collision on the model's performance is investigated. Then the effect of the manipulating the various parameters which give the EMDs in the EMD array their properties is investigated. The model's ability to detect impending collisions with different types of objects is then investigated. Lastly, the effect of neuron failure in the EMD array is investigated.

To test the model of the LGMD the model was implemented in the `neurosim` neural simulator. The EMD array was composed of 64x48 EMDs, the same amount as the neural group representing the output of the lamina model. The input to the preferred branches of the EMDs was a neural group of size 64x48 with a one-to-one connection topology with the EMDs in the EMD array. The input group of 64x48 neurons was made by taking a group of neurons of size 60x44 centred around the middle of the neural group of the output of the lamina model and expanding it to a size of 64x48 neurons. This expansion can be thought of in terms of EMD parameters as setting D equal to two. The front camera of AR.Drone 1.0 was used as the visual input to the lamina model.

In the first test the effect of the membrane persistence of the HS/VIS cell, which is the pre-synaptic neuron to the neuron responsible for the model's output, as well as the effect of the speed of approach to the impending collision would have on the model's output was investigated. This was done so that the underlying effects of both of these factors on the performance of the model could be understood in context of one another, to provide a better understanding of the model's performance and the factors that influence it.

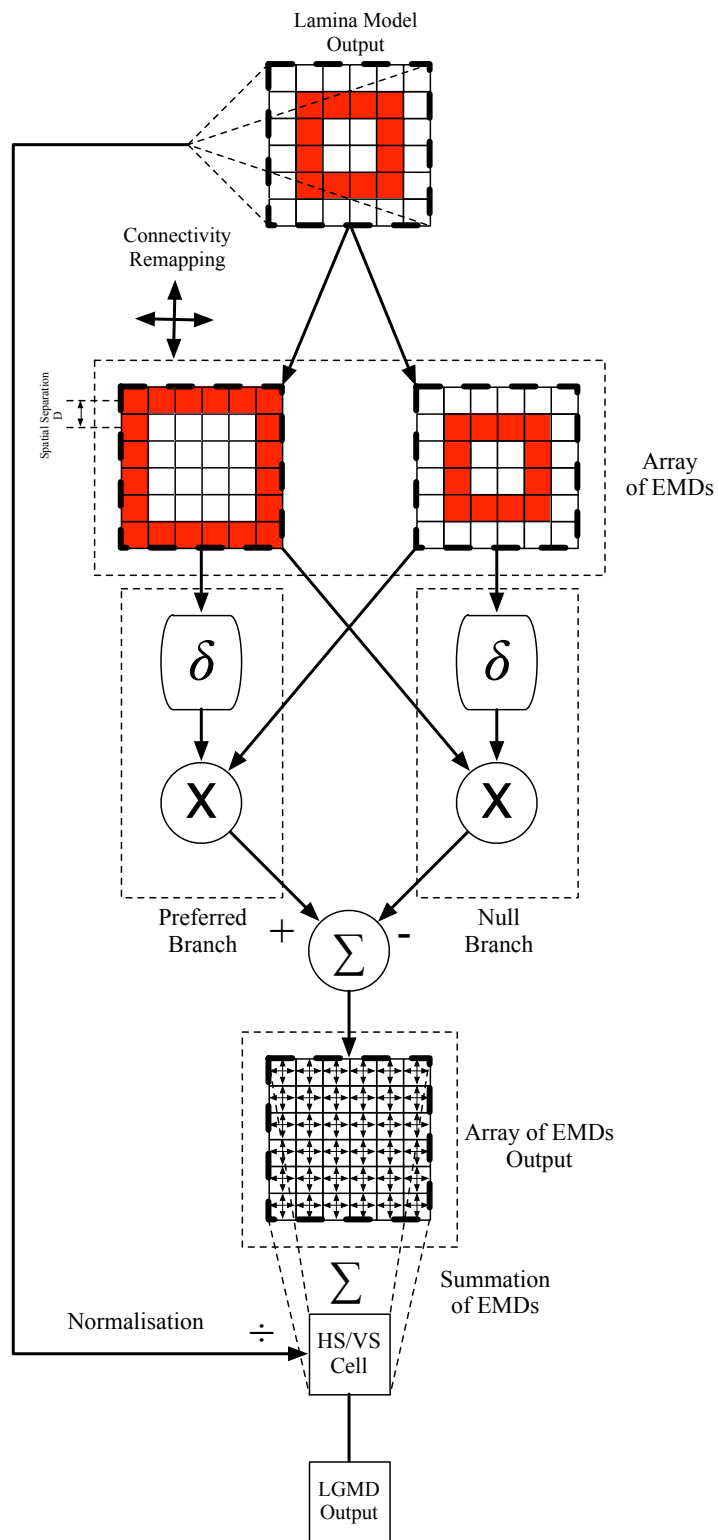


Figure 4.32 – Structure of the model describing the LGMD

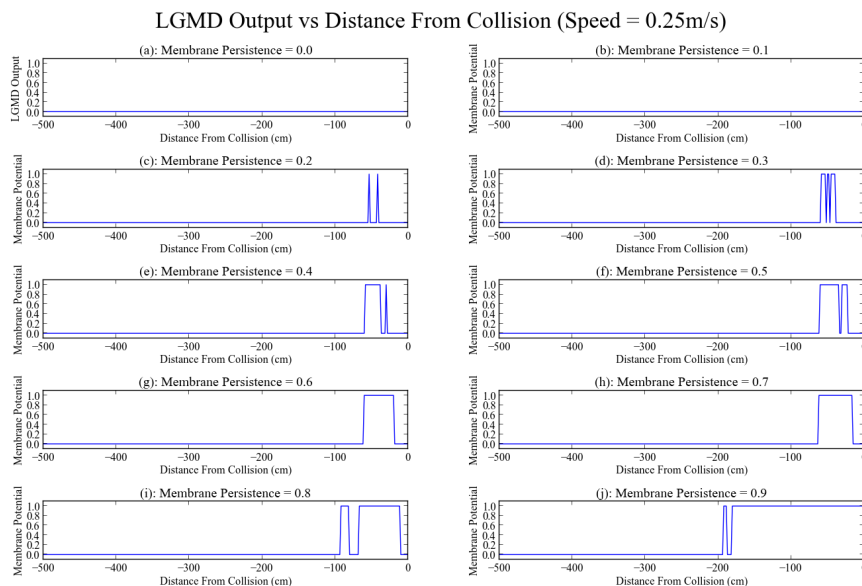


Figure 4.33 – Output of the LGMD model for different membrane persistence values of the HS/VIS cell while on a collision course at $0.25m/s$

To test the effect of the membrane persistence and the approach speed on the model's performance, the AR.Drone 1.0 was placed in the testing environment, described in section 3.4. The testing environment consisted of black A4 sheets of paper pasted against a white background. The AR.Drone 1.0 was placed 5m away from one of the walls of the testing area and moved at a constant speed towards it. The speeds investigated were $0.25m/s$, $0.5m/s$, $0.75m/s$ and $1m/s$. Nine identical models of the LGMD were implemented in the *iqr* neural simulator with the exception of the membrane persistence which was varied from 0.0 to 0.9 in increments of 0.1. The threshold potential for all the LGMDs output neurons was kept constant at a value of $v_t = 1v$. Both the firing of the LGMDs output neuron as well as the membrane potential of the HS/VIS cell was measured. For the speeds $0.25m/s$, $0.5m/s$, $0.75m/s$ and $1m/s$ the output of the LGMD model can be seen in figures 4.33, 4.34, 4.35 and 4.36 respectively and the membrane potential of the HS/VIS cells can be seen in figures 4.37, 4.38, 4.39 and 4.40 respectively.

The results of these tests indicate that the model is unable to detect impending collisions at low levels of membrane persistence when the the model's output neuron has a firing threshold of $v_t = 1v$. This can be ascribed to the normalisation of the HS/VIS cell, as discussed in section 4.3.3.2, used to make the model of the HS/VIS cell give an output that is independent of the amount of neural activity present in the output of the lamina model. The modification restricts the maximum value of the HS/VIS cell to one. This dependence of the model on membrane persistence is a desirable characteristic as the detection of a collision requires continual expansion of an object in the visual field, a strong indication that one is on a collision course. This reduces the risk that the model might detect an impending collision when high levels of expansion occur between just two consecutive frames, which is not necessarily a strong indicator of an impending collision.

From figures 4.33, 4.34, 4.35 and 4.36 it can be seen that the membrane persistence effects how early collisions are detected. The higher the membrane persistence, the earlier the LGMD model is able to detect a collision. The membrane potential is non-linearly related to the distance at which the impending collision is detected, as shown in figure 4.33.

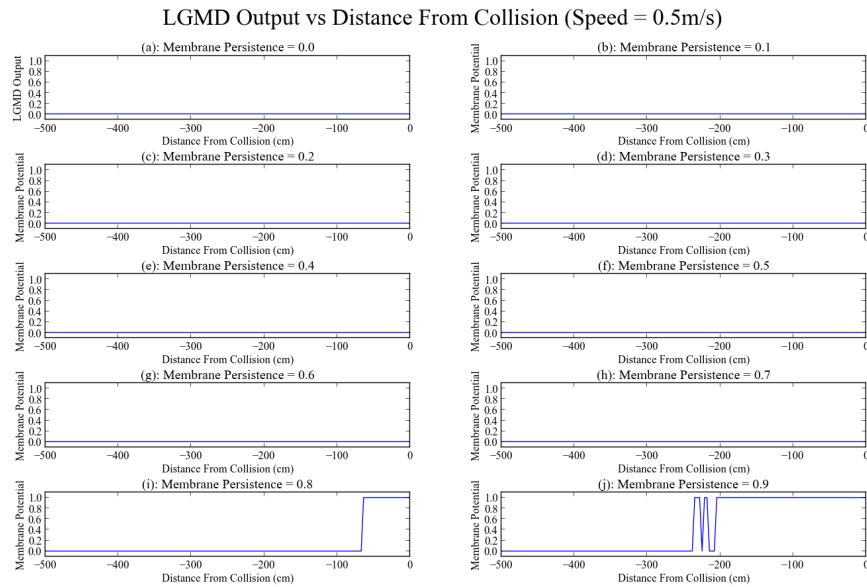


Figure 4.34 – Output of the LGMD model for different membrane persistence values of the HS/VIS cell while on a collision course at $0.5m/s$

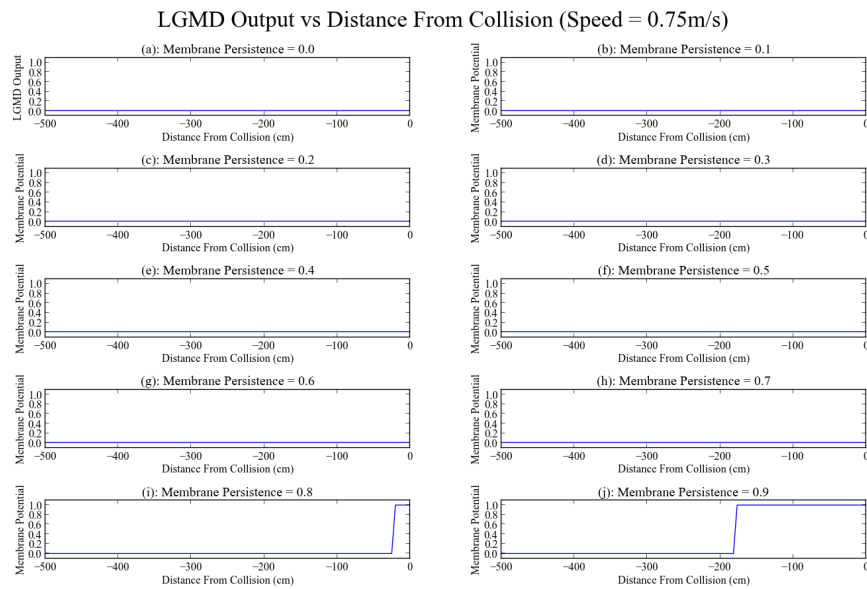


Figure 4.35 – Output of the LGMD model for different membrane persistence values of the HS/VIS cell while on a collision course at $0.75m/s$

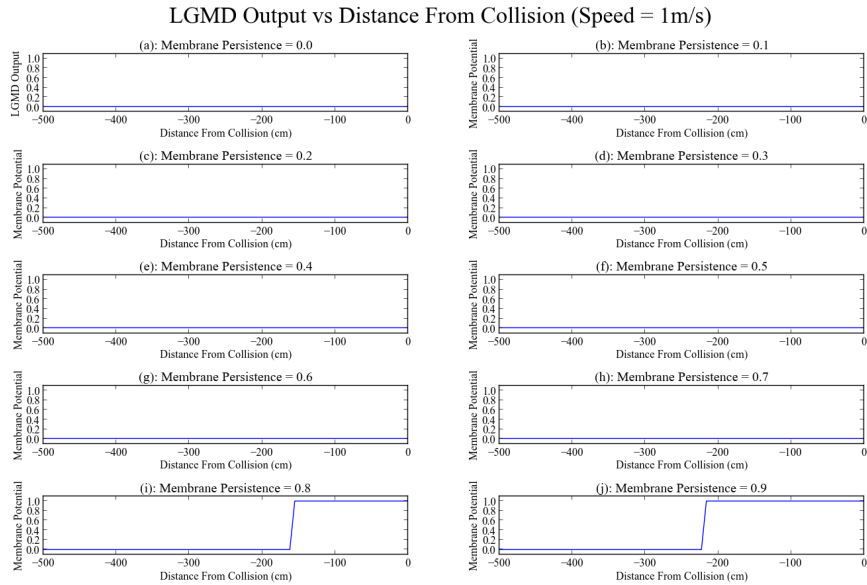


Figure 4.36 – Output of the LGMD model for different membrane persistence values of the HS/V_S cell while on a collision course at 1m/s

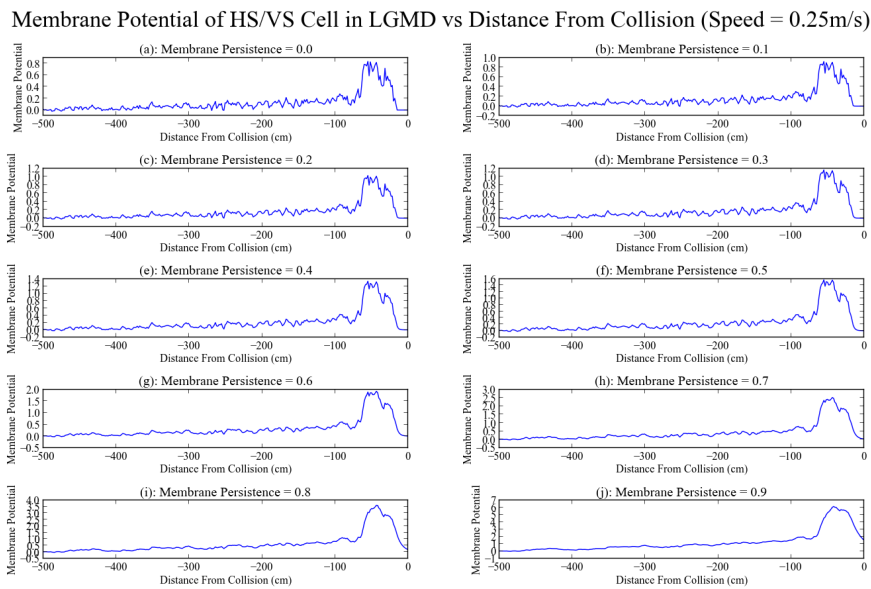


Figure 4.37 – Membrane voltage of the HS/V_S cell of the LGMD model for different membrane persistence values while on a collision course at 0.25m/s

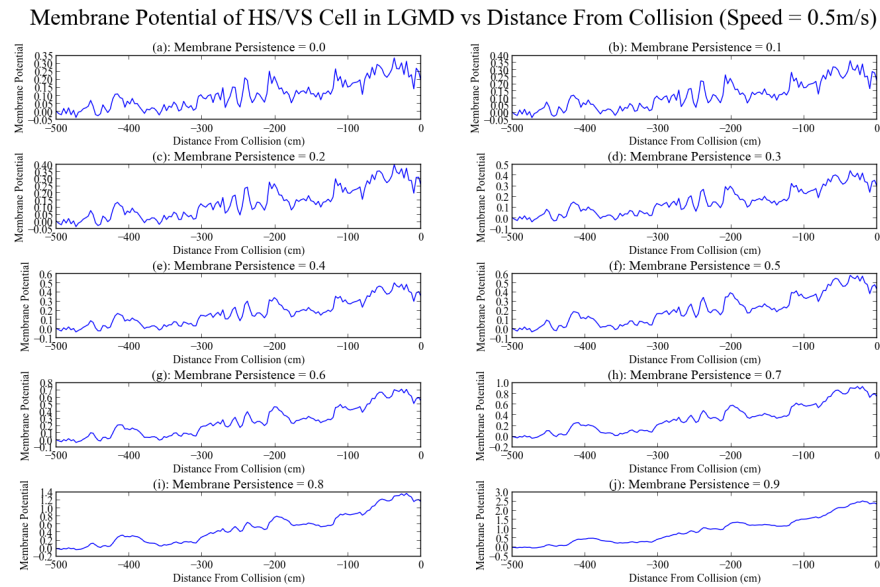


Figure 4.38 – Membrane voltage of the HS/V5 cell of the LGMD model for different membrane persistence values while on a collision course at 0.5m/s

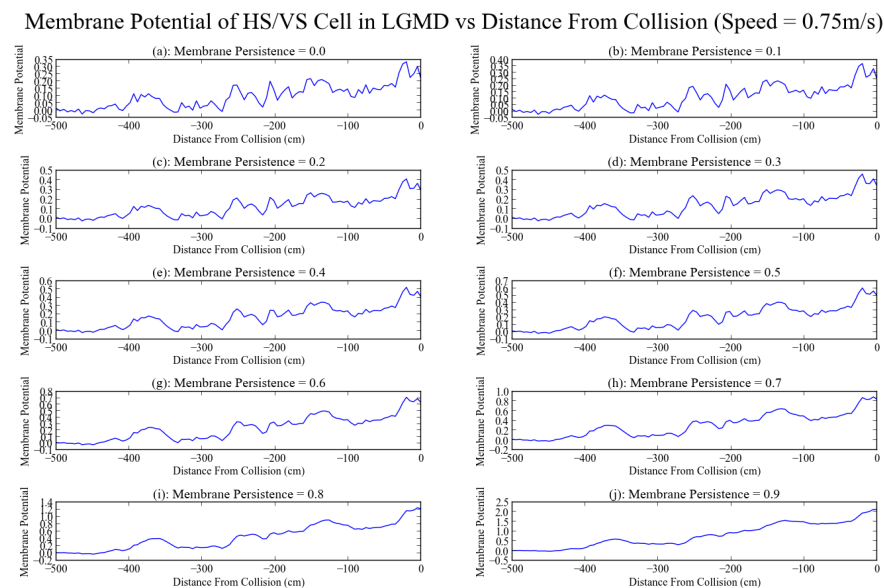


Figure 4.39 – Membrane voltage of the HS/V5 cell of the LGMD model for different membrane persistence values while on a collision course at 0.75m/s

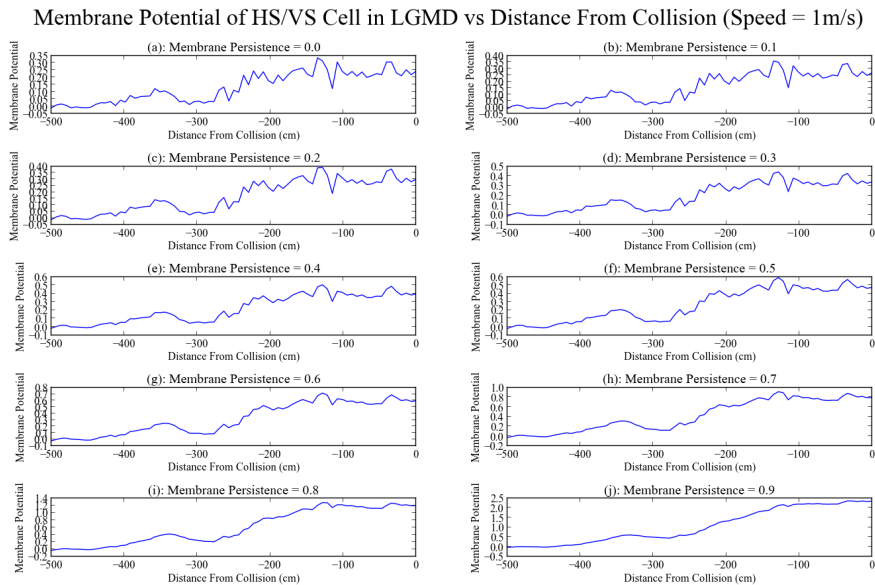


Figure 4.40 – Membrane voltage of the HS/VS cell of the LGMD model for different membrane persistence values while on a collision course at $1m/s$

These tests also reveal that the LGMD model is sensitive to collisions occurring at different speeds. From figure 4.37 it is seen that even for high membrane persistence values the membrane potential of the HS/VS cell when approaching a collision at $0.25m/s$ stays relatively low until 90cm before the collision. From equations 4.3.7 and 4.3.8 in section 4.3.2.2 this behaviour can be understood as an emergent property of the parameters of the EMDs used to detect expansion and contraction in the visual field. The application of this property allows the parameters in the EMDs to be manipulated so that the LGMD model can be made sensitive to different speeds.

In figure 4.37 it is seen that the membrane potential quickly increases from 90cm away from the impending collision and then starts to decrease 50cm from the impending collision. This can be explained by the sensitivity of the EMDs, as well as the perceived size of the objects in the visual field. The perceived expansion of an object moving towards an observer increases as the object gets closer to the observer. An example of this is if you place your hand at arms length in front of your face then bring it 5 cm closer to your eye. The hand appears to expand compared with other objects in the visual field. Repeating this procedure but starting with your hand 10cm from your eye and moving it 5 cm closer, results in the expansion of the hand appearing much larger, even though it was moved the same distance toward the eye. The membrane potential in figure 4.37 can now be understood. When approaching the collision, the perceived expansion corresponds strongly to the expansion to which the EMDs are sensitive. However, when the collision is 50cm away the perceived expansion becomes much greater than that to which the EMDs are sensitive, leading to less cross-correlation in the array of EMDs and a lower membrane potential in the HS/VS cell. This explains the results shown in figure 4.16, where the responses of differently tuned EMDs are shown to be symmetric around the speeds for which they were tuned. It is also seen that for speeds of $0.5m/s$, $0.75m/s$ and $1m/s$, the membrane potential of the HS/VS cell, seen in figures 4.38, 4.39 and 4.40, respectively, steadily increases but never reaches the peak values of the response at $0.25m/s$. This indicates that at the higher speeds the perceived expansion of the object on a collision course with the system never reaches the maximum expansion to which the EMDs are sensitive. It also explains why there is no

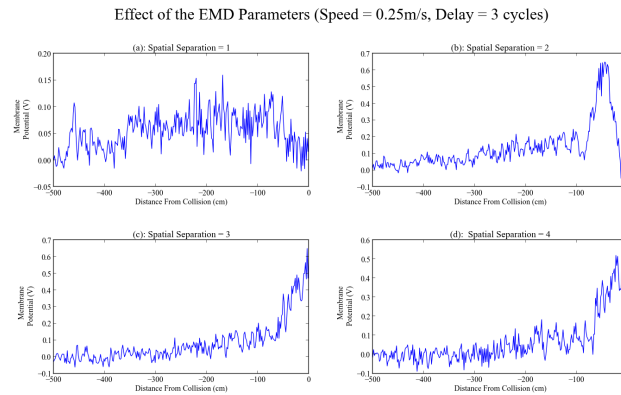


Figure 4.41 – Effect sensitivity of the EMDs on the membrane potential of the HS/VS cell of the LGMD

decrease in membrane potential immediately before the collision takes place at speeds of $0.5m/s$, $0.75m/s$ and $1m/s$.

Next, the effect of the parameters of the EMD, in particular the ratio $\frac{D}{\delta}$, described in section 4.3.3, is investigated. This was done by varying the spatial separation, D , of the EMDs between 1 and 4 while keeping the delay, δ , constant at 0.2s. The system was set on a collision course with one of the walls of the testing area at a constant speed of $0.25m/s$ and the membrane potential of the HS/VS cell of the model measured. The results of this test can be seen in figure 4.41.

From figure 4.41, it is seen that the parameters influencing the sensitivity of the EMDs also effect the membrane potential of the HS/VS cell and hence the performance of the entire model. This agrees with the observations made when examining the sensitivity of the HS/VS cell to different speeds. At a spatial separation of 1, the HS/VS cell was not able to detect motion reliably, because the perceived expansion never corresponded to the expansion to which the EMDs are sensitive. For a spatial separation of 2, figure 4.41b, it was seen that a collision is able to be detected. For spatial separations of 3 and 4, figures 4.41c and d, it was seen that the membrane potential of the HS/VS cell increases sharply just before the collision. This can be explained due to the object's perceived expansion approaching the amount of expansion which the EMDs are sensitive to but never reaching or exceeding it. This also explains why a decrease in membrane potential just before the collision is not seen in figure 4.41b.

Next, the effect of the threshold potential of the neuron defining the LGMD model's output is investigated. This was done by approaching a collision with one of the walls of the testing area at a speed of $0.25m/s$. The membrane persistence of the HS/VS cell in the LGMD was set to zero and the results are shown in figure 4.37a. The threshold voltage of the neuron defining the model's output was varied from 0.1v to 0.9v, in increments of 0.1v. The output of the LGMD can be seen in figure 4.42.

In figure 4.42 it is seen that the threshold voltage plays a role in both how well and how early a collision can be detected. For very low threshold voltages, figures 4.42a and b, the collision could not be detected. This can be attributed to the fact that for the model to signal an impending collision, low amounts of activity in the HS/VS cell are needed. This leads to external factors, such a lighting conditions, effecting the model's ability to detect a collision. For levels of membrane voltage above 0.3v it seems as if all collisions are able to be detected at approximately the same distance before the impending collisions. However, at threshold voltages of 0.9v and above no collision was able to be detected.

Effect of Threshold Potential on LGMD Model Output (Speed = 0.25m/s, Membrane Persistence = 0)

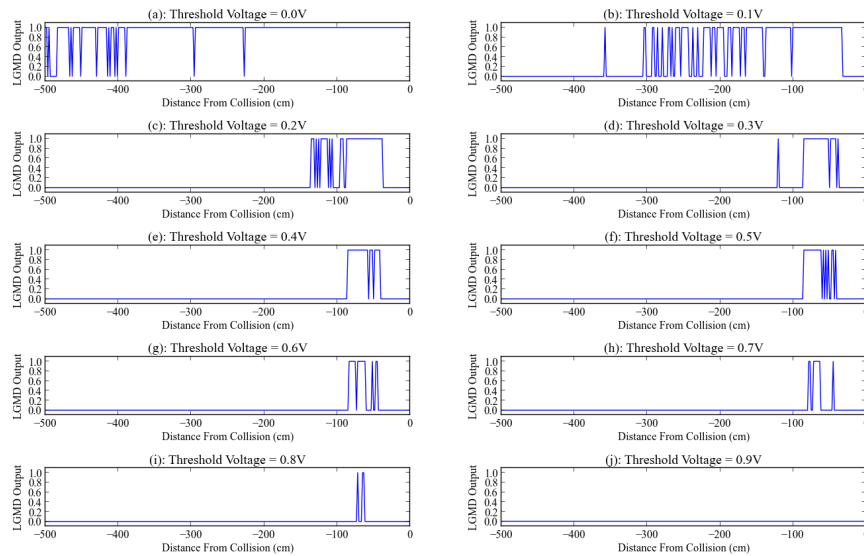


Figure 4.42 – Effect of V_t on the LGMD model's performance

Effect of Neuron Failure in the Preferred Branch of the LGMD EMD (Speed = 0.25m/s)

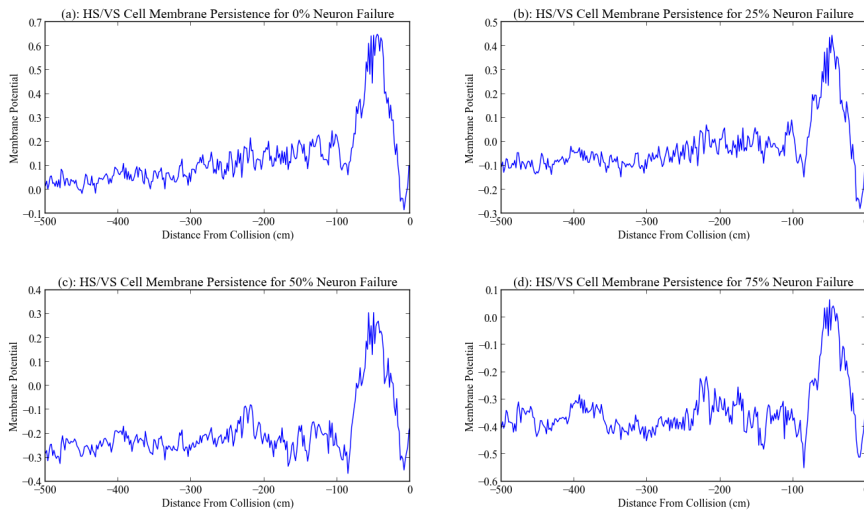


Figure 4.43 – Effect of neural failure in the preferred branch of the EMDs in the EMD array of the LGMD

The effect of neuron failure within the EMD array of the LGMD model was investigated by testing where components in the preferred branch, null branch and entire EMD had a defined probability of failing. First, the effect of neuron failure in the preferred branch of the EMD was investigated. The probability of failure was varied between 0 and 1 in increments of 0.25 and the effect of this on the membrane potential of the HS/Vs cell integrating the array of EMDs recorded. The results of these tests can be seen in figure 4.43.

From figure 4.43 it can be seen that the affect of neuron failure in the preferred branch

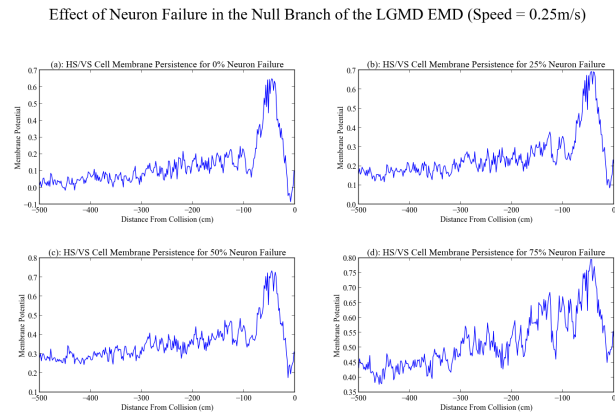


Figure 4.44 – Effect of neural failure in the null branch of the EMDs in the EMD array of the LGMD

of the EMD is the same as failure in the HS/VVS cell in section 4.3.3. Failure results in a negative bias in the membrane potential of the HS/VVS cell. The amount of bias depends on the probability of neuron failure. This can be attributed to the fact that neurons in the preferred branch do not detect cross-correlation when there is failure. Subtracting the null branch response from that of the preferred branch introduces the bias. It is also seen that the original shape of the response was maintained relatively well. Due to the fact that there were no constraints imposed on which preferred branch neurons failed (i.e. a neuron that either did or did not detect correlation might have failed), there is no way of calculating the exact bias introduced by the neuron failure or the exact change in shape of the plot of the HS/VVS cells membrane potential over time. The effect of failure in the null branch of the EMDs in the EMD array, figure 4.44, results in the same behaviour, except that a positive bias is introduced instead of a negative one.

The same method was used to investigate the effect of a total failure of the EMDs on the membrane potential of the HS/VVS cell in the LGMD. The results (figure 4.45) show that the effect of EMD failure on the HS/VVS cell membrane potential is similar to that described in section 4.3.3. The shape of the membrane potential of the HS/VVS cell is also maintained somewhat, with more distortion at higher levels of neuron failure. The amplitude of the response of the HS/VVS cell decreases as membrane failure increases. This is because the normalised signal coming from the lamina maintains its strength whereas the total amount of activity from the array of EMDs decreases due to the failure of individual EMDs.

4.5 Colour Extraction and Tracking

4.5.1 Introduction

This section describes the model designed to track a certain colour in the visual environment. The model was developed by using insight gained from working on the biologically-inspired models described previously. The model separates a specific colour, in this case red, from the visual environment and sends appropriate motor commands needed to keep the red object in the centre of the visual field.

Effect of Neuron Failure in the LGMD EMD (Speed = 0.25m/s)

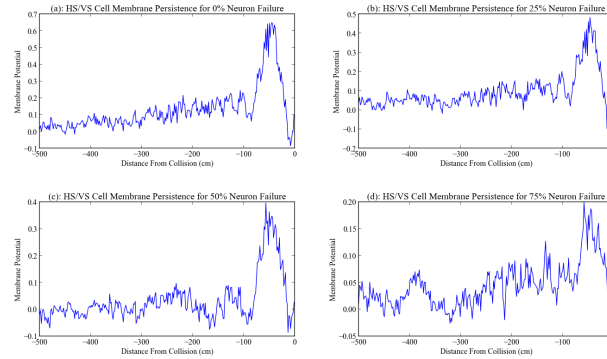


Figure 4.45 – Effect of total failure of EMDs within the EMD array on the membrane potential of the HS/VS cell

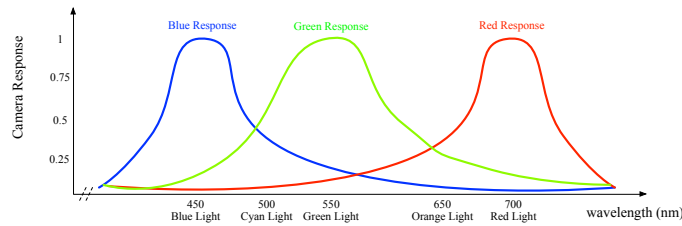


Figure 4.46 – Example of the RGB colour model

4.5.2 Colour Extraction

4.5.2.1 Introduction

The video stream from the AR.Drone is represented using the RGB colour model [76]. The use of the RGB colour model leads to some colours being represented as a combination of red, green and blue. This can be seen in figure 4.46 where the sensitivities of the three primary colour groups making up the RGB model, are plotted against the wavelength of the light. This causes the neurons representing the colour to be tracked, red in this case, to be detected in objects that would not necessarily be considered red by an observer. For example, in figure 4.46, orange can be seen to be a combination of red and green light. In this section a method to extract red from the other colours present in the image is developed.

4.5.2.2 Method

The visual input to the colour tracking model comes from the front camera of the AR.Drone 1.0. The video feed from the camera is in the form of the red, green and blue values of the current video frame and is passed to three separate neuron groups, each sensitive to a different colour.

To separate red effectively from green and blue, the outputs of the green and blue neuron groups is used to inhibit activity of the neuron group that forms the output of the colour extraction model. The excitation from the red neuron group is then passed to this new neuron group via an excitatory synapse. The ratio between the strength of the inhibitory and excitatory synaptic weights is the *GainRatio* and its formulation is shown in equation 4.5.1.

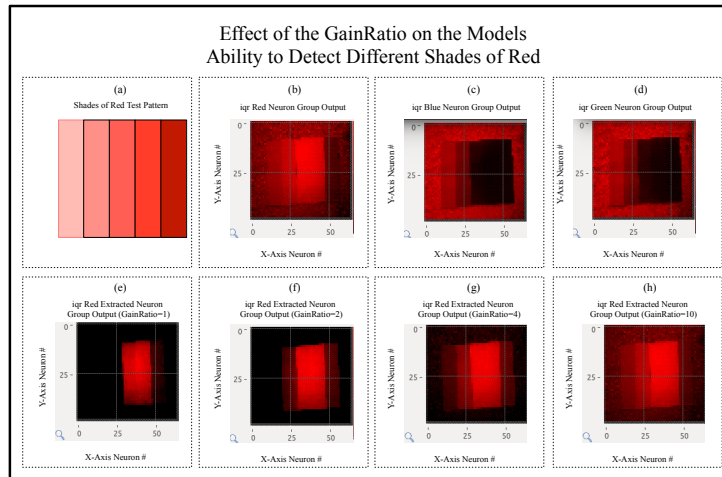


Figure 4.47 – Effect of the *GainRatio* on the colour extraction model’s performance

$$GainRatio = \frac{excitatorysynapticweight}{inhibitorysynapticweight} \quad (4.5.1)$$

The reasoning behind this connection topology is that for colours containing a combination of red and another colour, such as green in figure 4.46, the neuron activity that represents green will suppress the activity of the neurons representing red. The neurons in the group of neurons representing the image from which red is extracted will only show a positive membrane potential for objects whose only primary colour is red.

The membrane potential of the colour-extracted neuron group is connected to a group of the same size and is composed of leaky integrate-and-fire neurons, with a membrane persistence of 0. The reason for this is that the sensitivity to how red an object in the visual field must be before it causes the neurons representing it to fire, can be varied by adjusting the threshold potential of the LI&F neurons.

4.5.2.3 Results

In this section the colour extraction model is tested. First the ability of the model to differentiate red from green and blue is tested. Then the effect of the threshold potential of the LI&F neurons on the model’s ability to differentiate between different shades of red is tested.

The model was implemented on the `iqr` neural simulator, with visual input to the system coming from the front camera of the AR.Drone 1.0. The camera was pointed towards a piece of paper which consisted of horizontal stripes with varying intensities of red, as seen in figure 4.47a. The responses of the neuron groups representing the red, blue and green components of the image can be seen in figures 4.47b, c and d respectively. The *GainRatio* was then varied to see what effect it had on the colour extraction. The results can be seen in figures 4.47e, f and g and show that at a lower *GainRatio* the system is better able to detect red in the incoming image.

Next the effect that the threshold voltage of the LI&F neuron group representing the output had on the model’s ability to perform colour extraction was investigated. A *GainRatio* of 4 was used, as can be seen in figure 4.48b. The threshold potential values tested were $0.1v$, $0.25v$, $0.5v$ and $0.75v$ and the results are shown in figures 4.48c, d, e and f. It can be seen

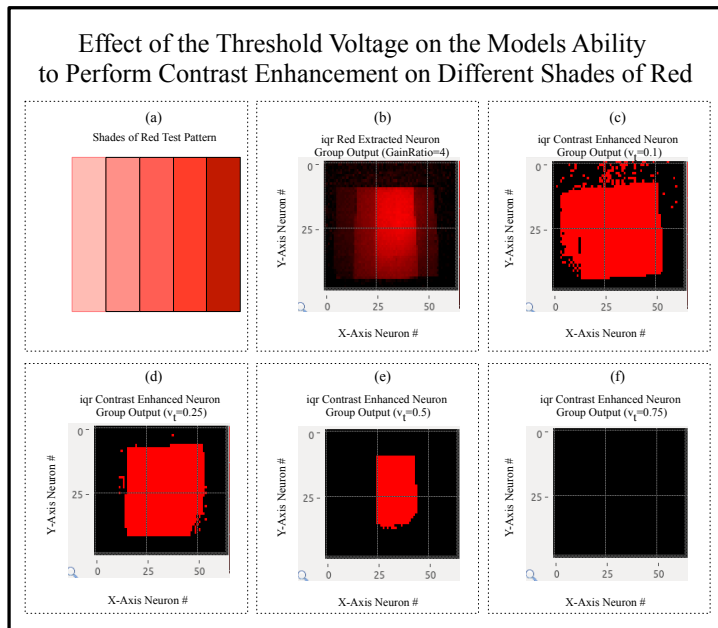


Figure 4.48 – Effect of the threshold voltage on the colour extraction model's performance

that at a very low threshold voltage, $0,1v$, excessive noise diminishes the model's ability to perform colour extraction. If the threshold voltage is increased to $0,25v$, the model is well able to separate light shades of red from the visual environment. When increasing the threshold voltage further, to $0,5v$, only the darker red shades are detected. At a threshold voltage of $0,75v$, no neurons fire.

These results show that both *GainRatio* and threshold voltage affect the model's performance. Both factors can be manipulated to determine which shade of red is detected. It was also seen that too high a threshold potential causes no colour to be detected.

4.5.3 Colour Tracking

4.5.3.1 Introduction

In this section a method to track the colour that has been extracted using the model developed in section 4.5.2 will be discussed. The model that is developed is independent of the size of the object in the visual field. The output of the model is a group of 4 neurons, each of which indicates the percentage of the object which is detected in the upper, lower, left and right sections of the visual field. The output of the model can then be used to send the commands to the system to keep the colour in the middle of the visual field. This has been done on the AR.Drone 1.0 and will be discussed in detail in section 5.2

4.5.3.2 Method

The visual input to the colour tracking model comes from the neuron group representing the output of the colour extraction model developed in section 4.5.2. This neuron group is divided into 4 sections, each representing a third of either the top, bottom, left or right of the video stream coming from the drone's camera. The excitation of all the neurons representing a particular section of the video stream are connected via excitatory synapses

to the output whose membrane potential indicates the percentage of the object being tracked that is within that section's visual field.

A normalisation technique is used to make the output neurons represent the percentage of the object that occurs in the section of visual field, rather than the total activity within that section. The normalisation method is similar to that used in the model of the HS/VS cell, but with some modifications. The first modification is that the input does not come from an array of EMDs but rather from the neurons that represent an image that has undergone colour extraction. The second is that, instead of a connection topology using the same presynaptic neurons that connect via excitatory synapses to the postsynaptic neuron, in the normalisation of the colour tracking model the activity of all the neurons representing the entire visual field is used. Since the only factor that will cause neural activity in the input to this model is an object with the particular colour being tracked, the activity of the neuron is dependent only on the percentage of the object found in the specific section of the visual field. This can be seen in equation 4.5.2.

$$\textit{TrackingOutput} = \frac{\textit{SectionActivity}}{\textit{AllActivity}} \quad (4.5.2)$$

4.5.3.3 Results

To test this model, it was implemented in the `theiqr` neural simulator. The model receives input from a user-generated pattern using the state manipulation panel of the `theiqr` neural simulator that simulates the input of the colour extraction model section 4.5.2.3. The size of the input is a 64x48 neuron group, which is the same size as the neuron group responsible for video input from the AR.Drone 1.0. The left and right sections of the visual field are composed of neuron groups of size 22x48 and the top and bottom sections are composed of neuron groups of size 64x16.

The model was first tested with the coloured object present only in one section of the input image. Next it was tested with the object in two sections of the image. Lastly, the effect of the object exceeding the space designated as the centre of the visual field was determined.

For the first test, the user input was a square of 10x10 neurons firing with an amplitude of 1. The square was first placed on the border between the centre and upper sections of the visual field so that there was no overlap between the two sections. The activity pattern of the group of neurons representing the output of the colour extraction model and the membrane potential of the neurons representing the output of the colour tracking model were measured. The results are shown in figure 4.49. The square group of firing neurons was then moved up so that 50% of the square was found in the region defining the centre and 50% was found in the section defining the top of the visual field, as shown in figure 4.50. Finally, the square group of firing neurons was shifted up so that all of the firing neurons were found in the section of the image defining the top of the visual field, shown in figure 4.51.

To test the model's capability when the size of the object exceeds the space designated to be the centre of the visual field, the user input to the model was a square group of neurons of size 40x40 centred in the visual field. Both the activity pattern of the group of neurons representing the output of the colour extraction model and the membrane potential of the neurons representing the output of the colour tracking model were measured, and the results are shown in figure 4.52. The model was found to behave as expected, able to determine the percentages of the object that are located in the top, bottom, left and right sections of the image.

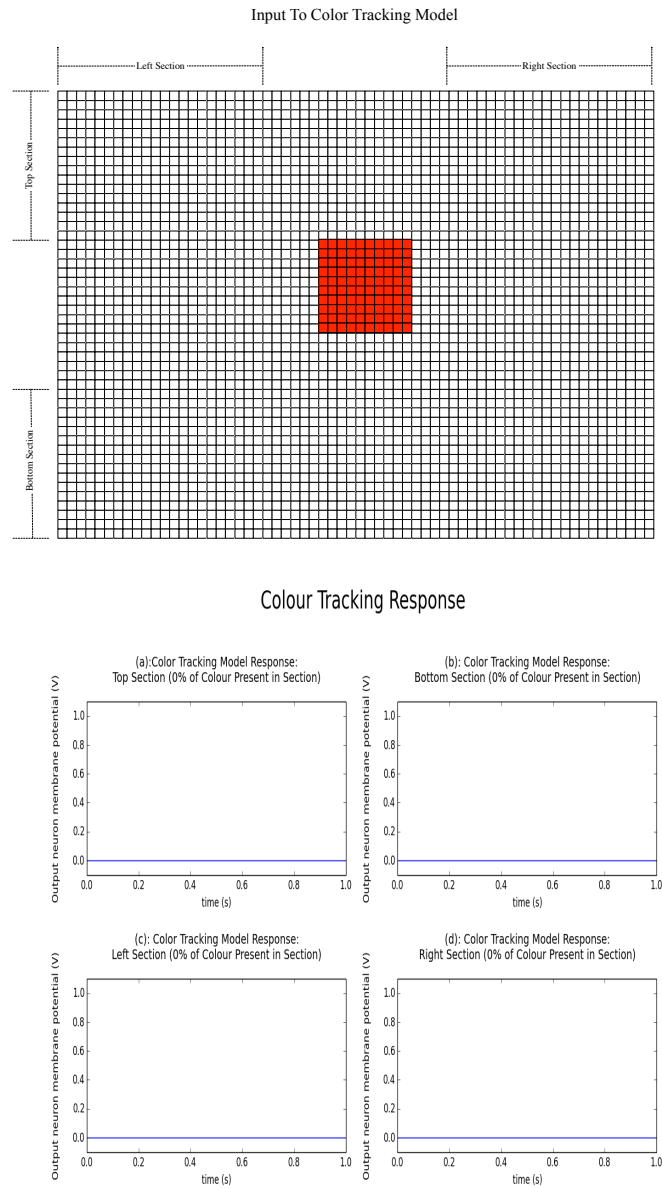


Figure 4.49 – Colour tracking model with no input to colour sensitive sections. Top: Neural activity. Bottom: Model output neuron voltages

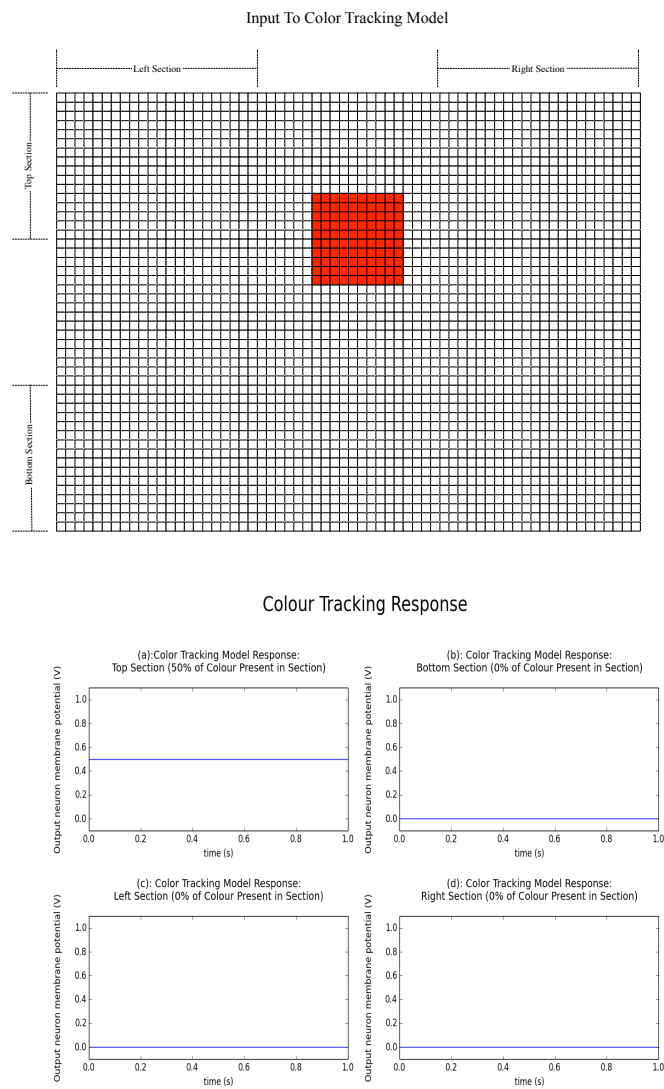


Figure 4.50 – Colour tracking model with 50% of input to colour sensitive section at the top of the image. Top: Neural activity. Bottom: Model output neuron voltage

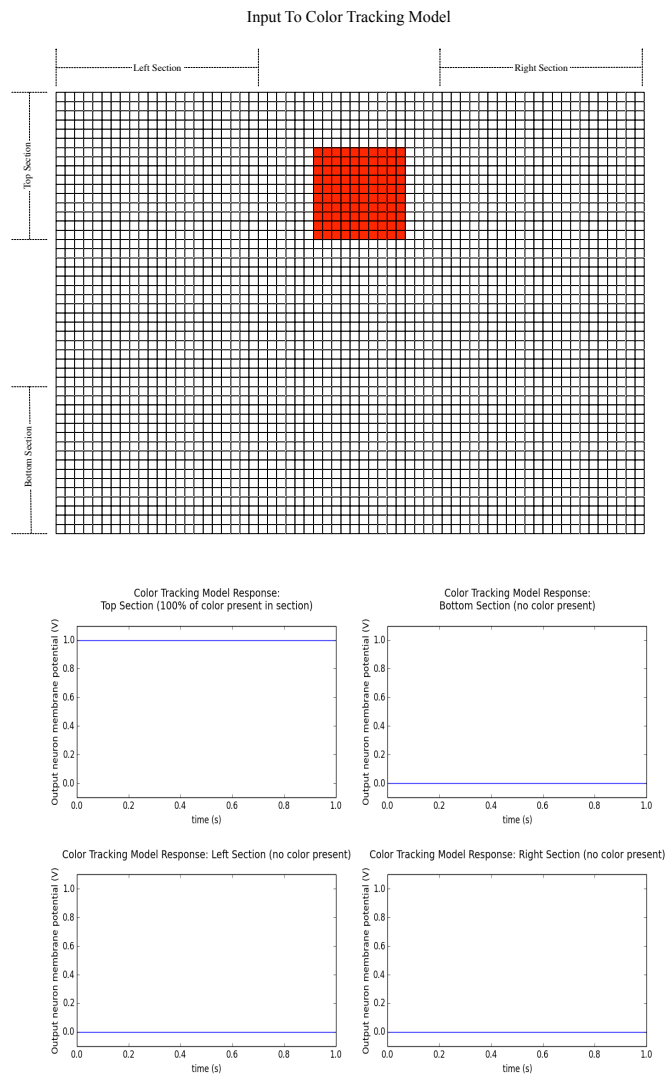


Figure 4.51 – Colour tracking model with 100% of input to colour sensitive section at the top of the image. Top: Neural activity. Bottom: Model output neuron voltage

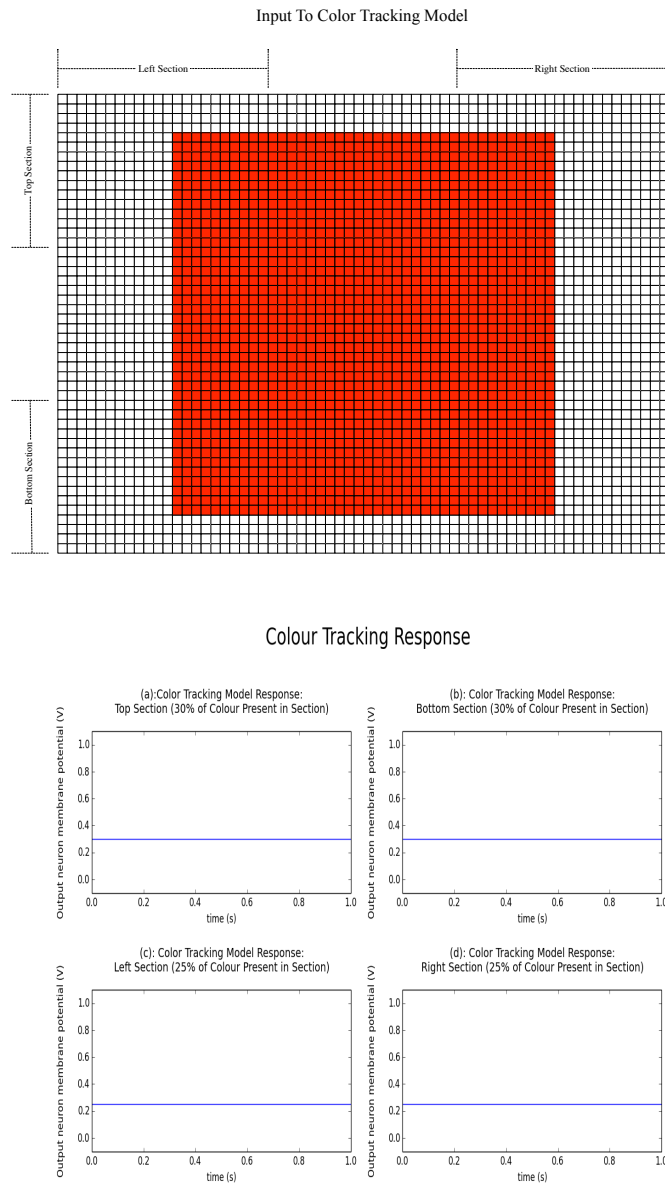


Figure 4.52 – Colour tracking model with a square 40 neurons in length and located in the centre of the frame given as the input. Top: Neural activity. Bottom: Model output neuron voltage

Chapter 5

Experimental Investigation

In this chapter the experiments performed to test the viability of using biologically inspired models for autonomous flight of a drone, are described. Using the hardware, software and data collection methods described in chapter 3, and the biologically inspired neural models of section 4, neural systems developed during the course of this research are implemented within the `iqr` neural simulator.

The first of these systems makes use of the models of the lamina: the elementary motion detector, HS/VS Cells and the LGMD. The purpose of the system is to test the feasibility of using biologically inspired models to fly the drone autonomously in a constrained environment. The constrained environment consisted of the 6 x 6m testing area, discussed in section 3.4.

The second system consists of the models for colour extraction and colour tracking. This system is required to track a red ball hanging from a string over an S-shaped path in the testing area as described in section 3.4. During the test, the ball is moved up and down. The drone is expected to maintain a constant following distance from the object, based on the perceived size of the object in the visual field.

5.1 Autonomous Flight in a Constrained Environment

5.1.1 Introduction

This experiment makes use of the lamina, elementary motion detector, HS/VS Cells and the LGMD to investigate if biologically inspired models are capable of flying the drone autonomously in a constrained environment. A successful experiment is defined by the system's ability to control the flight of the AR.Drone in the testing area, so that it does not collide with a wall, for a period of five minutes.

Before performing the experiments, the drone was flown with an unoptimised version of the system in place. During these flights the effect of the network parameters, i.e. the neuron parameters and synaptic weights, was investigated. This was done by manually manipulating the network parameters and observing the effect this had on the system's performance. This manual manipulation of parameters was done using `iqr`'s graphical user interface to change the values of various parameters of the neurons and synapses within the system. The results are discussed in section 5.1.5.

5.1.2 Motivation for Undertaking

This experiment was performed to investigate if biological models are able to control the flight of the AR.Drone in an environment of which it has no prior knowledge. If successful, this experiment will show that biologically inspired neural models can indeed be used for controlling the flight of an autonomous aircraft, while using only the camera as a universal sensor. Also, if the experiment is successful, it will encourage further development of new paradigms for the autonomous control of vehicles.

During the optimisation of the system, the effects of the network parameters on the system's behaviour as a whole can be analysed. By understanding the manner in which the structure and neural parameters affect the system's behaviour, better choices can be made when developing biological models to describe behaviour, as well as when creating the systems that make use of those models.

Furthermore, the implementation of the system on the drone allows us to identify the practical issues that might arise when implementing the biological models in real world systems.

5.1.3 Experimental Setup

The first step that is required for autonomous navigation in the constrained testing environment is to receive video input from the AR.Drone 1.0 and to pass it on to neural groups in the `iqr` neural simulator. For this, an `iqr` module that takes the incoming RGB values from the video stream received from the front camera of the drone, and converts them to neural activity, was used. This neural activity is then passed to three groups of linear threshold neurons, each of size 64x48 and representing one of the red, green or blue values of the image contained in the video stream of the drone. The excitation of each of these neural groups is then combined by passing their neural activity to a group of linear threshold neurons of size 64x48, representing the total RGB input received from the drone

Next, the signal from the RGB neural group is passed via excitatory connections to the neural model of the lamina that was described in section 4.2.2. The size of all neural groups in the lamina model is the same as that of the neural group representing the input received from the AR.Drone 1.0. The connection topology of the synapses connecting the RGB input to the neuron group representing the edge detected image can be seen in figure 4.3 of section 4.2.1. The inhibitory window width was chosen to be 5 neurons wide while the excitatory windows inner and outer value were 5 and 7 neurons respectively. These are the same values that were used for the testing of the system. The threshold voltage, v_t , of the leaky integrate-and-fire neurons that describe the contrast enhancement model of the lamina was chosen to be equal to one volt.

The output from the lamina is fed into two parallel models. The one simulates course stabilisation in the HS/VIS cells described in section 4.3. The other simulates the collision avoidance performed by the LGMD described in section 4.4.

The first neural structure in the course stabilisation model consists of three arrays of EMDs, each tuned for a different speed. The speed tuning is done through the connection topology of the network. Since the view angle of the AR.Drone 1.0 is 93° , and the neural group carrying the video stream from the camera is 64 neurons wide, equation 5.1.1 describes the angular separation of each neuron. A physical representation of this can be seen in figure 5.1.

$$NeuronAngularSeparation = \frac{ViewAngle}{\#Neurons} = \frac{93^\circ}{64} = 1.453125^\circ \quad (5.1.1)$$

From the angular separation and the time delay of the branches within the EMD, the angular speed can be calculated. Since `iqr` makes use of a discrete time paradigm between simulation cycles, the time delay can be calculated from the delay in simulation cycles. The system

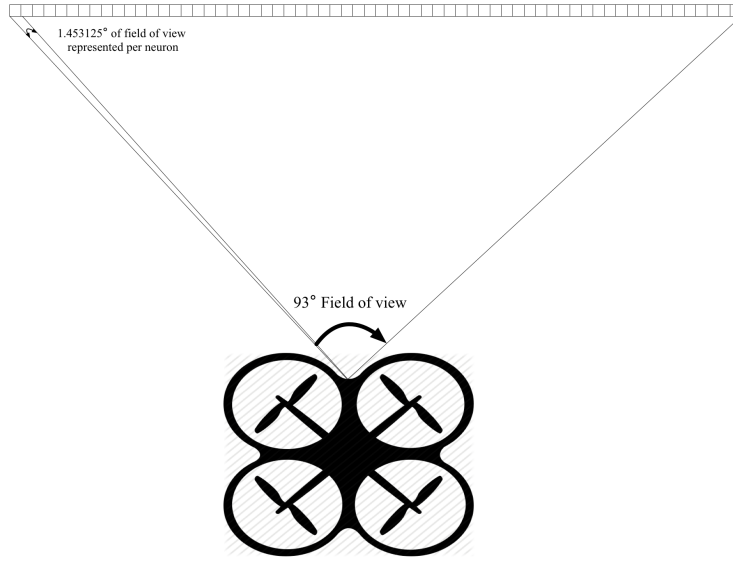


Figure 5.1 – View angle represented by each neuron in the drone's field of view

will run optimally at the frame rate of the drone's front camera that provides the input to the system. That frame rate is 15 frames per second as can be seen from the hardware specifications of the drone, given in section 3.2.2. The total time delay is a function of the number of simulation cycle delays and is given by equation 5.1.2.

$$TimeDelay(\#SimulationCycleDelays) = \frac{SimulationCycleDelays}{SimulationSpeed} \quad (5.1.2)$$

From the time delay, angular separation and simulation time, the angular speed for which the EMD has been tuned can then be calculated using equation 5.1.3.

$$EMD_{TunedSpeed} = \frac{NeuronAngularSeparation \cdot SimulationSpeed}{\#SimulationCycleDelays} \quad (5.1.3)$$

The three arrays of EMDs that receive their input from the lamina are sensitive to slow, medium or fast speeds. The inputs to EMDs representing the slow speed are spatially separated from one another by one neuron only, with the preferred branch receiving input from the leftmost neuron. The delay between inputs of the EMDs is 5 cycles, making them sensitive to motion at $4.359375 \frac{\circ}{s}$, with a preference for leftward motion. The EMDs tuned for medium and fast speeds are spatially separated by two and three neurons respectively, with the leftmost neurons being the input to the preferred. Both the medium and fast EMD have a delay of 5 cycles, making them sensitive to motion at $8.71875 \frac{\circ}{s}$ and $13.078125 \frac{\circ}{s}$, respectively, with a preference for leftward motion in both cases.

The responses of the three EMD arrays are then fed into three HS/VS cells to determine the whole field motion present at the particular speed to which the EMDs, that supply input to the HS/VS cells, are sensitive. The output neuron of the HS/VS cells has been converted to a linear threshold neuron, as done in section 4.3.4, so that the output of the HS/VS cell model can be input to the speed interpolation model. These HS/VS cells are equally spaced in terms of their sensitivity to angular speed and are also broadly tuned, thus satisfying the requirements for an input to the speed interpolation model described in section 4.3.4.

The outputs of the HS/VS cells are split by passing the signal coming from the neuron representing the output of the HS/VS cells to a linear threshold neuron via an excitatory

synapse, and to another linear threshold neuron via an inhibitory synapse. This allows the motion in the preferred direction to be represented by a positive membrane potential of the neuron connected via the excitatory synapse. Motion in the null direction is represented by the neuron connected via the inhibitory synapse. A threshold potential of $0.1v$ was used so that any small values which are not a strong indication of motion in that direction, are filtered out.

The responses of the neurons representing movement in the preferred and null directions are each projected independently onto separate sets of 7 projection neurons. These projection neurons represent angular speeds between $4.359375 \frac{\circ}{s}$ and $13.078125 \frac{\circ}{s}$ at an equal spacing of $1.24553571 \frac{\circ}{s}$.

The activities of the projection neurons of the preferred and null branches are then passed to their own neural models describing the winner takes all operation, seen in section 4.3.5. Each of the neurons used to represent the maximum vector, seen in equation 4.3.24, is connected to the neuron group responsible for making the drone move. The neurons representing the output the winner takes all operation on the input, which represent motion in the preferred direction (leftwise motion), are connected to the neuron responsible for making the drone go left. Similarly, the output of WTA structure that received its input from the null branch is connected to the neuron responsible for making the drone go right. The connection strength between the neurons representing the WTA matrix and the neuron responsible for the drone's movement is stronger for the neurons that represent a higher speed. This is done so that if a faster angular speed is detected, a stronger motor response is sent to the drone.

Although the outputs of both WTA models are used to send movement commands to the drone at the same time, it is assumed that there is no interference between the two outputs. This is due to the fact that when motion in one direction is detected by the HS/VS cell, there will be no motion detected in the other direction, when the output is split. The threshold potential of $0.1v$ of on the neurons that detect preferred and null direction motion also help to get rid of any small membrane potential excitement that is not an indicator of motion. Furthermore, the mechanism for converting neural activity to drone movement commands checks to see that opposing movement commands are handled in a sensible manner.

Collision avoidance for the system is implemented using the model of the LGMD. The model receives its inputs from the output of the model of the Lamina. First, the neuron group that represents the lamina model's output is split down the middle into two neuron groups of size 32×48 , which represent the left and right half of the original image.

Each of these new neuron groups is then connected to its own LGMD model (see section 4.4). There are 32×48 EMDs in the model of the LGMD. These EMDs receive input from the inner 28×44 neurons from the lamina model output that has been split. The 'image' that the signal from the 28×44 neurons, supplying input to the EMD array, is expanded to fit into a neuron group of size 32×48 . The delay of the EMDs, δ , in the EMD array used in the LGMD model, was chosen to be 3 simulation cycles.

The leaky integrate-and-fire neurons used to describe the output of the LGMD, are then connected to the neuron groups that are responsible for making the drone rotate left and right. The output of the LGMD model which receives input from neuron group that represents the left half of the visual field is connected to the neuron that controls the drone to rotate right, away from the collision. Similarly the output of the LGMD model that receives its input from the right half of the visual field is connected to the neuron responsible for rotation to the left.

The outputs of both of the LGMD models, representing the left and right field of view, are connected via excitatory synapses with synaptic weight of 0.5 to a leaky integrate-and-fire neuron with membrane persistence of 0 and a threshold potential of 1 volt. If both the LGMDs fire at the same time, which is possible in the cases of head-on collisions or when

heading into corners, this will cause the LI&F neuron to fire. The output of the LI&F is then connected via inhibitory synapses to the neurons responsible for the left and right rotation of the AR.Drone 1.0 causing the AR.Drone 1.0 not rotate when both LGMDs fire at the same time. The LI&F neuron is also connected via an excitatory synapse to the neuron that is responsible for making the drone move backwards, and thus away from away from the impending collision.

To test the system, the drone was placed in the centre of the testing facility, described in section 3.4. The drone was then allowed to fly freely for five minutes at a speed of $0.5 \frac{m}{s}$. This was done by attaching a neuron with a threshold potential of 0 to the neuron that controls forward motion in the group of neurons responsible for sending the control commands to the drone. The other flight control commands come from the output of the system that was described above. If the drone is able to navigate the testing environment for 5 minutes without crashing the test was considered successful. Ten tests were run, each time rotating the drones starting position 36° clockwise. As the drone uses the optical flow of its bottom camera to avoid any unwanted drifting while flying, the bottom camera was covered to test the ability of the biological model used for course stabilisations to stop this drifting. The system can be seen in figure 5.2.

5.1.4 Results

Factors that have a major effect on system performance are the synaptic weights and the neuron parameters specifically, the parameters that most effected the systems performance are the membrane persistence and threshold potential of the LGMD and the synaptic weights of the synapses connecting to the neurons used to send movement commands to the drone.

It was seen that the membrane persistence and threshold potential of the LGMD not only influenced the system's ability to detect and avoid collisions, but also influenced the how much of the testing area the drone explored. This can be understood by looking at the characteristics of the LGMDs output firing rate, figures 4.33, 4.34, 4.35, 4.36, as well as its membrane voltage, figures 4.37, 4.38, 4.39, 4.40. From these figures it is seen that the output of the LGMD is strongly dependent on the membrane persistence. This explains why the drone explores less of the available space in the testing area at higher levels of membrane persistence. It is due to impending collisions being detected sooner, thus triggering movements to steer away from these collisions before coming close to the walls of the testing area earlier. The neuron's membrane threshold affects the system similarly, and for the same reason a lower threshold potential means that a collision is sensed earlier, thus triggering an avoidance manoeuvre sooner, with the result that the drone explores less of the available space.

When looking at the synaptic weight of the synapses connected to any neuron in the neuron group responsible for sending motor commands to the drone the phenomenon of why small adjustments to the synaptic weight have a large effect on the system can be understood. For example consider the output of the WTA model in the course stabilisation system. The models output gives an indication of the actual speed at which objects in the visual field are moving. This output is connected, via excitatory synapses, to the neuron group responsible for sending motor commands to the drone to combat this motion. If the synaptic weights between the output of the WTA model and the neuron group responsible for motor output are not finely tuned, the drone overcompensates when movement is detected. This in turn causes the HS/VS cells to detect motion in the direction to which the drone is compensating and causes the drone to repeat the cycle.

Similar problems were seen with the LGMD, where sending too strong a motor response causes the drone to either rotate too much or fly backward, so the drone does not explore all parts of the testing area. Similarly, a low synaptic weight results in the drone crashing, even though a collision was correctly detected by the LGMD.

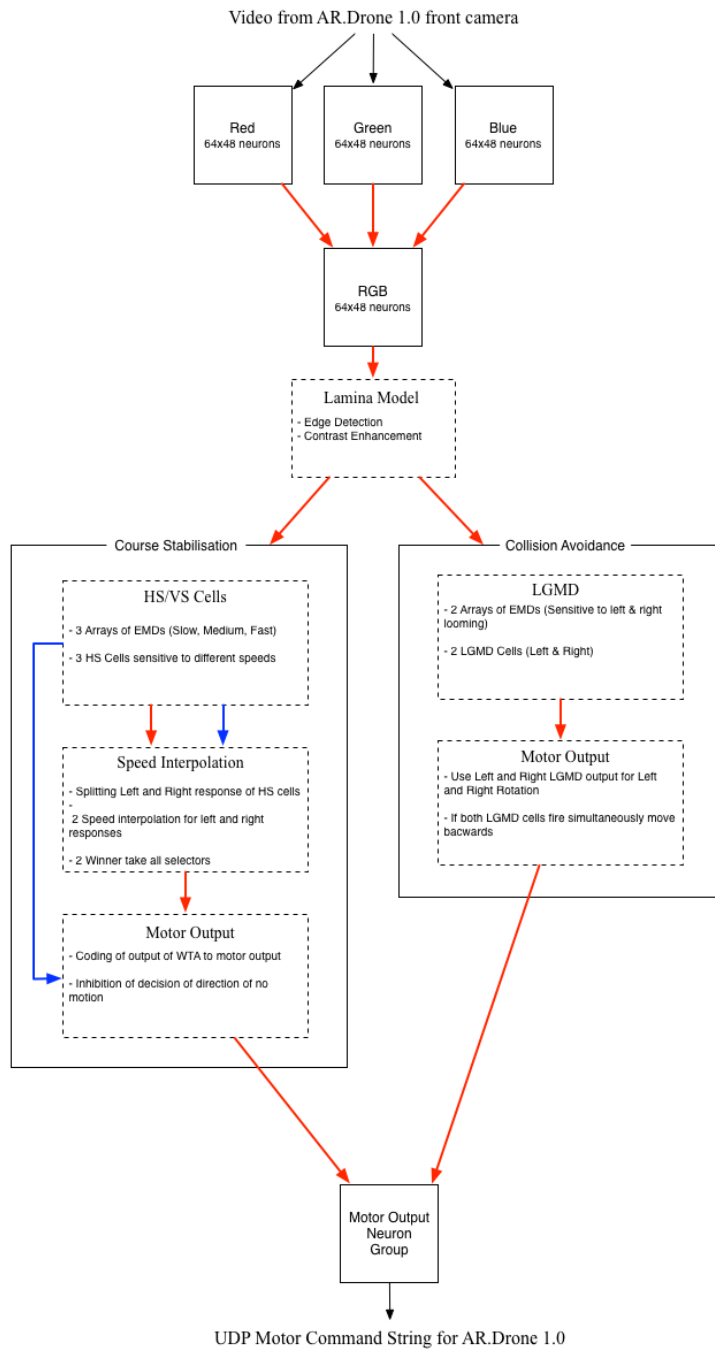


Figure 5.2 – System diagram for autonomous flight in a controlled environment experiment

The path taken by the AR.Drone for the 10 flight test runs can be seen in figure 5.3.

5.1.5 Interpretation of Results

It was seen that, by using biologically inspired methods for flight control, the drone was able to successfully navigate the testing area without colliding into a wall in 100% of the tests.

5.2 Colour Following

5.2.1 Introduction

In this experiment the colour extraction and tracking models developed in sections 4.5.2 and 4.5.3 are used to create a system that allows the drone to track and follow a coloured object detected in its visual field. The manner in which the system was evaluated was by suspending a red tennis ball by a string and tracing out an S-shaped curve, while also varying the height of the ball, and observing the ability of the drone to track the object. The experiment is considered to be successful if the system is able to follow the tennis ball through an S-shaped course while maintaining a reasonable following distance.

5.2.2 Motivation for Undertaking

This experiment is performed to investigate whether the models for colour extraction and tracking are able to control the drone's flight behaviour such that it is able to track a coloured object. Success of this experiment will show that biological models can be useful for goal-driven behaviour.

5.2.3 Experimental Setup

The first step in creating a system that enables the AR.Drone 1.0 to track and follow a coloured object, is to receive video input from the drone and pass it to neural groups with the *iqr* neural simulator. For this, an *aniqr* module which takes the incoming RGB values from the video stream received from the front camera of the drone, and converts them to neural activity, was used. This is done by creating three groups, each 64x48 neurons in size, of linear threshold neurons, section 3.1.2.4. Each has a membrane potential that depends on the intensity of a different primary colour (red, green or blue) in the image from the incoming video stream.

Next, the neural activity of the neuron groups representing red, green and blue are connected to another neuron group of linear threshold neurons, 64x48 neurons in size, using the connection topology defined in section 4.5.2. This forms the colour extraction section of the system. This neuron group is then used to supply input to the colour tracking model developed in section 4.5.3.

The outputs of the colour tracking model are connected to the neural group responsible for transforming neural activity to UDP control strings. The neuron that detects what percentage of the coloured object occurs within the top third of the image is connected to the neuron responsible for making the drone fly upwards. Similarly, connections are made to the neurons responsible for detecting the percentage of the coloured object detected in the left, right and bottom of the image.

A similar structure to that which detects what percentage of the colour-object is around the borders of the image, is used to determine the percentage of the coloured object within the centre of the image. This neuron is then connected to the neuron group which is responsible for making the drone move forward.

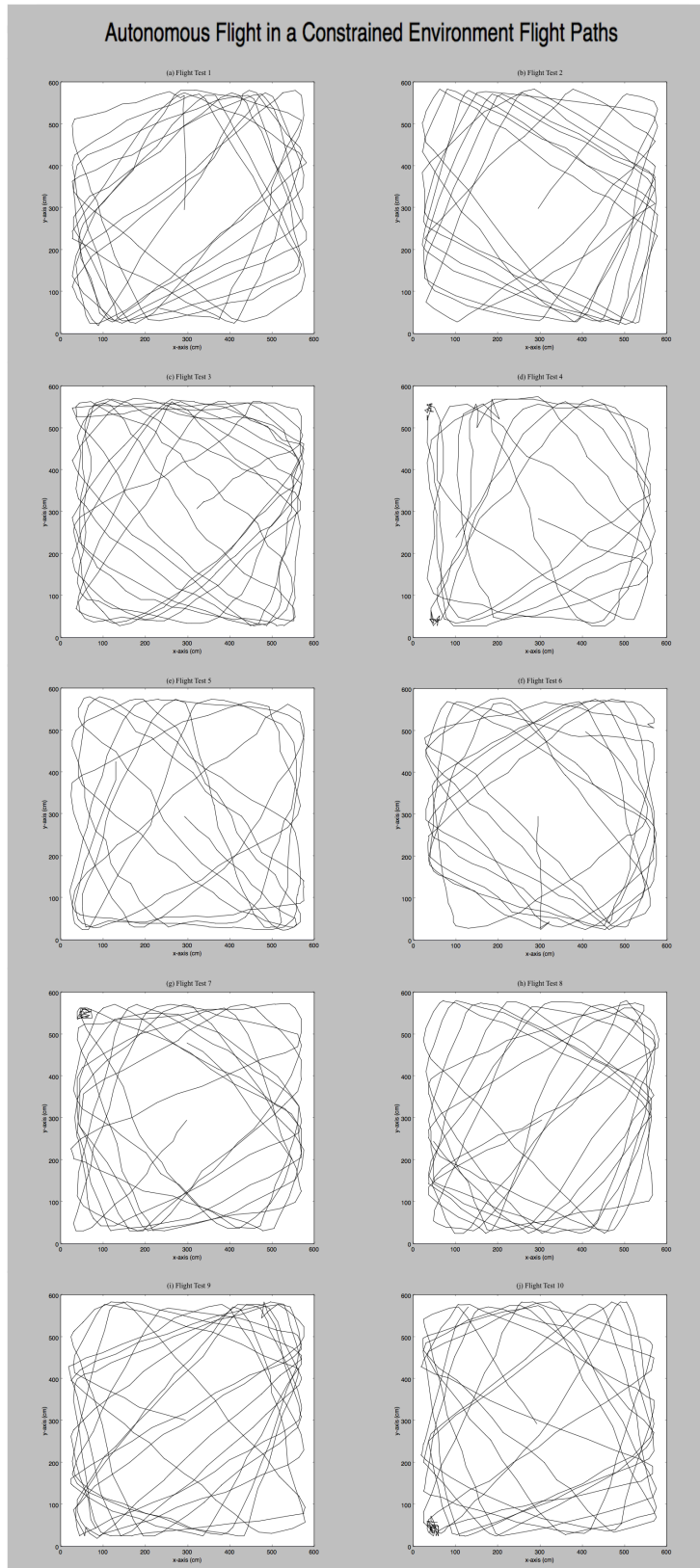


Figure 5.3 – Drone’s flight paths during the experiments performed to investigate autonomous flight in a constrained environment

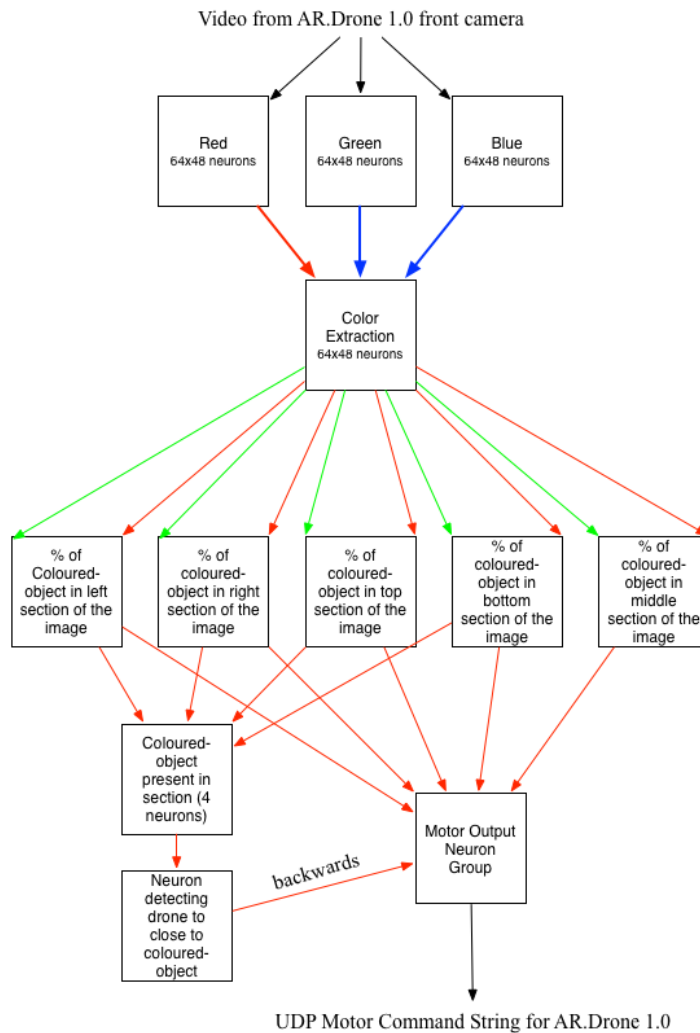


Figure 5.4 – The full colour following system

The responses of the four neurons that determine whether any part of the coloured object is located around the edge of the image, is passed to a group of 4 leaky integrate-and-fire neurons that represent the percentage of coloured object that was detected in that section of the image. These neurons fire with an amplitude of 1 and are connected, via excitatory synapses with strength 0.25, to another neuron with a threshold potential of 1. This neuron will thus fire only if the coloured object was present in all 4 bordering sections of the image. This neuron is connected to the neuron responsible for moving the drone backwards via an excitatory synapse with connection strength of 1. By exciting the neuron responsible for making the drone move backwards, by an amount greater than that of the neuron which is causing the drone to move forwards, allowing the drone to follow the system and maintain a following distance that depends on the perceived size of the object in the visual field. The entire system can be seen in figure ??.

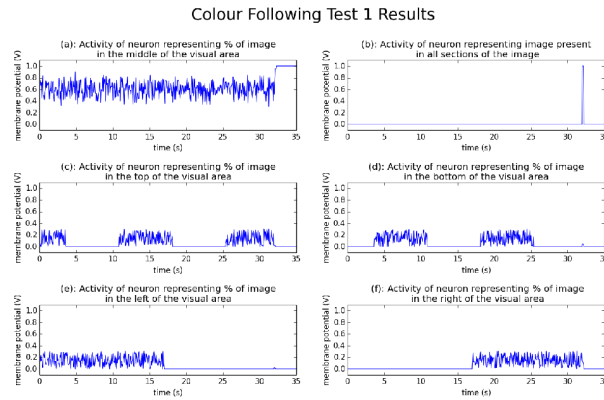


Figure 5.5 – Colour tracking experiment 1 results

5.2.4 Results

The system was tested by moving a ball horizontally over an S-shaped pattern, while simultaneously moving it up and down. This was repeated 10 times. A typical result from this experiment can be seen in figure 5.5. The results of the remaining 9 experiments can be seen in Appendix A. The figures show the membrane voltage of the neurons which represent the percentage of the object contained in a section of the image, as well as membrane voltage of the neuron responsible for detecting when the object is present in all sections of the image.

5.2.5 Interpretation of Results

From the results in section 5.2.4 it can be seen that the system is able to follow the coloured object correctly 70% of the time. In tests 2, 5 and 8, figures A.1, A.4 and A.7 respectively, it can be seen that the coloured object was no longer able to be tracked once it had moved out of the drone’s line of sight.

This can be explained by considering how the system operates. Motor commands are only sent to the drone based on the percentage of the object in the drone’s visual field. Once the object leaves the drone’s visual field, there is no longer any neural activity thus the drone has no way of sending movement commands.

For the other tests, figures 5.5, A.2, A.3, A.5, A.6, A.8 and A.9, it is seen that the system works as expected.

It was also seen that the model works as expected, maintaining a constant following distance from the object. This behaviour can be seen by looking towards the end of the test flight where the drone flies closer and closer to the object until the object is detected in all 4 borders of the visual field. At this point the system detects this and sends a motor command to the drone to move backwards, until 100 % of the object is found in the middle of the visual field.

Chapter 6

Conclusion and Recommendations

6.0.1 Introduction

This chapter presents the conclusions drawn from the experiments described in sections 5.1 and 5.2 and the biologically inspired models introduced in chapter 4. The chapter starts by commenting on the hardware's performance and gives some recommendations on means of improving the system at the hardware level. Next, comments are given on the experiment performed on autonomous flight in a constrained environment (section 5.1), summarising the system's performance and recommending further investigations that can be done, as well as ways in which it can be improved. The same suggestions are provided for the experiment performed on colour tracking described in section 5.2.

6.0.2 Hardware

While the AR.Drone 1.0 was a suitable platform for testing the viability of using biologically inspired models for autonomous flight control, hardware modifications are recommended that might significantly improve the functioning of the system and enhance the viability of using biological models in real world applications.

The first recommendation is to equip the system with a camera with a higher resolution and faster capture rate. It was hypothesised, when investigating the models of edge detection of section 4.2.1, and contrast enhancement of section 4.2.2 that increased resolution would allow smaller objects to be detected and avoided, which is essential if these models are to be used in real world environments. A faster frame rate would also allow the drone to fly at faster speeds while still being able to avoid objects.

The second recommendation is that the computations performed in the `iqr` neural simulator running on the base station should be performed on a chip housed in the drone itself. Recent advances in neuromorphic chips, such as IBM's TrueNorth [3], provide a good starting point in this regard, due to the power of the parallel processing power they offer as well as their energy efficiency.

A significant practical limitation of the current system is the short flight time. The system was seldom able to fly for longer than 10 minutes without it becoming necessary to replace the battery. A promising further adjustment would be the implementation of the system on a biologically inspired ornithopter, such as the DelFly [77].

6.0.3 Autonomous Flight in a Constrained Environment Using Biological Models

The experimental results presented in section 5.1 demonstrate the feasibility of using biological models for autonomous flight of a drone in a constrained environment.

The HS/VS cells model was able to detect motion in the whole visual field at the specific speed to which it was sensitive. The speed interpolation model was able to find the projected speed that corresponded most closely to the drone's actual speed and also able to keep the drone flying in a straight line.

The model of the LGMD was able to detect if the drone was on a collision course. The same LGMD model outputs signalling that the drone was on a collision course can be used to send motor commands enabling the drone steer away from the impending collision.

A very real improvement to the system would be to develop a method of optimising the synaptic weights and the neuron parameters. This could be done using a simulated environment and making use of an optimisation technique, such as the genetic algorithm, to find a good set of initial parameters. This set of initial parameters could then be further optimised using a supervised learning algorithm for spiking neural networks.

For practical applications, the system can also be made more robust. Models such as the winner take all, section 4.3.5, are currently very sensitive to neuron failure. Either more robust models should be developed to describe the operations performed by models which are sensitive to neuron failure, or redundancy should be built into the system. Even the model of the HS/VS cell which is able to detect motion when a large amount of neurons fail, does not confer robustness to the system since the models further down the hierarchical structure of the insects' brain may no longer be able to perform accurately.

In the current implementation the system does not search its environment in an optimal manner. The drone's behaviour is only to fly forward at a constant speed and make the required course stabilisation and collision avoidance manoeuvres that are determined by the biological models. A system for optimal searching, such as the Lévy Flight Pattern [78] could be used. This flight pattern optimises the success of random searches and has been shown to accurately mimic the behaviour of animals searching for food [79]. The inclusion of a model for an optimal foraging pattern would enable the system to be used for search driven tasks such as a search and rescue missions in environments too dangerous for humans.

Alongside the improvements discussed in this section and the improvements to the hardware discussed in section 6.0.2, the system also needs to be tested in real world environments to accurately assess the ability of biologically inspired models to control flight. Real world testing would reveal the true practical viability of these models.

6.0.4 Colour Tracking

In section 5.2 it is shown that a system using biologically inspired models is able to track coloured objects. While the system was able to track the chosen colour, this was true only if the colour was in its field of view, i.e. the system does not search for the colour. This poses a limitation on the system's abilities.

The first problem that needs to be addressed is the that the coloured object has to be detected before it can be tracked. In the current system this is not possible, so movement only occurs once the coloured object appears in the drone camera's field of view. To combat this, a flight pattern for optimal searching, such as the Lévy Flight Pattern [78] could be used. By implementing an optimal search pattern, the system would be able to find the coloured object autonomously and then track it. After the object is found the neural model responsible for the search can be deactivated and the colour tracking model used to control the flight behaviour could be activated.

The next issue that would need to be resolved is that the coloured object itself moves out of the field of view. This results in the drone hovering in place while waiting for the coloured object to reappear. Neural structures such as the HS/VS cells could be modified to respond to the optical flow of coloured objects in the visual field. Then, if motion is detected in a specific direction and speed, movement instructions can be sent to the drone to move in the appropriate direction even though the object is no longer in the field of view.

Additionally, the system might be improved in order to track more than colours, for example the faces or people. This could be achieved in two ways. The first would be to use a traditional image recognition algorithm that takes input from the drone's camera, computes the necessary algorithm and encodes the results into neural activity of the neuron in the iqr neural simulator that is then used to drive the motor outputs. An alternative other would be to use a supervised learning technique for spiking neural networks, such as the ReSuMe algorithm [80], to train the network to recognise the specific feature, faces for example, to be tracked and then use the output to send motor commands to the drone.

The colour tracking system could also be combined with the autonomous flight in a controlled environment system, detailed in section 5.1. This would allow for collisions to be avoided while still being able to track coloured objects. This could be useful for tasks where hard to reach areas need to be inspected. An example would be inspecting I-beams in construction by pointing a laser pointer at the location to be examined. This combination would allow the system to track the laser-pointers colour while still being able to avoid collisions with any obstacles. The benefit is that it would be unnecessary to hire skilled pilots to fly the drone when performing such inspections.

The system would also greatly benefit from finding a way to tune the synaptic weights and neuron parameters as discussed in section 6.0.3 and from the hardware improvements suggested in section 6.0.2.

Appendix A

Colour Tracking Experiment Results

Colour Following Test 2 Results

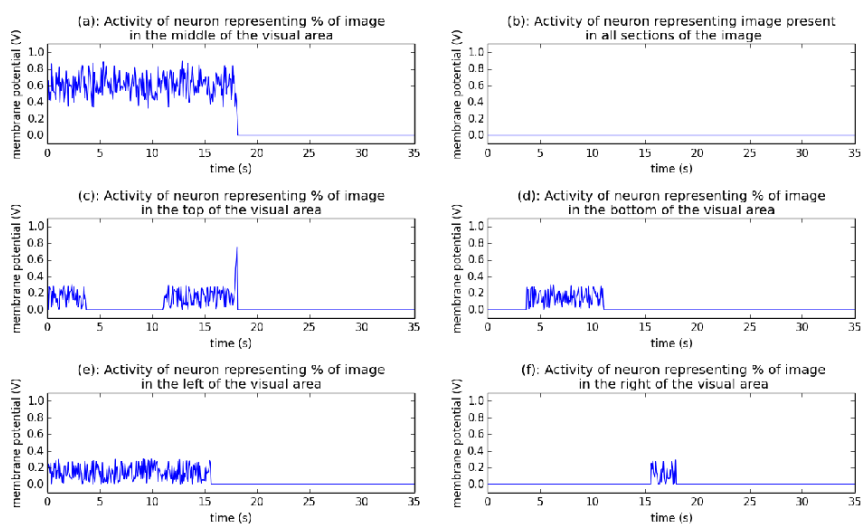


Figure A.1 – Colour tracking experiment 2 results

Colour Following Test 3 Results

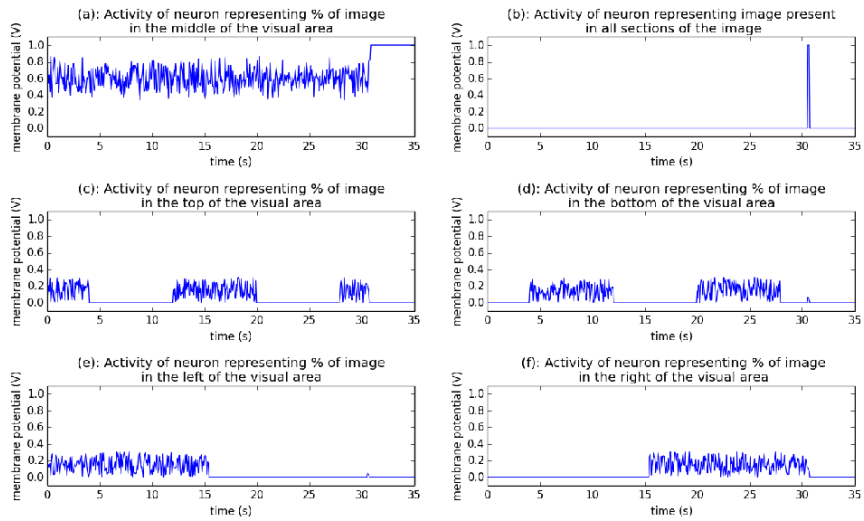


Figure A.2 – Colour tracking experiment 3 results

Colour Following Test 4 Results

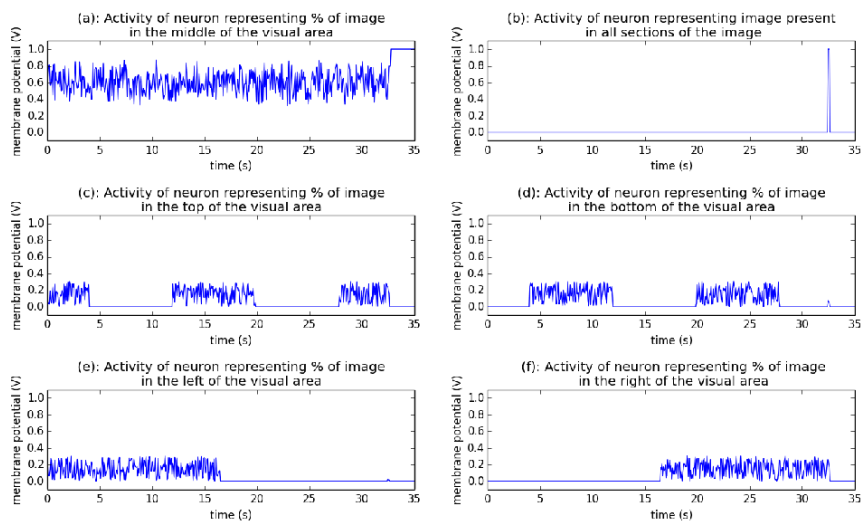


Figure A.3 – Colour tracking experiment 4 results

Colour Following Test 5 Results

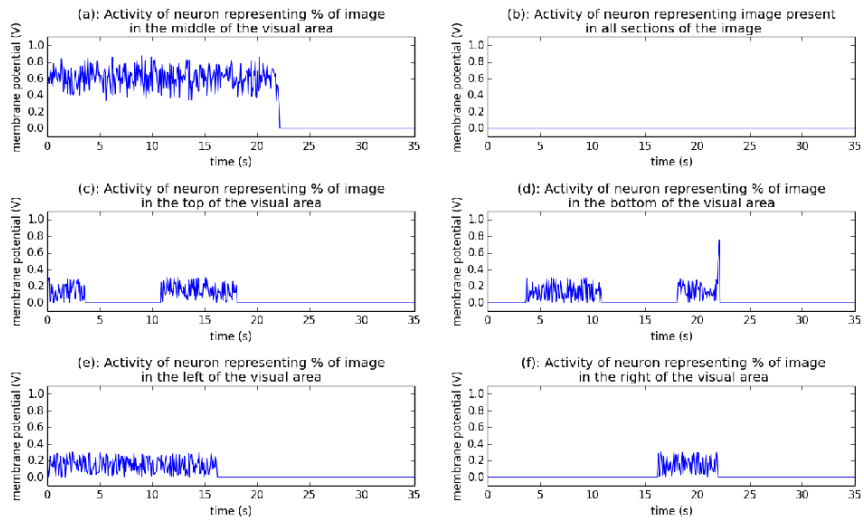


Figure A.4 – Colour tracking experiment 5 results

Colour Following Test 6 Results

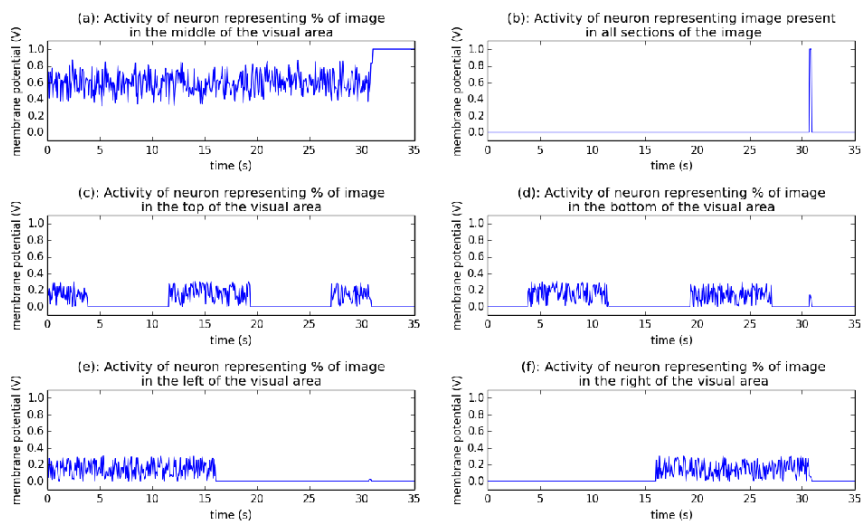


Figure A.5 – Colour tracking experiment 6 results

Colour Following Test 7 Results

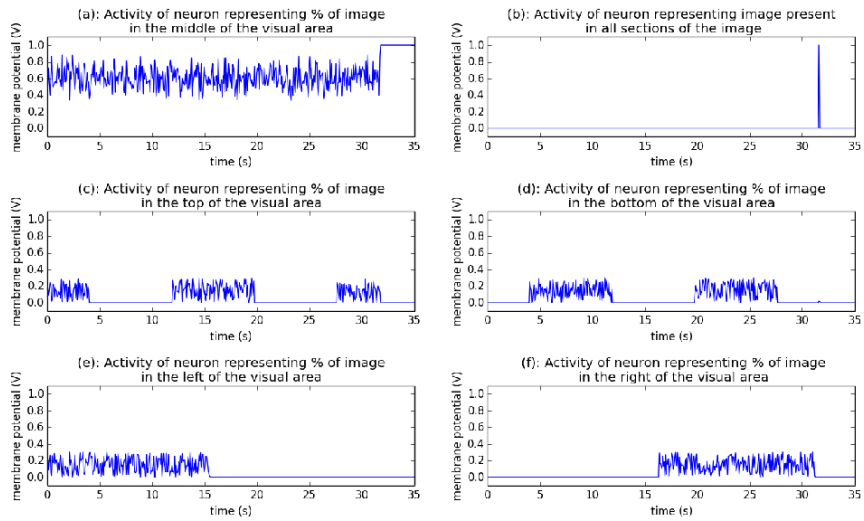


Figure A.6 – Colour tracking experiment 7 results

Colour Following Test 8 Results

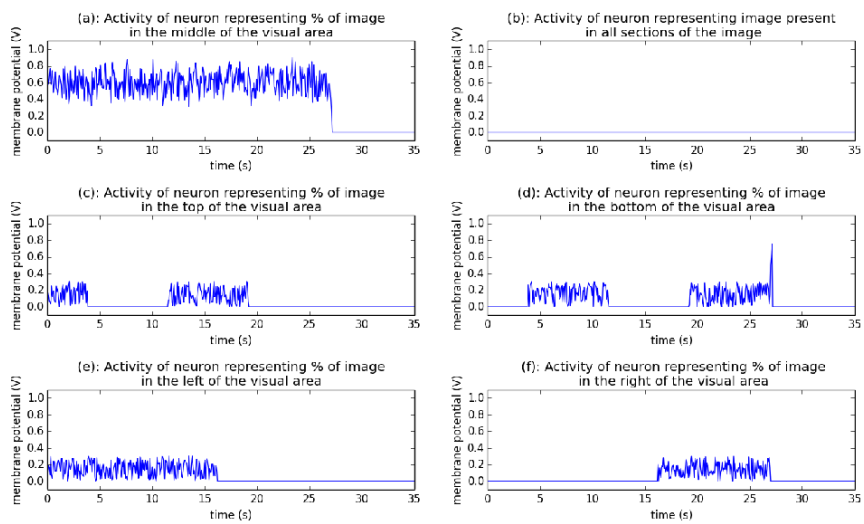


Figure A.7 – Colour tracking experiment 8 results

Colour Following Test 9 Results

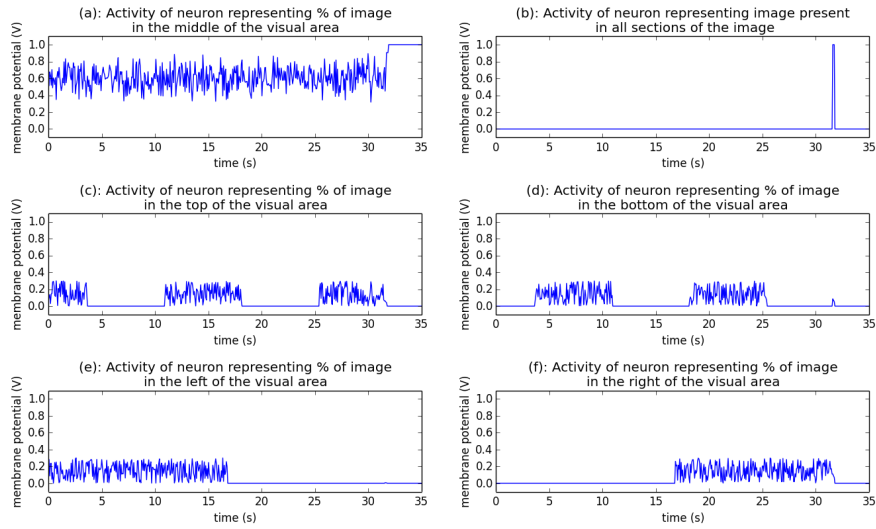


Figure A.8 – Colour tracking experiment 9 results

Colour Following Test 10 Results

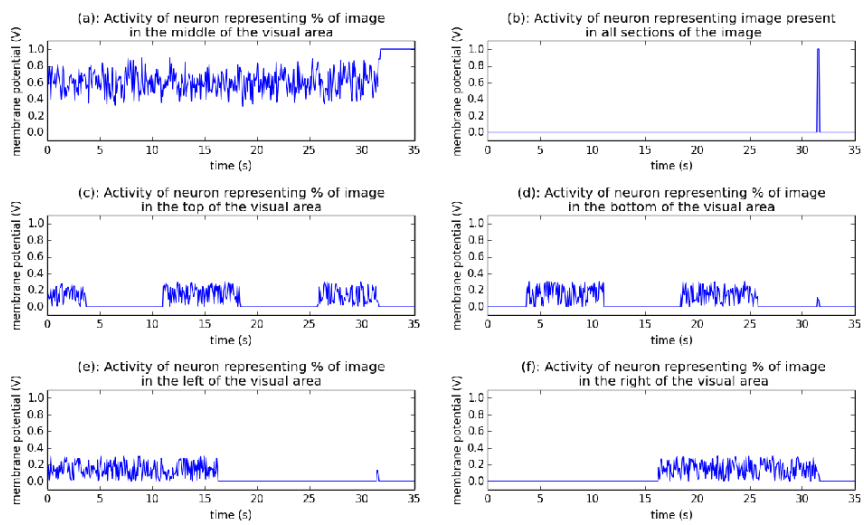


Figure A.9 – Colour tracking experiment 10 results

Bibliography

- [1] Hardie, R.C. and Stavenga, D.G.: *Facets of vision*. Springer-Verlag, 1989.
- [2] Oh, K.-S. and Jung, K.: Gpu implementation of neural networks. *Pattern Recognition*, vol. 37, no. 6, pp. 1311–1314, 2004.
- [3] Merolla, P.A., Arthur, J.V., Alvarez-Icaza, R., Cassidy, A.S., Sawada, J., Akopyan, F., Jackson, B.L., Imam, N., Guo, C., Nakamura, Y. *et al.*: A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, vol. 345, no. 6197, pp. 668–673, 2014.
- [4] Ababei, C., Maidee, P. and Bazargan, K.: Exploring potential benefits of 3d fpga integration. In: *Field programmable logic and application*, pp. 874–880. Springer, 2004.
- [5] Derks, E., Pastor, M.S. and Buydens, L.: Robustness analysis of radial base function and multi-layered feed-forward neural network models. *Chemometrics and Intelligent Laboratory Systems*, vol. 28, no. 1, pp. 49–60, 1995.
- [6] Tsang, W., Stone, A.L., Otten, D., Aldworth, Z.N., Daniel, T.L., Hildebrand, J.G., Levine, R.B. and Voldman, J.: Insect-machine interface: A carbon nanotube-enhanced flexible neural probe. *Journal of neuroscience methods*, vol. 204, no. 2, pp. 355–365, 2012.
- [7] Hampel, S., Chung, P., McKellar, C.E., Hall, D., Looger, L.L. and Simpson, J.H.: *Drosophila* brainbow: a recombinase-based fluorescence labeling technique to subdivide neural expression patterns. *nAture methods*, vol. 8, no. 3, pp. 253–259, 2011.
- [8] Girardin, C.C. and Galizia, C.G.: The " where " and " who " in brain science: probing brain networks with local perturbations. *Cognitive Computation*, vol. 4, no. 1, pp. 63–70, 2012.
- [9] Guo, P., Pollack, A.J., Varga, A.G., Martin, J.P. and Ritzmann, R.E.: Extracellular wire tetrode recording in brain of freely walking insects. *JoVE (Journal of Visualized Experiments)*, , no. 86, pp. e51337–e51337, 2014.
- [10] Ito, M., Masuda, N., Shinomiya, K., Endo, K. and Ito, K.: Systematic analysis of neural projections reveals clonal composition of the *drosophila* brain. *Current Biology*, vol. 23, no. 8, pp. 644–655, 2013.
- [11] O’Shea, M. and Williams, J.: The anatomy and output connection of a locust visual interneurone; the lobular giant movement detector (lgmd) neurone. *Journal of comparative physiology*, vol. 91, no. 3, pp. 257–266, 1974.
- [12] Rind, F.C. and Simmons, P.J.: Local circuit for the computation of object approach by an identified visual neuron in the locust. *Journal of Comparative Neurology*, vol. 395, no. 3, pp. 405–415, 1998.

- [13] Judge, S. and Rind, F.: The locust dcmd, a movement-detecting neurone tightly tuned to collision trajectories. *Journal of Experimental Biology*, vol. 200, no. 16, pp. 2209–2216, 1997.
- [14] Srinivasan, M., Zhang, S., Lehrer, M. and Collett, T.: Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology*, vol. 199, no. 1, pp. 237–244, 1996.
- [15] Götz, K.G.: Flight control in drosophila by visual perception of motion. *Kybernetik*, vol. 4, no. 6, pp. 199–208, 1968.
- [16] Hausen, K. and Egelhaaf, M.: Neural mechanisms of visual course control in insects. In: *Facets of vision*, pp. 391–424. Springer, 1989.
- [17] Wehner, R., Michel, B. and Antonsen, P.: Visual navigation in insects: coupling of egocentric and geocentric information. *The Journal of Experimental Biology*, vol. 199, no. 1, pp. 129–140, 1996.
- [18] Graham, P. and Collett, T.S.: View-based navigation in insects: how wood ants (*formica rufa* l.) look at and are guided by extended landmarks. *Journal of experimental Biology*, vol. 205, no. 16, pp. 2499–2509, 2002.
- [19] Collett, T.: Insect navigation en route to the goal: multiple strategies for the use of landmarks. *The Journal of Experimental Biology*, vol. 199, no. 1, pp. 227–235, 1996.
- [20] Posey, K.L., Jones, L.B., Cerda, R., Bajaj, M., Huynh, T., Hardin, P.E., Hardin, S.H. *et al.*: Survey of transcripts in the adult drosophila brain. *Genome Biol*, vol. 2, no. 3, 2001.
- [21] Strausfeld, N.J.: *Atlas of an insect brain*. Springer-Verlag Berlin, Heidelberg, New York, 1976.
- [22] Borst, A.: Drosophila's view on insect vision. *Current biology*, vol. 19, no. 1, pp. R36–R47, 2009.
- [23] Nilsson, D.-E.: Optics and evolution of the compound eye. In: *Facets of vision*, pp. 30–73. Springer, 1989.
- [24] Meinertzhagen, I.: Development of the compound eye and optic lobe of insects. *Developmental neurobiology of arthropods*, 1973.
- [25] Chapman, R.F.: *The insects: structure and function*. Cambridge university press, 1998.
- [26] Laughlin, S.B.: Form and function in retinal processing. *Trends in Neurosciences*, vol. 10, no. 11, pp. 478–483, 1987.
- [27] Rajesh, S., O'Carroll, D.C. and Abbott, D.: Elaborated reichardt correlators for velocity estimation tasks. In: *SPIE's International Symposium on Smart Materials, Nano-, and Micro-Smart Systems*, pp. 241–253. International Society for Optics and Photonics, 2002.
- [28] Douglass, J.K. and Strausfeld, N.J.: Visual motion detection circuits in flies: peripheral motion computation by identified small-field retinotopic neurons. *The Journal of neuroscience*, vol. 15, no. 8, pp. 5596–5611, 1995.
- [29] Douglass, J.K. and Strausfeld, N.J.: Anatomical organization of retinotopic motion-sensitive pathways in the optic lobes of flies. *Microscopy research and technique*, vol. 62, no. 2, pp. 132–150, 2003.

- [30] Rind, F.C. and Simmons, P.J.: Orthopteran dcmd neuron: a reevaluation of responses to moving objects. i. selective responses to approaching objects. *Journal of Neurophysiology*, vol. 68, pp. 1654–1654, 1992.
- [31] Simmons, P.J. and Rind, F.C.: Orthopteran dcmd neuron: a reevaluation of responses to moving objects. ii. critical cues for detecting approaching objects. *Journal of Neurophysiology*, vol. 68, pp. 1667–1667, 1992.
- [32] O’shea, M. and Rowell, C.: The neuronal basis of a sensory analyser, the acridid movement detector system. ii. response decrement, convergence, and the nature of the excitatory afferents to the fan-like dendrites of the lgmd. *Journal of Experimental Biology*, vol. 65, no. 2, pp. 289–308, 1976.
- [33] Rowell, C.F.: The orthopteran descending movement detector (dmd) neurones: a characterisation and review. *Zeitschrift für vergleichende Physiologie*, vol. 73, no. 2, pp. 167–194, 1971.
- [34] Bermudez i Badia, S. and Verschure, P.F.: A collision avoidance model based on the lobula giant movement detector (lgmd) neuron of the locust. In: *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 3, pp. 1757–1761. IEEE, 2004.
- [35] Gabbiani, F., Krapp, H.G., Koch, C. and Laurent, G.: Multiplicative computation in a visual neuron sensitive to looming. *Nature*, vol. 420, no. 6913, pp. 320–324, 2002.
- [36] Gabbiani, F., Krapp, H.G., Hatsopoulos, N., Mo, C.-H., Koch, C. and Laurent, G.: Multiplication and stimulus invariance in a looming-sensitive neuron. *Journal of Physiology-Paris*, vol. 98, no. 1, pp. 19–34, 2004.
- [37] Hausen, K.: Motion sensitive interneurons in the optomotor system of the fly. *Biological Cybernetics*, vol. 45, no. 2, pp. 143–156, 1982.
- [38] Hengstenberg, R.: Common visual response properties of giant vertical cells in the lobula plate of the blowflycalliphora. *Journal of comparative physiology*, vol. 149, no. 2, pp. 179–193, 1982.
- [39] Krapp, H.G., Hengstenberg, B. and Hengstenberg, R.: Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. *Journal of Neurophysiology*, vol. 79, no. 4, pp. 1902–1917, 1998.
- [40] Hodgkin, A. L.; Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. vol. 177, no. 4, pp. 500–544.
- [41] Bi, G.-Q. and Poo, M.-M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, vol. 18, no. 24, p. 10464, 1998.
- [42] Markram, H., Gerstner, W. and Sjöström, P.J.: Spike-timing-dependent plasticity: a comprehensive overview. *Frontiers in synaptic neuroscience*, vol. 4, 2012.
- [43] Doidge, N.: The brain that changes itself: stories of personal triumph from the frontiers of brain science/norman. 2007.
- [44] Sjöström, P.J., Rancz, E.A., Roth, A. and Häusser, M.: Dendritic excitability and synaptic plasticity. *Physiological reviews*, vol. 88, no. 2, pp. 769–840, 2008.
- [45] Song, S., Miller, K.D. and Abbott, L.F.: Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

- [46] Zhang, J.-C., Lau, P.-M. and Bi, G.-Q.: Gain in sensitivity and loss in temporal contrast of stdp by dopaminergic modulation at hippocampal synapses. *Proceedings of the National Academy of Sciences*, vol. 106, no. 31, pp. 13028–13033, 2009.
- [47] Petersen, C.C., Malenka, R.C., Nicoll, R.A. and Hopfield, J.J.: All-or-none potentiation at ca3-ca1 synapses. *Proceedings of the National Academy of Sciences*, vol. 95, no. 8, pp. 4732–4737, 1998.
- [48] O'Connor, D.H., Wittenberg, G.M. and Wang, S.S.-H.: Graded bidirectional synaptic plasticity is composed of switch-like unitary events. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9679–9684, 2005.
- [49] Lisman, J.E.: A mechanism for memory storage insensitive to molecular turnover: a bistable autophosphorylating kinase. *Proceedings of the National Academy of Sciences*, vol. 82, no. 9, pp. 3055–3057, 1985.
- [50] Tanaka, J.-i., Horike, Y., Matsuzaki, M., Miyazaki, T., Ellis-Davies, G.C. and Kasai, H.: Protein synthesis and neurotrophin-dependent structural plasticity of single dendritic spines. *Science*, vol. 319, no. 5870, pp. 1683–1687, 2008.
- [51] Lindén, H., Pettersen, K.H. and Einevoll, G.T.: Intrinsic dendritic filtering gives low-pass power spectra of local field potentials. *Journal of computational neuroscience*, vol. 29, no. 3, pp. 423–444, 2010.
- [52] Lapique, L.: Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *J. Physiol. Pathol. Gen.*, vol. 9, no. 1, pp. 620–635, 1907.
- [53] Hill, A.: Excitation and accommodation in nerve. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 119, no. 814, pp. 305–355, 1936.
- [54] McCulloch, W.S. and Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [55] Wilamowski, B.M.: Neural network architectures and learning algorithms. *Industrial Electronics Magazine, IEEE*, vol. 3, no. 4, pp. 56–63, 2009.
- [56] Bernardet, U. and Verschure, P.F.: iqr: A tool for the construction of multi-level simulations of brain and behaviour. *Neuroinformatics*, vol. 8, no. 2, pp. 113–134, 2010.
- [57] Bernardet, U., Blanchard, M. and Verschure, P.F.: Iqr: a distributed system for real-time real-world neuronal simulation. *Neurocomputing*, vol. 44, pp. 1043–1048, 2002.
- [58] Flyers, A.D.: Technical Specifications of AR.Drone 1.0. http://www.ardrone-flyers.com/wiki/Technical_Specifications_of_AR.Drone_1.0, 2014. [Online; accessed 17-November-2014].
- [59] Thompson, R.A.: *Haltere mediated flight stabilization in Diptera: Rate decoupling, sensory encoding, and control realization*. Ph.D. thesis, University of Florida, 2009.
- [60] Apple: MacBook Pro (13-inch, Mid 2012) - Technical Specifications. <http://support.apple.com/kb/sp649>, 2014. [Online; accessed 18-November-2014].
- [61] Ubuntu: MacBook Pro (13-inch, Mid 2012) - Technical Specifications. <http://releases.ubuntu.com/14.04/>, 2014. [Online; accessed 18-November-2014].
- [62] Project, Q.: QT Project. <http://qt-project.org/>, 2014. [Online; accessed 18-November-2014].

- [63] Bradski, G.: The opencv library. *Doctor Dobbs Journal*, vol. 25, no. 11, pp. 120–126, 2000.
- [64] Gopro hero3. <http://shop.gopro.com/EMEA/cameras/hero3-white/CHDHE-302-EU.html>. Accessed: 2015-06-15.
- [65] Gonzales, R.C. and Woods, R.E.: Digital image processing, 1993.
- [66] i Badia, S.B., Pyk, P. and Verschure, P.F.: A fly-locust based neuronal control system applied to an unmanned aerial vehicle: the invertebrate neuronal principles for course stabilization, altitude control and collision avoidance. *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 759–772, 2007.
- [67] Scott, E.K., Raabe, T. and Luo, L.: Structure of the vertical and horizontal system neurons of the lobula plate in drosophila. *Journal of Comparative Neurology*, vol. 454, no. 4, pp. 470–481, 2002.
- [68] Borst, A. and Haag, J.: Neural networks in the cockpit of the fly. *Journal of Comparative Physiology A*, vol. 188, no. 6, pp. 419–437, 2002.
- [69] Reichardt, W.: Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. *Sensory communication*, pp. 303–317, 1961.
- [70] Haag, J., Denk, W. and Borst, A.: Fly motion vision is based on reichardt detectors regardless of the signal-to-noise ratio. *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 46, pp. 16333–16338, 2004.
- [71] Angela, J.Y., Giese, M.A. and Poggio, T.A.: Biophysiological plausible implementations of the maximum operation. *Neural Computation*, vol. 14, no. 12, pp. 2857–2881, 2002.
- [72] Yuille, A.L. and Grzywacz, N.M.: A computational theory for the perception of coherent visual motion. *Nature*, vol. 333, no. 6168, pp. 71–74, 1988.
- [73] Feldman, J.A. and Ballard, D.H.: Connectionist models and their properties. *Cognitive science*, vol. 6, no. 3, pp. 205–254, 1982.
- [74] Ellias, S.A. and Grossberg, S.: Pattern formation, contrast control, and oscillations in the short term memory of shunting on-center off-surround networks. *Biological Cybernetics*, vol. 20, no. 2, pp. 69–98, 1975.
- [75] i Badia, S.B.: *The principles of insect navigation applied to flying and roving robots: from vision to olfaction*. na, 2007.
- [76] Poynton, C.: *Digital video and HD: Algorithms and Interfaces*. Elsevier, 2012.
- [77] Deng, S., Percin, M., van Oudheusden, B., Remes, B. and Bijl, H.: Experimental investigation on the aerodynamics of a bio-inspired flexible flapping wing micro air vehicle. *International Journal of Micro Air Vehicles*, vol. 6, no. 2, pp. 105–116, 2014.
- [78] Reynolds, A.M. and Rhodes, C.J.: The lévy flight paradigm: random search patterns and mechanisms. *Ecology*, vol. 90, no. 4, pp. 877–887, 2009.
- [79] Viswanathan, G., Buldyrev, S.V., Havlin, S., Da Luz, M., Raposo, E. and Stanley, H.E.: Optimizing the success of random searches. *Nature*, vol. 401, no. 6756, pp. 911–914, 1999.
- [80] Pomulak, F.: Supervised learning in spiking neural networks with resume method. *Phd, Poznan University of Technology*, vol. 46, p. 47, 2006.