

Methodologies for Scalable and Responsive Swarms and Networked Systems

by

Ruaridh Clark

A thesis presented in fulfilment of the requirements
for the degree of Doctor of Philosophy

2018

Department of Mechanical and Aerospace Engineering
University of Strathclyde

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by the University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in, or derived from, this thesis.

Acknowledgements

On the whole the experience of creating this dissertation was an enjoyable one, with Malcolm deserving the majority of the credit for this. I benefitted from the freedom I was given to control my research direction within a positive working environment that meant I rarely, if ever, dreaded a coming Monday. The general camaraderie of Malcolm's group has always been present; in the early days Chris, Pam and Giuliano were on hand to offer assistance and willing to put up with my slightly sardonic approach to daily life. Giuliano's research built the foundation of everything contained within these pages and his assistance was necessary in the beginning when at times I was more than a little lost. In the latter stages Ciara, Emma, Kristaps and Steve joined the office family and continued to foster our little community. Ciara, in particular, was always on hand to keep my ego in check and provide timely moral support. A mention is deserved for Wei and our many other office visitors who have contributed to office life and discussions around my work. Sasha also visited the office on one occasion and has been an excellent companion throughout the latter stages of this process; she took the time to review some of this work and the sections she did not cover are likely to contain the poorest grammar. Finally, it would be beyond remiss not to acknowledge the constant support of my family; my parents have been a foundation and safety blanket in everything I do, whilst my grandmother ensured I was well clothed and fed.

Abstract

Currently, engineers can only aspire to match nature's capabilities where the same control mechanism can scale from one hundred starlings to five thousand whilst maintaining properties such as cohesion and rapid response to predation. These aspirations have led to engineered swarms that rely on remote or human control to assemble formations, which therefore do not require leaders within the network that can communicate using inter-agent connections. As a consequence, it is primarily in theory where network leadership is considered, usually through the exploitation of a standard set of centrality metrics - such as PageRank, degree centrality or betweenness - ubiquitous across network science.

This dissertation attempts to lay the foundations for large, mobile, multi-agent swarms through a scalable control approach. This mechanism can achieve similar capabilities to that of a starling flock but with greater control over the swarm's movement. These criteria are met by defining control rules for the environment, using artificial kinematic fields, rather than for the individual; enabling agents to join or leave the swarm without a breakdown in cohesion or responsiveness - just like starlings. The functionality of this method, and the potential for swarm based applications, is demonstrated through a remote inspection case study. This control approach contributes to our rapidly increasing ability to create networked systems, but our understanding of how to control such complexity is not advancing as fast.

The identification of optimal network leaders is a significant step towards achieving a responsive, controllable, swarm. By not constraining the problem to a set number of leaders, the solution space - for anything other than small networks - is too vast to attempt an exhaustive search. The remedy for this leader selection problem has, so far, eluded researchers. Eigenvectors are presented here as key to solving this problem for determining optimal leadership when considering fast convergence to consensus. The

semi-analytical algorithms, developed herein, have $\mathcal{O}(n^3)$ time complexity and are found to perform as well as numerical optimisers with significantly greater complexity, $\mathcal{O}(n^4)$.

Eigenvectors map the dynamic response of a network where they are found to expose the most responsive communities that form in the wake of external perturbations. Comparisons, of eigenvector-based methods, with information flow simulations illustrate that analytical models can, in a computationally efficient manner, cast light on the interplay between leadership and topology. Whilst providing a greater understanding of the effectiveness and function of nature's networked systems, including the vastly complicated and responsive network of the human brain.

Contents

List of Symbols and Acronyms	vii
List of Figures	x
List of Tables	xix
1 Introduction	1
1.1 Natural Swarms	2
1.2 Engineered Swarms	3
1.3 Network Science	6
1.3.1 Network Leadership	7
1.4 Aims and Objectives	9
1.5 Dissertation Map	10
1.5.1 Contributions to Knowledge	10
1.6 Papers Authored	12
2 Scalable Swarm	13
2.1 Kinematic Field Definition	17
2.1.1 Vehicle Collision Avoidance	19
2.2 Hardware and Setup	22
2.3 Guidance and Control	23
2.3.1 Altitude Control	25
2.3.2 Attitude Control	26

2.3.3	Linear Control	26
2.4	Performance	27
2.4.1	Trajectory Errors	27
2.4.2	Control Improvements	28
2.4.3	Collision Avoidance	29
2.5	Case study: Remote Structural Inspection	31
2.5.1	Inspection Setup	31
2.5.2	Three-Dimensional Model Construction	33
2.5.2.1	Photogrammetry	33
2.5.3	Improvements & Automation	34
2.6	Summary	35
3	Responsive Swarms	37
3.1	Graph Theory	38
3.1.1	Consensus	40
3.1.2	Perturbation Driven Consensus	41
3.2	Constant Outdegree Networks	42
3.2.1	Consensus Speed Limit	43
3.2.2	Leadership Selection Algorithms	45
3.2.3	Power Optimisation	46
3.2.3.1	Results	47
3.2.4	Communities of Influence	50
3.2.4.1	Results	61
3.2.5	Influential Community Detection	71
3.2.6	Starlings - Predator Response	73
3.2.7	Discussion	76
3.3	Autocratic Leaders	77
3.3.1	Consensus Model with Autocratic Leaders	78

3.3.2	Leadership Selection Algorithm	84
3.3.2.1	Effectiveness of a Single Autocratic Leader	84
3.3.2.2	Effect of Redundant Connections	85
3.3.2.3	Nodes of Influence	85
3.3.3	Results	86
3.3.3.1	Nodes of Influence	86
3.3.4	Ratio of Perturbation to Edge Weighting	91
3.4	Variable Outdegree	95
3.4.1	Communities of Dynamic Response	95
3.4.1.1	Outdegree as a Marker for Influence	99
3.4.1.2	Leader Selection	101
3.5	Summary	104
4	Beyond the Swarm: Brain Connectomes	106
4.1	Caenorhabditis Elegans	108
4.1.1	Varshney's Intuitions	108
4.1.2	Bacik's Flow Model	112
4.1.3	Single Ablations	114
4.1.4	Signal Diffusion	118
4.2	Macaque	123
4.2.1	Hippocampus in the Queueing Network	123
4.2.2	An Eigenvector Perspective	124
4.3	Human	128
4.3.1	Spectral Analysis	129
4.3.2	Subject 113	130
4.3.2.1	Scan 1	130
4.3.2.2	Scan 2	135
4.3.3	Subject 679	140

4.4 Summary	142
5 Discussion	144
6 Future Work	146
7 Conclusions	148
Bibliography	150

List of Symbols and Acronyms

Constants and Variables

$(\cdot)_i$	Indicates an element of the vector (\cdot)
\langle, \rangle	Dot product
α, μ, ρ, H	Dimensionless scalar parameters
\hat{p}	Partition structure
λ	Eigenvalue
\mathcal{E}	Set of edges
\mathcal{G}	Graph
\mathcal{O}	Time Complexity
\mathcal{P}	Perturbation
\mathcal{V}	Set of nodes
μ	Mean
Ω	Shannon entropy
ψ	Yaw rate
φ	Roll angle
ϑ	Pitch angle
A	Adjacency matrix
B	Comparison matrix
C	Perturbation matrix
c	Constant or gains
c_1	Collision avoidance function
CV	Community variation
D	Degree matrix

LIST OF SYMBOLS AND ACRONYMS

h	Binary collision avoidance variable
k	Constant
L	Laplacian matrix
M	Generic matrix
m_e	Number of edges in the graph
N	Number of nodes in the graph
p	Power
P_{2-1}	Vehicle position vector
Q	Modularity
q	Probability
R	Radius
r^2	Coefficient of determination
sd	Standard deviation
V	Velocity vector
v	Velocity
γ	Vector of controlled flight variables
\mathbf{c}	Perturbation vector
\mathbf{s}	Network division vector
\mathbf{u}	Uniform vector
\mathbf{v}	Eigenvector
IR	Real number

Superscripts

l	Denotes a set from a subgraph
$*$	Normalised output
in	Indegree connections
out	Outdegree connections

Subscripts

\mathcal{P}	Perturbed vector or matrix
A	Belonging to adjacency matrix
i	i^{th} element

LIST OF SYMBOLS AND ACRONYMS

L	Belonging to Laplacian matrix
r	Perturbed node
s	Belonging to the system's matrix
s	Unperturbed node
U	Uniform vector or matrix
CoI^n	CoI resource allocation
$des1$	AKF desired vector

Acronyms

CDR	Communities of Dynamic Response
CoI	Communities of Influence
NNR	Nearest Neighbour
NoI	Nodes of Influence
OCN	Outgoing Connection Node
PN	Prominent Node

List of Figures

2.1	Rotating droplet of Paenibacillus vortex bacteria. (a) photo of droplet (b) velocity field representing bacteria motion. Figure reproduced from a paper by Czirák et al. (1996).	14
2.2	KMel Robotics quadrotor swarm display, image produced by KMel Robotics (2012).	15
2.3	Hopf Bifurcation vector field with streamlines as defined in eqs. (2.3) and (2.4).	19
2.4	Hopf Bifurcation vector field with streamlines as defined in eqs. (2.12) and (2.13).	19
2.5	Scheme for avoidance manoeuvre, based on the direction of travel, with multiple quadcopters.	20
2.6	Centre: H according to (2.17) where $\rho = 1000mm$ and $c_s = 15 \times 10^4 m^2$. Side: Instantaneous snapshots from above the modified kinematic field (centred on the target with streamlines displaying flow in the field) at different values of H ; rotating clockwise (1), reduced rotational field strength as $H \rightarrow 0$ (2), clockwise with reduced (3) and increased strength (4).	21
2.7	Parrot AR.Drone 2.0 with 7 reflective markers.	23
2.8	Vicon tracking system environment.	23
2.9	Software architecture for the control of multiple vehicles.	24
2.10	Control architecture scheme for one vehicle (dotted line indicates wireless data transfer; dashed line indicates causal effect).	25
2.11	Representative flight data from one vehicle during a two-vehicle flight with the altitude transitions marked above the plot.	29

2.12	Testing collision avoidance; two vehicles queuing behind the shell of another.	30
2.13	Comparison of flight data with collision avoidance function.	30
2.14	Three vehicles circling the ILW storage drum in the tracking environment.	31
2.15	Mean error flight ellipsoids at the five coverage bands for one vehicle completing the drum inspection.	32
2.16	(a) Image from inspection footage and (b) distortion corrected image.	34
2.17	Generated 3D model produced from two-vehicle flight footage.	34
2.18	Flowchart of a feasible fully automated version of the system. Green requires user input, red is autonomously executed and blue is a decision process.	35
3.1	75 node, 5 outdegree, random network where the numerically optimised leadership resource selection is represented with circled nodes (the diameter of each corresponds to the resources allocated); (a) \mathbf{v}_{L1} and \mathbf{v}_{L2} are the first and second left eigenvector of the network, (b) includes \mathbf{v}_{L3} and (c) represents the network where green lines mark bi-directional connections and grey are uni-directional.	48
3.2	Consensus speed ratio with respect to $\max_i((v_{L1})_i)$ for various resource allocation vectors over a range of outdegrees in a 50 node k -NNR network.	49
3.3	Geometrical interpretation of a 200 node, k -NNR, network with outdegree (k). v_{L2} and v_{L3} represent the 2 nd and 3 rd most dominant left eigenvectors.	51
3.4	Number of Leicht-Newman modules for a 100 node k -NNR network with a given outdegree. Circle colour dependent on the number of graphs with the same outdegree and number of modules present, with 10 graphs generated for each outdegree.	54

3.5	75 node, 5 outdegree, k -NNR network where the numerically optimised leadership resource selection is represented with circled nodes; (a) \mathbf{v}_{L1} and \mathbf{v}_{L2} are the first and second left eigenvector of the Laplacian matrix, (b) includes \mathbf{v}_{L3} and (c) represents the network where green lines mark bi-directional connections and grey are uni-directional.	55
3.6	75 node, 30 outdegree, k -NNR network where the numerically optimised leadership resource selection is represented with circled nodes; (a) \mathbf{v}_{L1} and \mathbf{v}_{L2} are the first and second left eigenvector of the Laplacian matrix, (b) includes \mathbf{v}_{L3} and (c) represents the network where green lines mark bi-directional connections and grey are uni-directional.	56
3.7	75 node, 5 outdegree, k -NNR network where the numerically optimised leadership resource selection is marked with circles (size corresponds to resource allocation) and the top four CoI communities are numbered; (a) \mathbf{v}_{A1} and \mathbf{v}_{A2} are the 1 st and 2 nd left eigenvector of the adjacency matrix, in the other plots the axes are for the adjacency after the removal of the most prominent node (b) from the 1 st CoI comm., (c) from the 1 st and 2 nd CoI comm., (d) from the 1 st , 2 nd and 3 rd CoI comm.	59
3.8	Power, p , required for Power Opt to match the consensus speed found by numerical optimisation.	60
3.9	75 node, 5 outdegree, k -NNR network with resource allocations highlighted: (a) the five, power optimised, CoI vectors are displayed; (b) the final optimised resource allocations are shown. Two-way connections are depicted in green with one-way in grey. Coloured circles overlay the network with the circle radius proportional to the resource allocation for the node at their centre.	62
3.10	Consensus Speed Ratio for k -NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.	63
3.11	Consensus Speed Ratio values for a 100 node k -NNR network, where $k = 5$, with randomly rewired links. The probability, q , of a link being randomly rewired is varied marked on the x-axis. The error bars mark the maximum and minimum deviation from the mean.	65

3.12 100 node k -NNR networks, where $k = 5$, with randomly rewired links. The probability, q , of a link being randomly rewired is varied marked on the x-axis. The error bars mark the maximum and minimum deviation from the mean. Consensus Speed Ratio values compare CoI and Power Opt approaches with the mean path length detailed on the other y-axis. 65

3.13 Consensus Speed Ratio for random networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean. 66

3.14 Consensus Speed Ratio for scale-free networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean. 67

3.15 Consensus speed comparison for different network structures, where $k = 10$, influenced by a near-optimal perturbation. The small-world connections were rewired with a probability of $q = 0.1$ 68

3.16 Solver time and difference between optimised consensus speed and $\max_i((v_{L1})_i)$ for large k -NNR networks with an outdegree of 30. A network was analysed at each 1000 node interval where $\max_i((v_{L1})_i) = \{31, 14, 10, 7, 5, 4, 4\} \times 10^{-4}$ for the analysed networks. 70

3.17 Geometrical interpretation of a 7000 node, k -NNR, network with outdegree (k). v_{L2} and v_{L3} represent the 2nd and 3rd most dominant left eigenvectors. 71

3.18 The CoI resource allocation is shown on this 50 node k -NNR network with coloured circles where each colour relates to a specific CoI vector (CoI 1 - red, CoI 2 - green, CoI 3 - magenta). The light brown shaded areas mark out the Leicht-Newman modules Leicht and Newman (2008) present in the graph. 72

3.19 1200 node starling k -NNR flock model, as defined in Young et al. (2013) where nodes are randomly distributed for a flock thickness of 0.2, analysis for (a) $k = 7$, (b) $k = 50$ and (c) $k = 100$ where coloured circles are centred on nodes and the circle's radius is proportional to the resource allocation. Circle colour key: red – CoI vector 1, green – CoI vector 2, magenta – CoI vector 3, black – CoI vector 4, blue – CoI resource allocation. 75

3.20	50 node, 5 outdegree, k -NNR network. Two-way connections are depicted in green with one-way in grey. Coloured circles are centred on nodes, with the circle radius dependent on resource allocation.	87
3.21	Consensus speed ratio for k -NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.	88
3.22	Run time comparison where the error bars mark the maximum and minimum deviation from the mean.	89
3.23	Consensus speed comparison between weighted and unweighted k -NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.	89
3.24	Lattice ring network with small world rewiring; 20 nodes, 4 outdegree with rewiring probability of $q = 0.1$. Rewired connections are in red, two-way in green and one-way in grey	90
3.25	Consensus speed ratio for lattice ring networks with small world rewiring probability of $q = 0.1$ and outdegree for all nodes set at 10. The error bars mark the maximum and minimum deviation from the mean.	91
3.26	For the weighted model, a numerically optimised perturbation is marked for a 250 node k -NNR network ($k = 8$) with a different edge weighting in each plot. The first three left eigenvectors of the Laplacian matrix are represented by v_{L1} , v_{L2} and v_{L3} . The blue circles mark nodes with perturbations applied where the diameter is proportional to the percentage of the total perturbation applied.	93
3.27	For the unweighted model, a numerically optimised perturbation is marked for a 250 node k -NNR network ($k = 8$) with a different edge weighting in each plot. The first three left eigenvectors of the Laplacian matrix are represented by v_{L1} , v_{L2} and v_{L3} . The blue circles mark nodes with perturbations applied where the diameter is proportional to the percentage of the total perturbation applied.	94
3.28	Comparison of communities - detected for a 50 node, 5 outdegree, k -NNR graph - by the Leicht-Newman algorithm (highlighted in light brown) with those detected by Algorithm 3.3 (node colour indicating community designation).	96

3.29	Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix for the 50 node k -NNR in Fig. 3.28. Community designation according to Algorithm 3.3 is noted using node colour and the most prominent community nodes have black outlines.	97
3.30	Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix for a 50 node random graph where outdegree varied between 3 and 7. Matching node colour indicates nodes belonging to the same community according to Algorithm 3.3.	99
3.31	Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix for a 50 node random graph where the outdegree varied between 3 and 7. The node colour relates to the node outdegree.	100
3.32	NNR graph with 50 nodes and an outdegree varying between 3 and 7: (a) Planar NNR graph layout based on x-y coordinates. Blue lines indicate two-way communication edges and grey indicates a one-way connection, with the red at the start of a line marking it as an outgoing edge. (b) & (c) Visualisation of node placement in eigenvector space, where v_{L2} and v_{L3} are the second and third left eigenvectors, respectively, of the Laplacian matrix. The dot colour denotes node outdegree.	102
3.33	NNR graph with 50 nodes and an outdegree varying between 3 and 7 where certain nodes are labelled with an ID number for reference: (a) Optimal resource allocation using a numerical optimiser. (b) & (c) Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix. Nodes are coloured according to community designation as defined by Algorithm 3.3. Certain nodes are labelled with an ID number for reference.	103
4.1	Map of neurons with longitudinal nerve tracts and commissures in the body. Image taken from WormAtlas (2013).	109
4.2	Eigenmode (eigenvector) v_3 corresponding to λ_3 . Image taken from Varshney et al. (2011).	110

4.3	Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are associated with λ_1 , λ_2 and λ_3 of the Laplacian matrix for the electrical junction network of the <i>C. elegans</i> . Community designation (see Algorithm 3.3) is noted using node colour.	111
4.4	Eigenmode (eigenvector) v_6 corresponding to λ_6	112
4.5	Visualisation of node placement in eigenvector space, where v_{L5} and v_{L6} are associated with λ_5 and λ_6 of the Laplacian matrix for the electrical junction network of the <i>C. elegans</i> . Community designation (see Algorithm 3.3) is noted using node colour.	113
4.6	Effect of single ablations on the make-up of partitions \mathcal{A} through to \mathcal{E} . Figure taken from Bacik et al. (2016).	115
4.7	Red '+'s and labels mark the nodes, in Fig. 4.6, that caused major disruption to the partitions \mathcal{A} to \mathcal{E} . Black '+'s mark the other nodes in the network. (a) v_{L1} , v_{L2} and v_{L3} are the first left eigenvectors. (b) v_{L1} against v_{L2} (c) zoomed in section (d) the Communities of Dynamic Response that a node belongs to is denoted with the colour of its dot.	117
4.8	Depiction of a simulated signal propagating through the <i>C. elegans</i> connectome from differing source neurons. Figure taken from Bacik et al. (2016).	118
4.9	Response of <i>C. elegans</i> connectome to a perturbation where red '+'s mark nodes highlighted in Fig. 4.8. Black '+'s mark the other nodes in the network.	120
4.10	Response of <i>C. elegans</i> connectome to a perturbation where red '+'s mark nodes highlighted in Fig. 4.8. Black '+'s mark the other nodes in the network.	121
4.11	Response of <i>C. elegans</i> connectome to a perturbation where red '+'s mark nodes highlighted in Fig. 4.8. Black '+'s mark the other nodes in the network.	122
4.12	v_{L1} and v_{L2} are the first left eigenvectors. Each '+' represents a neuronal area with some prominent areas marked on the plot.	125
4.13	Outgoing and incoming connections from the TFL, TFM and CA1 neuronal areas. Image taken from Mišić et al. (2014).	125

4.14 Eigenvector plots for subject 113 where (a) uses the first, third and seventh eigenvector to highlight the community structure of the first eigenvector and (b) employs the fifth, sixth and seventh eigenvector to highlight the community of the fifth eigenvector that is a *false* first eigenvector. The Communities of Dynamic Response are denoted by the colour of the node’s dot. 130

4.15 Subject 113 - scan 1: brain represented by x,y,z outlines from a surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,4,5,8,9) are displayed with a spectrum of colours from green (1) to blue (9). The most prominent node for each eigenvector is marked with a circle. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled. 131

4.16 Subject 113 - scan 1: brain represented by x,y,z outlines from a surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,4,5,8,9) are displayed with a spectrum of colours from green (1) to blue (9). The 3500 highest traffic edges are then overlaid in red (a) View from above; (b) View from behind; (c) View from the side. 133

4.17 Subject 113 - scan 2: brain represented by x,y,z outlines from a surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,3,5) are displayed with a spectrum of colours from green (1) to blue (5). The most prominent node for each eigenvector is marked with a cross. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled. 136

4.18 Subject 113: brain represented by x,y,z outlines from a surface model. Prominent nodes from scan 1 are displayed in yellow o’s with scan 2 marked in black x’s. The most prominent node for each eigenvector is marked with a circle or cross. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled. 137

4.19 Subject 113 - scan 2: brain represented by x,y,z outlines from a brain surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,3,5) are displayed with a spectrum of colours from green (1) to blue (9). The 3500 highest traffic edges are then overlaid in red (a) View from above; (b) View from behind; (c) View from the side. 139

4.20	Edge weight comparison between brain scan 1 and 2 for subject 113 from Landman et al. (2011).	140
4.21	Subject 679: brain represented by x,y,z outlines from a brain surface model. Prominent nodes from scan 1 are displayed with yellow o's with scan 2 marked with black x's. The most prominent node for each eigenvector is marked with a circle or cross. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled. .	141

List of Tables

2.1	Standard Deviation of Positional Error from Multiple (1 & 2 Vehicle) Drum Inspection Flights	28
3.1	Algorithm run time comparison for n nodes.	50
3.2	Convergence speed to consensus achieved when using a different number of CoI vectors for the 75 node, 5 outdegree, k -NNR network shown in Fig. 3.9.	63
3.3	Algorithm run time comparison for n nodes.	64
3.4	Algorithm run time comparison for N nodes.	88
4.1	Prominent nodes from Fig. 4.12.	126
4.2	Most traversed edges with node v_{L1} rankings.	127
4.3	Closest Power et al. (2011) ROI for each most prominent eigenvector node	135
4.4	Scan comparison of prominent nodes for subject 113.	138
4.5	Scan comparison of prominent nodes for subject 679.	142

Chapter 1

Introduction

Edmund Selous, a British ornithologist in the early 20th century, observed a starling murmuration where many hundreds of birds “*turned, wheeled, reversed the order of their flight, changed in one shimmer from brown to grey, from dark to light, as though all the individuals composing them had been component parts of one individual organism*”, Selous (1931). This was a remarkable sight but his conclusions, upon witnessing such behaviour, were also remarkable and echoed the well-known statement from Arthur C. Clarke (1973) that “*any suitably advanced technology is indistinguishable from magic*” . In this case, it was not magic but mental telepathy (“*some form of thought-transfer*”) that was deemed the only possible mechanism for achieving such instant synchronisation. Research into swarms has made significant progress since then. The synchronicity is still remarkable but more plausible explanations now exist, which are based on the propagation of agent interactions. However, many open questions on the precise nature of the mechanisms governing such swarms still need to be tackled. In particular, the role network topology plays in creating influential individuals is not fully understood. This dissertation is built upon these questions but does not constrain itself to just starlings, with these birds and other natural swarms also serving as inspiration in the development of control mechanisms for engineered swarms. The universality of network analysis is then revealed by utilising the tools developed for identifying influential swarm agents to investigate the most complex of networked systems: the brain.

1.1 Natural Swarms

Starlings are a well-known example of coordinated movement, but they are far from alone. The behaviours that enable starlings to flock are not unique, nor particularly rare, with all creatures exhibiting similar behaviour to varying degrees, as noted by Reynolds (1987). Schools of fish are another obvious example where, like starlings, the swarm provides protection from predation; extending the sensing range of an individual fish whilst making target selection difficult for the predator. The term coined by Vicsek and Zafeiris (2012) in their study of such coordinated phenomena is collective motion, where collective behaviour - introduced earlier by Vicsek (2001) - is inclusive of other examples of synchronisation, such as firefly light displays, and not just motion. Collective motion has been observed in a host of natural systems including macromolecules, bacteria colonies, cells, insects, fish, birds, mammals and humans. The defining features of collective motion are that agents in the swarm are fairly homogenous and move with approximately the same absolute velocity. To act as a swarm these agents must be capable of changing direction in response to fellow swarm agents, which are within a specific interaction range, to achieve alignment that is subject to noise.

Swarms are an attractive topic of study for engineers as the collective motion seen in nature has developed to be flexible, robust, scalable and responsive. Each of which is a desirable property, although often hard to attain, when designing a system.

Flexibility is marked by a capacity to handle different or changing environments. Şahin (2004) notes that ants are impressively flexible; acting independently when foraging but then coordinating when moving food too large to carry individually, while also being capable of larger scale coordination when forming chain structures to reach or pull desired objects.

Robustness is the ability to cope with the loss of individual agents, whilst remaining connected and able to accomplish the desired task. Again starlings are exemplary in this aspect, where Miller (2010) states that the approximately nearest neighbour structure - where each starling tracks a set number of its nearest neighbours even as flock density varies - gives the starling flock a remarkable elasticity. The flock is able to expand and contract dramatically and rapidly without splitting into disconnected groups. This set number also means that a starling will select a new neighbour to monitor if one of its neighbours is lost from view. In fact, Young et al. (2013) found

that starling flocks maximise robustness by always observing approximately 7 other flock members.

Scalability is one of the most remarkable aspects of collective motion, as the capacity to remain cohesive and responsive remains regardless of the number of participants. Herring are a particularly good example of this seemingly limitless capacity for system growth, where migrating herring can number in the millions with 17 mile long schools having been observed by Scheffer (1983).

Responsiveness - a key focus of this dissertation - captures the ability of the swarm to produce a fast coordinated response to stimulus. It is an essential performance metric for predator avoiding swarms, where information of incoming attack spread through the swarm rapidly to enable a coordinated but rapid manoeuvre as noted in Potts (1984); Treherne and Foster (1981).

1.2 Engineered Swarms

The first work to formally introduce the concept of swarm engineering was by Winfield et al. (2004) - although the term swarm engineering was first coined a few years earlier by Kazadi (2000). Winfield et al. highlighted the potential benefits of designing a distributed system, based upon swarm intelligence, and captured the challenges that emerge when the decision making and control is distributed across the system. The challenges include difficulty in predicting behaviour, when so many agents can potentially weild influence, and difficulty in control, if unexpected behaviours do emerge. The ability to design a swarm, or network, that produces a predictable response and can have its behaviour influenced, irrespective of its topological complexity, is at the core of this dissertation's contribution.

An engineered swarm is a group of autonomously operating robots that, based on their sensing capabilities and without access to a centralised controller, cooperate to complete a task. Brambilla et al. (2013) divides the different design methods for engineered swarms into behaviour-based design and automatic design methods.

Behaviour-based design methods focus on all robots implementing local rules, adjusting behaviour based on other members of the swarm, from which emergent swarm behaviour occurs. One of the most commonly used approaches is that of the finite state machine design, see Minsky (1967). This provides agents with a finite selection of possible behaviours that agents choose from based on sensory input. Therefore,

each agent is in a particular state at any point in time and transitions to a different state as a consequence of receiving stimulus. Probabilistic transitions are commonly used to determine when an agent has received enough stimulus to switch state, where a popular approach is that of the response threshold, developed by Granovetter (1978). This threshold guarantees that an agent remains in its state when a low stimulus is present, and ensures it switches state when receiving a large enough stimulus. For a stimulus that lies between these two options a function defines the probability that an agent will change its behaviour. De Lope et al. 2013 produced an illustrative paper for the state-of-the-art in multi-robot systems, where both a response threshold model and a reinforcement learning approach were shown to be effective in controlling multiple robots that must select and complete multiple heterogeneous tasks.

The reinforcement learning approach - used by de Lope et al. - also relies on a finite state machine, but it belongs to the category of automatic design methods alongside other popular methods such as evolutionary optimisation. Automatic design determines the final behaviour of the swarm through an autonomous trial and error approach that attempts to optimise the performance of the swarm. This approach essentially removes the human designer, providing an optimised performance but at the risk of creating unexpected and potentially undesired behaviour.

Virtual physics-based design falls into the behaviour-based design category and is of particular interest in this dissertation as it is the primary approach adopted herein for controlling swarm motion. The method is based on having every agent represented as a virtual particle that is influenced by virtual forces with those virtual forces then translated into the real world through an agent's propulsion system. In most implementations the virtual forces are exerted on agents by other agents in the swarm or by obstacles. An early example of this was the artificial potential field, pioneered by Khatib (1986), where a single manipulator was guided to a target point by a field of forces. The position of the target was a source of attraction whilst the obstacles present had repulsive surfaces associated with them. This control has since been applied in a swarm context with the most commonly applied function being the Lennard-Jones potential, introduced by Jones (1924), which approximates the interaction between a pair of neutral atoms or molecules. In a swarm context, applying this potential to each agent-pair interaction creates a potential well that maintains the distance between the agents. The approach detailed in this dissertation takes inspiration from the previous work by Jones but is a distinct method that defines a vector field for the operating environment, and will be referred to herein as an Artificial Kinematic Field (AKF). This method - where agents follow a vector flow just like particles being carried by

a river's current - is a deviation from most other virtual physics-based approaches - where agents are pulled towards an attractive source with obvious analogies to gravitational attraction. However, the benefits of both approaches are similar; the control laws are unaffected by the system's scale and the motion of agents are governed by mathematical rules that translate between sensory input and desired actuator output. These methods also benefit from provable stability, for the resulting agent behaviour, by using control theory, specifically Lyapunov stability theory. However, where AKFs excel is in their less disruptive approach to motion, allowing agents to flow around obstacles rather than just be repelled away from them. Therefore, AKFs are of particular interest when considering an engineered swarm that is able to emulate the smooth and coordinated motion performed by fish and birds. It should also be noted that the main challenge is not just achieving coordinated motion, which has been shown using virtual physics-based design back by Reynolds (1987), but it is having control and influence over the swarm so that it can be applied to a range of challenging applications.

A plethora of proposals have been put forward for engineered swarm applications. Mapping is a popular task for swarm robotics where one such example, developed by Kumar and Sahin (2003), uses a finite state machine approach based on ant foraging techniques to detect land mines with cooperative robots. Potential fields are also prominent in systems attempting to tackle the real-world; Bruemmer et al. (2002) enabled robots to detect a spillage and position themselves around the perimeter of the spill with potential fields acting on each robot. While Penders et al. (2011) developed a robotic swarm to assist firefighters by utilising artificial potential forces to keep the robots in contact with the firefighter but away from obstacles. Dangerous environments are an area where swarm applications have a strong case. The use of robots removes the risk of harm to humans who would have previously carried out the activity, see Marjovi et al. (2010) where a finite state machine approach is used to create a fire searching swarm for operation in an unknown environment. Another situation where humans may be exposed to risk is that of structural inspection, for example the structure requiring review could be in a radioactively contaminated area or an exposed section on an offshore oil platform. This is why remote structural inspection is demonstrated in Chapter 2 using multiple aerial vehicles. Many of these applications are promising but, to fully unlock the potential of engineered swarms, these systems should be capable of accommodating a vast number of agents. This is where the emerging field of network science becomes key as it will allow large systems with changeable topologies to be understood and controlled.

1.3 Network Science

For any swarm, natural or engineered, the interactions can be mapped to produce a networked system where agents share information through these interactions, whether that be data on position, velocity or any another parameter that can be agreed upon. Simplifying a swarm to its connections can, by using the tools of graph theory, provide powerful insights that are often obscured by the system's complexity.

Network science is a relatively new field, and according to Lewis (2011) a contentious one. It combines graph theory and control theory with applications across numerous disciplines. Despite the existence of doubt over its validity as a science, its history can be traced back to 1735 when graph theory was essentially created and applied to the Bridges of Königsberg problem (see Alexanderson (2006)) by Leonhard Euler, who solved how to circumnavigate all seven bridges of Königsberg by only crossing each once. This first application is an insight into the ubiquity of networks, and hence the wide applicability of graph theoretical approaches.

The solution to the Bridges of Königsberg problem did not trigger a surge of network science research instead there was a gap of over 200 years before any significant progress was made. Paul Erdős, in the 1950s, produced papers concentrating on random graphs with the notable development of the Erdős-Rényi model - where $G(n, p)$ is a random graph with n vertices where each possible edge has probability p of existing. The ability to model random graphs is the basis of many network analysis methods where comparison with a random graph can reveal order. Modularity - a concept that will be prominent later in this dissertation - is a prime example of this, where communities are defined as groups of nodes that are more densely connected than would be expected in a random graph. The next significant work followed over a decade later with Stanley Milgram's development of the small-world concept. Milgram's work and then the development of scale-free networks by Barabási and Albert (1999), in the late 1990s, are two of the most wide reaching developments in network science and, as such, will be discussed in greater detail in the following paragraphs.

The concept of a small-world was introduced by Milgram who created the notion of *six degrees of separation* with his US postal experiment. Participants from Nebraska and Kansas were asked to forward a letter to a person closer to the final destination, which was Cambridge and Boston, Massachusetts. The person receiving the letter was instructed to do the same, with the aim being that the letter would eventually find its

way to Massachusetts at which point Milgram was to be notified. Ignoring the lost letters, the resulting number of hops was on average 5.2 (see Travers and Milgram (1967) where the hops ranged from 2 to 10) and as a result of some questionable rounding (plus a film starring Will Smith (1993)) the concept of six degrees of separation was placed into the public consciousness. Recently, Facebook have updated this average number of hops for the more connected age we now live in; Edunov et al. (2016) finding an average of 3.57 degrees of separation. Obviously this only applies to those people who have Facebook accounts but the principle remains the same, that despite having a huge graph, the diameter - the longest distance between two nodes - remains startlingly small. This is a phenomenon seen in many organically developed networks, such as brain networks - see Bassett and Bullmore (2006) - where this structure tends to minimise wiring costs while supporting high dynamical complexity.

Another significant topological discovery is that of scale-free networks, where the distribution of connections is highly non-random with the curve following a power law. This law defines the probability that a node has exactly k links with the following equation: $P(k) \sim k^{-\gamma}$ where γ is a parameter that defines the connection distribution with scale-free networks most often characterised by $2 < \gamma < 3$ (see Barabási and Albert (1999) for examples). Hence, most of a network's nodes have a low number of connections whilst a small selection are very well connected. Barabási (2009) summarises the discovery of scale-free networks where the initial work focused on how both the interlinked pages of the world wide web (WWW) and then the interconnected hardware of the internet, which supports the WWW, are both topologically scale-free. This topology has been rigorously examined, revealing a number of interesting properties; Pastor-Satorras and Vespignani (2001) found that viruses can spread easily in scale-free networks. Albert et al. (2000) noted that this topology is highly vulnerable to targeted attacks but Cohen et al. (2000) found that they are, at the same time, resistant to random node deletion.

1.3.1 Network Leadership

Although not focused on either the small-world or scale-free properties of networks, this dissertation does consider the role of network topology and hence inevitably crosses paths with these two most universal of examples in network science literature. The primary focus on networks, detailed herein, is that of optimal leader detection i.e. uncovering which nodes if perturbed will achieve the fastest convergence of the whole

network to a new state of consensus. Returning to Edmund Selous, our perplexed ornithologist, who observed - correctly this time - that starlings could not possibly have a single leader or employ sentinels. These were popular theories, at that time, for explaining the coordinated motion, where sentinels were proposed as birds whose job it was to detect danger and tell the rest of the flock. We now know that consensus can be achieved despite the absence of a central authority, but that does not mean every node leverages the same amount of influence. Often there are instigators of transition that, although not designated as leaders, have the connections to spread information effectively and in essence lead the network in that moment. Much of this dissertation is dedicated to the detection of these effective influencers in the network, a task that has only been briefly addressed before now with most other research focused on problems such as the minimum number of leaders required to fully control a system, see Liu et al. (2011), or leader selection for optimising other metrics such as network robustness, observability and controllability with notable works by Liu et al. (2011); Lozano et al. (2008); Ni et al. (2013); Rahmani et al. (2009). Another common approach is to consider an engineered system with a designated leader that the other agents follow, such as in Ying et al. (2014) work on undirected networks that examined how quickly each node could guide the rest of the network to a new state of consensus. Ying found that a node's ability to lead fast consensus was correlated with its degree. This dissertation goes further in considering directed networks where the ratio of indegree and outdegree - number of incoming connections versus number of outgoing connections - proves to be a crucial factor.

In nature, swarm leaders are chosen for a variety of reasons and are not necessarily those best placed in the network to lead. Attanasi et al. (2015) and Herbert-Read et al. (2015) observed that starlings on the edge of the flock will trigger predator avoidance as they will detect the predator first. These leaders may be more likely to be at the edge of the flock but the randomness of the predator attack means that the leaders in a starling swarm are not necessarily those that will achieve the fastest response. For an engineered system, leaders can be selected that are best placed within a network to achieve optimal convergence speed to consensus. For small networks, where there are a limited and computable number of options, this is a simple enough problem to solve numerically. However, as the network grows the combinatorial nature of the problem results in a rapid increase of the possible options and, hence, the solution space.

The final aspect of leadership that bears consideration is that in practice it comes in many forms. A variety of leadership styles are employed by people who usually select their style with the aim of achieving a desired performance from their team. Therefore

it is worth considering whether a leadership approach can be selected to optimise a network's response in systems aiming for consensus. While starlings function well with what appears to be a flat leadership structure (egalitarian) there are still indications from nature that there are benefits to be found in varying, if not the style of leadership, the hierarchy of those in the network. Nagy et al. (2010) found that homing pigeons not only determine their spatial positions but weight their neighbour monitoring based on a well-defined hierarchy. The authors go on to suggest that such a hierarchy, where pigeons pay more and less attention to others based on their seniority within the group, produces a more efficient flock than would be achieved with an egalitarian leadership structure. As will become clear when progressing through this dissertation, it is invariably a two way street, nature can inspire and prompt effective design but in attempting to optimise an engineered solution we can often shine a light back on nature's design revealing the mechanisms that make for a scalable and responsive system.

1.4 Aims and Objectives

This dissertation concerns itself with systems that have no central controller and where there is limited communication between agents, i.e. agents are not able to communicate with all other members of the network at the same time (all-to-all communication). The main objectives of this work fall under three observations on naturally existing systems and are as follows:

- ◇ Natural swarms display remarkable scalability:
 - Produce a framework that could support a scalable, autonomous, engineered swarms.
- ◇ Large schools and flocks maintain their ability to respond and avoid predators:
 - Discover what makes agents in a network more influential and what role topology plays in creating these influential leaders.
 - Find out how to identify these agents in any given network.
 - Uncover effective topologies for creating a highly responsive system.
- ◇ Networks are ubiquitous:
 - Can the lessons learnt from highly responsive swarming systems be applied to other responsive networks?

1.5 Dissertation Map

Chapter 2 will introduce a method for controlling scalable engineered swarms by using artificial kinematic vector fields combined and collision avoidance manoeuvres. This scheme is validated through tests in a laboratory environment using multiple quadcopters. A case study is presented to highlight a future application for such a system.

Chapter 3 considers how to achieve a fast network response from a network by identifying and influencing effective leaders. Semi-analytical algorithms are developed herein to define a perturbation that maximises a system's convergence speed to consensus. The algorithms rely on the influence of the first left eigenvector of the Laplacian matrix, where the eigenvector is a known metric for a node's ability to spread information. The effectiveness of autocratic leadership in improving the speed of the system's response is also studied. Finally, the dynamic response and leadership selection in the more complex case of variable outdegree networks is considered. A simple but effective method, that is again dependent on a system's eigenvectors, is developed. This chapter includes a number of proofs, with the algorithms developed also supported by simulations and toy examples.

Chapter 4 focuses on the large scale and highly responsive networks of brain connectomes. The methods developed in the previous chapter for swarm control are now applied in the analysis of a network. Analysis is undertaken on a number of directed animal brain networks. The results produced are then compared with previous studies that have numerically modelled the information flow in brain networks. These studies validate the results and highlight that useful insights, previously inaccessible via traditional graph theory approaches, can be gained by taking a perturbation driven consensus approach. Finishing with a demonstration of the analytical capabilities developed through a study of large human connectomes.

1.5.1 Contributions to Knowledge

- ◇ Developing, applying and testing a vector field approach for multiple aerial vehicles.

In particular, the creation of a novel, distributed and scalable control scheme that could accommodate a large engineered swarm.

- Created a scalable control scheme using an artificial kinematic field. (Section 2.1)
 - Applied this scheme in a laboratory environment with multiple quadcopters. (Section 2.4)
 - Part of the novelty lies in the less disruptive vehicle avoidance mechanism, which was also demonstrated in a laboratory environment. (Section 2.1.1)
- ◇ Using a perturbation driven consensus approach to create optimal leader selection methods, but also demonstrating its applicability in the analysis of existing network systems by revealing new insights.
- Created effective and efficient algorithms that can find the optimal perturbation distribution in constant outdegree networks, and hence identify the effective leader(s). (Section 3.2.2)
 - Highlighted when the first left eigenvector is an effective map of node influence but also discovered topologies for which it is not accurate at predicting influence. (Section 3.2.4)
 - Revealed how starlings facilitate fast response to predator attack by relying on a limited number of topologically determined interactions. (Section 3.2.6)
 - Proved that for a complete graph autocratic leadership will always produce a faster convergence rate to consensus than egalitarian leadership, in the absence of noise. (Section 3.3)
 - Developed an effective and novel approach to community detection and leader selection for variable outdegree networks. (Section 3.4)
 - Demonstrated the effectiveness of analytical graph theoretic approaches on connectome analysis through comparison with results reliant on numerical methods. (Section 4.1 & 4.2)
 - Performed eigenvector-based analysis of human brain networks, identifying influential pathways in the brain that were used to distinguish between subjects. (Section 4.3)

1.6 Papers Authored

Chapter 2 - Scalable Swarm

Ruaridh Clark, Giuliano Punzo, Charles MacLeod, Gordon Dobie, Rahul Suman, Gary Bolton, Stephen Pierce, Malcolm Macdonald. (2017). Autonomous and scalable control for remote inspection with multiple aerial vehicles. *Robotics and Autonomous Systems*, **87**: 258–268.

- Using an Airborne Swarm for Remote Inspection [video]: https://www.youtube.com/watch?v=M_mhS_rUjbU

Ruaridh Clark, Giuliano Punzo, Charles MacLeod, Gordon Dobie, Rahul Suman, Stephen Pierce, Malcolm Macdonald. (2014). Autonomous swarm testbed with multiple quadcopters. *1st World Congress on Unmanned Systems Engineering, 2014-WCUSEng*, ACM.

Ruaridh Clark, Giuliano Punzo, Charles MacLeod, Gordon Dobie, Rahul Suman, Stephen Pierce, Malcolm Macdonald, Gary Bolton. (2015). 3D model generation using an airborne swarm. *AIP Conference Proceedings*, **1650**: 1460–1467.

Chapter 3 - Responsive Swarm

Ruaridh Clark, Giuliano Punzo, Malcolm Macdonald. (2016). Consensus speed optimisation with finite leadership perturbation in k-nearest neighbour networks. *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pp. 879–884.

Ruaridh Clark, Giuliano Punzo, Kristaps Baumanis, Malcolm Macdonald. (2016). Consensus speed maximisation in engineered swarms with autocratic leaders. *Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering*, ACM.

Chapter 5 - Brain Analysis

Ruaridh Clark, Malcolm Macdonald. An eigenvector perspective on responsive systems; from flocks to brains. In Preparation *PLOS Computational Biology*

Chapter 2

Scalable Swarm

The goal of engineering a swarm is to produce an autonomously operating system that can accommodate additional agents without having to alter its control scheme. It is this capacity for scaling that may well form the foundation of future autonomous swarms, where leadership and control become more challenging issues with the potential for vast size and complexity. To begin our journey towards such lofty ambitions, it is wise to take a first step with a simpler and more constrained system. As a result, this chapter will concentrate on circling; this is a common swarm behaviour with its continuous looping pattern often observed in natural swarming systems, where individuals can remain in motion but the swarm's position is essentially static. One of the most familiar examples is performed by schooling fish, often as a response to predator attack. Bumann et al. (1997) demonstrated that there is a significantly higher risk to fish leading a school and therefore an obvious solution for a fish, upon finding itself in a leading position, is to follow the tail of the group. This results in a circling motion that achieves a more even spread of predation risk amongst members of the school. Circling is a behaviour that is far from exclusive to predator response and can be displayed by far simpler organisms such as bacterial colonies. These colonies have ring shaped trajectories, as noted in Vicsek (2001) where Fig. 2.1 was sourced and displays a single rotating droplet containing the *Paenibacillus vortex* bacteria alongside the corresponding velocity field.

A final case of circling in nature is an illustrative example of the potential for undesirable behaviour in complex swarming systems. Army ants display a circling behaviour that can result in the death of all the ants involved. This ant mill, or death spiral, is usually a consequence of a group of blind army ants becoming disconnected from

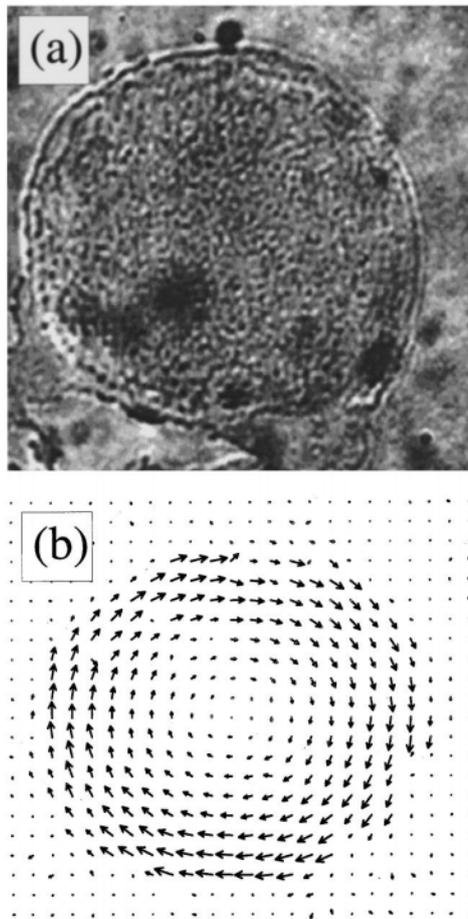


Figure 2.1: Rotating droplet of *Paenibacillus vortex* bacteria. (a) photo of droplet (b) velocity field representing bacteria motion. Figure reproduced from a paper by Czirák et al. (1996).

University of Pennsylvania, (see Fig. 2.2) by KMeL Robotics with their video, *A Swarm of Nano Quadrotors* (2012), that has, at the time of writing, been seen over 8.38 million times. Kushleyev et al. (2013) explains how their control scheme operated, stating that it relies on intermediate waypoints to guide the vehicles to their target, with the control decentralised to a certain extent by splitting the swarm into groups that follow a given waypoint. This is a step towards a fully distributed system, but would not satisfy the aim of this dissertation for a fully autonomous and distributed control scheme.

A prominent approach to swarm control is to calculate many possible trajectories, then assess the collision risk of each before optimisation routines reduce the options and

the rest of the swarm. If these disconnected ants begin to follow the splinter group's own pheromone trail a circling pattern is formed, much like schooling fish, but, unlike the fish, the ants have no way to escape. A consequence of the ants' blindness is a complete reliance on pheromones for navigation and without a disruption to the circling pattern they will eventually die from exhaustion, as noted by Beebe (1921), when he recorded a 350 m diameter spiral. Such examples motivate engineers to design systems that are not susceptible to unexpected and undesired behaviour.

The deterministic control scheme developed in this chapter shall not look too dissimilar from that shown in Fig. 2.1 (b), which is in contrast to most other multi-vehicle coordination work where precise control of every agent is employed. This includes the impressive displays at the General Robotics, Automation, Sensing & Perception (GRASP) laboratory, Univer-



Figure 2.2: KMel Robotics quadrotor swarm display, image produced by KMel Robotics (2012).

make a final selection. An example of this control scheme is detailed in Shanmugavel et al. (2010) where cooperative path planning is undertaken to ensure that all vehicles arrive at their target simultaneously. These methods can be effective when given a small enough swarm or a large amount of computing resources but the vector field approach, proposed in this chapter, is a computationally light and scalable control scheme for multiple unmanned aerial vehicles (UAVs). It is designed for implementation in a distributed manner but like the Kushleyev et al. (2013) swarm it is not fully decentralised as the distributed scheme is implemented in a laboratory environment. A central controller is, therefore, setup to simulate multiple decentralised agents with access to localisation information.

Decentralised control is achievable by using a vector field (often referred to as a Lyapunov vector field), which is a virtual physics-based method built from analytical functions that create a guidance law. Artificial potential functions (APFs) were mentioned in Section 1.2 as a popular and similar virtual physics-based design method, with this form of control extensively researched and successfully demonstrated for real-world applications. One such example is Leonard et al. (2007) where a fleet of underwater gliders were controlled with APFs, by attaching an attractive potential to virtual leaders (bodies) that have trajectories generated centrally before being broadcast to the gliders.

To progress beyond this APF based work, a control scheme is desired that is more representative of what can be seen in nature. Therefore, a method is needed that does not depend on the calculation of waypoints or trajectories for virtual leaders but still facilitates smooth, non-disrupted, motion. This chapter shall focus on defining the

agents' response to their position within an environment using vector fields. The work presented does not progress to the point of considering unknown environments but a control scheme will be applied that enables a circling swarm behaviour that can be demonstrated in a laboratory setting. To meet these criteria APF methods come close, but a potentially less disruptive approach is that of Artificial Kinematic Fields (AKFs). AKFs are defined herein to be a vector field that is created without considering an associated potential function. Frew et al. (2008) demonstrated that a vector field could produce a stable convergence to a continuous circling trajectory, referred to as a limit cycle. One such method for achieving a limit cycle is a Hopf bifurcation, where a local bifurcation about a fixed point of a dynamical system generates a limit cycle. A simple Hopf bifurcation, defined by Bennet and McInnes (2009), for an x-y coordinate frame is:

$$\dot{x} = \mu x + y - x(x^2 + y^2) \quad (2.1)$$

$$\dot{y} = -x + \mu y - y(x^2 + y^2) \quad (2.2)$$

where these equations produce a circling limit cycle behaviour. Frew et al. (2008) and Lawrence et al. (2008) considered the control of UAVs as an application of vector fields with limit cycles. Both of these works consider a more constrained flight control case, by using fixed wing aircraft instead of quadcopters, and also did not progress to real-world testing where noise and disturbances affect the effectiveness of kinematic field control.

Bennet and McInnes (2009) proved that the Hopf bifurcation is linearly stable, with the term $\dot{z} = -\alpha z$ included for maintaining the vehicle's altitude. This stability was proven by investigating the eigenvalues for the Jacobian matrix of the system, where for $\mu < 0$ and $\alpha > 0$ a linearly stable spiral forms and for $\mu > 0$ a bifurcation occurs with an oscillatory limit cycle appearing. Note, the radius of the limit cycle grows larger as μ is increased. Lyapunov second theorem was then employed by Bennet to demonstrate that nonlinear stability was also present. Bennet defined the Lyapunov function, L , as $L = \frac{1}{2} \sum_i x_i^2$, which results in $\frac{dL}{dt} = \sum_i [\rho_i^2 (\mu - \rho_i^2) - \alpha z_i^2]$ where ρ is the radial position. From this it could be seen that for $\mu > 0$ and $\alpha > 0$, $\dot{L} > 0$ if $\rho_i^2 < \mu$ and $\dot{L} < 0$ if $\rho_i^2 > \mu$. Therefore, the system is attracted to a limit cycle of radius, $\rho_i = \mu$, in the x-y plane.

Frew et al. (2008), Lawrence et al. (2008), and Bennet and McInnes (2009) form the foundation of what is developed in this chapter, where the main challenges concern

adapting theoretical control mechanisms for application and incorporating a suitable vehicle collision avoidance mechanism.

2.1 Kinematic Field Definition

The kinematic field is defined in the horizontal plane with the swarm driven to circle around the origin of the global reference frame. The fundamental structure of the field is a modified version of the Hopf bifurcation function from Bennet and McInnes (2009), which was introduced previously. This function allows the radius at which circling takes place to be defined in an x-y plane, where x and y are distances from the centre of the field, as

$$\dot{x} = c_1^o(y + \mu x) - Rx\sqrt{x^2 + y^2} \quad (2.3)$$

$$\dot{y} = c_1^o(-x + \mu y) - Ry\sqrt{x^2 + y^2} \quad (2.4)$$

where R defines the radius of the limit cycle trajectory, c_1^o is a constant, \dot{x} and \dot{y} are desired velocities, and μ is a dimensionless scalar parameter that was discussed in the previous section.

The benefit of writing the Hopf bifurcation in the form shown in eqs. 2.3 and 2.4 is that μ can be defined as follows

$$\mu = \frac{R^2}{c_1^o} \quad (2.5)$$

to guarantee a circular trajectory of radius R around the centre. This can be verified by transforming eqs. (2.3) and (2.4) into polar coordinates (r, θ) and showing that the radial velocity is always null at a distance R from the centre ($R = r = \sqrt{x^2 + y^2}$).

$$\dot{x} = c_1^o(y + \mu x) - R^2x = c_1^oy \quad (2.6)$$

$$\dot{y} = c_1^o(-x + \mu y) - R^2y = -c_1^ox \quad (2.7)$$

$$\begin{aligned} \dot{r} &= \dot{x} \cos \theta - \dot{y} \sin \theta \\ &= c_1^o(y \cos \theta - x \sin \theta) \\ &= c_1^o(R \sin \theta \cos \theta - R \cos \theta \sin \theta) \end{aligned} \quad (2.8)$$

$$\dot{r} = 0. \quad (2.9)$$

It can also be easily verified that, along the circular trajectory, the tangential velocity

(v_θ) is constant

$$\begin{aligned} v_\theta = r \dot{\theta} &= -\dot{x} \sin \theta - \dot{y} \cos \theta \\ &= -c_1^o (y \sin \theta + x \cos \theta) \\ &= -c_1^o R (\sin^2 \theta + \cos^2 \theta) \end{aligned} \quad (2.10)$$

$$\dot{r} = -c_1^o R. \quad (2.11)$$

In theory a vehicle starting at any point in the field will fall into a limit cycle, see eq. 2.3. In practise momentum gathered by a vehicle beginning outside the limit cycle may result in it overshooting and possibly passing straight over the central point. This is addressed here by the addition of a function that provides a stronger control action close to the central point, while effectively leaving the characteristics of the field, produced from eqs. (2.3) and (2.4), unaltered. This function is a radial field in the form $1/(1 + \sqrt{x^2 + y^2})$, which increases the repulsion from the centre while decreasing the attraction at large distances, thus making approaching manoeuvres smoother and preventing overshoots in the direction of the central point. The resulting field is described by

$$\dot{x} = \frac{c_2}{c_U + \sqrt{x^2 + y^2}} [c_1^o (y + \mu x) - Rx \sqrt{x^2 + y^2}] \quad (2.12)$$

$$\dot{y} = \frac{c_2}{c_U + \sqrt{x^2 + y^2}} [c_1^o (-x + \mu y) - Ry \sqrt{x^2 + y^2}] \quad (2.13)$$

where c_2 is a constant used to scale the whole expression as appropriate to fit its output within the control architecture and c_U is a unitary constant with dimensions in metres. The constant c_2 performs a scaling function for all the vectors in the field, but if it is negative then the vehicles will be repelled from the desired radius rather than attracted. In the next section, c_1 is introduced to replace c_1^o where it will influence the rotation of the field and enable collision avoidance manoeuvres. The values for the constants used in this work were found empirically, for use with the Parrot AR.Drone 2.0 (2015) in the laboratory environment as will be discussed in Section 2.2, but importantly do not need to be varied when more than one drone is used. For reference, the values are $c_1^o = 3 \times 10^5 \text{ m}^2$, $c_2 = 1.5 \times 10^{-7} \text{ m}^{-1} \text{ s}^{-1}$ and $R = 1.2 \text{ m}$.

In Fig. 2.3 the field produced from eqs. (2.3) & (2.4) where streamlines starting at the edge of the plot have the largest vectors. In Fig. 2.4 the field produced from eqs. (2.12) & (2.13) is shown where it can be seen that streamlines have a similar vector magnitude and point towards the limit cycle. The streamlines exist in an ideal simulated scenario, but the field in Fig. 2.4 has better protection against vehicles attempting to pass over

the central point as well as a reduced gradient for agents travelling in from outside the limit cycle.

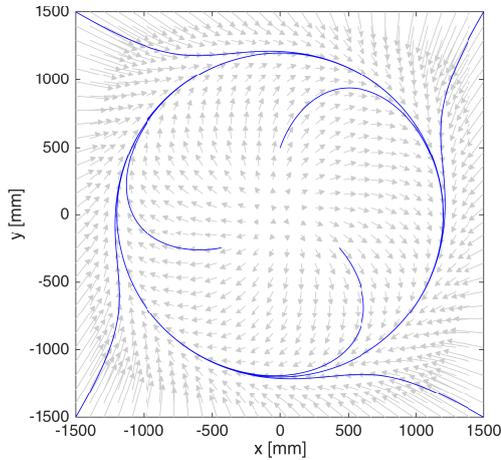


Figure 2.3: Hopf Bifurcation vector field with streamlines as defined in eqs. (2.3) and (2.4).

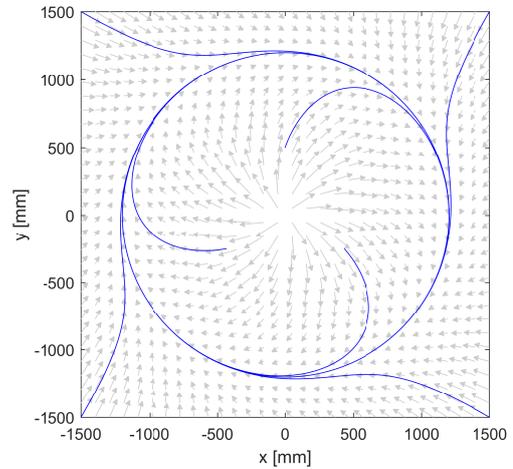


Figure 2.4: Hopf Bifurcation vector field with streamlines as defined in eqs. (2.12) and (2.13).

2.1.1 Vehicle Collision Avoidance

A popular way to perform collision avoidance, in multi-agent systems, is a form of APF where there is a mutual repulsive potential associated with each agent, see for example Bennet and McInnes (2009); Leonard and Fiorelli (2001); McCamish et al. (2010); Vasile et al. (2011). This way each agent alters the kinematic field by producing a short-range repulsive action on the other agents. This is an efficient but crude mechanism for performing collision avoidance as the trajectories generated only consider collision prevention. This creates the possibility that the APF can prevent or reverse the agents from proceeding in their desired direction, which may have a detrimental effect on system performance. Instead of an APF, a less disruptive approach will be introduced that blends well with the global kinematic field by altering an agent's field when collision avoidance is necessary.

The novel approach, to vehicle collision avoidance that is presented in this section, modifies a vehicle's kinematic field when it approaches another vehicle, thereby reducing the magnitude of the field's rotating component. In order to be effective only the trailing vehicle (with respect to the direction of the field's rotation) is inhibited. Identification of this vehicle is achieved by considering the scalar product of the relative position vector with the desired velocity vector. Fig. 2.5 depicts a collision avoid-

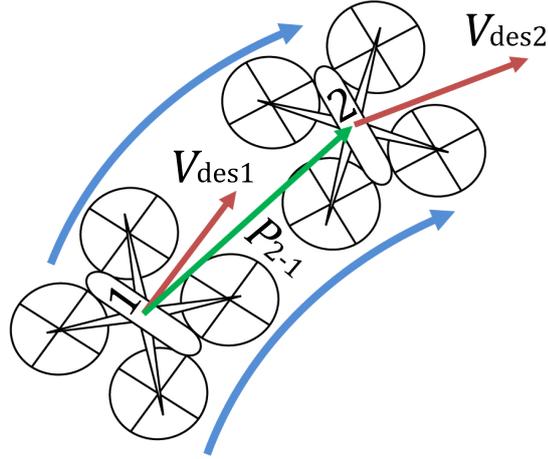


Figure 2.5: Scheme for avoidance manoeuvre, based on the direction of travel, with multiple quadcopters.

ance scenario for two quadcopters with a binary variable, h , defined on the basis of the following scalar product

$$V_{des1} \cdot P_{2-1} \geq 0 \rightarrow h = 1 \quad (2.14)$$

$$V_{des1} \cdot P_{2-1} < 0 \rightarrow h = 0 \quad (2.15)$$

where P_{2-1} is the position vector of vehicle 2 with respect to vehicle 1 in the global reference frame and V_{des1} is the desired velocity vector, as defined by the kinematic field, for vehicle 1. This enables the kinematic field to be modified asymmetrically - i.e. only the trailing vehicle, where $h = 1$, is affected.

The desired velocity of vehicle 1, as calculated in eqs. (2.12) and (2.13), is filtered to create the asymmetrically modified kinematic field. This asymmetry is achieved by replacing the constant c_1^o with the following function:

$$c_1 = hH c_1^o - (1 - h) c_1^o \quad (2.16)$$

where H is dimensionless and scales the rotational component of the field as a function of P_{2-1} and c_1^o is a constant. Eq. 2.16, given that h is either 0 or 1, essentially represents a binary choice between $H c_1^o$ when $h = 1$ and c_1^o when $h = 0$. This change does not affect the radial velocity at distance R from the target centre, which remains null, with the calculation of μ in eq. (2.5) updated with c_1 .

This asymmetrically modified field only occurs when vehicles are in close proximity. The H term that governs this proximity enables a threshold distance between two ve-

hicles to be defined, whereby passing this point results in a switch of direction for the rotational component of the global kinematic field, affecting that drone, as depicted in Fig. 2.6. The modified field enables station keeping, relative to the leading vehicle and at the defined distance from the target, until the leading vehicle moves on. The function used is in the form

$$H = \frac{|P_{2-1}| - \rho}{||P_{2-1}| - \rho|} + \frac{\rho - |P_{2-1}|}{|\rho - |P_{2-1}||} e^{-\frac{(|P_{2-1}| - \rho)^2}{c_s}} \quad (2.17)$$

where ρ defines the threshold distance between vehicles and c_s is an empirically sought scaling factor that influences the gradient of the function ($c_s = 15 \times 10^4 \text{ m}^2$). The value of c_s may require adjustment, for example to ensure collision avoidance while operating in a more turbulent environment where the vehicles may deviate further from the path defined by the kinematic field. Fig. 2.6 details how the scaling of the kinematic field's rotational component affects the modified global field for one vehicle. This figure highlights that H equals 0 at the threshold distance ($\rho = 1000 \text{ mm}$), which was chosen as a result of tests in the laboratory environment where turbulence was able to

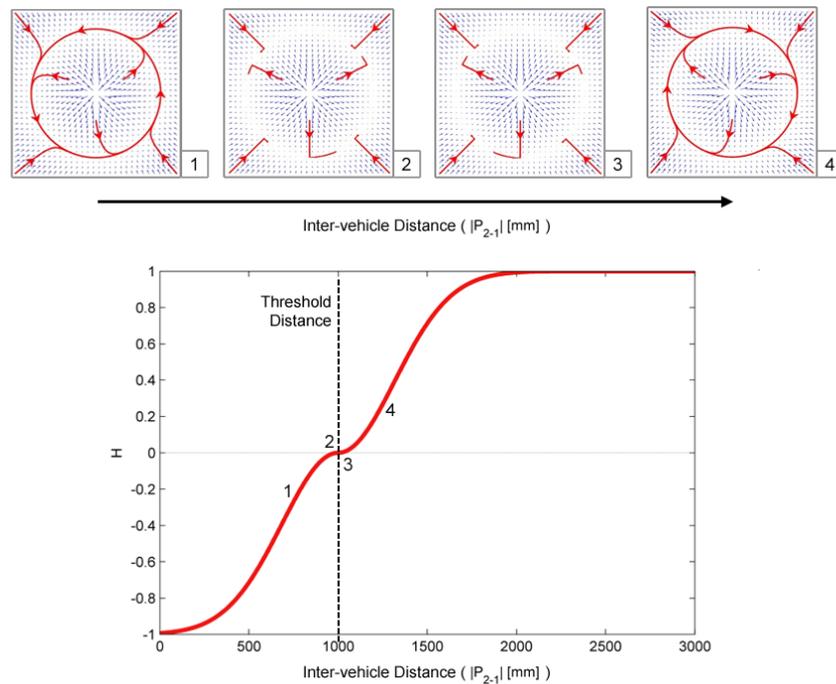


Figure 2.6: Centre: H according to (2.17) where $\rho = 1000\text{mm}$ and $c_s = 15 \times 10^4 \text{ m}^2$. Side: Instantaneous snapshots from above the modified kinematic field (centred on the target with streamlines displaying flow in the field) at different values of H ; rotating clockwise (1), reduced rotational field strength as $H \rightarrow 0$ (2), clockwise with reduced (3) and increased strength (4).

dominate the vehicle's motion when the inter-vehicle distance was less than a metre. The rotational component of the kinematic field is shown in Fig. 2.6 to act in opposite directions either side of this threshold distance. Each vehicle only considers the closest vehicle ahead of it when modifying its kinematic field and in the case that one vehicle holds its position, all of the following vehicles will form a queue along the circular trajectory.

Any final implementation of this system would require the ability to bypass agents that become stranded due to malfunction. Therefore, an alteration to the kinematic field is proposed whereby the radius of the limit cycle shall be increased as the inter-vehicle distance reduces. In effect, this allows the following vehicle to pass around the stationary one and then return to the usual radius of the limit cycle. Such an approach was not feasible given the limited size of the laboratory used (see the following section) but this would be a useful progression for a more robust system.

2.2 Hardware and Setup

To demonstrate the proposed control scheme, a series of laboratory trials were carried out using multiple aerial vehicles. These vehicles are commercially available quadcopters, Parrot AR.Drones 2.0 (see Parrot (2015) for further details), with the following capabilities:

- ◇ Embedded 720p, 30 fps, camera that has a 92° wide angled lens, which produces JPEG compressed images.
 - A rolling shutter is used that can result in significant motion blur.
- ◇ Lithium-Ion Polymer (1,000 mAh) battery provides 10–12 minutes of flight time.
- ◇ Four-rotor helicopter, or quadcopter, with a wingspan of 517 mm, see Fig. 2.7.
- ◇ Four brushless 28,500 RPM motors with long blades of radius 98.5 mm.
- ◇ Capable of transmitting status and configuration information - including battery charge, control state (landed, flying etc.), attitude and speed estimation.
- ◇ Video is recorded onto an on-board USB memory storage device.

The system tests carried out were all performed at the University of Strathclyde, in collaboration with the Centre for Ultrasonic Engineering:

- ◇ Vehicles are tracked from above by a six camera Vicon MX motion capture system, developed by Vicon (2015) and displayed in Fig. 2.8.
 - Positional tracking coverage is provided for a volume of approximately $6 \times 3 \times 3$ m, referred to from here on as the test volume.
 - 6 degrees of freedom information for an estimated error of less than ± 3 mm throughout the test volume, see Dobie et al. (2013).
 - Vehicles are identified by a unique pattern of three or more spherical (14 mm diameter) markers, see Fig. 2.7, that reflect infrared light and are affixed to the frame.
- ◇ Vehicles controlled through a computer interface connected over a 65Mbps, IEEE 802.11n (Wi-Fi), network.
 - Positional information is passed over the wireless network at 100 Hz to the computer that implements a distributed controller for each vehicle.

2.3 Guidance and Control

The software architecture is built upon open source code, developed in C# by Balanukhin (2013), to control a solitary Parrot AR.Drone that was then adapted to command multiple vehicles through a Graphical User Interface (GUI). The control scheme



Figure 2.7: Parrot AR.Drone 2.0 with 7 reflective markers.

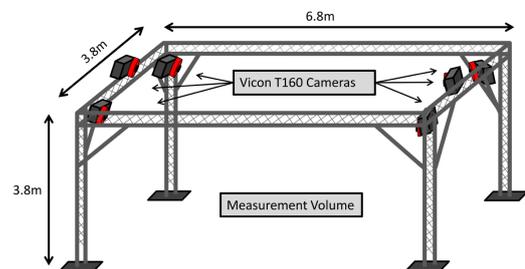


Figure 2.8: Vicon tracking system environment.

was centralised to one computer, but emulates the operations of an autonomous and distributed system whilst alleviating the quadcopters of the control algorithms computational load. The software architecture is depicted in Fig. 2.9; programme capabilities include autonomous and manual control of the vehicles while displaying the status and navigation data through the GUI. In this work, a maximum of three vehicles are used, but the architecture is designed to accommodate far larger numbers. Fig. 2.9 shows that the distributed controller and GUI have their command outputs processed by the Command Sender before being passed onto the vehicles. This allows manual commands to be passed to the vehicles while they are in autonomous flight; enabling, for example, the initiation of video recording or a switch from autonomous mode into manual flight control. The Command Sender passes on commands at 30 Hz to each vehicle, as recommended in the Parrot developer guide Piskorski et al. (2012), to ensure smooth flight. The commands, including yaw rate, vertical velocity, pitch and roll angle, are transformed to 32-bit integers according to the IEEE754 standard before being transformed to an ASCII string and passed on to the vehicle control software through a User Datagram Protocol (UDP) port.

A kinematic field provides a highly nonlinear guidance law, which is a function of the vehicle's position with respect to a central target, that is mapped to the control action through a linear controller. This arrangement generates smooth trajectories for the vehicles, with the control architecture illustrated in Fig. 2.10. The desktop com-

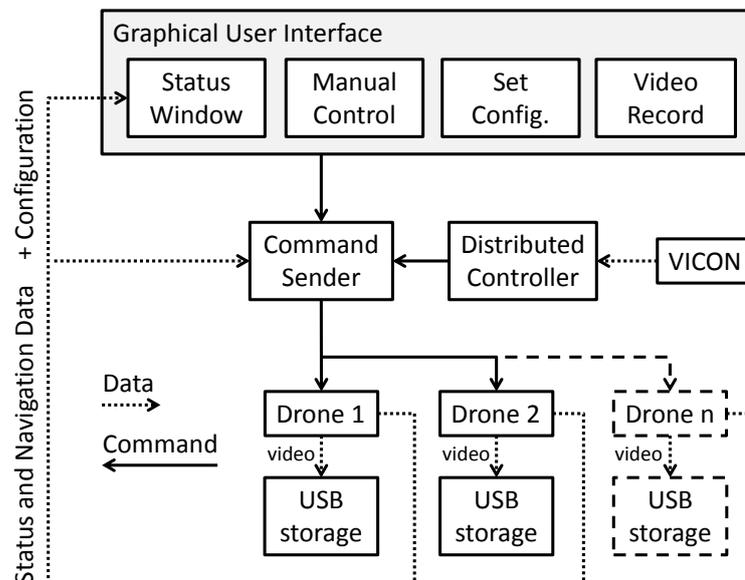


Figure 2.9: Software architecture for the control of multiple vehicles.

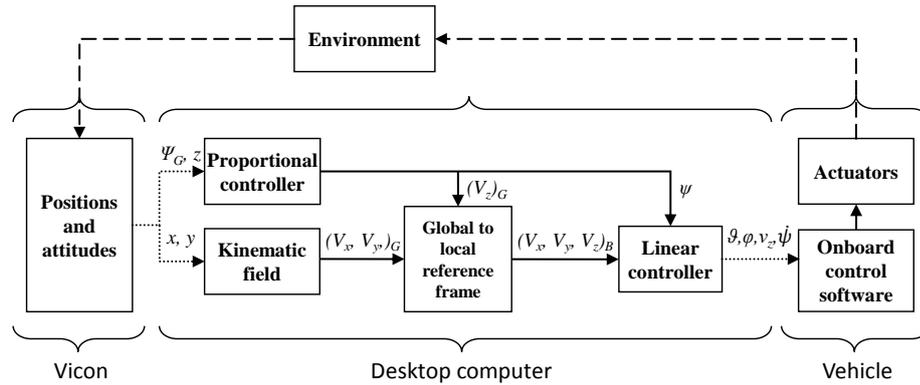


Figure 2.10: Control architecture scheme for one vehicle (dotted line indicates wireless data transfer; dashed line indicates causal effect).

puter is provided with the vehicle’s own position and the relative positions of the other vehicles by the Vicon system. Based on this information, the local kinematic field of each vehicle is computed to produce the desired velocities in the horizontal plane of the external reference frame (i.e. the flight volume). These desired velocities are then converted to the vehicle centred (local) reference frame, before they are passed to a linear controller. This controller provides the pitch and roll angles to the on-board controller with desired yaw and vertical speeds also supplied. This in turn commands the motors to execute the requested manoeuvre. The control scheme is designed to be highly scalable if the vehicles have sufficient on-board processing capabilities and position tracking. The on-board processing is limited, for the vehicles used in the following work (Parrot AR.Drone 2.0), with the control handled, in a distributed manner, by a central processor. The specifics of this scheme are outlined in the following paragraphs.

2.3.1 Altitude Control

A proportional controller is implemented to control the altitude of the vehicles, which operates in conjunction with the quadcopter’s on-board, ultrasound dependant, altitude controller. The output of the proportional controller is converted from the global to the body reference frame as shown in Fig. 2.10. The requirements for altitude control were simple with the vehicle decreasing its height by a set interval after target coverage was complete at the current altitude, allowing a simple z -axis waypoint to be used in conjunction with this controller to transition to different altitudes. To determine if

complete coverage has been achieved at any given altitude, each vehicle has to have knowledge of where the vehicle ahead of it started on the current coverage altitude. This knowledge would allow the trailing vehicle to reach that starting point before transitioning to another altitude. In this way the control scheme remains easily scalable with collision avoidance and coverage determination requiring each vehicle to only observe the one ahead of it.

2.3.2 Attitude Control

For a typical quadcopter, pitch and roll angles are coupled with the forward and lateral motion respectively. This design enables forward or side force components to be produced by tilting the vehicle. When no forward or side movements are commanded, the vehicle hovers and in this phase the attitude is controlled in closed loop by the on-board controller only. This is overridden by control commands when altering the yaw angle, which is controlled in the same closed loop manner as the altitude. For the inspection task, discussed in the following case study, the attitude controller keeps the vehicle's x -axis pointing in the direction of the target whilst the quadcopter manoeuvres around it. This is a requirement as the Parrot AR.Drone has a rigidly held front facing camera. As a consequence, the desired azimuth changes with position. This is defined as

$$\psi_{des(i)} = \text{atan2}(y_i, x_i) \pm \pi \quad (2.18)$$

where x_i and y_i are the coordinates of the vehicle in the global reference frame that is centred on the target and $\text{atan2}(y_i, x_i)$ is similar to calculating the arctangent of y_i/x_i , except that the signs of both arguments are used to determine the quadrant of the result in the range $[-\pi, \pi]$. To set the desired vehicle angle towards the centre of the field $\pm\pi$ is applied to ensure the result remains within the range $[-\pi, \pi]$. The error in the actual angle is then mapped to an angular rate through a linear controller that selects the shortest rotation direction to reach the desired angle.

2.3.3 Linear Control

The linear controller maps the desired velocity of each vehicle to commanded pitch and roll angles. The desired velocity vector is decomposed along its forward and lateral components in the body reference frame and these are scaled by a proportional

controller. The result is then filtered to output in the range $[-1, 1]$, required for the AR.Drone on-board software, by using the hyperbolic tangent function

$$\gamma^* = \tanh(\gamma) \quad (2.19)$$

where γ is the vector of the controlled variables (including the roll angle φ , pitch angle ϑ , vertical velocity v_z and yaw rate ψ) and γ^* is the normalised output.

A proportional controller is used to map from desired forward and lateral velocities according to the kinematic field, vertical velocity and azimuth angle to commanded pitch angle, roll angle, vertical velocity and yaw rate. The controller is expressed by

$$\begin{pmatrix} \vartheta \\ \varphi \\ v_z \\ \psi \end{pmatrix} = \begin{pmatrix} c_\vartheta v_{des(x)} \\ c_\varphi v_{des(y)} \\ c_z (Z_{des} - z) \\ c_\psi (\Psi_{des} - \psi) \end{pmatrix} \quad (2.20)$$

where, $v_{des(x)}$ and $v_{des(y)}$ are the forward and lateral velocities in the body reference frame produced by the kinematic field, Ψ_{des} is the desired azimuth angle that varies with time, ψ is the actual one, and c_ϑ , c_φ , c_z , c_ψ are the gains of the proportional controller. The values of the gains were sought empirically and specifically to achieve smooth dynamics with the Parrot AR.Drones with $c_\vartheta = 0.7 \text{ s}^{-1}$, $c_\varphi = 0.7 \text{ s}^{-1}$, $c_z = 1 \text{ s}^{-1}$ and $c_\psi = 1.5$ used. The difference in the gains defined are a result of the quadcopter design with similar control movements resulting in the same gains for c_ϑ and c_φ .

2.4 Performance

The focus of the AKF approach is to generate smooth trajectories automatically. This field then enables autonomous flight for multiple vehicles simultaneously. However, the trajectory errors shall provide context for the restriction of the current approach and the progress that can be made in future work.

2.4.1 Trajectory Errors

A circling flight, where the vehicle is required to complete five orbits around a central object, starts at 1.5 m altitude and descends 0.3 m after each orbit, finishing at 0.3 m altitude. This approach helps to quantify the effect that ground proximity has on flight

errors with this trajectory directly applicable to the case study that will be discussed shortly.

The standard deviation of the error for each band is detailed in Table 2.1 where the errors in height position indicate that flights at lower altitude were more susceptible to error (the error ellipsoids are visualised later in this chapter in Fig. 2.15). It is assumed that the ground effect - downwash from the quadcopter blades - is responsible for the larger errors with a more turbulent flight environment present when flying close to the floor. A similar trend can be seen in the radial error for one vehicle but in the

Table 2.1: Standard Deviation of Positional Error from Multiple (1 & 2 Vehicle) Drum Inspection Flights

Nominal Height [mm]	1500	1200	900	600	300
1 Vehicle: Mean of radial error	134	168	202	219	197
Standard Deviation [mm]					
1 Vehicle: Mean of height error	20	65	65	65	89
Standard Deviation [mm]					
2 Vehicles: Mean of radial error	205	167	201	160	206
Standard Deviation [mm]					
2 Vehicles: Mean of height error	38	67	75	85	104
Standard Deviation [mm]					

two-vehicle case this trend is not obvious, which is probably a result of increased air turbulence resulting in noisier data.

The large radial error in the two-vehicle case, at height 1500 mm, is likely due to the lack of control on the starting positions of the drones combined with the increased turbulence when compared with a solitary drone. The plot in Fig. 2.11 supports this claim with the initial positions seen to be offset from the radial path and the control method unable to prevent overshoot when attempting to maintain a radial distance. It also appears that, in this case at least, the time to complete an orbit decreases as the vehicles descend to lower altitudes with this fast coverage again likely to be a consequence of the ground effect, a common aerodynamic phenomenon.

2.4.2 Control Improvements

More precise positional control, than has been presented here, is possible with way-point based trajectories, but a reduced level of precision is accepted to enable a dis-

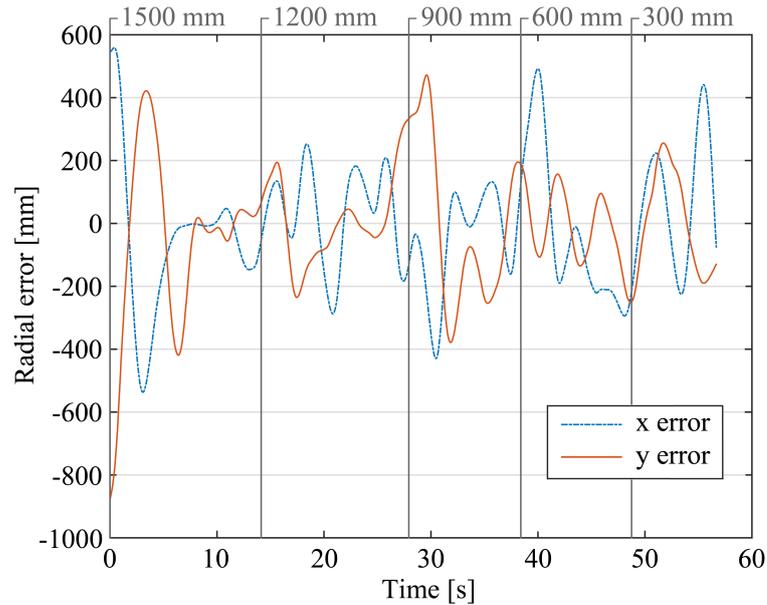


Figure 2.11: Representative flight data from one vehicle during a two-vehicle flight with the altitude transitions marked above the plot.

tributed and more scalable system. That is not to say that the errors cannot be reduced as the proportional (P) controller, described in Section 2.3, can be substituted with a proportional and derivative (PD) controller to enhance the control performance and, in particular, reduce the overshoot. Currently the vehicles fly at a relatively low speed, which allows the desired pitch and roll to be commanded sufficiently with only the AKF-defined velocity (see Eq. 2.20) rather than its difference with respect to the actual velocity. The introduction of a PD controller would instead consider the error in velocity in the horizontal plane and its derivative. The error along the vertical axis and the yaw angle could also be controlled by a more straightforward PD controller.

2.4.3 Collision Avoidance

The collision avoidance mechanism is required even when only two-vehicles are flying in the test volume; the variable starting positions and turbulent flight environment can, within such a limited space, result in vehicle-vehicle collisions. The collision avoidance can be used to form a queue of vehicles, as displayed in Fig. 2.12; when the leading vehicle remains stationary, the other vehicles will attempt to remain at a threshold distance from one and other. In nominal operation the collision avoidance will have a reduced influence from the queuing scenario, where the approaching vehicle will slow or stop briefly before resuming normal operation when the inter-vehicle distance grows

sufficiently large.

In Fig. 2.13, a sample of the angular velocity of the trailing vehicle (determined from the angular position as recorded by Vicon during a two-vehicle flight) is compared with the collision avoidance function H , see Fig. 2.6. It can be seen that there is a delay between command and execution but it is also clear that the rotational component of the vehicle's velocity decreases and begins rotating in the opposite direction in the places where the H falls below zero.

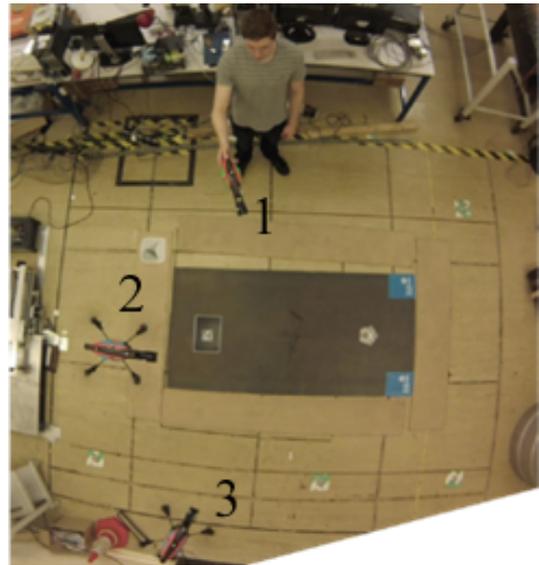


Figure 2.12: Testing collision avoidance; two vehicles queuing behind the shell of another.

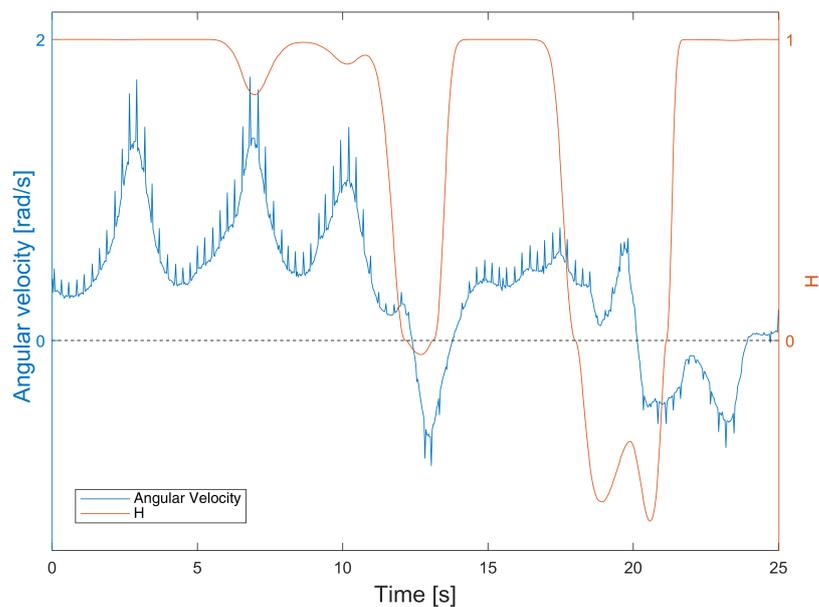


Figure 2.13: Comparison of flight data with collision avoidance function.

2.5 Case study: Remote Structural Inspection

The proposed kinematic field based control scheme was demonstrated with an application of an engineered swarm; remote structural inspection. This application employs autonomously or remotely controlled vehicles to inspect structural assets such as tanks, flare stacks, chimneys, and wind turbines. In this particular case, aerial vehicles (quadcopters) are used to autonomously and visually assess the state of a structure. Examples of visual structural inspection, that use aerial vehicles, include Fraundorfer (2015) who presented a 3D reconstruction of a building's facade from aerial imaging, while Ortiz et al. (2014) demonstrated an autonomous approach to the visual inspection of metallic vessels to identify coating breakdown, corrosion, and cracks. Remote inspection has also received interest from Khanna et al. (2015) in reviewing agricultural fields and from Remondino et al. (2011) in recording sights of archaeological interest.

2.5.1 Inspection Setup

For this case study an inspection of a nuclear intermediate level waste (ILW) storage drum was undertaken, see Fig. 2.14. The drum is a waste packaging and encapsulation plant liquor drum, from the Sellafield nuclear reprocessing site and constructed from 316 Stainless Steel, that has a diameter of 800 mm and a height of 1200 mm. The test volume allows for the drum to be circled by the Parrot AR.Drones, but the number of

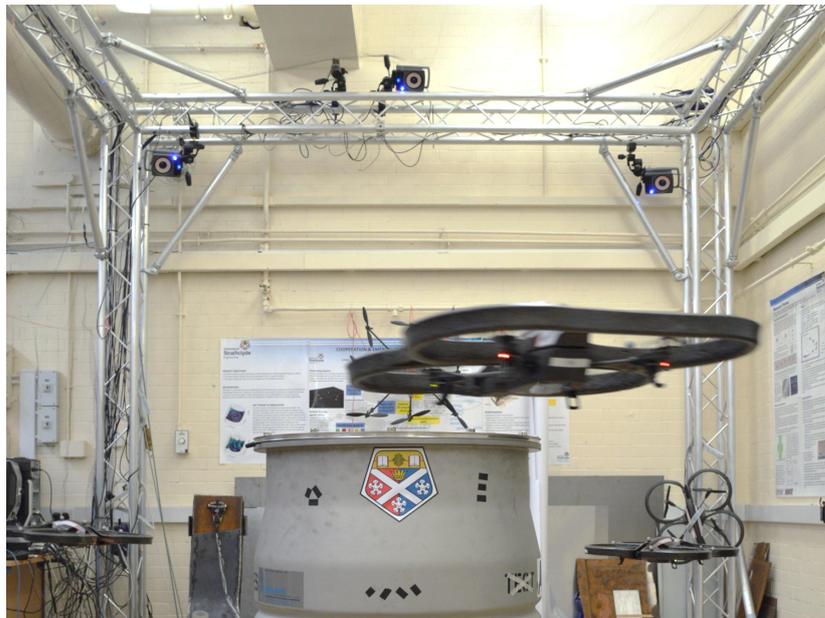


Figure 2.14: Three vehicles circling the ILW storage drum in the tracking environment.

vehicles used was restricted due to the volume constraints with the inspections carried out by two vehicles.

The Parrot AR.Drones are equipped with an ultrasound sensor for performing on-board height stabilisation control. Only two frequencies are available for the ultrasound, therefore a three-vehicle system in close proximity, although tested in the test volume and capable of operating, is prone to disruptive ultrasonic sensor interference. Even in a two vehicle scenario, the variation in starting conditions and the limited test volume for the two vehicles often results in some, usually minor, collision avoidance being required.

To complete the inspection, two vehicles takeoff and then enter autonomous flight; rising above the top of the drum to the first coverage band at a height of 1.5 m. Coverage is achieved by using coverage bands, as mentioned in Section 2.3.1, whereby each vehicle registers the position of the vehicle ahead and once all the vehicles have reached their registered position they transition to another band to repeat the process. For this case each band was separated by 30 cm with the vehicles landing after completing the final band at a height of 30 cm above the floor. Each band took 10-15 seconds to complete with five bands used for the drum inspection, with the bands and drum shown in Fig. 2.15 where the error ellipsoids relate to the positional errors presented in Section 2.4.1.

The visual inspection is carried out by the 720p HD camera, rigidly incorporated into the AR.Drone's main structure, with in-flight video recorded onto an on-board USB memory storage device. This device is then removed and the footage processed after

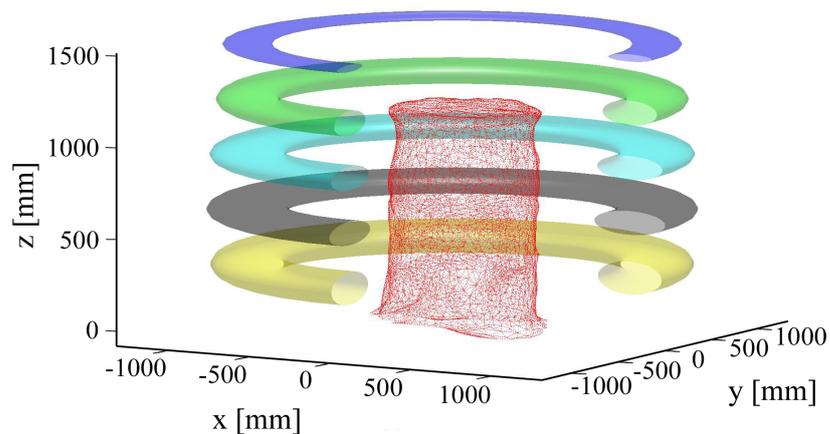


Figure 2.15: Mean error flight ellipsoids at the five coverage bands for one vehicle completing the drum inspection.

the flight.

2.5.2 Three-Dimensional Model Construction

The inspection can be completed using the aforementioned control methods but the collected data needs to be processed before reviewing the structural state of the target. An intuitive method for displaying visual inspection data is to map it into a three-dimensional (3D) space to allow a reviewer to explore an object to check for faults and failures. Photogrammetry analysis can enable the creation of a 3D model from the inspection images, providing an estimation of the geometry as well as a reviewable environment. This model construction shall be briefly described to give context to the challenges of using a swarm for such an application.

State-of-the-art image-based reconstruction systems derive markers or features from the texture information present in the images and by triangulating markers, creating point clouds corresponding to the geometry of the scene. This method is attractive when dealing with remote environments and has been previously investigated in the literature El Kahi et al. (2011); Hansen et al. (2013); Harris and Stephens (1988); Shi and Tomasi (1994). In particular for robotic inspection Dobie et al. (2013) but also for archaeological work Brutto and Meli (2012); Kersten and Lindstaedt (2012), where the reconstruction tools were the same as those adopted in this work. The system detailed here uses this reconstruction method to enable a remote inspection that then provides data review in a similar manner to an in-person inspection.

2.5.2.1 Photogrammetry

The recorded video footage is processed to enable the creation of a 3D model using photogrammetry analysis carried out by Autodesk's 123D Catch software Autodesk (2015). The challenges included selecting frames from the video recorded that were clear and detailed present with the footage prone to suffering from motion blur. These frames then had to be corrected for the distortion caused by the lens of the Parrot AR.Drone 2.0. A calibration was carried out, as described by Azad et al. (2008), that enables objects that appear curved because of the lens distortion to be straightened to more accurately represent their real shape. Fig. 2.16 (a) shows a vehicle-recorded image and (b) is the post-processed output, where some of the image at the borders of (a) has been lost in the distortion correction process.

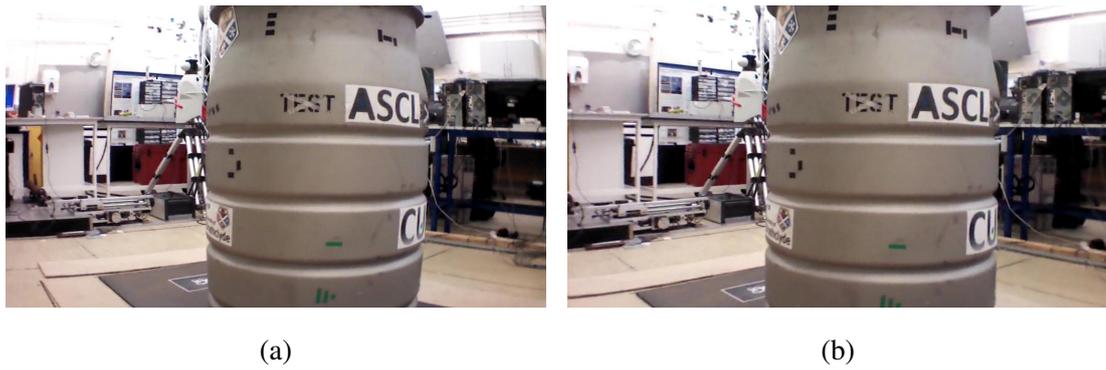


Figure 2.16: (a) Image from inspection footage and (b) distortion corrected image.

A 3D surface-meshed model of the drum’s lid and side is created as a product of the image matching and stitching with the recorded images mapped to the 3D mesh point cloud. The final textured CAD model, seen in Fig. 2.17, was made possible by manual input during the stitching process, where points could be selected that appeared in at least three images. The manual inputs and artificial markers had an influence on the final model by ensuring there were no gaps in the model construction and to prevent image placement errors that could produce significant but localised errors.

2.5.3 Improvements & Automation

To inspect a large outdoor structure, such as a chimney, the Parrot AR.Drones could be replaced by commercial grade inspection vehicles such as the Falcon 8 from *Ascending Technologies* (2015) and the Vicon MX positioning system could be replaced by GPS or for improved accuracy, the work of Misra and Enge (2006) on differential GPS.



Figure 2.17: Generated 3D model produced from two-vehicle flight footage.

Pose uncertainty would increase when comparing a GPS tracking system with that of Vicon, but the control scheme can be adapted to compensate, with the collision avoidance threshold distance increased as well as increasing the safety margin on the set distance to target.

2D-2D feature correspondence data was added manually to aid the reconstruction process used in 123D Catch. This step was necessary due to the low texture associated with the target, which in turn was compounded by the compression used by the on-board camera. A fully automated system could be realised through modifications to the hardware and algorithms with a possible version of such a system presented in Fig. 2.18. It is envisioned that this system would carry on generating trajectories until making an autonomous decision on whether coverage was complete, with target coverage requirements being manually inputted before the flight. The use of wireless data transfer and a higher resolution imager, with some form of image stabilisation, would enable full automation of the model generation process. Errors in the final model could also be reduced by automatically combining localisation information to ensure the accuracy of image location prior to image stitching. Finally, a combination of automated and manual inspection of the final data would probably be necessary to complete the inspection.

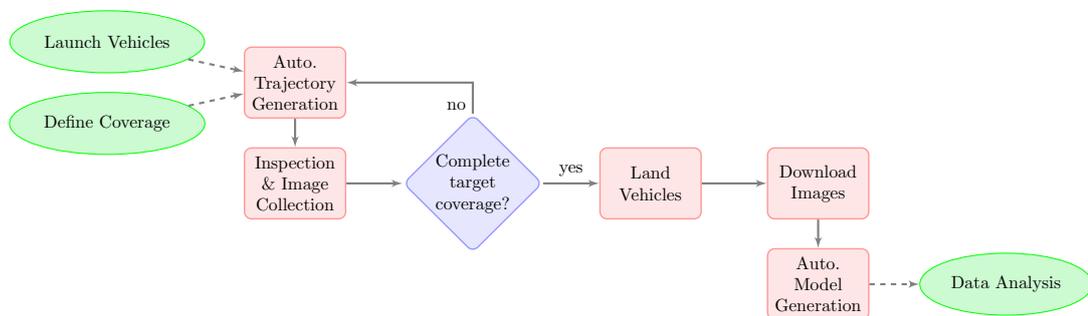


Figure 2.18: Flowchart of a feasible fully automated version of the system. Green requires user input, red is autonomously executed and blue is a decision process.

2.6 Summary

A scalable and autonomous control scheme has been presented for creating a circling swarm behaviour. The proposed technique benefits from the relative simplicity of the control scheme as the trajectory does not have to be re-evaluated if the vehicle diverts from the planned course or requires another process to return the vehicle to the initial

path before continuing. The key and novel aspects of this distributable and scalable control are to be found in the vehicle and target collision avoidance developments. The scheme has been implemented for three vehicles, where the collision avoidance was tested with the vehicles forming a queue while maintaining their defined distance from the target, and validated through a remote inspection case study where two-vehicles performed an autonomous inspection on a nuclear ILW storage drum. The results achieved highlight the potential for autonomous trajectory generation that, in theory and in the absence of spatial constraints, could be increased in scale whilst still producing a similar performance to the smaller systems demonstrated in this chapter. The technological constraints, highlighted and discussed in this chapter, are not the only stumbling blocks to realising a vast swarming system. Our understanding of how to identify or create agents, which can effectively lead the whole system, is also in need of advancement.

Chapter 3

Responsive Swarms

The previous chapter established a mechanism for creating a large engineered swarm that could approximate the performance and scale of natural swarming systems. Increasing scale without a rigid structure leads to a more complex system that in turn opens the door to unexpected behaviour and complete failure. An excellent example of a complex system behaving unexpectedly is the development of a neural network - an autonomous behaviour design approach - based Artificial Intelligence (AI) for playing the Super Smash Bros. video game by Firoiu et al. (2017). Neural network AIs operate like a black box with designers not fully aware of the reason decisions are made, but they often prove effective as in this case where the AI played and beat ten of the world's top ranked players (ranked between 16th and 70th). However, and this is of vital importance when considering an engineered system, the lack of system understanding can lead to failure despite a good performance in nominal conditions. In this case, if the opposition player remain crouched at the side of the battle area for a long enough the AI would “behave very oddly, refusing to attack and eventually KO-ing itself by falling off the other side of the stage”. While the ramifications of a video game character falling to their demise are minimal, a real world robotic system could pose a more substantial threat if it were to behave erratically. The removal of the black box of control is, therefore, essential.

When considering a swarming system, the product of agent interactions could lead to unexpected behaviours and would represent a black box if uninvestigated. This chapter endeavours to understand how these interactions can lead to consensus by using graph theoretic tools to see past the obscuring tangle of connections and reveal how nodes are influenced by the state of others in the network.

3.1 Graph Theory

This chapter will lean heavily on the definitions and tools of graph theory with this section serving as a brief introduction to some of the more pertinent aspects. Graphs come in many forms but an important distinction, for this work, is that of directed and undirected graphs. An undirected graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where there is a set of \mathcal{V} nodes and \mathcal{E} edges, which are unordered pairs of elements of \mathcal{V} i.e. the connection between two nodes can be traversed in either direction. A directed graph has a similar definition but the edges are ordered pair of elements of \mathcal{V} . The undirected graph can be considered to be a special case of the directed graph, where an edge $(i, j) \in \mathcal{E}$ denotes a connection between agent i and agent j . The undirected graph edge (i, j) corresponds in the directed graph to edges (i, j) and (j, i) . Graph theory allows for self edges $(i, i) \in \mathcal{E}$, but these shall not be considered in the work detailed here.

The degree of a node is defined, for an undirected graph, as the number of edges connected to that node. For directed graphs, indegree and outdegree are the terms used; indegree - number of connections entering a node, outdegree - number of connections leaving a node.

A path is a sequence of edges in a graph. That is, there are connections that connect the node at the start of the path to the one at the end. A subset of paths are cycles, where the same node is at the start and end of the path. An important definition is that of connectedness for undirected graphs, where they are connected if there is a path between every pair of nodes. Directed graphs, on the other hand, can either be strongly or weakly connected. To be strongly connected there has to be a sequence of nodes that are connected with each edge directed in the same direction. To be weakly connected there only has to be a sequence of nodes connected by edges where the direction of the edges does not matter. Another similar concept, which is necessary for consensus (see Lemma 3.1), is spanning trees. A spanning tree occurs when every node in \mathcal{V} can be reached by a single node. For the undirected case, the existence of a spanning tree means the graph is connected and vice versa. For the directed case, a spanning tree is only evidence of a weakly connected graph and, therefore, does not guarantee strong connectedness. A specific case for a directed graph is an arborescence (or rooted directed spanning tree) where there is a root node that has only outgoing connections with all other network nodes only having one parent. A parent node is defined here as a node that has an outgoing connection to any number of child nodes. A directed spanning tree does not require its root node to have no incoming connections, nor

are the other nodes required to have only one parent but the arborescence exists as a subgraph $(\mathcal{V}', \mathcal{E}')$ of $(\mathcal{V}, \mathcal{E})$ of this tree.

The adjacency matrix, A , is a square $N \times N$ matrix when representing a graph of N nodes. This matrix captures the network's connections where $a_{ij} > 0$ (a_{ij} is the ij entry of the graph adjacency matrix) if there exists a directed edge from node i to j and 0 otherwise. Variable edge weights contain information on the relative strength of interactions, whilst uniform edge weighting either only represent the presence of a connection or is a result of all the edges having the same information carrying capacity. For an undirected graph, the adjacency matrix is symmetric with an edge $(i, j) \in \mathcal{E}$ resulting in $a_{ij} = a_{ji} > 0$.

The Laplacian matrix is composed of the adjacency matrix and the degree matrix, D , as

$$L = D - A$$

where the degree matrix is a diagonal matrix where the i^{th} diagonal element is the out-degree of node i , which is equivalent to summing the elements of row i of A . This makes L diagonally dominant, as the off-diagonal elements are nonpositive and the diagonal elements nonnegative. The Laplacian has many useful properties that emanate from the fact the row sum is always zero. For instance, this row sum property assures that there is at least one zero eigenvalue of L that is associated with a connected component of the graph. In the case where there is more than one zero eigenvalue, the number of zero eigenvalues represent the number of connected components of the graph. Dhal et al. (2014) notes that the dominant eigenvalue (zero eigenvalue in this case) is real and, for a strongly connected graph, has an algebraic multiplicity of 1 (referred to as a simple eigenvalue). When there is a single zero eigenvalue there is only one component, known as the giant component. For directed graphs the situation is slightly more complicated where a giant component will always be present in strongly connected graphs. A giant component may also be present for weakly connected graphs but it is not guaranteed; for example, at least two graph components will be present in a weakly connected graph with two nodes that have no outdegree.

Gershgorin's disc theorem can be used to understand the position of the non-zero eigenvalues. This theorem can be used to show that the Laplacian matrix is positive semidefinite, i.e. all the matrix eigenvalues are nonnegative. When plotted with real and imaginary axes every eigenvalue of a complex, square matrix lie within the Gershgorin discs. Each disc is associated with a row of the square matrix and has a radius, $R_i = \sum_{j \neq i} |a_{ij}|$, that is equal to the sum of the absolute value of the off-diagonal

elements. The centre of each disc is the diagonal element of the row, therefore there will be N discs for an $N \times N$ matrix. Since the diagonal elements of a Laplacian matrix are invariable nonnegative, then none of the Gershgorin discs are centred in the negative real part of the plane. Now consider that the sum of the off-diagonal elements is never greater in magnitude than the diagonal element, this prevents this disc from ever crossing into the negative real plane. Therefore all the eigenvalues must be nonnegative and for $-L$ all the eigenvalues are nonpositive, which is a necessary condition for a stable system. It is also possible to say that, in the undirected case, $\lambda_1 = 0 \geq \lambda_2 \geq \lambda_i \geq \dots \geq \lambda_N$ where λ_i is the i^{th} smallest eigenvalue of L . The smallest, non-zero eigenvalue, is the algebraic connectivity (λ_2) that quantifies the convergence rate of consensus for a system. Directed graphs are similar but there can be complex eigenvalues - that occur in conjugate pairs - therefore it can only be said that $\text{IR}(\lambda_1) = 0 \geq \text{IR}(\lambda_2) \geq \text{IR}(\lambda_i) \geq \dots \geq \text{IR}(\lambda_N)$. In the directed case the smallest non-zero eigenvalue, λ_2 , continues to capture the convergence rate to consensus, with this metric vital to the majority of work to follow where consensus, and the nodes most effective at influencing it, shall be explored.

3.1.1 Consensus

Consensus is a process that begins with every agent having an initial value, which differs from some or all of the other agents in the network or from a target value. When successful a system converges to one value for all agents, by exchanging information with neighbours in an attempt to minimise the difference between their values. The consensus process is fundamental to many networked systems and has been investigated in relation to flocks and swarms, applied in distributed computing and in algorithms like PageRank, see Page et al. (1999). In the case of flocking, Jadbabaie et al. (2003) applied consensus analysis to explain the emergence of alignment in a simple flocking model developed by Vicsek et al. (1995).

As mentioned in the previous section, the negated Laplacian provides a stable system with the collective dynamics of linear consensus expressed as

$$\dot{x} = -Lx, \quad (3.1)$$

which Murray (2003) proved will guarantee that consensus is reached asymptotically when the graph is connected. Writing this consensus protocol in another way, by

looking at it from the point of agent interactions, as

$$\dot{x}_i = \sum_{j=1}^N a_{ij}(x_j - x_i) \quad (3.2)$$

reveals how the Laplacian matrix captures the averaging process conducted by agents to achieve consensus.

3.1.2 Perturbation Driven Consensus

Perturbation driven consensus is a critical aspect of design for a multi-agent system that is required to respond rapidly to external stimuli. It also appears to be a slightly neglected topic of study, as most similar work has concentrated on the effectiveness of selected leaders rather than applying a variable perturbation to one or many agents in the network. The networks considered herein have N agents connected via local communication with a static, time-invariant, topology. This is represented with a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, that is at least weakly connected with one giant component. A uniform signal $\mathbf{u} = \mathbf{u}[\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}]^T \in \mathbb{R}^N$ is supplied to all agents with different positive gains c_i , where $i = 1, 2, \dots, N$. The dynamics of this system are defined as

$$\dot{x}_i = \sum_{j=1}^N a_{ij}(x_j - x_i) + c_i(u - x_i) \quad (3.3)$$

where x_i is the state of the i^{th} agent and u is the scalar target value that all agents must achieve. The resource allocation, c_i , ranges from 0 to 1, where $\sum_i c_i = 1$, and scales the comparison between the uniform input signal, u , and the current state x_i .

The global dynamics of the network can be expressed as

$$\dot{\mathbf{x}} = -L\mathbf{x} + C(\mathbf{u} - \mathbf{x}) \quad (3.4)$$

where C is the perturbation matrix, $C = \text{diag}(\mathbf{c}) = \text{diag}(c_1, \dots, c_N)$. Spanning trees, highlighted previously as vital for consensus, have been explored by Jadbabaie et al. (2003); Mesbahi and Egerstedt (2010); Shao et al. (2015) who produced the following lemma.

Lemma 3.1. *For a directed network \mathcal{G} defined by Eq. (3.4), consensus could be achieved if \mathcal{G} contains a directed spanning tree, that is, a graph that does not contain a directed*

cycle and there exists a vertex u such that for any $w \in \mathcal{V}$, there exists a directed path from u to w .

The essence of Lemma 3.1 implies that for a directed network, the sufficient condition for achievement of consensus is that each agent is reachable from the input u through a directed path.

Eq. (3.3) can then be transformed to

$$\dot{\mathbf{x}} = -(L + C)\mathbf{x} + C\mathbf{u} . \quad (3.5)$$

Consider the following coordinate change

$$\mathbf{v} = \mathbf{x} - (L + C)^{-1}C\mathbf{u} . \quad (3.6)$$

Applying this affine transformation to Eq. (3.5) will remove the $C\mathbf{u}$ term with the model becoming

$$\dot{\mathbf{v}} = -(L + C)\mathbf{v} . \quad (3.7)$$

Punzo (2013) demonstrated that for the system $(-L - C)$ the perturbation would be non-singular; always producing a stable response, with the eigenvalues remaining negative. The non-singular property being necessary to satisfy $(L + C)^{-1}$ in Eq. (3.6). There would be no zero eigenvalue as $(-L - C)$ is Hurwitz with Punzo (2013) proving this for when the diagonal perturbation $C = \text{diag}(\mathbf{c})$ has nonnegative entries \mathbf{c} as long as $\langle c, v \rangle \neq 0$ where v is the first left eigenvector of L (the eigenvector associated with the smallest eigenvalue in magnitude). This stipulation essentially ensures that all i nodes without an outdegree are supplied with a perturbation resource, $\mathbf{c}_i \neq 0$, otherwise the perturbation would have no effect. The conclusion of this work was that a perturbation supplied to a node that is the root of a spanning tree, i.e. globally observable, will influence all of its child nodes and eventually result, in the absence of noise, in a graph reaching consensus.

3.2 Constant Outdegree Networks

The networks considered in this section are constrained to be k -outdegree (constant outdegree). A similar network structure is often seen in nature with starlings and schooling fish, which are found to have network connections that form according to

topological rather than metric distance, see Ballerini et al. (2008). Such a topological rule would therefore seem suitable and implementable for a scalable robotic system where each robot could be instructed to monitor k neighbours.

The constant outdegree constraint also produces useful mathematical properties, such as a uniform first right eigenvector (FRE). For a given system - the outdegree of node A represents node A following node B's state, and the indegree represents node B monitoring node A's state - the first left eigenvector (FLE) can be colloquially described as a ranking metric in terms of a nodes influence in the network. High ranking nodes, according to the FLE, are those that collate information from across the network. While the FRE could be seen as ranking a nodes influenceability, i.e. how easy it is for others in the network to influence that particular node. High ranking nodes, according to the FRE, are those who receive information from sources across the network. The constant outdegree, and associated uniform FRE, means that all nodes have the same influenceability but this does not mean that all nodes will be equally effective network leaders. It does, however, mean that analysis relying on only the left eigenvector can be an effective strategy for detecting optimal network leaders. This would not necessarily be the case if outdegree values varied enough to cause some nodes to receive far less information and, as a result, become harder to lead to a new consensus state.

The adjacency matrix was previously introduced, see Section 3.1, but shall now be more specifically defined for the k -outdegree case to be $a_{ij} = 1/k$ if there exists a directed edge from node i to j , and 0 otherwise. Each row of the adjacency matrix will, therefore, have the same sum total,

$$\sum_j a_{ij} = 1 \quad \forall i, j \in \mathcal{V}. \quad (3.8)$$

3.2.1 Consensus Speed Limit

A definition for the theoretical upper bound for convergence rate to consensus (often referred to herein as consensus speed), for a perturbation driven system, is defined in this section. The upper bound is defined for graphs irrespective of whether that bound can be achieved through an optimal perturbation. It is worth mentioning that throughout this dissertation the notation $(\cdot)_i$ is used to indicate an element of the vector (\cdot) whenever the vector's notation already includes a subscript or the vector is the result of an operation indicated in the brackets.

Theorem 3.1. *Let L be the Laplacian of a connected, directed graph with outdegree of k for all nodes (k -outdegree) and $C = \text{diag}([c_1, \dots, c_P])$ be the diagonal perturbation matrix consisting of non-negative entries, where $\sum_i c_i = 1$. Then, the limits for the smallest eigenvalue of the perturbed Laplacian system, $S = -L - C$, is*

$$-\max_i((v_{L1})_i) < \lambda_{S1} < 0 \quad (3.9)$$

where $(v_{L1})_i \forall i \in \mathcal{V}$ is an element of the FLE of L with $(v_{L1})_i \geq 0$.

Theorem 3.1 can be proved as follows:

Proof. The first left eigenvector of S , \mathbf{v}_{S1} , is defined as $\mathbf{v}_{S1}^\top S = \lambda_{S1} \mathbf{v}_{S1}^\top$ from which the dominant eigenvalue can be found, when $\sum_i (v_{S1})_i = 1$, to be

$$\lambda_{S1} = \sum_i (\lambda_{S1} \mathbf{v}_{S1}^\top)_i = \sum_i (\mathbf{v}_{S1}^\top S)_i. \quad (3.10)$$

Since L is a Laplacian matrix, $\sum_i (\mathbf{v}_{S1}^\top L)_i = 0$, the diagonal perturbation matrix, C , can be substituted into Eq. (3.10) as

$$\lambda_{S1} = \sum_i (\mathbf{v}_{S1}^\top S)_i = -\sum_i (\mathbf{v}_{S1}^\top C)_i = -\sum_i (v_{S1})_i c_i. \quad (3.11)$$

The maximum dominant eigenvalue can then be defined as

$$\max(\lambda_{S1}) = \max(-\sum_i (v_{S1})_i c_i) = -\max_i((v_{S1})_i \times 1). \quad (3.12)$$

Defining $\max(\lambda_{S1}) = -\max_i((v_{S1})_i)$ is rather trivial when considering the $\sum_i c_i = 1$ constraint, whereby $\max(\lambda_{S1}) = 1 \times -\max_i((v_{S1})_i)$. In the case where e elements of \mathbf{v}_{S1} are equal to $\max_i((v_{S1})_i)$, then $c_i = 1/e$ for those i corresponding to $\max_i((v_{S1})_i)$ and 0 elsewhere.

An equation that approximates the shift, $\delta\lambda_1$, in a distinct eigenvalue of a generic square matrix for small perturbations is defined by Deif (1995) as

$$\delta\lambda_{S1} \approx \frac{\mathbf{v}_{L1}^\top \delta S \mathbf{r}_{L1}}{\mathbf{v}_{L1}^\top \mathbf{r}_{L1}} = -\frac{\mathbf{v}_{L1}^\top C \mathbf{r}_{L1}}{\mathbf{v}_{L1}^\top \mathbf{r}_{L1}} = \frac{\mathbf{v}_{L1}^\top S \mathbf{r}_{L1}}{\mathbf{v}_{L1}^\top \mathbf{r}_{L1}}, \quad (3.13)$$

since $\mathbf{v}_{L1}^\top L = 0$ and where \mathbf{v}_{L1} and \mathbf{r}_{L1} are the left and right eigenvectors respectively

of L corresponding to the eigenvalue, λ_{L1} . Given that

$$\lambda_{S1} = \frac{\mathbf{v}_{S1}^T \mathbf{S} \mathbf{r}_S}{\mathbf{v}_{S1}^T \mathbf{r}_S} \quad (3.14)$$

and $\delta\lambda_{S1} = \lambda_{S1} - \lambda_{L1} = \lambda_{S1}$ it can be seen, with reference to Eq. (3.13), that $\mathbf{v}_{S1} \approx \mathbf{v}_{L1}$ and $\mathbf{r}_{S1} \approx \mathbf{r}_{L1}$ for small perturbations.

For larger perturbations, Eq. (3.13) is no longer valid with $\mathbf{v}_{S1} \neq \mathbf{v}_{L1}$ and $\mathbf{r}_{S1} \neq \mathbf{r}_{L1}$. Consider that

$$\mathbf{v}_{L1}^T L = \lambda_{L1} \mathbf{v}_{L1}^T = \mathbf{0}$$

and

$$(\mathbf{v}_{L1}^T \mathbf{S})_i = (\mathbf{v}_{L1}^T L)_i = 0 \quad (3.15)$$

for i where $c_i = 0$. Now consider that

$$(\mathbf{v}_{S1}^T \mathbf{S})_i = (\lambda_{S1} \mathbf{v}_{S1}^T)_i < 0 \quad (3.16)$$

as long as i is a globally reachable node.

The optimal perturbation, $c_i = 1/e$, was implicitly defined in Eq. (3.12) for e nodes whose eigenvector entry, $(v_{S1})_i$, equals $\max_i((v_{S1})_i)$. Comparing eqs. 3.15 and 3.16 it can be seen that $(v_{S1})_i > (v_{L1})_i$, where $c_i = 0$. Due to the $\sum_i (v_{L1})_i = \sum_i (v_{S1})_i = 1$ constraint, it follows that for this optimal perturbation scenario $(v_{S1})_i < (v_{L1})_i$ for i corresponding to $c_i = 1/e$. Hence, the limit for the magnitude of the smallest eigenvalue of the perturbed Laplacian is $\max_i((v_{L1})_i)$ and is approached when $(v_{S1})_i \rightarrow (v_{L1})_i$. \square

3.2.2 Leadership Selection Algorithms

The allocation of leadership in networks is a continuing field of research, where brute force searches can guarantee good results but become computationally challenging for large networks. Most work in this area has focused on the leader-follower scenario, usually by selecting a set number of nodes as leadership candidates. Ying et al. (2014) took such an approach by developing a consensus centrality metric with the aim of achieving fast consensus driven by a leader. Degree centrality can be taken as a rough guide to leadership selection with Ying et al. (2014) also highlighting its potential as a leader selection metric. The work presented in this dissertation is not directly comparable with works similar to Ying et al. (2014) that are concerned with selecting a set number of leaders, but it is more applicable to a variety of real-world problems

where influence can be targeted in a variable manner.

The work of Punzo et al. (2016) on resource allocation is directly comparable to the problem considered here, where they demonstrated that the FLE of the Laplacian matrix is a good strategy for allocating variable resources, in certain cases, for leading a network to consensus. However, Punzo et al. (2016) also noted that a numerical optimiser, using sequential quadratic programming methods with an active-set algorithm, see MathWorks (2015a), is able to consistently uncover better allocations. The main drawback of using such an optimiser is the expansion of the search space, since it is dependent on the network size. This results in $\mathcal{O}(N)$ operations that calculate the matrix eigenvalue, at a cost of $\mathcal{O}(N^3)$ according to Stewart (2000), producing a total run time of $\mathcal{O}(N^4)$. Two semi-analytical strategies shall be presented, for graphs composed of a single component, that have reduced computational run times but can produce similar results to fully numerical attempts. Firstly, the Power Optimisation approach - for when all the nodes in the graph can be led effectively by a few central nodes - and then the Communities of Influence method - for when the graph is connected but there exists groups of isolated nodes that require local leaders to achieve a fast network-wide response. The method is referred to as communities as it finds the community that respond to these local leaders that exert a strong local influence but not necessarily any significant global network influence.

3.2.3 Power Optimisation

The Power Optimisation (Power Opt) strategy is a newly developed semi-analytical approach that focuses resources on the most effective leaders by raising the FLE to some power, p , according to

$$\mathbf{c} = \frac{\mathbf{v}_{L1}^p}{\sum_i (v_{L1})_i^p} \quad (3.17)$$

where \mathbf{c} is the resource allocation vector, \mathbf{v}_{L1} is the first left eigenvector of the Laplacian matrix and \mathbf{v}_{L1}^p an element-wise operation. Eq. (3.17) maximises λ_{S1} by locally changing p in \mathbb{R} . When $p \rightarrow 0$ the resource allocation, c , approaches a uniform vector state. The Power Opt method iteratively reduces the resources to less influential nodes while increasing those assigned to the most prominent, as p is increased while maintaining $\sum_i c_i = 1$, until the convergence rate stops increasing. For a large value of p , $c_i > 0$ for any nodes that have an eigenvector entry $\mathbf{v}_{L1i} \approx \max_i((v_{L1})_i)$ and $c_i \approx 0$ for all others. An iterative search along the line can reveal a large enough p to achieve these values for c_i and maximise the convergence rate to consensus. For networks,

where a close approach to this upper limit is not possible, the search space appears to always be convex with a single maximum occurring between the low and high p values. An iterative line search is again employed to solve for these networks, which is then followed by the bisection method (see Kaw et al. (2008) for details) when increasing the p value further would result in a reduction of the convergence rate. The search space of the Power Opt method is dependent on the power, p , and does not grow with an increasing number of nodes, N . Hence, the eigenvalue calculation is the dominant process resulting in $\mathcal{O}(N^3)$ complexity for the Power Opt method. This is obviously an improvement on the $\mathcal{O}(N^4)$ resulting from the purely numerical process.

In Fig. 3.1 the 75 nodes of an Erdős-Rényi random network, where each node has an outdegree of 5, is displayed as a geometrical interpretation with each point's coordinates defined by the three most dominant left eigenvectors, associated with the three smallest eigenvalues in magnitude of the Laplacian. A similar network visualisation has been used for the networks in the KONECT database produced by Kunegis (2016), but these were not used for any analytical purpose. The leadership resource selection is defined by the numerical optimiser that applies a sequential quadratic programming method*. Comparing the node sequence selected by the numerical approach with the first left eigenvector (FLE) indicates that the FLE is an effective indicator of influence for this graph. The results of this comparison in Fig. 3.1 showed that for this network the numerical approach selected 14 nodes to be supplied with resources, these 14 were the 14 top ranked nodes as determined by the FLE. It was, however, not a perfect match with a couple of deviations in the order of the most influential nodes, with the 3rd & 4th nodes swapping places and the 10th, 12th & 13th changing order. The node ranking according to the FLE (\mathbf{v}_{L1}) is a close approximation of the ranking according to the numerical allocation for the graphs investigated. This similarity is the basis of the Power Optimisation's effectiveness as it just varies the resource allocation and not the node sequence. This shall be shown to not always be the case when the presence of multiple distinct communities creates multiple significant modes of system response, with \mathbf{v}_{L2} and \mathbf{v}_{L3} also becoming relevant to the optimal system response.

3.2.3.1 Results

The consensus speed limit, introduced in section 3.2.1, is not used for comparison in most of the following analyses as, in many cases, the upper speed limit is far in excess of what can actually be achieved. Only in certain topological conditions does

*implemented with the fmincon algorithm in MATLAB by MathWorks (2015a).

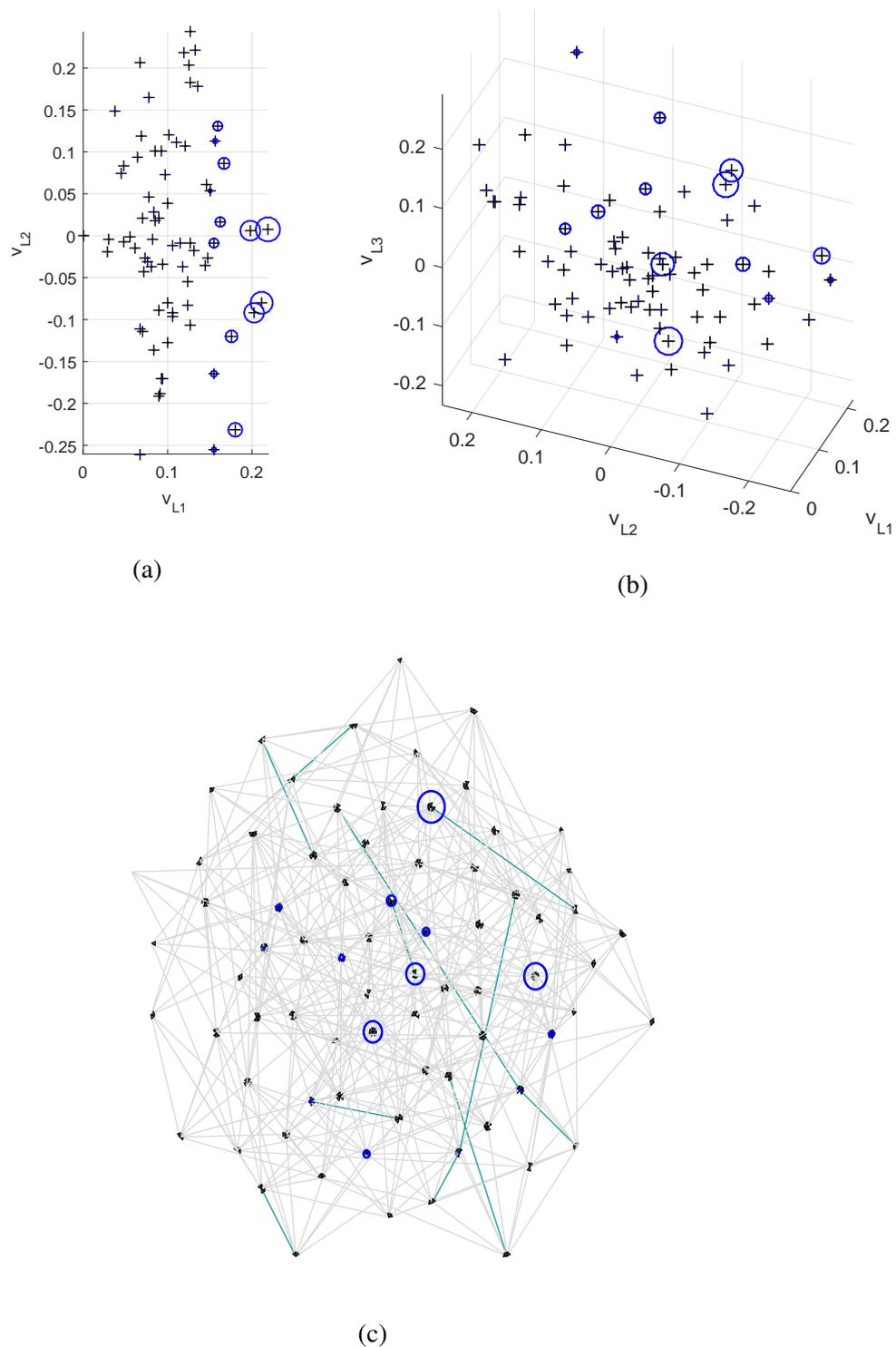


Figure 3.1: 75 node, 5 outdegree, random network where the numerically optimised leadership resource selection is represented with circled nodes (the diameter of each corresponds to the resources allocated); (a) \mathbf{v}_{L1} and \mathbf{v}_{L2} are the first and second left eigenvector of the network, (b) includes \mathbf{v}_{L3} and (c) represents the network where green lines mark bi-directional connections and grey are uni-directional.

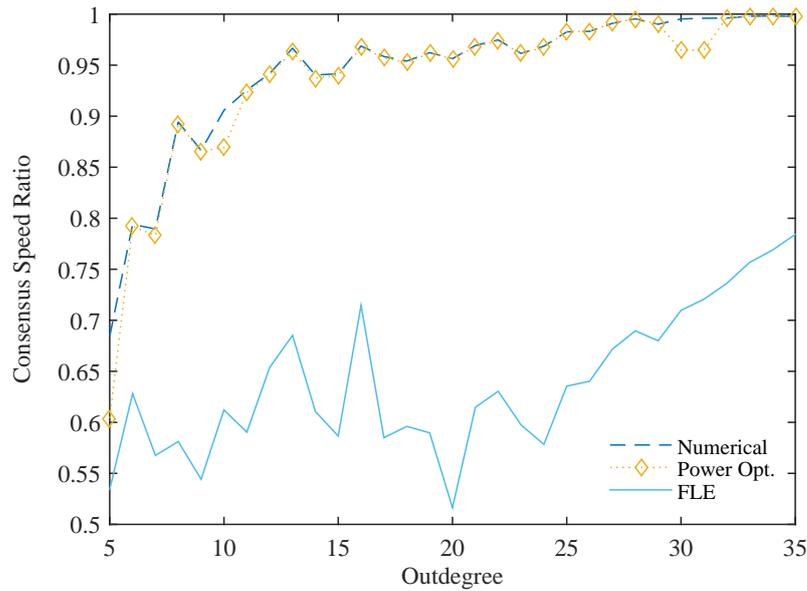


Figure 3.2: Consensus speed ratio with respect to $\max_i((v_{L1})_i)$ for various resource allocation vectors over a range of outdegrees in a 50 node k -NNR network.

an optimal benchmark match the result of the upper limit, see Fig. 3.2. Hence the validity of Power Opt method was supported through comparisons with a numerical optimiser, described in Section 3.2.2, that is used in this dissertation to represent the optimal benchmark. In reality, the optimiser cannot guarantee optimality as it may find a locally optimal solution but given the size of the networks used it should be similar to the globally optimal result.

As stated in Theorem 3.1, the unobtainable limit for the smallest eigenvalue in magnitude of the perturbed, negated, Laplacian is equal to the largest element of the FLE, $\max_i((v_{L1})_i)$. The results in Fig. 3.2 approach this consensus speed limit at high outdegrees for a 50 node network, where nodes have been randomly distributed in a plane before applying k -Nearest Neighbour (k -NNR) connection rules for a range of outdegrees. The consensus speeds for the numerical and Power Opt approaches are seen to converge with, but never exceed, the $\max_i((v_{L1})_i)$, achieving better results than the unmodified FLE that was used as a *good* leadership allocation by Punzo et al. (2016).

Fig. 3.2 demonstrates that the Power Opt method can be effective. In this case it appears to match the numerical result for most of the outdegrees tested and as the outdegree increases the convergence speed achieved approaches the theoretical maximum. The Power Opt is, therefore, effective for this network but the main benefit lies in its efficiency, which is significantly better than the numerical optimiser. Table 3.1 details

the worst case run times (asymptotic complexity) alongside trend lines for actual run time - assessed from runs with a 1000 node, $k = 10$, k -NNR network - with the coefficient of determination, R^2 , displayed to show the accuracy of the trend line fit. This shows the efficiency of Power Opt with the trend for the numerical optimiser well over an order of magnitude higher than it is for the Power Opt.

Table 3.1: Algorithm run time comparison for n nodes.

Algorithm	Asymptotic Complexity	Actual Trend [s]	R^2
Numerical	$\mathcal{O}(n^4)$	$5 \times 10^{-7}n^{3.51}$	0.998
Power Opt	$\mathcal{O}(n^3)$	$1 \times 10^{-5}n^{2.06}$	0.982

The Power Opt method does not always match the numerical optimiser in Fig. 3.2 (see outdegree 5, 10, 30 and 31). The reason for this sub-optimal performance is captured in the following section. Networks are revealed to sometimes contain multiple influential communities and the Power Opt method is seen to perform best when only a single prominent community is present.

3.2.4 Communities of Influence

Well-connected networks, such as graphs where all nodes have a high outdegree, are unlikely to have groups of nodes that are significantly isolated from the network as a whole. For lower outdegree graphs, where due to the k -NNR topology edges connect primarily to local proximity nodes, groups can form that have a high local clustering but with few paths connecting these clustered nodes to the rest of the graph. This emergence of local hubs is especially common with k -Nearest Neighbour (k -NNR) topologies, where nodes connect to a set number of other nodes that are nearest to them (unlike distance based connection metrics there is no threshold on the maximum distance of a connection just a set number of outgoing connections). High and low outdegree, in the context of this work and specifically for k -NNR graphs, is defined based on the community structure that results. By displaying the network, in Fig 3.3, as a geometrical interpretation, with each point represented by the second and third most dominant eigenvectors, the difference between low outdegree, where multiple communities form, and high outdegree, where there is no longer a clear community structure, can be seen. The FLE is not used in this depiction as all elements are positive and,

therefore, the community structure is less clear than using the following eigenvectors that include positive and negative values. For the cases shown in Fig 3.3, the boundary between high and low outdegree is not clear but $k \geq 14$ would be referred to as high outdegree with a more homogenous distribution of nodes, while for $k = 5$ there is a clear structure with lines of nodes that are indicative of multiple communities. This structure can also be seen to a lesser extent in $k = 8$ and 11; why this indicates the presence of distinct communities is the subject of further discussion in Section 3.4. Before that, it is worth considering established community detection metrics.

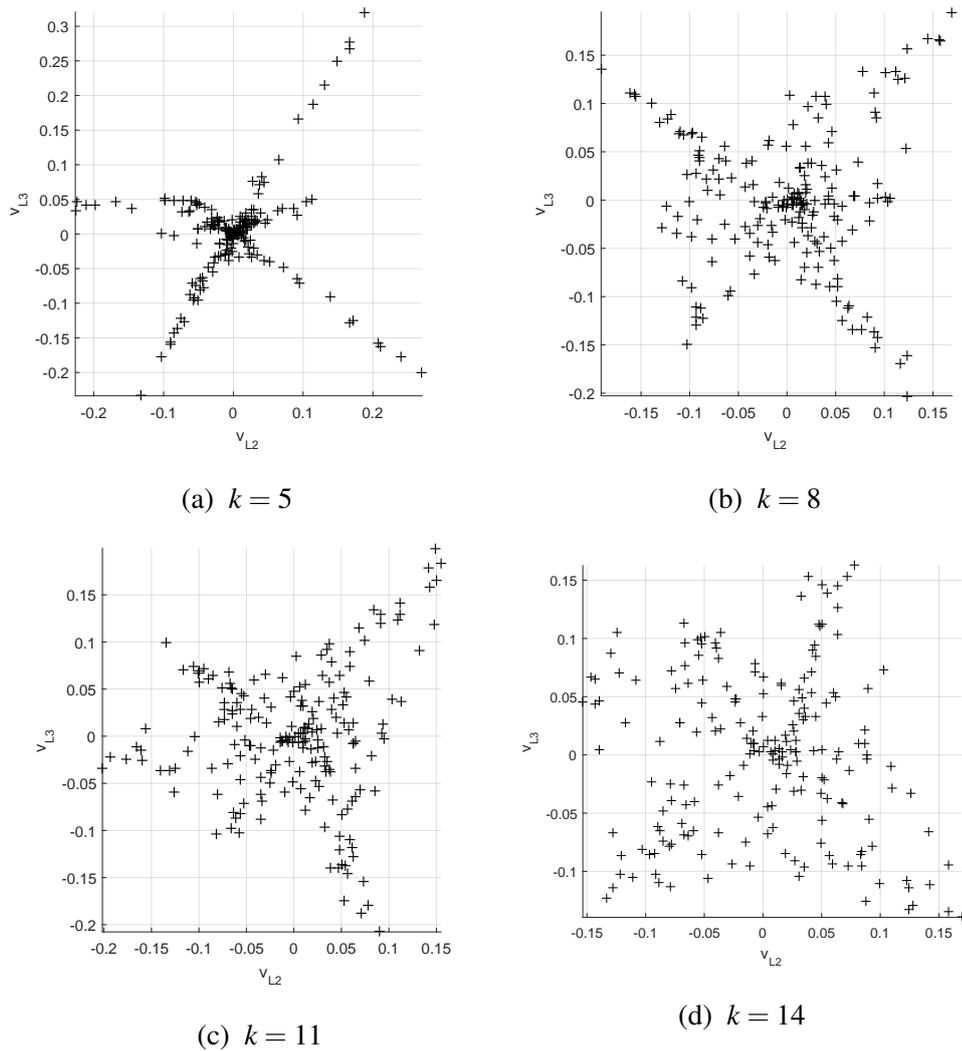


Figure 3.3: Geometrical interpretation of a 200 node, k -NNR, network with outdegree (k). v_{L2} and v_{L3} represent the 2nd and 3rd most dominant left eigenvectors.

Leicht and Newman (2008) developed a community detection algorithm that, for directed networks, maximises a modularity function, which takes into account edge di-

rection, for any number of possible divisions in a network (discussed later). The communities detected with the Leicht-Newman method shall be referred from hereon - as they were by Leicht and Newman, and for the sake of clarity - as modules. For high outdegree k -NNR networks, fewer non-overlapping modules are found (e.g. 2 modules for a 100 node k -NNR network with an outdegree of 50) than in lower outdegree scenarios where many modules are present (e.g. 10 modules for 100 node network with an outdegree of 5), as is displayed in Fig. 3.4.

Modularity

Modularity is defined as the fraction of a graph's edges that lie within a given group of nodes minus the expected fraction if the edges were distributed randomly. This can be defined for both undirected and directed graphs, where the directed case takes into account the edge direction. A positive value for modularity indicates the presence of a module of nodes where the number of edges exceeds that which would be expected from a random distribution. A negative modularity value is also possible and would indicate fewer edges than expected. The modularity is calculated by randomising the edges while maintaining the same outdegree for each node and then using this graph as the expected graph with which the actual one can be compared to produce a modularity value.

The simplest way to calculate modularity is for a network divided into two groups by a vector \mathbf{s} , where $s_i = 1$ if it belongs to one group and $s_i = -1$ if it belongs to the other. The expected number of edges, for a random graph, is captured by $k_i^{\text{in}}k_j^{\text{out}}/m$ where k^{in} and k^{out} denote the indegree and outdegree for the labelled vertices, either i or j in this

example. The modularity can then be calculated as

$$\begin{aligned}
 Q &= \frac{1}{2m_e} \sum_{ij} \left[A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m_e} \right] (s_i s_j + 1) \\
 &= \frac{1}{2m_e} \sum_{ij} s_i B_{ij} s_j \\
 Q &= \frac{1}{2m_e} \mathbf{s}^\top B \mathbf{s} \tag{3.18}
 \end{aligned}$$

where s_i is an element of the vector \mathbf{s} , m_e is the number of edges in the network and $B = A - \frac{(k^{\text{out}} k^{\text{in}})^\top}{m_e}$, which performs the comparison between the actual network and a random graph with the same indegree and outdegree associated with each node.

To split the network into modules, eq. (3.18) is updated by replacing B with the symmetrical $B + B^\top$ that has useful properties, such as eigenvectors that form an orthonormal basis, which allows for $\mathbf{s} = \sum_i a_i \mathbf{v}_{B_i}$ with $a_i = \mathbf{v}_{B_i}^\top \cdot \mathbf{s}$. Then

$$\begin{aligned}
 Q &= \frac{1}{2m_e} \sum_i a_i \mathbf{v}_{B_i}^\top B \sum_j a_j \mathbf{v}_{B_j} \\
 Q &= \frac{1}{2m_e} \sum_i \lambda_i (\mathbf{v}_{B_i}^\top \cdot \mathbf{s})^2, \tag{3.19}
 \end{aligned}$$

where λ_i is the eigenvalue of $B + B^\top$ corresponding to eigenvector \mathbf{v}_{B_i} . Under a normalisation constraint, for \mathbf{s} , the maximum of Q is achieved when \mathbf{s} is parallel to the leading eigenvector \mathbf{v}_{B_1} . The module detection works by maximising $\mathbf{v}_{B_1} \cdot \mathbf{s}$ that, given the constraints acting on \mathbf{s} , results in $(\mathbf{v}_{B_1})_i > 0$ when $s_i = 1$ and $(\mathbf{v}_{B_1})_i < 0$ when $s_i = -1$. This process works for splitting a network into two groups that maximise Q . But scaling this process to multiple modules requires a slight modification to the equations presented. Essentially a module is partitioned and then each newly created module is partitioned, if this partitioning results in an increase in Q , with this process only halting when no further partitioning of any module will result in an increased Q . This process is described in detail by Leicht and Newman (2008).

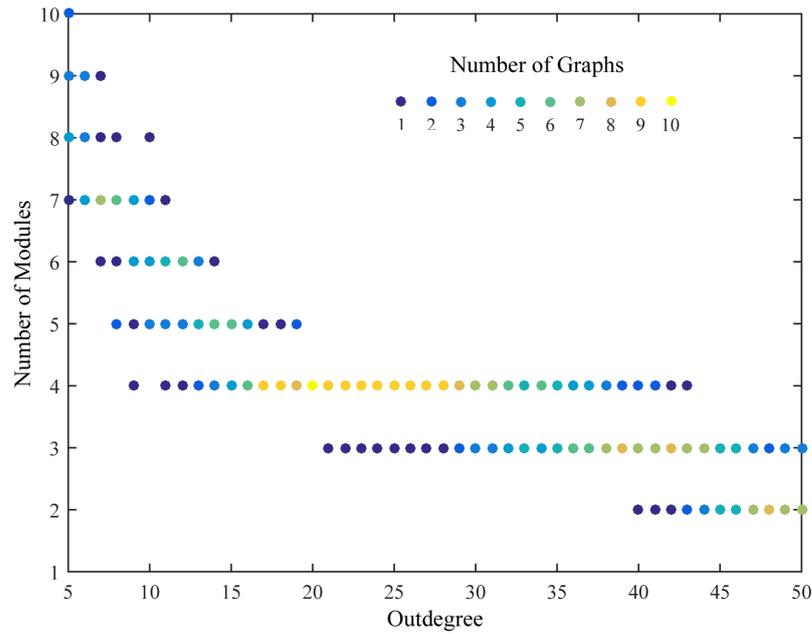


Figure 3.4: Number of Leicht-Newman modules for a 100 node k -NNR network with a given outdegree. Circle colour dependent on the number of graphs with the same outdegree and number of modules present, with 10 graphs generated for each outdegree.

A geometrical interpretation of the network, using the first three left eigenvectors, is used to reveal how to optimally perturb a network to achieve the fastest convergence rate to consensus in Fig. 3.5. This is a good example of distinct communities in a 75 node, 5 outdegree, k -NNR network where the modularity for this graph is high at 0.691. This analysis reveals a clear structure, with Fig. 3.5 (a) and (b) showing three distinct lines of nodes emanating from the origin. This structure is not present in the 75 node, 5 outdegree, random graph, from Fig. 3.1 (a) and (b), with this lack of community/module structure explained by a significant decrease in modularity to 0.2544. The first left eigenvector, \mathbf{v}_{L1} , continues to be an important marker for effective leadership. But, in contrast with the random case, the sequence of nodes according to the numerically optimised resource distribution - represented by the size of the blue circles centred on selected nodes - is very different to the sequence according to \mathbf{v}_{L1} .

Fig. 3.6 is used to demonstrate that multiple distinct communities will disappear in k -NNR topologies, as well, when the outdegree increases. This is supported by the findings in Fig. 3.4 where the number of Leicht-Newman communities decrease with increasing outdegree. The modularity is also lower than the $k = 5$ random case (Fig. 3.1) with 0.219 recorded.

The nodes, with the largest numerically optimised resource assignment in Fig. 3.5,

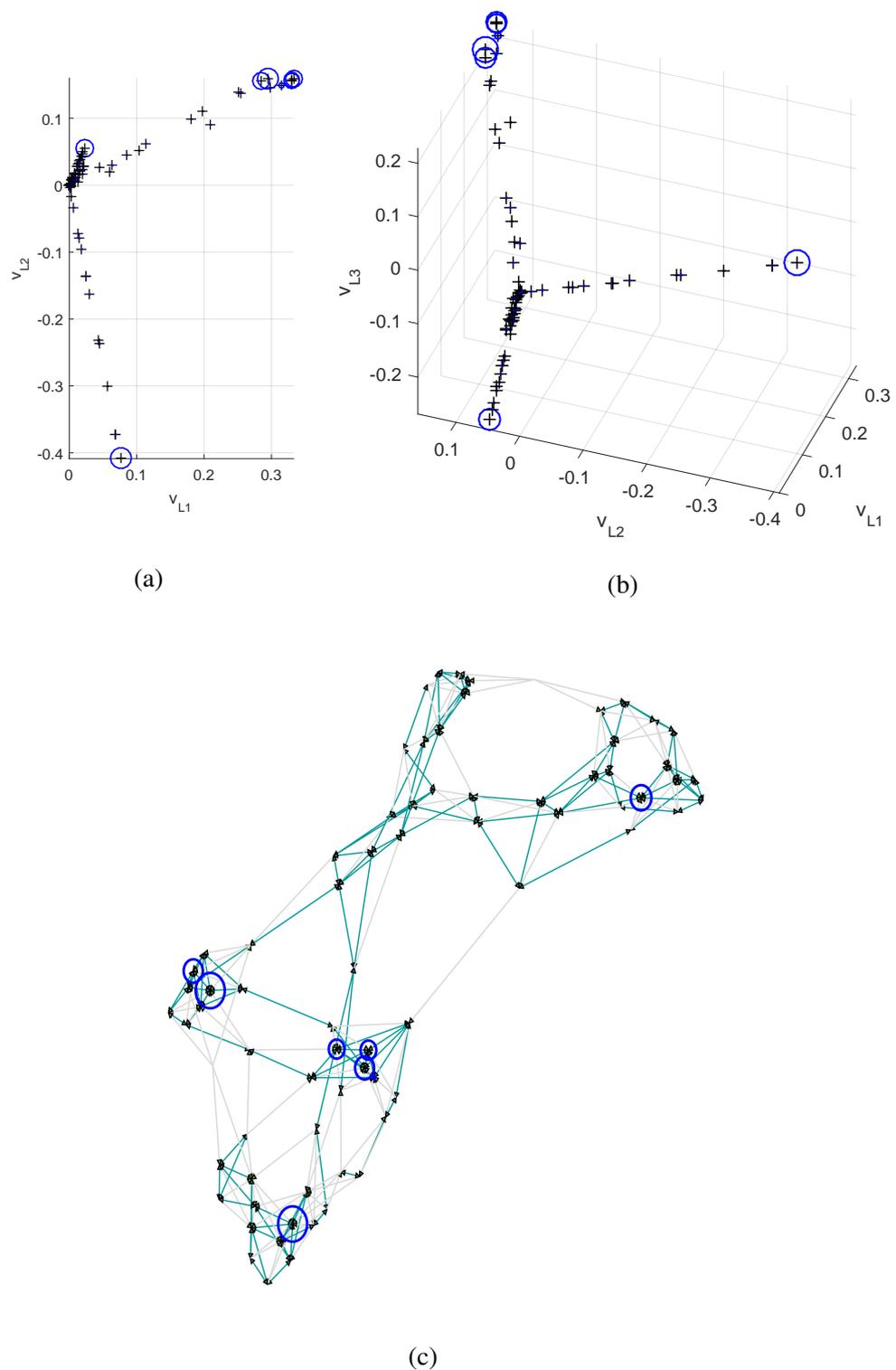


Figure 3.5: 75 node, 5 outdegree, k -NNR network where the numerically optimised leadership resource selection is represented with circled nodes; (a) v_{L1} and v_{L2} are the first and second left eigenvector of the Laplacian matrix, (b) includes v_{L3} and (c) represents the network where green lines mark bi-directional connections and grey are uni-directional.

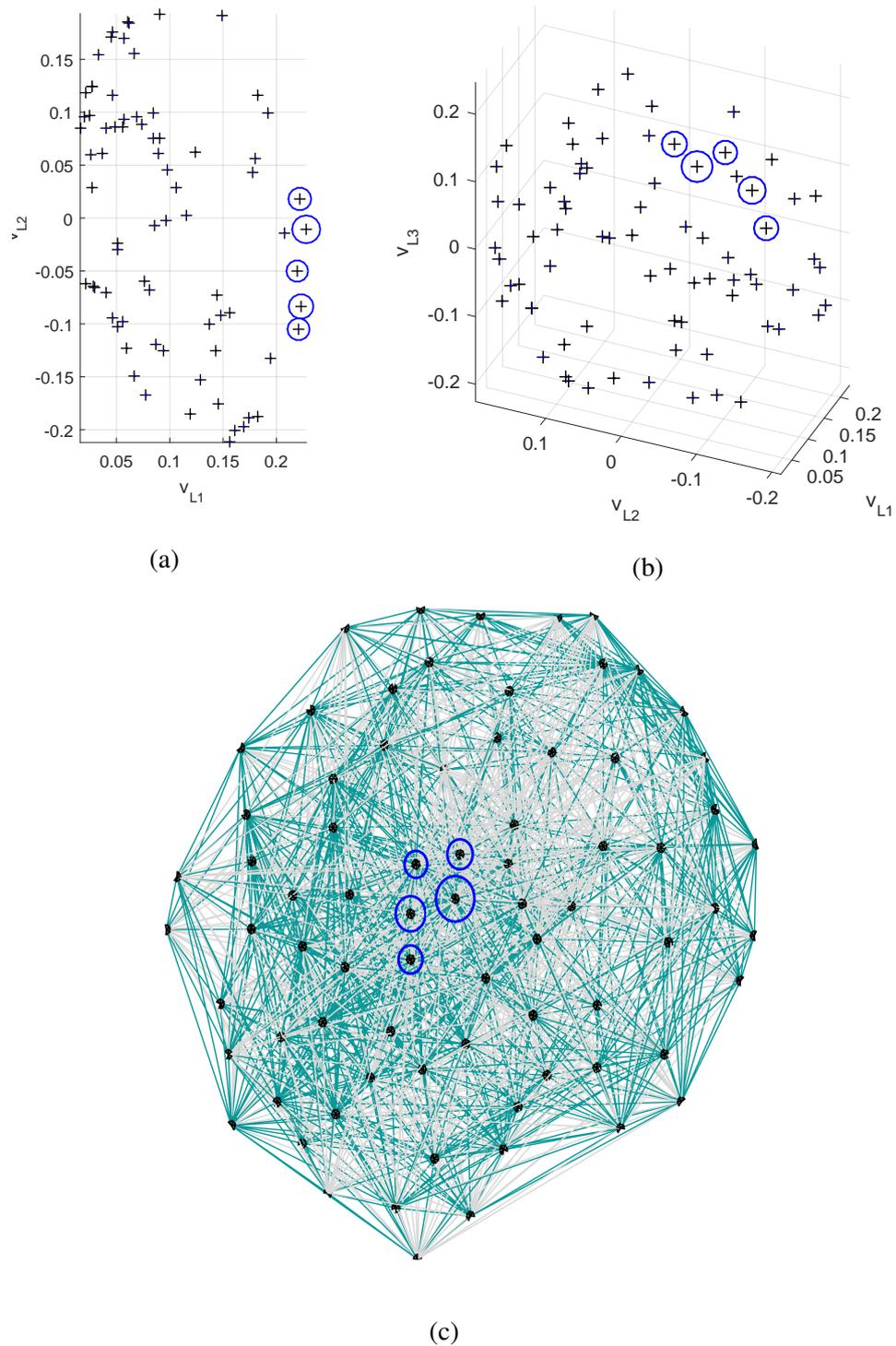


Figure 3.6: 75 node, 30 outdegree, k -NNR network where the numerically optimised leadership resource selection is represented with circled nodes; (a) \mathbf{v}_{L1} and \mathbf{v}_{L2} are the first and second left eigenvector of the Laplacian matrix, (b) includes \mathbf{v}_{L3} and (c) represents the network where green lines mark bi-directional connections and grey are uni-directional.

are located at the end of each prominent line of nodes that leads back to the plot's origin (i.e. those furthest from the origin). If the Power Opt approach was applied to this network it would not be able to match the numerically optimised result seen in the Fig. 3.5. The Power Opt approach would appear to be more effective if applied only to the nodes that belong to each line separately with the resulting allocations then combined. To this end the Communities of Influence (CoI) method was developed, as an extension of the Power Opt approach, for when the modularity decreases and the Power Opt method is no longer effective.

The CoI algorithm will now be described with the first steps focused on defining CoI vectors. It begins with the calculation of the FLE of the adjacency matrix (A) - not the Laplacian as will be discussed later - that is set as the first CoI vector. The adjacency matrix is then modified by removing all connections from the most prominent node as indicated by the FLE. A new FLE of A is found for this modified matrix, this is now the second CoI vector, with the adjacency again modified and the process repeated. By performing this node removal, the Power Opt approach can be essentially repeated for the separate lines in the eigenvector plot (as discussed in reference to Fig. 3.5) before combining the resource allocations.

The algorithm is presented in more detail in Algorithm 3.1, with the Power Opt approach from eq. (3.17) used to attribute resources for each CoI vector and a numerical optimiser, employing a sequential quadratic programming method*, used to decide the relative importance of the power optimised CoI vectors.

In Fig. 3.7 the effect of removing all connections to and from the most prominent node of the top four CoI vectors, as described in Algorithm 3.1, is displayed using the geometrical representation of a 75 node k -NNR network where $k = 5$ (as previously displayed in Fig. 3.5). It can be seen that this process reduces the prominent node's (those numbered and circled) first left eigenvector (v_{A1}) entry. Most of the nodes that are lined up behind a prominent node in Fig. 3.7 (a) also have their v_{A1} entry reduced after the connections are removed from their prominent node. This is evident from the loss of a line of nodes in plots (b) and then (c).

In Algorithm 3.1, the number of communities required to find a near-optimal leadership varies depending on the topology in question. Five communities of influence were deemed to be a sufficient upper limit for all the networks examined in this chapter, but for large sparse networks more communities may well be necessary. In Section 3.4 a

*implemented with the `fminunc` algorithm in MATLAB MathWorks (2015b)

Algorithm 3.1 CoI**procedure** FINITE LEADERSHIP OPTIMISATION

Calculate the FLE, $\mathbf{v}_{A1} = [v_1, v_2, \dots, v_N]$, for the adjacency matrix, $A = (a_{ij}) \in \mathbb{R}^{N \times N}$.

for $m = 1$ to $n \in \mathbb{R}^+$ **do**

Define the resource vector, $\mathbf{c}_m = \frac{(\mathbf{v}_{A1})^p}{\sum_i (\mathbf{v}_{A1})^p} \forall i \in \mathcal{V}$,

where $p = p_m \in \mathbb{R}^+$.

$\forall i \in \mathcal{V}, j \in \arg \max_k v_k, a_{ij} = a_{ji} = 0$.

Calculate \mathbf{v}_{A1} for the updated adjacency matrix.

end for

Numerical optimiser maximises $|\lambda_{S1}|$ where

$C = \text{diag}(\mathbf{c}_{CoI^n})$ with $r_m \in \mathbb{R}^+$ a weighting variable for the different resource vectors.

$\mathbf{c}_{CoI^n} = f(\mathbf{c}_1, \dots, \mathbf{c}_m, p_1, \dots, p_m, r_1, \dots, r_{m-1})$ with the function $f(\dots)$ defined in Eq. (3.20).

end procedure

method is presented for evaluating how many distinct lines are present in the eigenvector plot, but this work did not go as far as to implement such analysis into the CoI optimisation. This could be incorporated by using it to set the upper bound for the number of communities (i.e. CoI vectors) to be considered in the CoI analysis. The number of CoI vectors used is important as it affects number of variables in the optimisation and, as a consequence, the computational time. This can be seen for n CoI vectors, where the final resource allocation is produced by combining these vectors, as

$$\mathbf{c}_{CoI^n} = \frac{\mathbf{c}_1 + \sum_{i=1}^{n-1} \frac{\mathbf{c}_{i+1}}{r_i}}{1 + \sum_{i=1}^{n-1} \frac{1}{r_i}} \quad (3.20)$$

where the denominator, with weighting variables $\mathbf{r} = \{r_1, \dots, r_{n-1}\}$ that are initially set to 1, scaling the combined vectors to ensure $\sum_i (\mathbf{c}_{CoI^n})_i = 1$. The initial guesses for the powers (p_1, \dots, p_n) were based on analysis of densely connected k -NNR networks. Ten networks were analysed between 50 and 150 nodes in size at each node interval (of size ten) with an outdegree of 30, the results using the Power Opt method are shown in Fig. 3.8. This analysis uncovered that the optimiser selected a high power, p , between 45 and 220 for the best leadership resource distribution. A high power usually translates to fewer nodes being provided with leadership resources. The CoI essentially combines multiple Power Opt vectors for the multiple communities of influence in the network, therefore the initial guesses for the powers (p_1, \dots, p_n) were set at 50 to help avoid local minima and reduce computational effort. The presence of more variables in

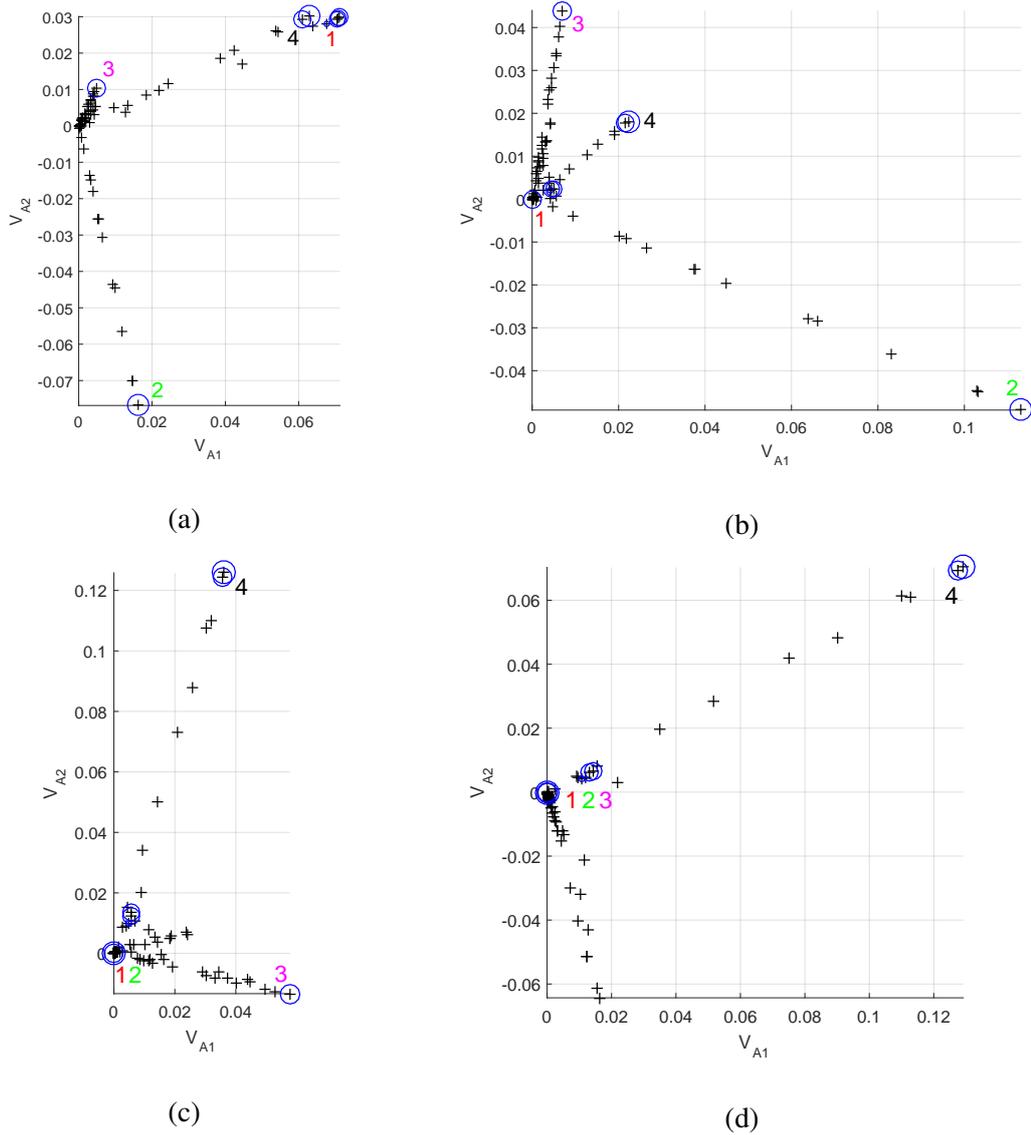


Figure 3.7: 75 node, 5 outdegree, k -NNR network where the numerically optimised leadership resource selection is marked with circles (size corresponds to resource allocation) and the top four CoI communities are numbered; (a) v_{A1} and v_{A2} are the 1st and 2nd left eigenvector of the adjacency matrix, in the other plots the axes are for the adjacency after the removal of the most prominent node (b) from the 1st CoI comm., (c) from the 1st and 2nd CoI comm., (d) from the 1st, 2nd and 3rd CoI comm.

the optimisation increases the search space and the algorithm run time when compared with the Power Opt method, however the run time remains defined by the eigenvalue calculation. The worst case for CoI is therefore also $\mathcal{O}(n^3)$, but the difference in actual computational time is explored in the next section.

For constant outdegree networks the FLE of the adjacency and Laplacian matrix are the same but their eigenvalues differ. The dominant eigenvalue for the adjacency matrix,

and therefore also the Laplacian matrix, is real and, for a strongly connected graph, strictly dominant with an algebraic multiplicity of 1, see Dhal et al. (2014). The FLE corresponds to this dominant eigenvalue that for the adjacency matrix is the largest eigenvalue, while for the Laplacian it is the smallest eigenvalue in magnitude. For the CoI algorithm; the adjacency matrix is used as the graph has to be modified, with node removals, into a variable outdegree network and then analysed again by calculating the FLE. To understand why the adjacency is used, consider the following simplification; in comparing these matrices the FLE of the adjacency and Laplacian can be represented as the *Indegree* and the ratio of *Indegree* to *Outdegree* (I/O) respectively. Therefore, considering a node with both an in and out connection to a removed node, i.e. a node that is a member of the most influential community, the relationship will change to $Indegree-1$ and $((I-1)/(O-1))$. This suggests that for the adjacency matrix the influence metric (value of FLE element v_i) will decrease. Whereas for the Laplacian the influence will increase since $(I/O) < ((I-1)/(O-1))$, if $Indegree > Outdegree$ as could be expected when the node in question is part of an influential community. The adjacency matrix is, therefore, more effective in detecting the less prominent communities, with the FLE of the Laplacian susceptible to only detecting nodes that were part of the original, most prominent, community. It is also possible for nodes that are part of the influential community to only have an outgoing connection to the previously most

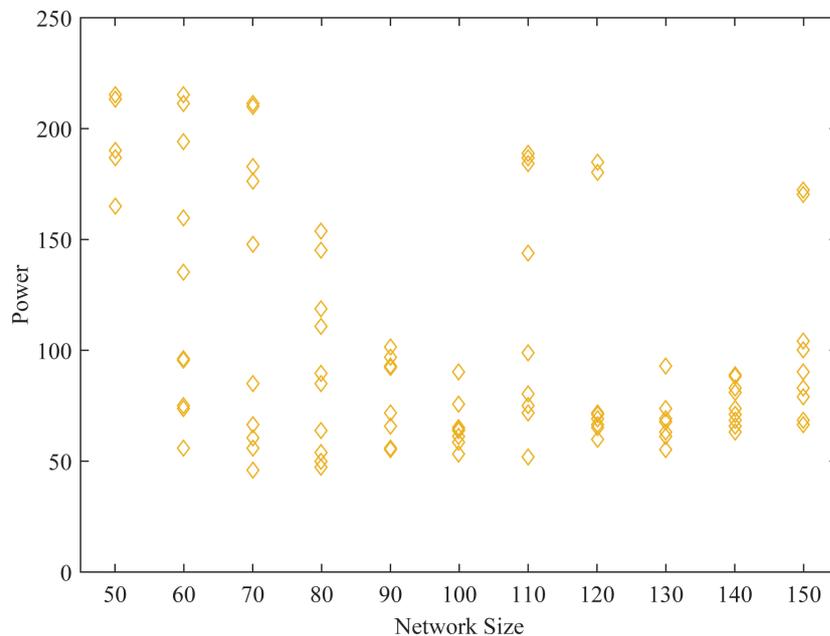


Figure 3.8: Power, p , required for Power Opt to match the consensus speed found by numerical optimisation.

influential but now removed node, which would result in no change for the adjacency case with no change in Indegree but a larger increase in the ratio for the Laplacian case, with the ratio becoming $(I/(O - 1))$.

3.2.4.1 Results

The validity of the algorithms developed was supported through comparisons with the numerical optimiser, described in Section 3.2.2, that is used in this dissertation to represent the optimal benchmark. In reality, the optimiser cannot guarantee optimality as it may find a locally optimal solution but given the size of the networks used it should be similar to the globally optimal result. This claim is supported by the results shown here, where the numerical optimiser can be seen to find more sub-optimal solutions as the network size grows (see Fig. 3.13). The methods developed herein are shown to be effective, through comparison with the optimal benchmark, for a number of k -outdegree topologies, including k -NNR with small-world rewiring, random and scale-free graphs. The consensus speed limit, introduced in section 3.2.1, is not used for comparison in most of the following analyses as in many cases the upper speed limit is far in excess of what can actually be achieved. Only in certain topological conditions does the optimal benchmark match the result of the upper limit.

In Fig. 3.9 (a), a toy example of a 75 node network depicts the power optimised CoI vectors (see Eq. (3.17)) as defined by the CoI optimisation method. This is the same network as is displayed in Fig. 3.7. Five CoI vectors are displayed in (a) with CoI 2, 3 and 5 only having resources allocated to their most prominent node. To complete the CoI optimisation a gain is applied to each of the CoI vectors. In Fig. 3.9 (b) the CoI and numerically optimised resource allocations are presented together. It can be seen from this figure that the vectors, CoI 5, is not allocated with any of the CoI resource allocation, which indicates that the gains selected removed resources from these vectors with a combination of the other four power optimised vectors providing the final allocation. The CoI and numerical optimisation approaches resulted in very similar allocations, which is evident when comparing the orange and blue circles marking their resource selections. The consensus speed is also compared with the values indicated in table 3.2, where it is clear that using fewer than four CoI vectors in this example yields a suboptimal response.

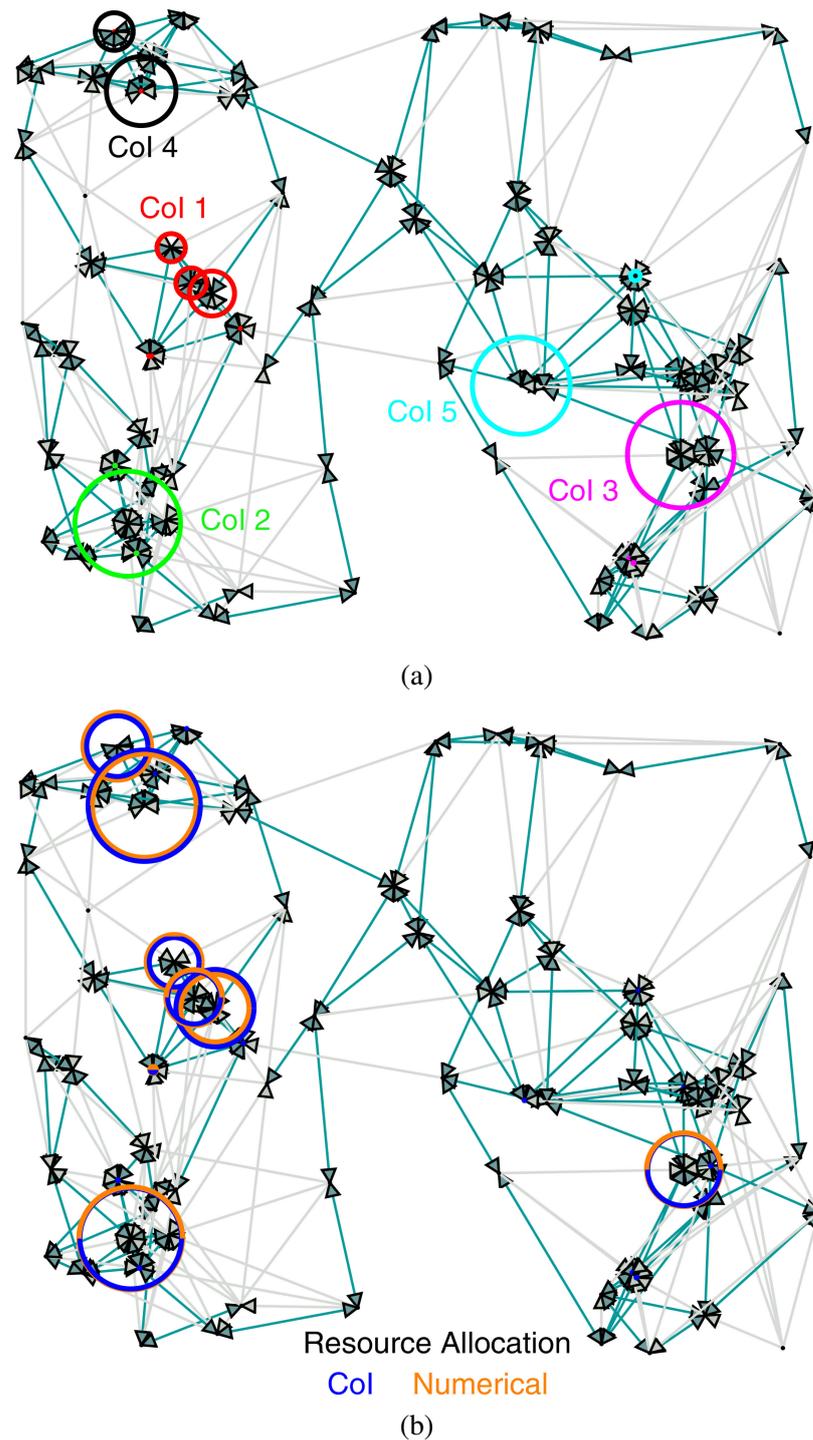


Figure 3.9: 75 node, 5 outdegree, k -NNR network with resource allocations highlighted: (a) the five, power optimised, CoI vectors are displayed; (b) the final optimised resource allocations are shown. Two-way connections are depicted in green with one-way in grey. Coloured circles overlay the network with the circle radius proportional to the resource allocation for the node at their centre.

Table 3.2: Convergence speed to consensus achieved when using a different number of CoI vectors for the 75 node, 5 outdegree, k -NNR network shown in Fig. 3.9.

No. of CoI Vectors	Convergence Speed (λ)
1	0.320
2	0.356
3	0.389
4	0.405
5	0.405

k-NNR

In Fig. 3.10 the Power Opt and Communities of Influence (using five communities) were compared with the numerical optimiser, defined previously, for k -NNR networks with randomly distributed nodes. There were ten networks at each interval between 100 and 900. The Consensus Speed Ratio is defined in reference to the numerical approach where a ratio value greater than 1 indicates a faster consensus speed than the numerical result. The CoI results appear to improve with respect to the numerical as the networks grow larger. The worst case run times (asymptotic complexity) are detailed

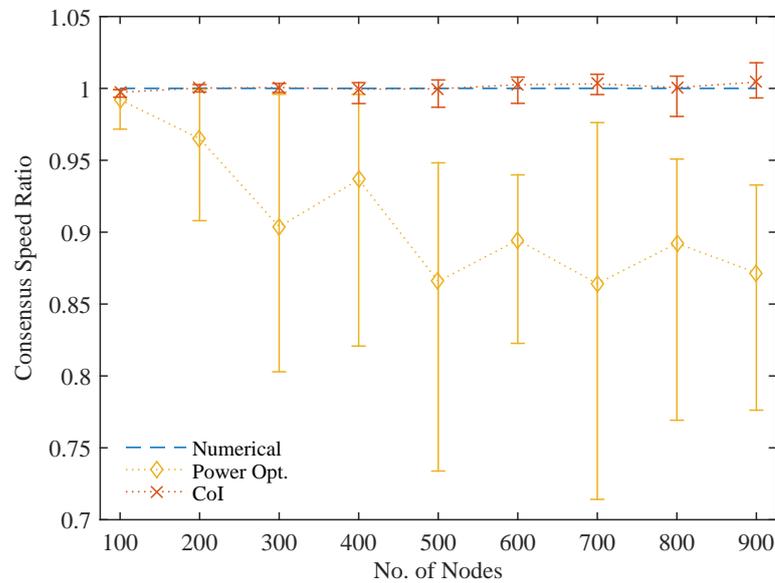


Figure 3.10: Consensus Speed Ratio for k -NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.

in Table 3.3 for comparison with the trend lines from the data presented in Fig. 3.10. These trend lines translate to the computational time, required for a 1000 node k -NNR network, being two orders of magnitude higher for the numerical method when compared with the five community CoI analysis. The coefficient of determination, r^2 , is displayed in the table to show the accuracy of the trend line fit.

Table 3.3: Algorithm run time comparison for n nodes.

Algorithm	Asymptotic Complexity	Actual Trend [s]	r^2
Numerical	$\mathcal{O}(n^4)$	$5 \times 10^{-7}n^{3.51}$	0.998
CoI	$\mathcal{O}(n^3)$	$2 \times 10^{-4}n^{2.10}$	0.993
Power Opt	$\mathcal{O}(n^3)$	$1 \times 10^{-5}n^{2.06}$	0.982

k-NNR with Small-world

The Power Opt and Communities of Influence (using five communities) were compared with the numerical optimiser, defined previously, for k -NNR networks with randomly created small-world connections where $k = 5$. These are presented in Fig. 3.11 where the probability of links being rewired and becoming small-world connections is displayed on the x-axis. From this figure, it can be noted that only the Power Optimisation's performance changes as the rewiring probability increases.

For small-world graphs, it has been noted that the mean path length will decrease when links are randomly rewired, see Newman (2000). The mean path length is defined as the average number of steps when considering the shortest path between all possible combinations of two nodes in the network. This phenomena is also displayed in Fig. 3.12 where the same models used in Fig. 3.11 are compared. Fig. 3.12 shows that as the percentage of randomly rewired links increases the Power Opt results converge with those of the five community CoI approach. The Consensus Speed Ratio, in Fig. 3.12, is defined in reference to the five community CoI method where a ratio value less than 1 indicates a slower consensus speed than the CoI result. The Power Opt approach would be the same as a one community CoI analysis, therefore it can be seen that as the rewiring probability increases the number of influential communities decreases until only one remains.

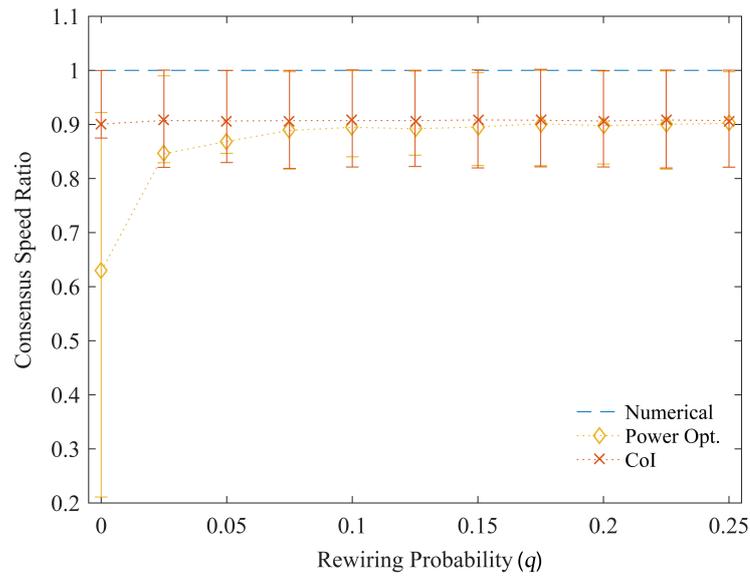


Figure 3.11: Consensus Speed Ratio values for a 100 node k -NNR network, where $k = 5$, with randomly rewired links. The probability, q , of a link being randomly rewired is varied marked on the x-axis. The error bars mark the maximum and minimum deviation from the mean.

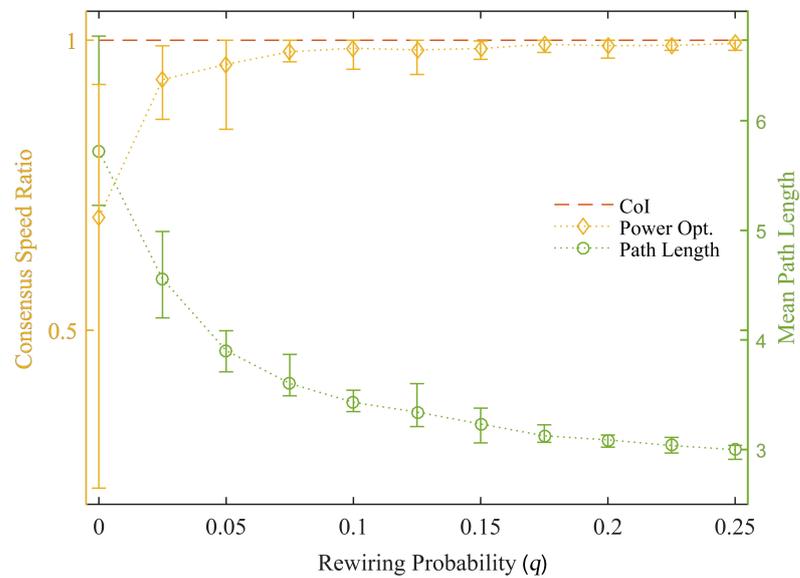


Figure 3.12: 100 node k -NNR networks, where $k = 5$, with randomly rewired links. The probability, q , of a link being randomly rewired is varied marked on the x-axis. The error bars mark the maximum and minimum deviation from the mean. Consensus Speed Ratio values compare CoI and Power Opt approaches with the mean path length detailed on the other y-axis.

k-Outdegree Random

Fig. 3.13 displays the Power Opt and CoI methods compared with the numerical optimiser, for ten Erdős-Rényi random networks at each 100 node interval, with the outdegree set at 10 for all nodes. The Consensus Speed Ratio is again defined in reference to the numerical approach where a ratio value greater than 1 indicates a faster consensus speed than the numerical result. The CoI and Power Opt approaches improve, in comparison with the numerical optimiser, as the networks grow in size with the CoI always outperforming the numerical for networks that are 500 nodes or larger. While the Power Opt is similarly superior for networks of 800 nodes or greater. The difference between the CoI and Power Opt remains fairly constant as the network size grows. This is possibly an indicator that it is the numerical optimiser that underperforms as the network increases in size. By their very definition, random graphs have minimal community structure and, as a consequence, are likely to create many sub-optimal local minima that could prevent a numerical optimiser from succeeding. The larger graphs amplify this problem by having a larger solution space to traverse while searching for the optimal result.

k-Outdegree Scale-free

Networks were generated that have scale-free properties but also conform to the k -outdegree constraints, i.e. the indegree distribution follows a power law where a small

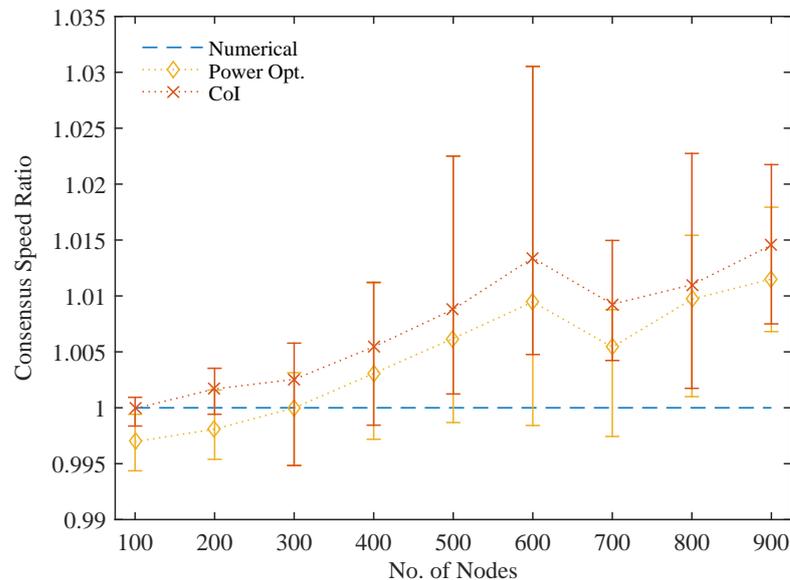


Figure 3.13: Consensus Speed Ratio for random networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.

number of nodes have the majority of the inward connections while most nodes have few inward connections and the outdegree of all nodes is kept constant. This was achieved using an adapted version of the algorithm presented in Prettejohn et al. (2011) and analysed for $k = 10$.

For the networks analysed, the Power Opt, CoI and numerical approaches achieve similar results, as depicted in Fig. 3.14. The semi-analytical methods are consistently, but only slightly, outperformed by the numerical approach. This is notable as it differs from the trends seen in Fig. 3.10 and 3.13 where the CoI method performed better, with respect to the numerical, as the network size increased. These differences highlight the respective strengths of the different approaches. The allocation of the perturbation is more constrained in the CoI and Power Opt algorithms with vectors scaled or combined to produce the final allocation, as opposed to the numerical method that alters each value to find an optimal allocation. However, the CoI approach, in particular, appears to be more effective at uncovering the most effective leader nodes when the network becomes larger. Therefore, it could be anticipated that for scale-free networks, where there are only a few well-connected nodes that would be effective leaders, that the numerical approach would perform better. In contrast, the k -NNR and random cases produce graphs where the leaders are more difficult to identify and, therefore, the semi-analytical methods can outperform the numerical.

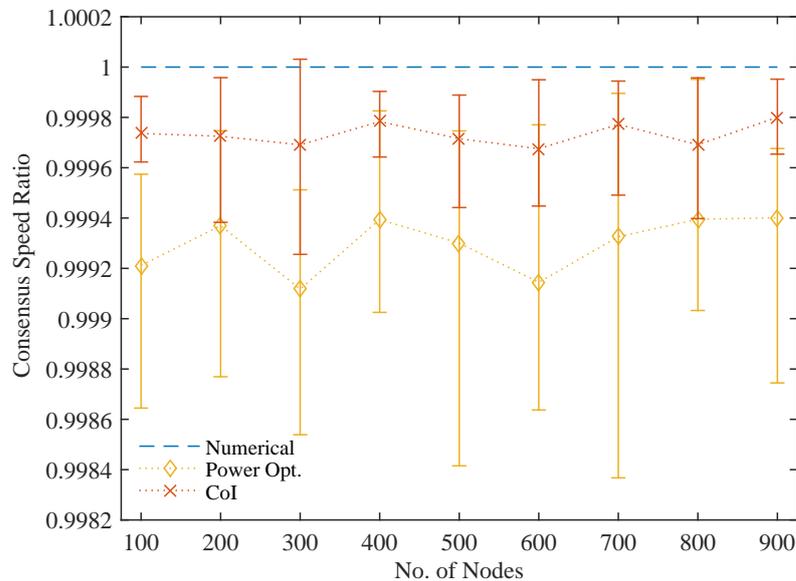


Figure 3.14: Consensus Speed Ratio for scale-free networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.

An optimal approach would be to combine the node detecting capability with the resource allocating capability of the fully numerical approach. But, as with most of the methods considered, it is always a trade-off between performance and algorithm run-time.

k-Outdegree Ring Lattice with Small-world

A regular ring lattice is a graph laid out in a circle with $k/2$ outward connections on either side of each node. Small-world rewiring is then applied as in Section 3.2.4.1. The leadership identification is simple for these graphs with the Power Opt able to find the optimal leadership resource each time. Therefore, the graph is only included in the following section on network comparison.

Network Type Comparison

In Fig. 3.15 the networks, considered so far, are compared for an optimised perturbation found using the CoI optimisation method. The results presented here appear to, on the surface, contradict previous work by Olfati-Saber (2005) where small-world networks are stated to be ideal candidates for ultrafast information networks, however Fig. 3.15 shows that both k -NNR and ring lattice networks with small-world rewiring

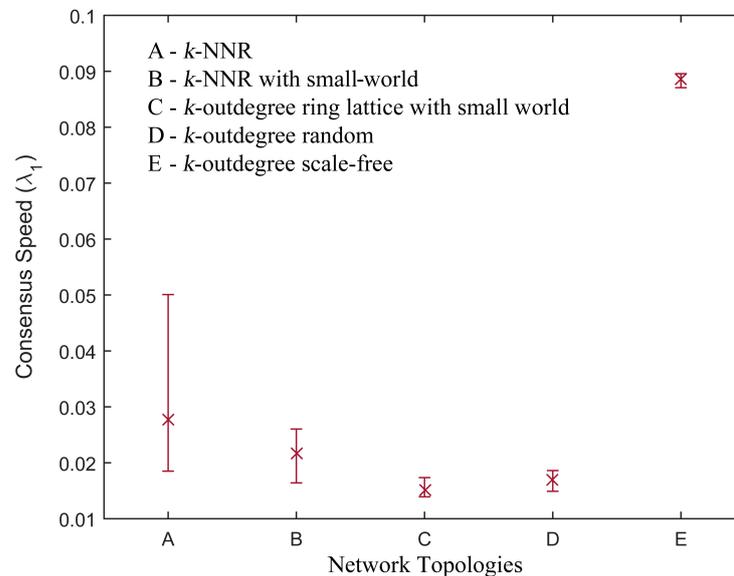


Figure 3.15: Consensus speed comparison for different network structures, where $k = 10$, influenced by a near-optimal perturbation. The small-world connections were rewired with a probability of $q = 0.1$.

perform comparatively poorly. This highlights that there is a difference in the topology requirements for fast consensus when information is being spread by a perturbation compared with when nodes are averaging their state in the absence of a perturbation, as considered by Olfati-Saber (2005).

In a node averaging situation, with no external perturbation, the ideal topology occurs when every node is connected to every other (all-to-all). Introduce a perturbation and an all-to-all topology produces a slow response as there is a larger system inertia to overcome; making it more difficult to transition the whole network to a new state, as displayed by Punzo et al. (2016). Therefore a network, where the majority of nodes have few connections but these connections are arranged to produce short paths to the leader nodes (those directly affected by the perturbation), will perform well in achieving perturbation driven consensus. This topological description is similar to that of a scale-free network, which is demonstrated in Fig. 3.15 where scale-free performs significantly better than the other networks considered.

Fig. 3.15 also demonstrates that distinct community structure can produce a good performance, with the k -NNR network outperforming the small-world and random graphs in the majority of cases. This is less effective than the scale-free networks, but a similar principle is at work. As discussed, scale-free networks are effective due to the short paths between leader nodes (those receiving a perturbation) and every other node in the graph. A short path between leaders and every other node in the graph is also achieved for k -NNR, as each prominent hub is supplied with a perturbation and all the nodes within the hub are well connected to each other. k -NNR probably performs less well as the perturbation resource is limited and has to be divided to reach multiple hubs. This reduces the rate of response from nodes in the k -NNR network when compared with the scale-free graphs where an optimal allocation usually requires fewer nodes to be supplied with a perturbation than in k -NNR graph examples.

Large Networks

For large networks ($N \geq 1000$) a comparison of the Power Opt and CoI methods with the numerical optimiser benchmark was not feasible, due to the computational time required. The upper limit for consensus speed, $\max_i((v_{L1})_i)$ as stated in Theorem 3.1, was used instead as a benchmark to show, in Fig. 3.16, that near-optimal results were still being achieved. Unlike the previous results in this Chapter, the upper limit could be used as the high outdegree of 30 would enable the results to approach the limit's value. For previous analyses, especially those with relatively low outdegrees, the limit would have greatly exceeded the convergence speed achieved by an optimal allocation.

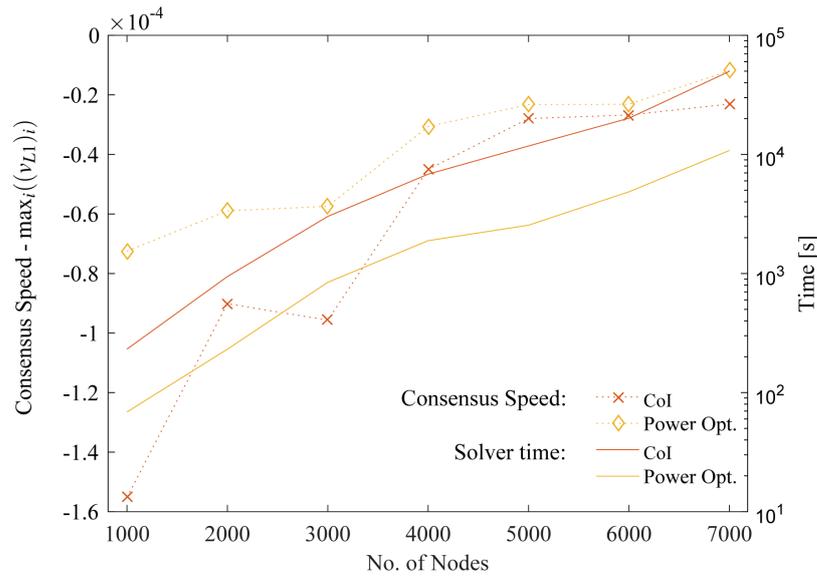


Figure 3.16: Solver time and difference between optimised consensus speed and $\max_i((v_{L1})_i)$ for large k -NNR networks with an outdegree of 30. A network was analysed at each 1000 node interval where $\max_i((v_{L1})_i) = \{31, 14, 10, 7, 5, 4, 4\} \times 10^{-4}$ for the analysed networks.

In Fig. 3.16 the Power Opt outperforms the CoI vector, due to the CoI method relying on a numerical optimiser, that often finds suboptimal local minima when analysing such a large network. The Power Opt method is effective even with very large networks (10^3 nodes) where the calculation times for the numerical optimiser would be extremely long. The trend in Table 3.3 predicted that a 7000 node network would take approximately 166 days to evaluate with the numerical optimiser and computational hardware used for this dissertation.

In the case of a high outdegree network, the CoI vector only requires the contribution of one community, as discussed, and hence closely matches the resource allocation generated by the Power Opt approach. This is the case in Fig. 3.16 where an outdegree of 30 is large enough for Power Opt analysis to find a near-optimum resource allocation. The explanation for this can be understood by looking at the comparison in Fig. 3.17 of the same distribution of 7000 nodes with (a) an outdegree of $k = 5$ and (b) $k = 30$ assigned according to k -NNR connection rules. This comparison emphasises that, although there are lines extruding from the central collection of nodes in the $k = 30$ case, these lines are short in comparison with the low outdegree example. Therefore, these 7000 node examples are presenting the same results as the 75 node examples from Fig. 3.5 & 3.6. The $k = 5$ graph exhibited clear lines from the origin of the eigenvector

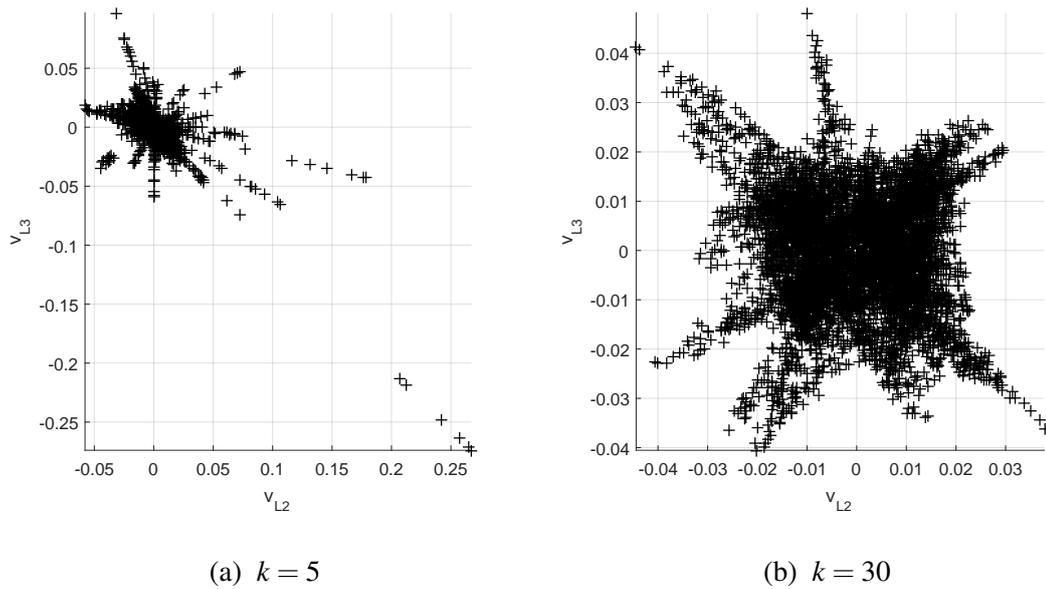


Figure 3.17: Geometrical interpretation of a 7000 node, k -NNR, network with outdegree (k). v_{L2} and v_{L3} represent the 2nd and 3rd most dominant left eigenvectors.

plot (Fig. 3.17) with contributions from multiple communities of influence required to find an optimal resource distribution. In the $k = 30$ graph the nodes were more clustered together and this resulted in only one community of influence being required to achieve an optimal allocation.

3.2.5 Influential Community Detection

The CoI approach also provides a form of community detection; when compared with the modules detected by the Leicht-Newman algorithm (2008) the CoI algorithm found that the different CoI vectors (see Algorithm 3.1) had their most prominent nodes - in all but one case out of the 540 analysed - located in separate Leicht-Newman modules. The graphs considered were k -NNR between 100 – 900 nodes with set outdegrees for all nodes ranging from 5 to 10. The graphs were produced at 100 node intervals with 60 considered at each interval. Fig. 3.4 has already shown that the number of modules present does not tend to vary by more than 3 for 100 node k -NNR with the same out-degree. For 20 of these cases some of the leading nodes in the four top communities, as determined by CoI, were in the same module, as determined by the Leicht-Newman algorithm. However, in only one of these 20 cases did the CoI algorithm assign leadership resources to the first group of leading nodes and not the second that inhabited the same Leicht-Newman module. The one exception was a 900 node graph, with an

outdegree of 7, where the first and second CoI determined communities resided in the same 90 node Leicht-Newman module. Arguably this case was a result of the Leicht-Newman algorithm failing to detect a community, as there was only a single outgoing (observing) connection connecting the group of nodes in the first CoI to the rest of the module.

The toy example introduced in Fig. 3.18 shall be used here and throughout the following sections because of its clarity in displaying community divisions. In this section it highlights that the CoI method allocates network leaders to separate modules in the network. This figure reveals that there is some correlation between the modules defined by Leicht-Newman and the leaders identified by the CoI method. It can be conjectured, given this finding, that when the CoI method allocates resources to multiple CoI vectors the network has multiple distinct communities that are more effectively lead from within rather than by nodes that are central to the network as a whole.

This brief investigation into the use of the CoI approach to perform community detection is developed upon and expanded later in this chapter, see Section 3.4, where communities are defined with reference to first three eigenvectors of the system.

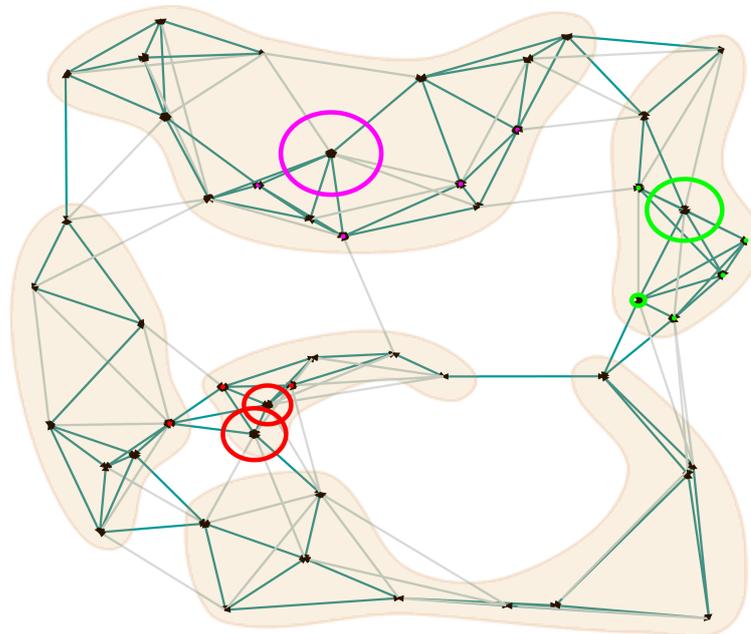


Figure 3.18: The CoI resource allocation is shown on this 50 node k -NNR network with coloured circles where each colour relates to a specific CoI vector (CoI 1 - red, CoI 2 - green, CoI 3 - magenta). The light brown shaded areas mark out the Leicht-Newman modules Leicht and Newman (2008) present in the graph.

3.2.6 Starlings - Predator Response

In this dissertation's introduction starlings were discussed and noted for their remarkable ability to form and maintain large flocks. In particular, Ballerini et al. (2008) found that starlings have approximately seven connections at all times regardless of the flock density. The reason for this connection rule is unclear, but Cavagna et al. (2010) later proposed two possible reasons for why starlings maintain this k neighbour topology. The first of these suggestions was cognitive limit; a claim that is supported by research on other birds where Emmerton and Delius (1993) performed experiments on the cognitive limits of pigeons and found 7 to be the upper limit. The experiment worked by training the pigeons to always select the higher number of dots - that were projected onto a piece of paper - with the main finding being that pigeons were able to differentiate between dot pairs until they reached a seven and eight dot comparison where they failed to pick successfully. If starlings had a similar cognitive limit then tracking seven birds may just be a consequence of their processing capacity. The second reason that Cavagna et al. (2010) proposed was that of optimal information spreading where the unproven claim is that this number has evolved as it produces the most effective graph for transferring information across the flock. The work of Young et al. (2013) was also noted in the introduction and it suggests that seven connections per bird provides the starlings with a more robust flock. The true reason for why $k \approx 7$ has emerged is still a matter of debate, but this section will present further evidence for why a relatively low outdegree has emerged as an effective limit for flocking starlings.

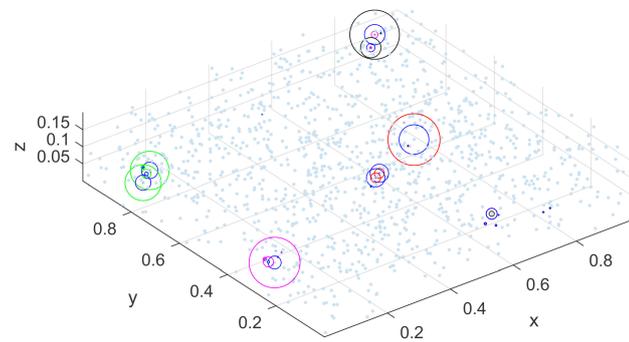
Consensus has been used to study starlings before, Shang and Bouffanais (2014) concluded that convergence rates increases as k increases for a k -NNR graph reaching agreement on alignment. This would seem to imply that starlings could more effectively spread information, of approaching predators, by observing more of their fellow birds. However, the work by Shang and Bouffanais considered consensus in the absence of a perturbation and, as noted in section 3.2.4.1, the topologies that achieve fast consensus in the presence of a perturbation are significantly different from those in the unperturbed scenario. Therefore, by examining static networks from a perturbation driven consensus point-of-view, it will be revealed that increasing k will not result in faster consensus. Instead the benefits of $k = 7$ are revealed, through comparison with higher outdegree models, as it enables multiple distinct communities to form that shall be shown to be helpful for the starlings' attempts to avoid predators.

The networks analysed thus far have all been planar graphs but starling formations, which inspired this research, obviously exist in three-dimensional space. Starling

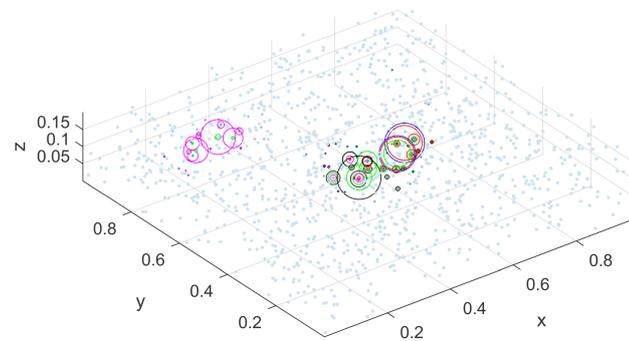
flocks are stated in Young's 2013 work to tend towards a thicknesses of between 0.13 and 0.27, where flock thickness is defined as the ratio of smallest to largest dimension of an ellipsoid having the same principal moments of inertia as the flock. In Fig. 3.19 three examples of 1200 node networks are presented that represent starling flocks with a thickness of 0.2, where the flocks were modelled in the same manner as Young et al. (2013) by randomly distributing nodes from a uniform distribution within a rectangular prism.

The leadership distribution examples in Fig. 3.19 are representative of the graphs analysed. The average perturbation driven convergence rate (from 10 randomly generated starling distributions) was $\lambda_{S1} = 0.0040$ for $k = 7$, and $\lambda_{S1} = 0.0019$ for both $k = 50$ & $k = 100$. This follows the trend found in the analysis of planar k -NNR graphs. The network response is faster for the lower k networks but also plateaus as the outdegree is increased with little difference between the response times of $k = 50$ and $k = 100$. What is of particular interest, when considering starling flocks, is the position of the optimal leaders according to CoI. Fig. 3.19 reproduces a common finding from the planar network analyses, where it was seen that the optimal leaders tend to be centrally located when the distinct community structure is removed by increasing the outdegree (see Fig. 3.5 and 3.6) as would be the case for $k = 50$ and $k = 100$. While for lower k outdegree examples, where the distinct community structure is more prominent, the optimal leaders are spread out. Considering an actual starling flock, this means that the most influential leaders are placed nearer the edges of the flock and will be some of the first birds to react to a predator attack as noted by Attanasi et al. (2015) and Herbert-Read et al. (2015). Therefore, the $k = 7$ topological rule appears to promote effective evasion of predators when compared with the slower responding high k out-degree flock where none of the, centrally located, optimal leaders are likely to trigger a predator evasion manoeuvre.

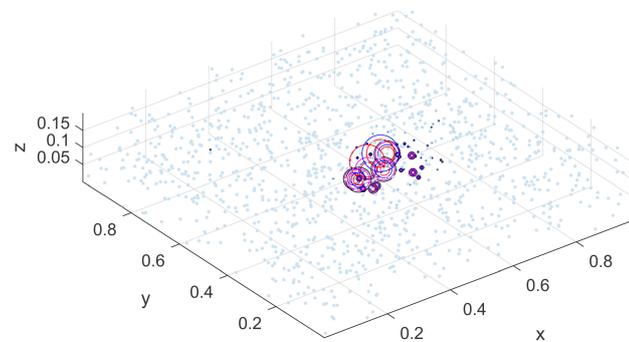
It is possible to conjecture that without such well-defined multiple community structure the starling response would be more uniform and rigid. This would probably be easier to track, predict and attack than the $k = 7$ network, where starlings maintain a cohesive unit but the shape fluctuates due to different communities responding at different rates triggered primarily by their local leaders. $k = 7$ therefore enables a fast and fluctuating response while assuring connectivity with a sufficiently large outdegree. The upper bound for k that ensures connectivity is defined by Balister et al. (2005), in their work on connectivity of k -NNR graphs, and stated as $0.9967 \log N$ where N is the number of nodes in the network. For a network of 1200 nodes, as considered here, $k \geq 5.11$ would assure connectivity.



(a)



(b)



(c)

Figure 3.19: 1200 node starling k -NNR flock model, as defined in Young et al. (2013) where nodes are randomly distributed for a flock thickness of 0.2, analysis for (a) $k = 7$, (b) $k = 50$ and (c) $k = 100$ where coloured circles are centred on nodes and the circle's radius is proportional to the resource allocation. Circle colour key: red – CoI vector 1, green – CoI vector 2, magenta – CoI vector 3, black – CoI vector 4, blue – CoI resource allocation.

3.2.7 Discussion

Much of the focus of this chapter, so far, has been on k -NNR networks which, by taking inspiration from starlings, could be an appropriate topology for developing a fast responding mobile multi-agent system if applied with an optimal perturbation. However, it could be argued that starling flocks have little to gain from achieving fast perturbation driven convergence to consensus. The CoI algorithm has revealed that the optimal solution for low outdegree k -NNR networks is to supply a perturbation to separate communities in the graph that essentially reach consensus on their own with limited input from nodes outside of their own community. Therefore a starling system, where the perturbation is isolated to one region of the graph (a predator will only be spotted by the birds in the local vicinity of the attack), employs a topology that is not aiming to reach a fast consensus. Instead a fast local response will be triggered by those in immediate danger, followed by a slower response from the other communities that shall ensure the flock remains connected.

Again considering a multi-agent system, factors other than consensus speed would still need to be considered before determining the ideal network topology. For example it is likely that a scale-free topology, despite producing faster perturbation led consensus, will be less resilient to targeted node removal, with its dependence on a few network hub nodes that have a large number of inward connections making it vulnerable. Whereas a k -NNR graph could withstand the removal of more influential nodes since there is a more even distribution of connections and hence greater redundancy.

Finally, it is clear from the results that the semi-analytical methods are most beneficial when employed on large networks. The difference in algorithm run time increasing as the network size grows, as noted in Table 3.3, which is an obvious consequence of the difference in time complexities for the algorithms used. Less predictable was the CoI method outperforming the numerical optimiser for larger outdegree cases, see Fig 3.13 for k -outdegree random networks and Fig. 3.10 for k -NNR. This reveals that the numerical optimiser, for larger graphs, is more susceptible to settling into local minimum, far from the global minimum. This failure occurs more often with the k -outdegree random graphs where the lack of structure produces a more complicated solution space for the optimiser to search.

3.3 Autocratic Leaders

This section considers the case where leaders are given more influence by paying less regard to the rest of the network when following a perturbation. These leaders that pay less attention to other nodes in the network are referred to as *Autocratic Leaders*.

The networks considered here are directed with natural swarms such as starlings employing directed communication. If considering an engineered system, duplex communication would be an option but there are benefits to using directed information flow in the system. One such advantage was found by Olfati-Saber and Murray (2004) who demonstrated that the communication/sensing cost of protocols, where the information flow is directed, is smaller than that of their undirected counterparts. Another advantage is that a directed graph can facilitate faster response to perturbations. A robust network, where redundant connections are an asset, should benefit from undirected edges as this is a simple way to build redundancy. In contrast, the focus of this chapter is on responsive systems and, as stated previously, the minimum requirement for consensus is a directed spanning tree. Therefore redundant connections, which go against the flow of information from a leader node, are likely to slow down system consensus and reduce system responsiveness. This claim shall be investigated and justified in Section 3.3.2.2.

To create autocratic leaders the edge weightings have to be adjusted for edges departing from leader nodes. Omidi and Abdollahi (2013) and Xiao and Boyd (2004) highlighted in their work that adjusting edge weightings based on the degree of the vertices involved can improve the convergence speed to consensus. For this section, the edge weighting shall only be adjusted for edges that depart from nodes that receive a leadership perturbation. The edge weighting is then dependant on the amount of resources that node is given. Once again nature can provide a source of inspiration; Nagy et al. (2010) found that flocking homing pigeons determine their spatial position and weight their neighbour monitoring based on a well-defined hierarchy. Nagy et al. (2010) also suggest that weighting the connections based on hierarchy, rather than having an egalitarian structure, may produce a more efficient flock. In this section the mathematical framework developed in the previous section shall be further developed to test whether a more efficient response could be achieved by an artificial swarm with constant outdegree (k -outdegree) topologies, where the hierarchy is created from the leadership resource allocation.

3.3.1 Consensus Model with Autocratic Leaders

As in Section 3.2, the networks considered here have N agents connected via local communication with a static, time-invariant, topology. The dynamics of this system were previously defined in Eq. (3.3) but have now been updated, to include the edge weighting on leader nodes to create autocratic leaders, as

$$\dot{x}_i = \sum_{j=1}^N (1 - c_i) a_{ij} (x_j - x_i) + c_i (u - x_i) \quad (3.21)$$

where x_i is the state of the i^{th} agent, $\sum_i c_i = 1$ and u is the scalar target value that all agents must achieve. a_{ij} is the ij entry of the graph's normalised adjacency matrix. For a network with the same outdegree for all nodes (k -outdegree), $a_{ij} = 1/k$ if there exists a directed edge from node i to j and 0 otherwise.

Translating Eq. (3.21) into the global dynamics of the system gives

$$\dot{\mathbf{x}} = -[I - C]L\mathbf{x} + C(\mathbf{u} - \mathbf{x}) \quad (3.22)$$

where I is an identity matrix, C is the perturbation matrix, $C = \text{diag}(c_1, \dots, c_p)$, and L is the normalised Laplacian matrix of graph \mathcal{G} . The normalised Laplacian is defined as $L = D - A$ where $D = \text{diag}(\mathbf{1})$ is the degree matrix of \mathcal{G} , with elements $d_i = \sum_j a_{ij}$. This Laplacian matrix, weighted and unweighted, always has at least one zero eigenvalue, $\lambda_1 = 0$, where the smallest, non-zero, eigenvalue (λ_2) represents the consensus speed.

Eq. (3.22) can be transformed to

$$\dot{\mathbf{x}} = -([I - C]L + C)\mathbf{x} + C\mathbf{u}.$$

The $C\mathbf{u}$ can be removed by considering

$$\mathbf{v} = \mathbf{x} - ([I - C]L + C)^{-1}C\mathbf{u},$$

which is possible since applying a perturbation guarantees that all eigenvalues are non-zero, as noted for Eq. (3.6), making $-([I - C]L + C)$ nonsingular. A coordinate change (affine transformation) can then be implemented with the model now represented by

$$\dot{\mathbf{v}} = -([I - C]L + C)\mathbf{v}. \quad (3.23)$$

For the perturbed, negated Laplacian ($-[I - C]L - C$) the smallest eigenvalue in magni-

tude is negative (and non-zero) with λ_1 representing the convergence rate to consensus. It is worth noting that this makes λ_1 both the smallest eigenvalue in magnitude and the largest eigenvalue of the system.

In the autocratic leader model $(-[I - C]L - C)$ an individual agent, i , has its attention focused on only its neighbours ($c_i = 0$), only the external perturbation ($c_i = 1$), or on a weighted combination of neighbour (internal) and external perturbation ($0 > c_i > 1$). In contrast, the previous $-L - C$ model makes no adjustment to the Laplacian with the perturbation always being additional to the internal neighbour monitoring. It is proved here for both models, when the perturbation is uniform, that the smallest eigenvalue in magnitude becomes $\lambda_1 = -1/N$.

Theorem 3.2. *Let L be the Laplacian of a directed graph that is at least weakly connected and C_U the uniform perturbation matrix of N entries where $C_U = \text{diag}([c, \dots, c]) = \text{diag}([\frac{1}{N}, \dots, \frac{1}{N}])$. Then, the smallest eigenvalue in magnitude of the negated, unweighted $(-L)$ and weighted $(-[I - C_U]L)$ Laplacian perturbed by the matrix $-C_U$ is*

$$\lambda_{(-L-C_U)\mathbf{1}} = \lambda_{(-[I-C_U]L-C_U)\mathbf{1}} = -c = -\frac{1}{N}. \quad (3.24)$$

Theorem 3.2 can be proved as follows:

Proof. Hammack et al. (2011) states that the right eigenvector of the Laplacian associated with the largest eigenvalue, $\lambda_1 = 0$, is the uniform vector, $\mathbf{1} = [1, 1, \dots, 1]^T \in \mathbb{R}^N$. When the Laplacian is perturbed by a matrix C_U , that is composed of a uniform vector along the diagonal, then the dominant right eigenvector remains uniform. This can be shown by first considering the eigenvalue and eigenvector relationships for the unperturbed Laplacian, which is

$$L\mathbf{x}_1 = \lambda\mathbf{x}_1$$

and the uniformly perturbed Laplacian

$$(L + \mathcal{P}I)\mathbf{v}_1 = \lambda_{L\mathcal{P}}\mathbf{v}_1,$$

where \mathbf{x}_1 and \mathbf{v}_1 are first right eigenvectors, \mathcal{P} is a constant and $\lambda_{L\mathcal{P}}$ is an eigenvalue of $(L + \mathcal{P}I)$. Then consider that

$$(L - \lambda I)\mathbf{x}_1 = 0$$

$$(L + \mathcal{P}I - \mathcal{P}I - \lambda I)\mathbf{x}_1 = 0$$

$$[(L + \mathcal{P}I) - (\lambda + \mathcal{P})I]\mathbf{x}_1 = 0,$$

which shows that $\lambda + \mathcal{P}$ is an eigenvalue of $(L + \mathcal{P}I)$ and that \mathbf{x}_1 is an eigenvector of $(L + \mathcal{P}I)$. Therefore, $\lambda_{L\mathcal{P}} = \lambda + \mathcal{P}$ and $\mathbf{x}_1 = \mathbf{v}_1$ where both eigenvectors are uniform. This applies for both the weighted and unweighted model where λ_1 is the real eigenvalue of smallest magnitude for a directed graph, since $0 > \Re(\lambda_1) \geq \Re(\lambda_2) \geq \dots \geq \Re(\lambda_N)$ for any perturbed Laplacian.

Given that $M_U \mathbf{x}_1 = \lambda_1 \mathbf{x}_1$, where \mathbf{x}_1 is uniform, it is then clear that

$$\lambda_1 = \sum_j (M_U)_{ij} = -c = -\frac{1}{N} \quad (3.25)$$

where $i, j \in \mathcal{V}$ and M_U represents either the weighted or unweighted Laplacian perturbed by a matrix C_U as the eigenvector and the row sums are the same for both cases. \square

Theorem 3.2 shows that, for a non-uniform perturbation, the weighted model will achieve a larger magnitude of λ_1 , and hence a faster consensus speed, than the unweighted model.

Theorem 3.3. *Let L be the Laplacian of a complete graph and $C_{\mathcal{P}}$ the non-uniform, diagonal, limited perturbation matrix of N entries. Then, the smallest eigenvalue in magnitude of the perturbed, negated, weighted Laplacian $(-[I - C_{\mathcal{P}}]L - C_{\mathcal{P}})$ will always be greater than that of the unweighted $(-L - C_{\mathcal{P}})$ model.*

Theorem 3.3 can be proved as follows:

Proof. Theorem 3.2 establishes that λ_1 will be the same for both the weighted and unweighted models influenced by a uniform perturbation. Introducing \mathcal{P} as a deviation from the uniformly perturbed state, which takes the form of a subtraction from an element of the uniform perturbation, $C_U = \text{diag}(\mathbf{c})$, to create a non-uniform perturbation matrix, $C_{\mathcal{P}}$. The matrix M_U is the uniformly perturbed matrix, affected by C_U , while $M_{\mathcal{P}}$ is a non-uniformly perturbed matrix, affected by $C_{\mathcal{P}}$.

Applying the addition of \mathcal{P} to any entry of \mathbf{c} will affect the row sum of the system matrix, except when a node has no outdegree where it will remain the same,

$$\sum_j (M_{\mathcal{P}})_{rj} < \sum_j (M_U)_{rj} \leq 0 \quad (3.26)$$

for both the weighted and unweighted model where $M_{\mathcal{P}}$ is the matrix M_U that has been altered by applying a deviation, \mathcal{P} , to entry r of \mathbf{c} . For a deviation \mathcal{P} , given Eq. (3.26),

$$(M_{\mathcal{P}}\mathbf{x}_1)_r < (M_U\mathbf{x}_1)_r < 0$$

for element r of the vector produced. It is known that $M_U\mathbf{x} = \lambda\mathbf{x}$ and, as stated before, the right eigenvector for the uniformly perturbed scenario will be a uniform vector, such as \mathbf{x}_1 . Therefore $M_{\mathcal{P}}\mathbf{x}_1 \neq h\mathbf{x}_1$ where $h \in \mathbb{R}^-$, when $M_U\mathbf{x}_1 = h\mathbf{x}_1$ and given Eq. (3.26). To converge towards the solution, $M_{\mathcal{P}}\mathbf{x}_{\mathcal{P}} = h\mathbf{x}_{\mathcal{P}}$, the varied entry of $\mathbf{x}_{\mathcal{P}}$ (referred to as x_r) must decrease for any non-uniform perturbation as $|(M_{\mathcal{P}}\mathbf{x}_1)_r| > |(M_U\mathbf{x}_1)_r|$. As a result,

$$x_r < x_s \quad \forall \{s \in \mathcal{V} \mid s \neq r\} \quad (3.27)$$

when $M_{\mathcal{P}}\mathbf{x}_{\mathcal{P}} = h\mathbf{x}_{\mathcal{P}}$.

It shall be shown that for the non-uniform perturbation scenario $(x_r)_{\mathbf{w}} < (x_r)_{\mathbf{uw}}$ and $(x_s)_{\mathbf{w}} > (x_s)_{\mathbf{uw}}$ by, again, considering how to converge towards the equality $M_{\mathcal{P}}\mathbf{x}_{\mathcal{P}} = h\mathbf{x}_{\mathcal{P}}$ where $h \in \mathbb{R}^-$. Convergence to equality is pursued by varying x_r with $-z$, where $\mathbf{x}_{\mathcal{P}}$ is an initially positive uniform vector. For clarity the variation $\sqrt{\quad}$ is applied to \mathbf{c} for only the first entry, and hence affects the first element of the vector in Eq. (3.29), therefore $-z$ is only employed on the first element of $\mathbf{x}_{\mathcal{P}}$.

$$\begin{aligned} M_{\mathcal{P}}\mathbf{x}_{\mathcal{P}}(z) &= \begin{bmatrix} (M_{\mathcal{P}})_{1,1} & \dots & (M_{\mathcal{P}})_{1,N} \\ \vdots & \ddots & \vdots \\ (M_{\mathcal{P}})_{N,1} & \dots & (M_{\mathcal{P}})_{N,N} \end{bmatrix} \begin{bmatrix} (\mathbf{x}_{\mathcal{P}})_1 - z \\ \vdots \\ (\mathbf{x}_{\mathcal{P}})_N \end{bmatrix} \\ &= \begin{bmatrix} (M_{\mathcal{P}})_{1,1}((\mathbf{x}_{\mathcal{P}})_1 - z) + (M_{\mathcal{P}})_{1,2}((\mathbf{x}_{\mathcal{P}})_2) + \dots \\ \vdots \\ (M_{\mathcal{P}})_{N,1}((\mathbf{x}_{\mathcal{P}})_N - z) + (M_{\mathcal{P}})_{N,2}((\mathbf{x}_{\mathcal{P}})_2) + \dots \end{bmatrix} \end{aligned} \quad (3.28)$$

The derivative of $M_{\mathcal{P}}\mathbf{x}_{\mathcal{P}}(z)$ is

$$\frac{d(M_{\mathcal{P}}\mathbf{x}_{\mathcal{P}}(z))}{dz} = \begin{bmatrix} -(M_{\mathcal{P}})_{1,r} \\ -(M_{\mathcal{P}})_{2,r} \\ \vdots \\ -(M_{\mathcal{P}})_{N,r} \end{bmatrix} = \begin{bmatrix} |(M_{\mathcal{P}})_{1,1}| \\ -|(M_{\mathcal{P}})_{2,1}| \\ \vdots \\ -|(M_{\mathcal{P}})_{N,1}| \end{bmatrix} \quad (3.29)$$

where $r = 1$ and the sign of the $M_{\mathcal{P}}$ entries are highlighted in the final expression since the diagonal entries are positive and the off-diagonal entries negative. Given that

$|(M_{\mathcal{P}})_{1,1}|_{\mathbf{w}} < |(M_{\mathcal{P}})_{1,1}|_{\mathbf{uw}}$, when the same \mathcal{P} is applied to the first entry of \mathbf{c} , and all the other elements remain unaltered, then

$$\left(\frac{d(M_{\mathcal{P}\mathbf{x}_{\mathcal{P}}}(z))_{\mathbf{w}}}{dz}\right)_r < \left(\frac{d(M_{\mathcal{P}\mathbf{x}_{\mathcal{P}}}(z))_{\mathbf{uw}}}{dz}\right)_r. \quad (3.30)$$

This indicates that, for the same \mathcal{P} and z variation, $M_{\mathcal{P}\mathbf{x}_{\mathcal{P}}}(z)$ will change less for the weighted model than the unweighted. Therefore a smaller variation, z , is required to achieve $M_{\mathcal{P}\mathbf{x}_{\mathcal{P}}}(z) = h\mathbf{x}_{\mathcal{P}}(z)$, when using the same value of \mathcal{P} , for the weighted model than the unweighted, hence

$$(x_r(z))_{\mathbf{w}} > (x_r(z))_{\mathbf{uw}} \quad (3.31)$$

and when normalising the right eigenvector $(x_s(z))_{\mathbf{w}} < (x_s(z))_{\mathbf{uw}}$.

An example of the unweighted model, $(M_{\mathcal{P}})_{\mathbf{UW}}$, and weighted model, $(M_{\mathcal{P}})_{\mathbf{W}}$, are shown below where each is affected by a perturbation $C_{\mathcal{P}} = \text{diag}([(c_{\mathcal{P}})_1, \dots, (c_{\mathcal{P}})_N])$ and where $a_{i,j}$ is an element of the adjacency matrix

$$(M_{\mathcal{P}})_{\mathbf{UW}} = \begin{bmatrix} -1 - (c_{\mathcal{P}})_1 & \dots & a_{1,N} \\ \vdots & \ddots & \vdots \\ a_{N,1} & \dots & -1 - (c_{\mathcal{P}})_N \end{bmatrix} \quad (3.32)$$

$$(M_{\mathcal{P}})_{\mathbf{W}} = \begin{bmatrix} -1 & \dots & a_{1,N}(1 - (c_{\mathcal{P}})_1) \\ \vdots & \ddots & \vdots \\ a_{N,1}(1 - (c_{\mathcal{P}})_N) & \dots & -1 \end{bmatrix}. \quad (3.33)$$

For a complete graph, i.e. all-to-all communication, considering Eq. (3.32) for the unweighted graph, $x_r = (x_r(z))_{\mathbf{uw}}$ and $x_{s'}$ represents all the $(x_s(z))_{\mathbf{uw}}$ terms as the $\sum_j a_{ij} = 1$ constraint in place, then

$$\begin{aligned} M_{\mathcal{P}\mathbf{x}_{\mathcal{P}}} &= \lambda \mathbf{x}_{\mathcal{P}} \\ (M_{\mathcal{P}\mathbf{x}_{\mathcal{P}}})_1 &= \lambda (\mathbf{x}_{\mathcal{P}})_1 \\ (-1 - c_{\mathcal{P}})(x_r) + x_{s'} &= \lambda (x_r) \\ \lambda_{\mathbf{uw}} &= -1 + \frac{x_{s'}}{x_r} - c_{\mathcal{P}} \end{aligned} \quad (3.34)$$

Eq. (3.31) showed that x_r would be less in the weighted case. To account for this it is defined with respect to the unweighted case as follows, $(x_r)_{\mathbf{w}} = x_r - \kappa$ where $\kappa \in \mathbb{R}^+$,

and taking into account the normalisation of the right eigenvector $(x_{s'})_{\mathbf{w}} = x_{s'} + \kappa$. Therefore, in a complete graph, i.e. all-to-all communication, considering Eq. (3.33) for the weighted graph

$$\begin{aligned} (M_{\mathcal{P}\mathbf{x}\mathcal{P}})_1 &= \lambda(\mathbf{x}_{\mathcal{P}})_1 \\ (-1)(x_r - \kappa) + (1 - c_{\mathcal{P}})(x_{s'} + \kappa) &= \lambda(x_r - \kappa) \\ \lambda_{\mathbf{w}} &= -1 + \frac{x_{s'} + \kappa}{x_r - \kappa} - c_{\mathcal{P}} \frac{x_{s'} + \kappa}{x_r - \kappa} \end{aligned} \quad (3.35)$$

Eq. (3.27) shows that,

$$\frac{x_{s'} + \kappa}{x_r - \kappa} > \frac{x_{s'}}{x_r} > 1$$

and it is known from Punzo (2013) that $0 > \lambda_{\mathbf{uw}}$, therefore

$$\begin{aligned} 0 &> -1 + \frac{x_{s'}}{x_r} - c_{\mathcal{P}} \\ c_{\mathcal{P}} &> -1 + \frac{x_{s'}}{x_r} \\ c_{\mathcal{P}} \frac{x_{s'}}{x_r} &> c_{\mathcal{P}} > -1 + \frac{x_{s'}}{x_r} \end{aligned}$$

Now the unweighted and weighted cases can be compared where $c_{\mathcal{P}} \frac{x_{s'}}{x_r}$ is included for clarity

$$\begin{aligned} \left| -1 + \frac{x_{s'}}{x_r} - c_{\mathcal{P}} \right| &< \left| -1 + \frac{x_{s'}}{x_r} - c_{\mathcal{P}} \frac{x_{s'}}{x_r} \right| < \left| -1 + \frac{x_{s'} + \kappa}{x_r - \kappa} - c_{\mathcal{P}} \frac{x_{s'} + \kappa}{x_r - \kappa} \right|. \\ |\lambda_{\mathbf{uw}}| &< |\lambda_{\mathbf{w}}| \end{aligned}$$

□

In the case of a directed k -outdegree graph, that is at least weakly connected, the convergence speed in the weighted case will almost certainly be faster than the unweighted case. This can not be as clearly shown as it was in the complete graph but the same approach as Theorem 3.3 can be taken. Eq. (3.34) remains the same for non-complete graphs but there is a change in Eq. (3.35). Specifically that $(x_{s'})_{\mathbf{w}} = x_{s'} + \kappa$ is no longer a valid assumption as not all of x_s entries will be involved in the calculation of $(M_{\mathcal{P}\mathbf{x}\mathcal{P}})_1$. $x_{s'}$ is merely a comparative tool between the unweighted and weighted cases, therefore it does not change. The most likely replacement for this expression in the k -outdegree case would be $(x_{s'})_{\mathbf{w}} = x_{s'} + \frac{\kappa}{n}$ where $n \in \mathbb{R}^+$. However, it is possible

that $(x_{st})_{\mathbf{w}} = x_{st} - \frac{\kappa}{n}$ or $(x_{st})_{\mathbf{w}} = x_{st}$. Only in the negative κ scenario could the logic of Theorem 3.3 be called into question as

$$\frac{x_{st}}{x_r - \kappa} > \frac{x_{st}}{x_r}.$$

3.3.2 Leadership Selection Algorithm

Employing the Communities of Influence method to optimise the selection of autocratic leaders produced a pattern of selection whereby only one leader per community was ever given a perturbation resource. This differed from the unweighted case where the perturbation resource would be spread among a number of nodes in a community of influence. There is mathematical support for the allocation of perturbation resources to an individual leader, especially in the case that this leader is autocratic with a high outdegree.

3.3.2.1 Effectiveness of a Single Autocratic Leader

Merikoski and Virtanen (1997) states that for a generic matrix, $M \in \mathbb{R}^{N \times N}$, with real and ordered eigenvalues, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$, the bounds for the most prominent eigenvalue are

$$\mu + \frac{sd}{(N-1)^{\frac{1}{2}}} \geq \lambda_1 \geq \mu + s(N-1)^{\frac{1}{2}}, \quad (3.36)$$

where $\mu = \frac{\text{tr } M}{N}$, i.e. trace of M divided by the number of vertices, is the mean of the eigenvalues and

$$sd = \sqrt{\frac{\text{tr } M^2 - (\text{tr } M)^2/N}{N}} \quad (3.37)$$

is the standard deviation. The equality on the left of Eq. (3.36) occurs if and only if $\lambda_1 = \lambda_2 = \dots = \lambda_{N-1}$ and on the right if and only if $\lambda_2 = \lambda_3 = \dots = \lambda_N$.

Taking λ_1 in Eq. (3.36) as the convergence speed to consensus for a perturbed Laplacian. Then for the weighted model, where the perturbation alters the off-diagonal and not the diagonal elements, of a constant outdegree graph, the trace of M is always $\text{tr } M = N$ given the adjacency matrix constraints used (see Eq. (3.8)). Only the standard deviation in Eq. (3.36) will vary when the perturbation is changed. A perturbation that minimises $\text{tr } M^2$ will ensure the minimal possible deviation of λ_1 from the mean, where the mean is maximum value achievable for λ_1 . To understand why minimising $\text{tr } M^2$ minimises the standard deviation, consider a simple multiplication $y = (1 - c) * c$.

This equation represents a simplification of the resource allocation in a 2 node system maximum and minimum difference in the split are 1:0 and 0.5:0.5 respectively, hence $(1 - c) + c = 1$ and $0.5 \geq c \geq 1$. The maximum value of y is found when $c = 0.5$, i.e. equal distribution of resources, while the minimum is achieved when $c = 1$. The same can be said for applying perturbations in the weighted case. For a weighted $(-[I - C_{\mathcal{P}}]L - C_{\mathcal{P}})$ and complete graph, setting all the off-diagonal value to zero (i.e. providing $c_i = 1$ to a node) will produce the smallest $\text{tr } M^2$. For a non-complete graph, the situation is not as simple but it does provide important insights. The higher the outdegree, i.e. the closer the situation is to the complete graph, the more impact the weighted model has on the system response. It is possible to conceive of a case where there is no benefit from supplying a node with perturbation resources. This would be a non- k -outdegree scenario where the leader had no outward connections. It can be concluded that for complete graphs a single autocratic leader will be most effective and for non-complete graphs with a high outdegree this will continue to be an optimal strategy. When the outdegree is not high the case changes, as will be seen in the following sections, where it is most effective to have an individual community (rather than whole network) leader is more common when considering autocratic leaders (i.e. the weighted model).

3.3.2.2 Effect of Redundant Connections

Eq. 3.36 & 3.37 can also reveal aspects of optimal structural design for a network. It was mentioned, briefly, that redundant connections, i.e. those counter to the flow of information from the most prominent, would likely reduce the responsiveness of the system. This can now be understood from a mathematical point of view, as these redundant connections will influence the $\text{tr } M^2$ value. If there are only uni-directional edges then $(M^2)_{ii} = (m_{ii})^2$, after a bi-directional edge is added $(M^2)_{ii} = (m_{ii})^2 + m_{ij}m_{ji}$. Therefore, redundant connections will increase $\text{tr } M^2$ and, as a result, decrease the systems responsiveness that is represented by $|\lambda_1|$.

3.3.2.3 Nodes of Influence

A Communities of Influence (CoI) algorithm has been described previously for the unweighted model, where vectors representing different influential communities were combined to produce an optimal leadership resource allocation. In this section a similar approach for the weighted model, Nodes of Influence (NoI), shall be presented

where limited resources are distributed amongst multiple nodes, each of which have been identified as the most influential in their community. These communities being defined by the first left eigenvector (FLE) of the adjacency matrix and then manipulated versions of that matrix. By only having to consider a single node from each community of influence the NoI approach becomes less computational expensive than the CoI algorithm. The NoI approach is defined here in Algorithm 3.2.

Algorithm 3.2 NoI – Variable Resources

procedure VARIABLE RESOURCES OPTIMISATION

 Calculate the FLE, \mathbf{v}_1 , for the adjacency matrix,

$$A = (a_{ij}) \in \mathbb{R}^{N \times N}.$$

for $m = 1 : n \in \mathbb{R}^+$ **do**

 Select leader node, r_m , that corresponds to

$$\max(v_i) \forall i \in \mathcal{V}.$$

$$a_{ir_m} = 0 \text{ and } a_{r_mi} = 0 \forall i \in \mathcal{V}.$$

 Calculate \mathbf{v}_{m+1} for the updated adjacency matrix.

end for

 Quasi-Newton solver* maximising $|\lambda_1(-[I - C]L - C)|$

 by varying c_l where $l \in \mathbf{r} = \{r_1, \dots, r_n\}$.

end procedure

Once the leader nodes have been identified a Quasi-Newton algorithm based optimiser* is used to determine the resource distribution amongst these selected nodes. For the outdegrees and network sizes investigated in this paper 5 leading nodes were found, through multiple tests on a range of scenarios, to be sufficient when seeking near-optimal leadership.

3.3.3 Results

This section shall detail the results obtained by comparing a numerical optimiser, which uses sequential quadratic programming methods[†], with the Nodes of Influence algorithm.

3.3.3.1 Nodes of Influence

In Fig. 3.20 a toy example of a 50 node k -NNR network ($k = 5$), previously introduced in Section 3.2.5, depicts the most influential node from each of the five communities

*implemented with the `fminunc` algorithm in MATLAB MathWorks (2015b)

†implemented with the `fmincon` algorithm in MATLAB by MathWorks (2015a).

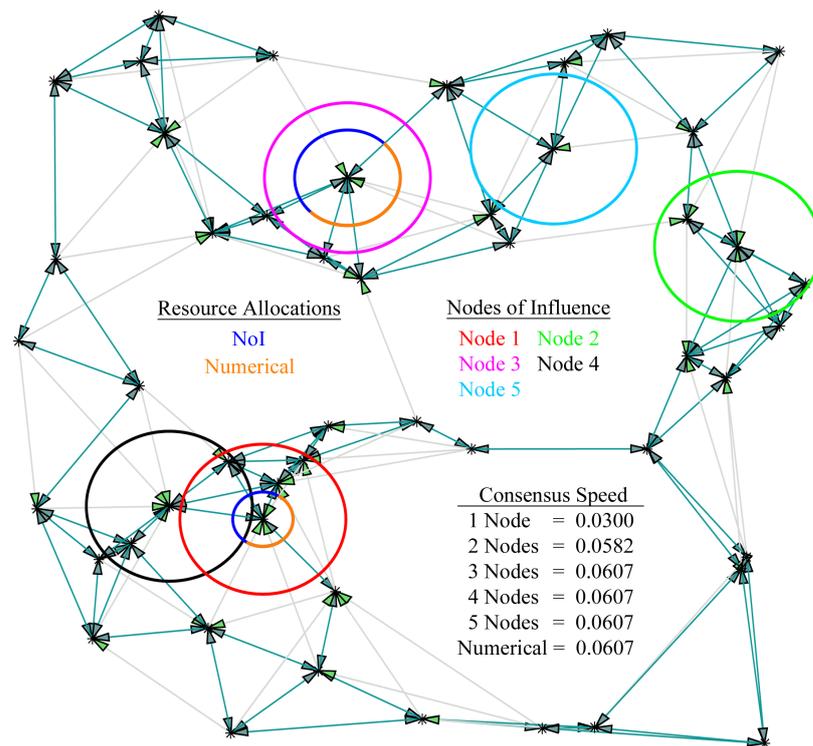


Figure 3.20: 50 node, 5 outdegree, k -NNR network. Two-way connections are depicted in green with one-way in grey. Coloured circles are centred on nodes, with the circle radius dependent on resource allocation.

of influence detected. The relative influence of each node for a specific community is proportional to the radius of the associated circle. The perturbation resource allocations, NoI and Numerical, are also detailed in the figure and seen to produce identical results with the blue and orange circles overlapping. In the unweighted case a perturbation was supplied to three communities, with two of these enlisting multiple nodes to spread the perturbation. But, as discussed, the weighted model requires only one leader from each community for this magnitude of perturbation. It is also worth noting that the NoI method allocated a perturbation to the first and third most prominent communities with the third community receiving the largest perturbation. The community order is often directly correlated with network influence but, as in this case, that is not always true.

k-NNR

For a k -NNR graph, which is at least weakly connected, the NoI vector produces on average a superior result when compared to the numerical optimiser for networks of

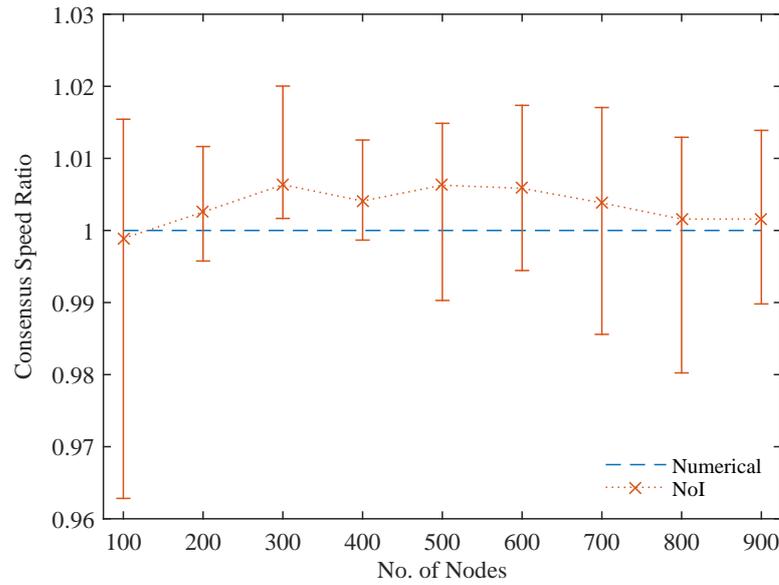


Figure 3.21: Consensus speed ratio for k -NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.

200 nodes or greater. This is displayed in Fig. 3.21 for different network sizes with 25 graphs examined at each interval.

In Fig. 3.22 the run time comparison between the two methods highlights the difference in calculation complexity, with the numerical approach requiring greater than an order of magnitude more time to find an, on average, inferior result than the NoI method. For both methods the worst case calculation complexity is mainly defined by the eigenvalue computation, that is defined as $\mathcal{O}(N^3)$ by Stewart (2000), with the numerical also requiring a search space of N and hence becoming $\mathcal{O}(N^4)$. These calculation complexities were then compared with the actual trends from Fig. 3.22 and detailed in Table 3.4 alongside the coefficient of determination (R^2) for the trend.

Table 3.4: Algorithm run time comparison for N nodes.

Algorithm	Complexity	Actual Trend	r^2
Numerical	$\mathcal{O}(N^4)$	$2 \times 10^{-6}N^{3.3}$	0.9974
NoI	$\mathcal{O}(N^3)$	$9 \times 10^{-4}N^{1.8}$	0.9949

The maximum consensus speeds found in Fig. 3.21 can then be compared by analysing the same graphs with the CoI method for the unweighted model ($-L - C$) and the NoI method for the weighted model ($-[I - C]L - C$), as presented in Fig. 3.23. These

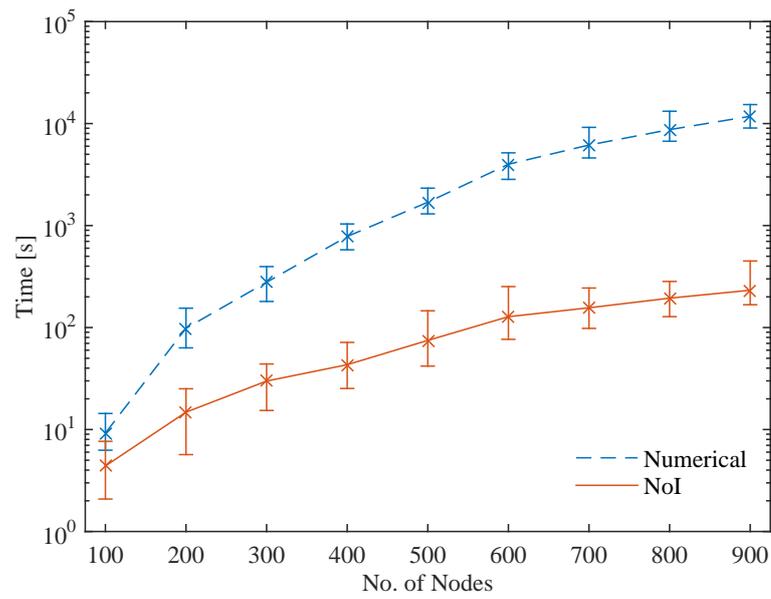


Figure 3.22: Run time comparison where the error bars mark the maximum and minimum deviation from the mean.

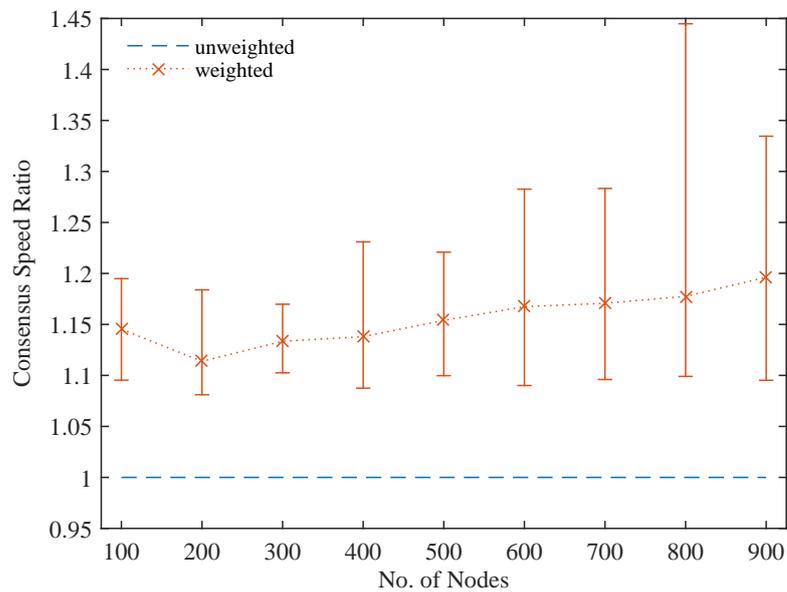


Figure 3.23: Consensus speed comparison between weighted and unweighted k -NNR networks with outdegree set at 10. The error bars mark the maximum and minimum deviation from the mean.

results agree with the proof of Theorem 3.3 by showing that the weighted model always achieves faster consensus speeds than the unweighted.

Lattice Ring with Small World Rewiring

A regular ring lattice is a graph layed out in a circle with $k/2$ outward connections on either side of each node, as displayed in Fig. 3.24. Small world rewiring is then applied, with a set probability that each link in the graph has a possibility of being rewired to end at a different node but retain the same source. This network was analysed for a rewiring probability of $q = 0.1$ with the results displayed in Fig. 3.25. Even with 10% of the connections being rewired, this network only ever contains one influential community. Therefore the full NoI approach is not required, instead the most prominent node of the FLE is always selected to lead. Despite this network requiring a simpler leader selection process, the complexity of the problem is highlighted by the numerical optimiser uncovering on occasion only local minima that are far from the global optimum, as shown in Fig. 3.25 for 300, 700 and 800 node networks. The analytical basis of the NoI and CoI algorithms is a strength as it guards against results that are only locally optimal and far from the system's global optimal.

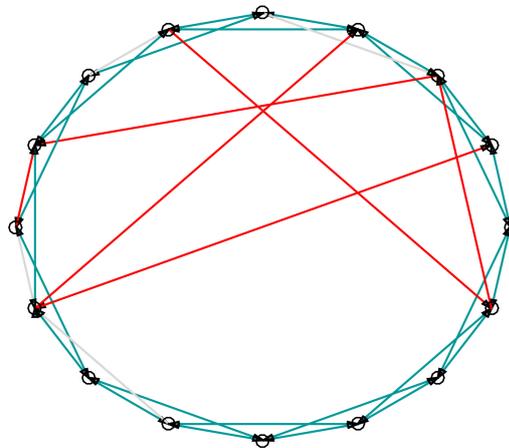


Figure 3.24: Lattice ring network with small world rewiring; 20 nodes, 4 outdegree with rewiring probability of $q = 0.1$. Rewired connections are in red, two-way in green and one-way in grey

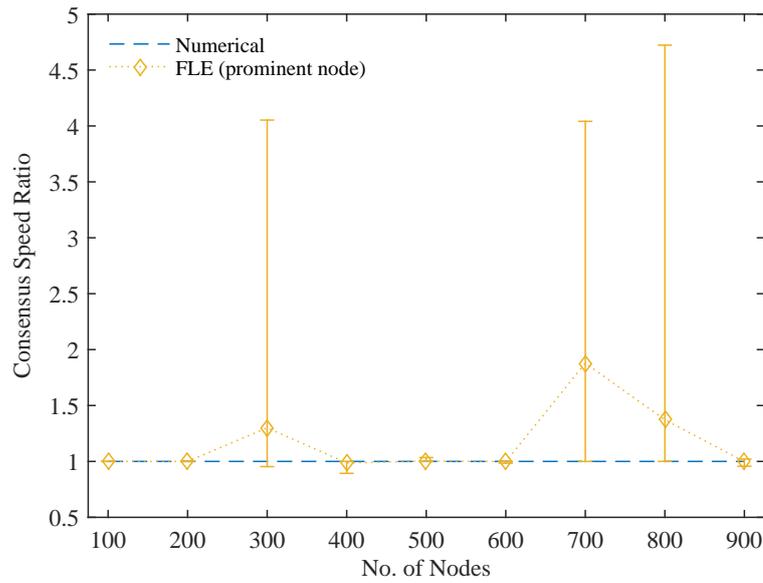


Figure 3.25: Consensus speed ratio for lattice ring networks with small world rewiring probability of $q = 0.1$ and outdegree for all nodes set at 10. The error bars mark the maximum and minimum deviation from the mean.

3.3.4 Ratio of Perturbation to Edge Weighting

This section presented the NoI algorithm, which selects only one leader from each community of influence when applying the weighted model ($-[I - C_{\mathcal{P}}]L - C_{\mathcal{P}}$). In the examples shown previously, selecting one leader from each community of influence was sufficient for an optimal convergence to consensus. This does not appear to be true for all cases, in particular when alterations are made to the ratio of perturbation magnitude to adjacency matrix weightings, i.e. the individual edge weightings ($\omega = a_{ij}$ for a connection from i to j). All edges in the adjacency matrix have the same weighting, ω . Figure 3.26 demonstrates that when this ratio is altered, for all edges in the weighted model, the (numerically found) optimal strategy selects multiple leaders from each community of influence. Therefore, NoI does not generate the optimal leadership allocation in these altered ratio scenarios.

The reason for this change in leadership distribution is proposed to be as follows; when the perturbation becomes large, relative to the capacity of edges in the graph, a bottleneck forms that delays the spread of information. For a large enough perturbation it will become more effective to distribute the perturbation across multiple nodes in a community. Eventually, by decreasing the edge weighting, the most effective approach

becomes that of providing all nodes in the graph with the same perturbation. This claim is supported by the example in Fig. 3.26 where the numerically optimised leadership selection is shown, for a network of autocratic leaders, to require multiple leaders per community in the $\omega = 0.1$ and 0.01 cases. As mentioned, the NoI method would not find a near-optimal result for these cases (as more than one leader was selected by the numerical optimiser for each community), but the more general approach provided by CoI, introduced in the previous chapter, is still valid.

It is also worth observing that varying the perturbation magnitude has a similar effect on the unweighted leadership model, as shown in Fig. 3.27, where increasing the weighting for all edges results in only a single leader being selected from each community. Both Fig. 3.26 and 3.27 are similar for the $\omega = 0.1$ & 0.01 cases. But when $\omega = 1$ the difference is more obvious with the autocratic case only selecting one leader from two communities while the egalitarian model has contributions from 5 communities with multiple nodes in two of those communities supplied with a perturbation.

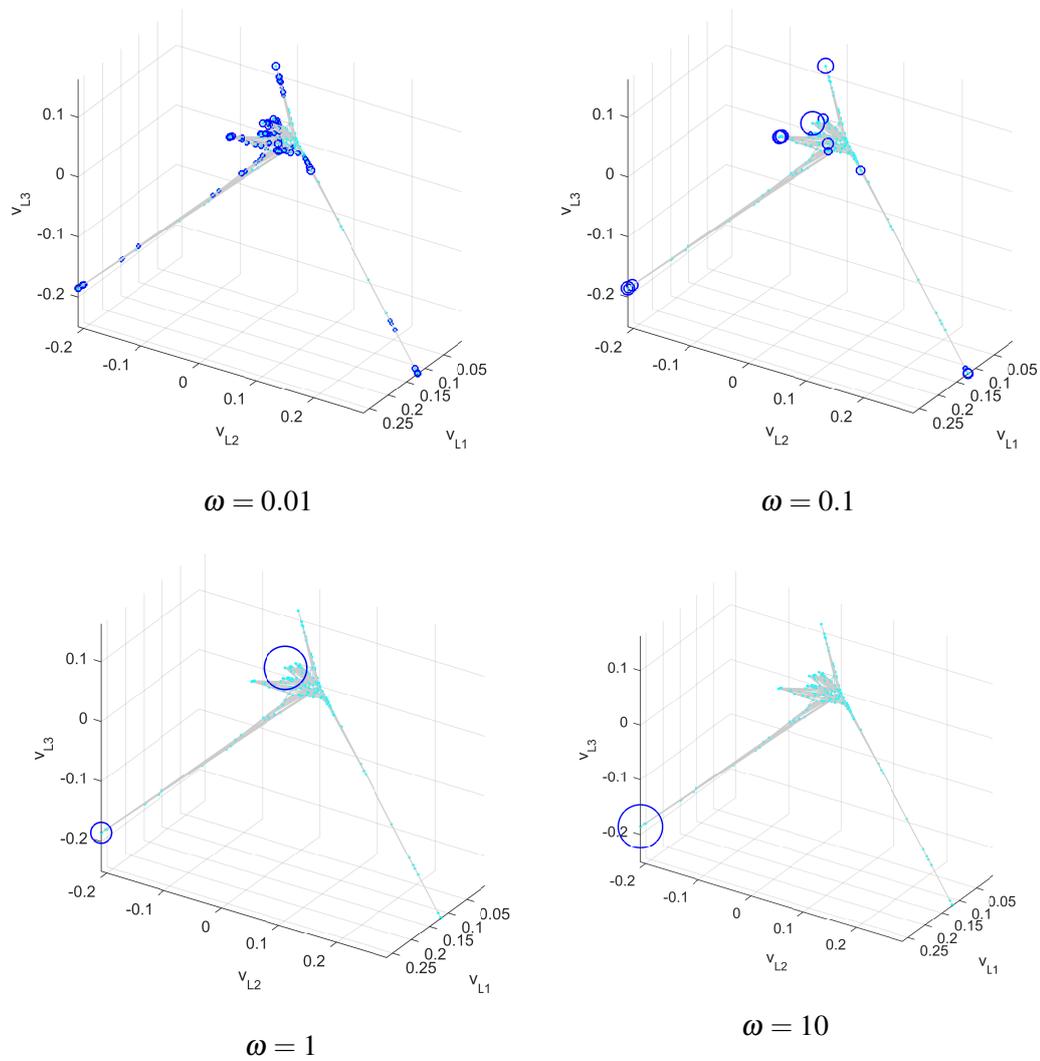


Figure 3.26: For the weighted model, a numerically optimised perturbation is marked for a 250 node k -NNR network ($k = 8$) with a different edge weighting in each plot. The first three left eigenvectors of the Laplacian matrix are represented by v_{L1} , v_{L2} and v_{L3} . The blue circles mark nodes with perturbations applied where the diameter is proportional to the percentage of the total perturbation applied.

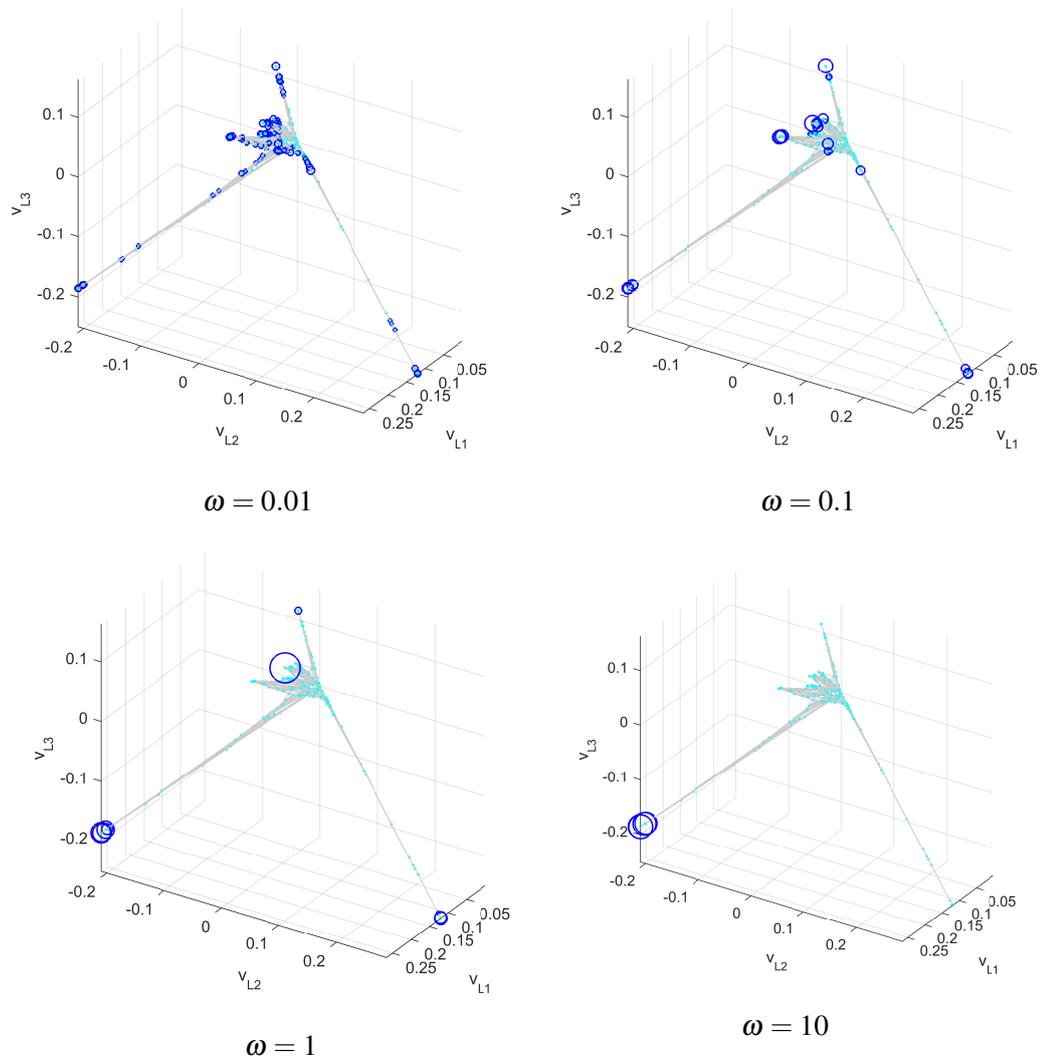


Figure 3.27: For the unweighted model, a numerically optimised perturbation is marked for a 250 node k -NNR network ($k = 8$) with a different edge weighting in each plot. The first three left eigenvectors of the Laplacian matrix are represented by v_{L1} , v_{L2} and v_{L3} . The blue circles mark nodes with perturbations applied where the diameter is proportional to the percentage of the total perturbation applied.

3.4 Variable Outdegree

Up until now this dissertation has focused on constant, k , outdegree networks but this section shall consider the optimal perturbation problem applied to variable outdegree networks. Constant outdegree being a subset of variable outdegree. In Section 3.2 constant outdegree networks were discussed that have a uniform first right eigenvector; enabling the first left eigenvector (FLE) to be effective at highlighting nodes for leading fast convergence to consensus. But when the outdegree is variable this produces a more challenging environment for optimal leadership detection; it becomes not only important to identify the most influential nodes (as highlighted by the FLE) but it may also be relevant to identify and concentrate resources on those that are hard to influence. It is worth reiterating that the optimisation metric is that of convergence rate to consensus, where consensus requires the whole graph to reach the new state. The issue can be summarised, colloquially, by saying that the variable outdegree problem needs to ensure that *no one is left behind* while enabling the most influential nodes to lead.

3.4.1 Communities of Dynamic Response

The Leicht-Newman community detection algorithm has been introduced previously as a community detection method that compares the density of connections present with the density expected from a network where the edges are randomly allocated. This section presents the Communities of Dynamic Response community detection method, which operates on directed networks and - unlike the Leicht-Newman method - is not heuristic as it is indirectly based on the structural layout of the network. Communities are instead determined by considering the prominent modes of dynamic response, i.e. the system's eigenvectors. A comparison of the communities found by the Leicht-Newman algorithm and the Communities of Dynamic Response is shown in Fig 3.28 for a toy example, prior to the Communities of Dynamic Response algorithm being formally defined. This figure indicates a similarity in community division between the two methods but also that there are clear differences. The dynamic response of the network finding fewer communities (marked with blue, orange and red coloured nodes) than the Leicht-Newman algorithm (marked out by light green shading). The Leicht-Newman communities are predominantly associated with one community of dynamic response, with one exception. This exception sees a Leicht-Newman community split

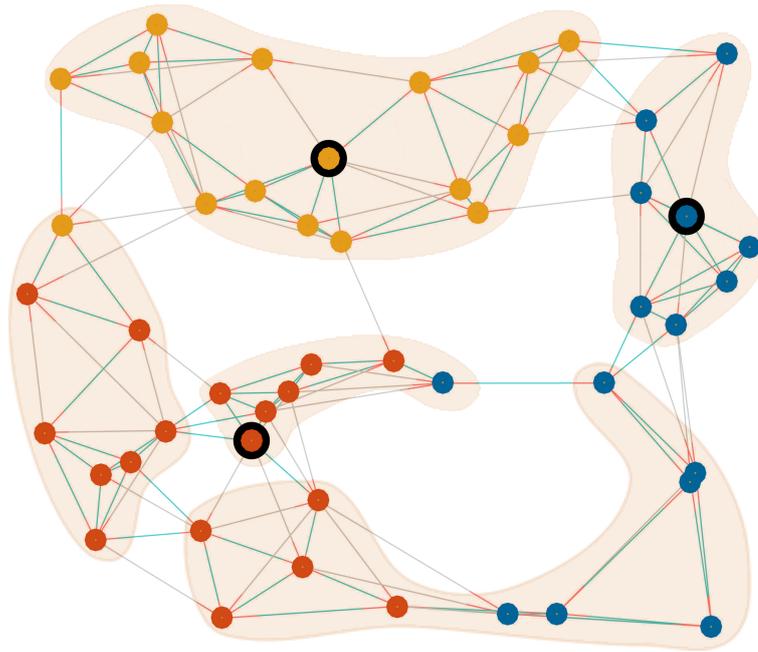


Figure 3.28: Comparison of communities - detected for a 50 node, 5 outdegree, k -NNR graph - by the Leicht-Newman algorithm (highlighted in light brown) with those detected by Algorithm 3.3 (node colour indicating community designation).

between two dynamic response communities. The question that needs to be answered is: what is the point in having a different and differing community detection method?

The Leicht-Newman algorithm highlights a group of nodes, in the centre of Fig. 3.28, as the smallest grouping, but as is known from Fig. 3.18 this community is home to the most influential nodes for this network, as determined by the FLE. Therefore, these nodes wield the most influence over the whole network, but seen through the lens of the Leicht-Newman approach they appear to be the most isolated clique. The communities of dynamic response, in comparison, show that these nodes have a wider reaching influence, and that this network is likely to respond as three distinct groups. It is also worth reiterating that the Leicht-Newman detection algorithm is a heuristic process, that relies on probability, and, therefore, the communities it detects may vary if repeated analyses are conducted. The Communities of Dynamic Response algorithm is, however, based solely on the graph's eigenvectors that do not change without changing the graph first.

The Communities of Dynamic Response algorithm is defined by considering the first three left eigenvectors of the directed Laplacian matrix of a network. These three eigenvectors can be visualised by plotting each node according to its value in the eigenvectors, as depicted in Fig. 3.29 (a) & (b). This reveals the communities that consist

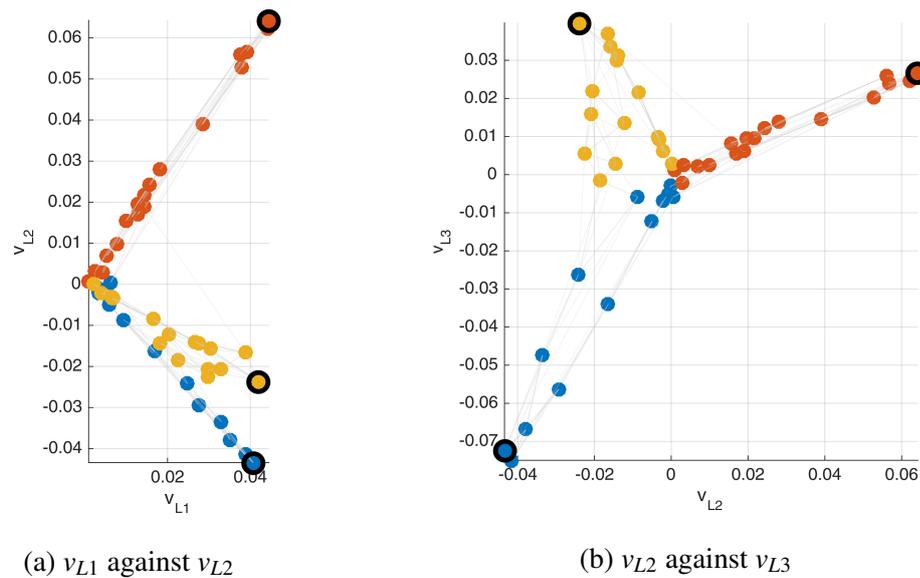


Figure 3.29: Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix for the 50 node k -NNR in Fig. 3.28. Community designation according to Algorithm 3.3 is noted using node colour and the most prominent community nodes have black outlines.

of nodes between the most prominent node (which has the largest value according to the FLE for its community) and the origin of the plot. Nodes in the same dynamic response community are part of a connected path in the graph that leads to this prominent node. When considering dynamic response of the whole network the FLE is of most importance, as it captures the biggest and most cohesive (in the same direction) system response, and, hence, is used to determine the most prominent nodes. These prominent nodes in Fig. 3.28 & 3.29 are marked with a black outline and are the same as those identified previously by CoI algorithms in Fig. 3.18.

A more detailed description of the Communities of Dynamic Response algorithm is now given in Algorithm 3.3.

Algorithm 3.3 Detecting communities of dynamic response**procedure** COMMUNITY DETECTION

Find the first three, normalised, eigenvectors of the Laplacian, $L \in \mathbb{R}^{N \times N}$, (associated with the three smallest eigenvalues in magnitude) \mathbf{v}_{L1} , \mathbf{v}_{L2} and \mathbf{v}_{L3} .

for $i = 1$ to N **do**

Set $\mathbf{e}_i = [(\mathbf{v}_{L1})_i, (\mathbf{v}_{L2})_i, (\mathbf{v}_{L3})_i]$ and $S_i = |\mathbf{e}_i|$.

end for

Sort N nodes by S , where I is the index and $I(1)$ is the node with $\max(S)$.

Set $n = 1$.

for $i = 1$ to N **do**

Set $n_I = I(i)$

if node $I(i)$ has an outdegree > 0 . **then**

o_I is a node that has an outward connection ending at $I(i)$.

$Sl_{o_I}^{o \rightarrow I(i)}$ is the scalar projection of \mathbf{e}_{o_I} onto $\mathbf{e}_{I(i)}$ (i.e. $(\mathbf{e}_{I(i)} \cdot \mathbf{e}_{o_I}) / S_{I(i)}$).

if $S_{I(i)} > Sl_{o_I}^{o \rightarrow I(i)} \forall o_K$. **then**

Add all o_I nodes to $C_n = \{I(i), o_1, \dots, o_m\}$, where n is the community number and m is the number of o_I nodes.

Set $list = C_n$.

Set $Sl_n^{list}(1) = \{S_{I(i)}, Sl_1^{o_I \rightarrow I(i)}, \dots, Sl_m^{o_I \rightarrow I(i)}\}$.

while $|list| > 0$ **do**

Set $K = list(1)$.

for $i = 1$ to indegree of j **do**

o_K is a node that has an outward connection ending at K .

$Sl_{o_K}^{o_K \rightarrow I(i)}$ is the scalar projection of \mathbf{e}_{o_K} onto $\mathbf{e}_{I(i)}$.

$Sl_K^{o_K \rightarrow I(i)}$ is the scalar projection of \mathbf{e}_K onto $\mathbf{e}_{I(i)}$.

Find S_{comp} by subtracting $Sl_{o_K}^{o_K \rightarrow I(i)}$ from $Sl_K^{o_K \rightarrow I(i)}$.

if $S_{comp} > 0$ and $o_K \notin C_n$ **then**

Add o_K to C_n and $list$.

Add $Sl_K^{o_K \rightarrow I(i)}$ to Sl_n^{list} .

end if

end for

Remove $list(1)$ from $list$.

end while

Set $n = n + 1$

end if

end if

end for

for $i = 1$ to N **do**

if node i is present in multiple communities (C_n) **then**

$(Sl_n^{list})_i$ is node i 's entry in Sl_n^{list} .

Remove node i 's entry from Sl_n^{list} and $C_n \forall n$ where

$(Sl_n^{list})_i < \max((Sl_n^{list})_i)$.

end if

end for

end procedure

3.4.1.1 Outdegree as a Marker for Influence

In Fig. 3.30 the left eigenvectors are again used to display a network, this time a 50 node graph where connections have been randomly assigned and nodes are constrained to have an outdegree from 3 to 7. Unlike the previous example network, the communities of dynamic response are less clearly defined for this network and often composed of only a few nodes. This is not unexpected as, has been previously discussed, a random graph will usually result in a less well defined community structure. In fact, using the definition of Leicht-Newman community detection, there should be only one community present. In contrast there are many communities of dynamic response detected. The presence of so many dynamic response communities can be understood by looking at Fig. 3.31. Observe that the community leaders (nodes of a specific colour that are furthest from the plot's origin) are primarily those with the lowest outdegree. Since there are no clearly defined communities the influential nodes are primarily those that are hardest to lead, i.e. the nodes with the lowest outdegree. This relates back to the challenge of driving a variable outdegree network to consensus where a main part of the focus is that *no one is left behind*. This claim can be supported by taking the same approach as Section 3.3.2.1, where the effectiveness of a single autocratic leader was investigated.

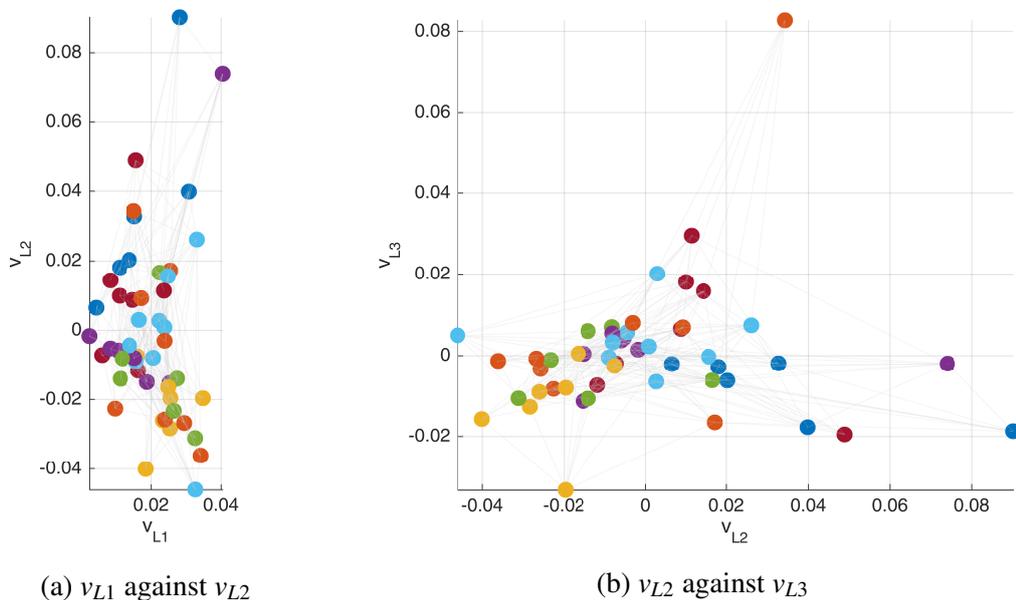


Figure 3.30: Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix for a 50 node random graph where outdegree varied between 3 and 7. Matching node colour indicates nodes belonging to the same community according to Algorithm 3.3.

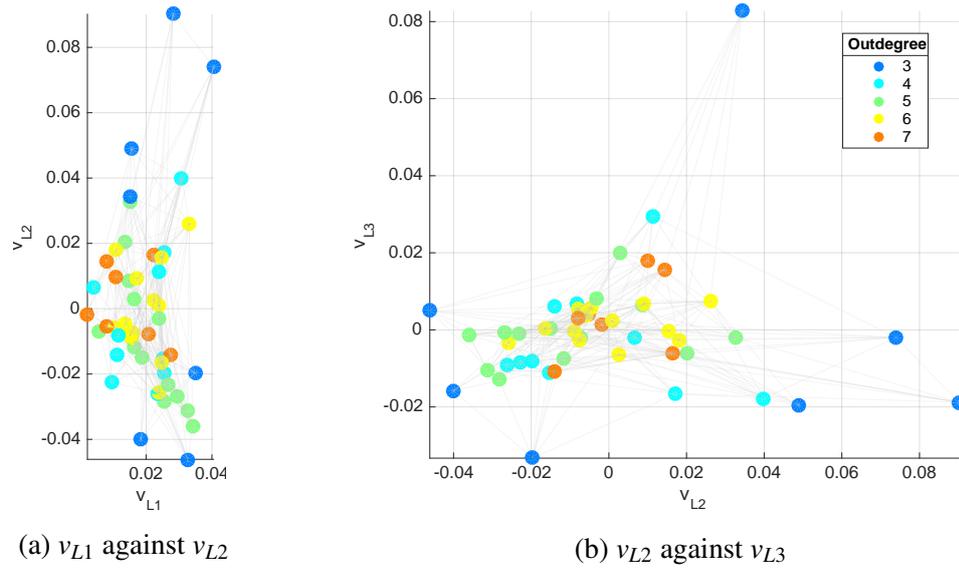


Figure 3.31: Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix for a 50 node random graph where the outdegree varied between 3 and 7. The node colour relates to the node outdegree.

Eq. 3.36 and 3.37 show that, for the unweighted model with variable outdegree, the mean is always $m = \frac{\text{tr } L + \sum_i c_i i}{N} \forall i \in \mathcal{V}$ where $M = -L - C$ and $c_i i$ is a diagonal element of C . As seen before, the trace of M^2 (i.e. $\text{tr } M^2$) is the only value that can be optimised when the magnitude of the perturbation is constrained. To maximise the magnitude of λ_1 a minimal value of s is required. As discussed in Section 3.3.2.1, for the weighted model, reducing the outdegree of a node reduces the trace of M^2 . This relationship is similar in the unweighted model.

Consider that, for the unweighted case, the diagonal element $(M)_{ii} = \sum_j a_{ij} \forall j \in \mathcal{V}$, then $(M^2)_{ii} = ((M)_{ii})^2 + \dots$ where $((M)_{ii})^2$ becomes an increasingly dominant component of $(M^2)_{ii}$ as the outdegree of i increases. Since the perturbation has to be added to a diagonal element of M , that is then squared, adding it to the smallest element (i.e. the row with the lowest outdegree) will result in the smallest increase in $\text{tr } M^2$. This is not the whole picture but supports the finding that many low outdegree nodes are selected as influential leaders, especially in graphs that lack clear community structure, such as in Fig. 3.30.

For completeness, the case of a nearest neighbour graph with variable outdegree, which is constrained to be from 3 to 7, is considered and depicted in Fig. 3.32. Despite the addition of variable outdegree the nearest neighbour network is still clearly divided into communities of dynamic response. The outdegree remains related to influence,

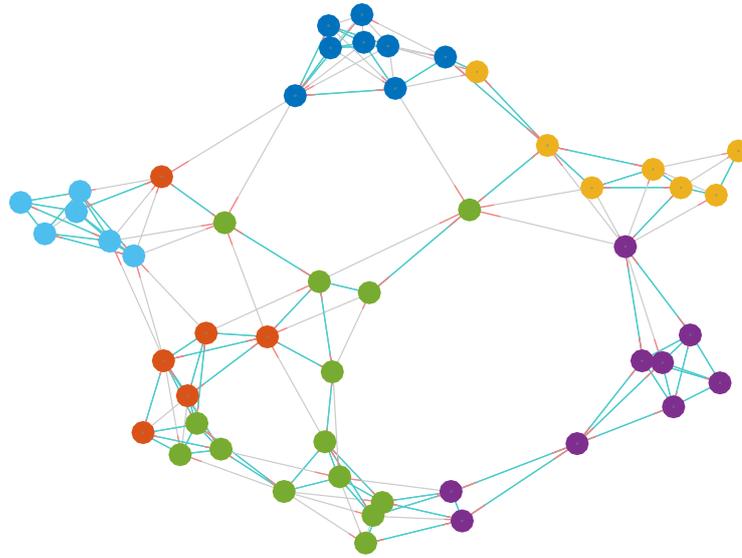
as in the random graph example, with the prominent community nodes all having a low, either 3 or 4, outdegree as depicted in Fig. 3.32 (b) & (c). But, unlike the random case, there are low outdegree nodes close to the plot's origin, which indicates that these nodes are less influential. Therefore, the relationship between outdegree is not as clear cut, as it was in the random case, with position within the community structure also an influencing factor.

3.4.1.2 Leader Selection

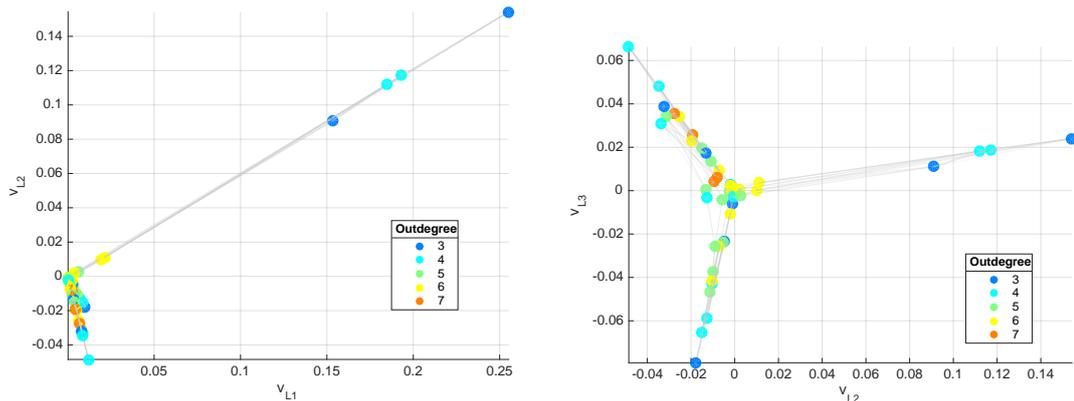
In Section 3.2.4, the CoI approach was compared with a numerical optimiser. Fig. 3.33 (a) presents a similar comparison of the CoI method versus the numerical optimiser where both are attempting to assign optimal resources/perturbations for driving the network to a new consensus state.

Fig. 3.33 (a) demonstrates that for a variable outdegree network the CoI approach (leading CoI nodes are marked out by different coloured circles) fails to find the numerically optimal leadership nodes (marked out with orange circles that represent applied perturbation). The leading CoI nodes 34, 15 and 41 are not given any resources and nodes 13, 43 and 49 are not detected as being influential. In contrast to the CoI method's success, it can be seen that the prominent nodes from Fig. 3.33 (b) and (c) - i.e. nodes 13, 28, 43, 45 and 49 that have the highest v_{L1} value for their community - are the only nodes allocated with the numerically optimal perturbation. It is interesting to note by observing nodes 43 & 49, and 13 & 28, that when two communities are aligned in a similar direction on the eigenvector plot it is the prominent node that is furthest from the origin and has the greatest v_{L1} value that is assigned with more resources. Another noteworthy point is that the magnitude of a node's v_{L1} value is not proportional to the resources allocated to it; for example, node 45 has the greatest v_{L1} value but not the largest allocation of resources.

Fig. 3.33 highlights the potential for the Communities of Dynamic Response method to be adapted into a computationally light leadership selection algorithm. The calculation of the first three left eigenvectors of the network would be used to select the prominent nodes in the communities found on the eigenvector plot as leaders. This could be done in a similar fashion to the CoI and NoI approaches, where an optimiser allocates resources but the search space is drastically reduced by only considering a select group of nodes.



(a) The communities of dynamic response are marked using different coloured nodes

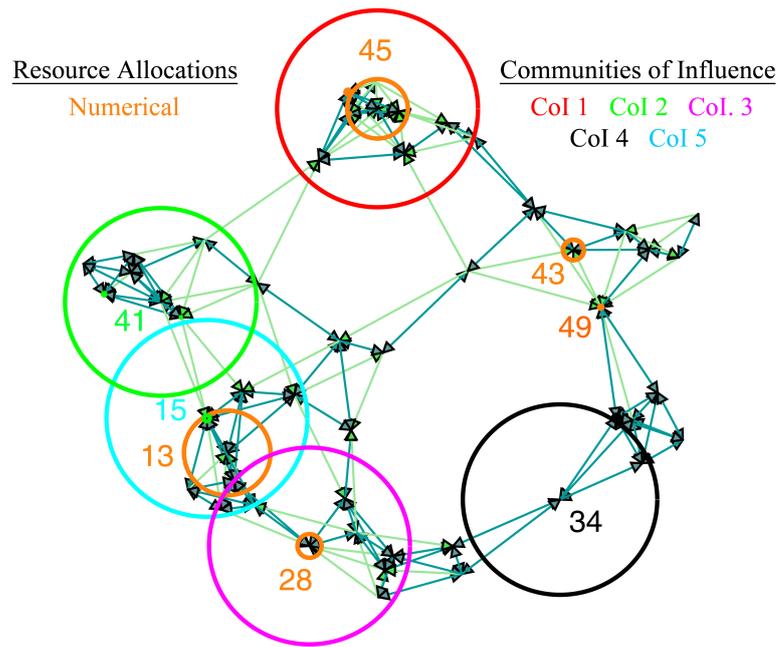


(b) v_{L1} against v_{L2} with node outdegree marked by colour

(c) v_{L2} against v_{L3} with node outdegree marked by colour

Figure 3.32: NNR graph with 50 nodes and an outdegree varying between 3 and 7: (a) Planar NNR graph layout based on x - y coordinates. Blue lines indicate two-way communication edges and grey indicates a one-way connection, with the red at the start of a line marking it as an outgoing edge. (b) & (c) Visualisation of node placement in eigenvector space, where v_{L2} and v_{L3} are the second and third left eigenvectors, respectively, of the Laplacian matrix. The dot colour denotes node outdegree.

The Communities of Dynamic Response method also excels as the most widely applicable leadership detection (optimal perturbation allocation) algorithm developed in this dissertation. Since variable outdegree networks have boundless topological lay-



(a) Network layout with the radius of the orange circles - centred on nodes - proportional to the resource/perturbation applied to that node. Five leading nodes from the Communities of Influence, see Algorithm 3.1, are also marked by a coloured circle.

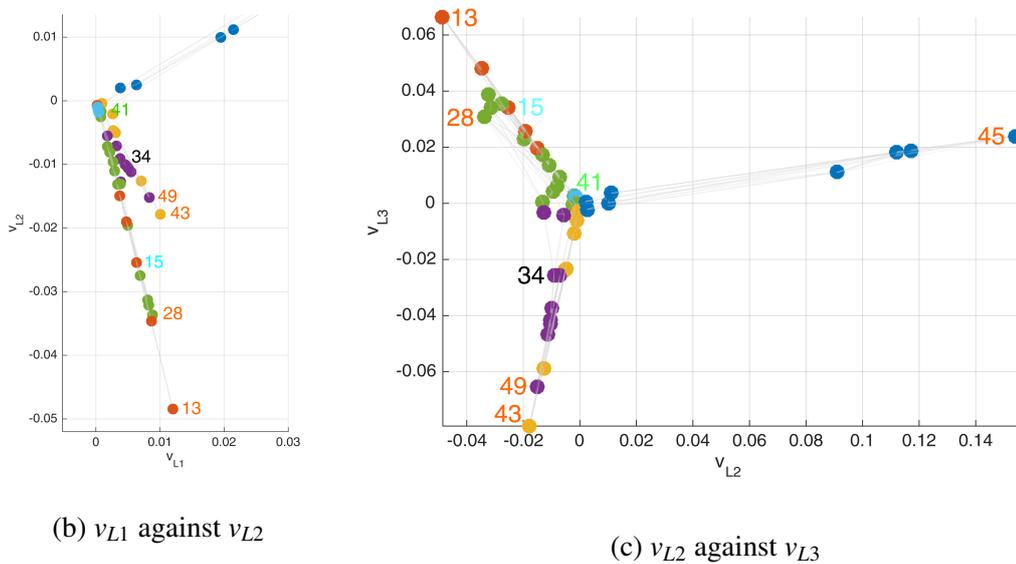


Figure 3.33: NNR graph with 50 nodes and an outdegree varying between 3 and 7 where certain nodes are labelled with an ID number for reference: (a) Optimal resource allocation using a numerical optimiser. (b) & (c) Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are the first three left eigenvectors of the Laplacian matrix. Nodes are coloured according to community designation as defined by Algorithm 3.3. Certain nodes are labelled with an ID number for reference.

outs and analysis can be applied to many other systems and applications, whereas much of the focus here has been on networks such as k -NNR that are strongly relevant to swarming systems. To this end, the following chapter shall move beyond developing decision-making and control methods for future swarming systems and explore the potential of applying such algorithms as analytical tools on existing networks.

3.5 Summary

A number of semi-analytical methods have been presented, for constant outdegree networks, that are able to select an optimal perturbation for leading a network rapidly to consensus. The first of these methods, the Power Opt approach, relies solely on the first left eigenvector (FLE) of the Laplacian matrix that is then scaled to realise an optimal allocation. This method was shown to operate particularly well, matching a numerical optimiser's performance, in high outdegree networks where there is only one dominant community present. When multiple influential communities are present, such as in low outdegree k -NNR networks, another method of leadership selection was developed; the Communities of Influence (CoI) approach. This approach is also reliant on the FLE but crucially also investigates the FLEs of manipulated versions of the adjacency matrix before employing an optimiser to produce an optimal leadership allocation.

The influence of leadership style on convergence speed to consensus was considered by contrasting egalitarian leaders with autocratic ones. Egalitarian leaders are those that receive a perturbation but also have their state mediated by those they are connected to. Autocratic leaders give greater weighting to the perturbation; in the case of a single leader receiving all of the perturbation resources they will ignore all of their neighbours and follow only the perturbation. Autocratic leaders were found to be more effective in leading a fast convergence to consensus when a non-uniform perturbation was applied. Autocratic leaders are often more effective when there is only one per community, therefore, the Nodes of Influence (NoI) was developed to allocate leadership to individual nodes in the most influential communities. This method was demonstrated to be sub-optimal when the ratio of perturbation magnitude to edge weighting was altered; specifically when increasing the perturbation or decreasing the edge weighting enough multiple autocratic leaders in the same community became optimal.

Finally the case of variable outdegree networks, of which constant outdegree networks are a subset, was considered. By combining the first three left eigenvectors, and con-

sidering what nodes were connected, it was possible to detect communities of dynamic response; i.e. the dominant communities that form in response to optimal perturbations. The most prominent nodes in these newly defined communities were also identifiable, according to the FLE, and hence the most effective leaders for fast consensus could be found. This is a useful discovery as the Power Opt, CoI and NoI methods are not effective on variable outdegree networks, where the first right eigenvector is no longer a uniform vector.

Chapter 4

Beyond the Swarm: Brain Connectomes

In the previous chapter systems were considered where nodes followed the actions of their neighbours. This meant that if well-connected nodes changed their state then information on this change of state would spread quickly to the whole swarm. The challenge, that has been successfully addressed by starlings, is to achieve high responsiveness as the scale of the swarm increase. The current chapter shall consider a network that is as vast as any known to humankind but one that is required to respond rapidly to stimulus; the brain. These networks are commonly referred to as connectomes with the scale far exceeding anything considered thus far. But that is not the only difference in this chapter, as a slight change of perspective will be required. For connectomes, the connections are defined so that, the direction of the edge corresponds to the direction of information flow. An effective leader node, as defined previously, is likely to have many edges pointing towards it. In the connectome scenario, a so-called ‘effective leader’ would become a bottleneck for information in the graph. This ‘leader node’ remains highly influential in the graph and critical to delivering the connectome’s function, but now acts as a collator of information, rather than an effective leader of nodes. The analytical capabilities, developed thus far, will be shown to provide insights into the key regions and the functions that these regions enable.

The human brain is often cited as being the most complex known object, due to the estimated 100 billion neurons with 100 trillion synaptic connections Braun et al. (2015). Our ability to map neurons and their connections is limited but ever improving. For smaller (non-human) brains, tracing the flow of, for example, silver along neuronal tracts in a brain can provide a map of connections (see Heimer and Robards (2013)

for more information). These networks are referred to as structural connectomes as the brain's underlying wiring structure is mapped out. But these processes are far too slow and labour intensive for more complicated and intricate brain structures. More advanced methods have been developed that use electro-magnetic scans, such as diffusion magnetic resonance imaging and functional magnetic resonance imaging (fMRI), to create maps that encompass millions of neurons (see Van Den Heuvel and Pol (2010) for a review on mapping resting-state brains with fMRI). These methods still require refinement and have been used, primarily, to produce undirected graphs. The directionality of the links is often unknown as it is only the co-activation of functionally related brain regions that is noted, which does not consider which region activated first and, therefore, does not highlight edge direction. These approaches do not produce a map of the underlying structure, but instead reveals regions that have a function together, therefore these graphs are called functional connectomes.

It is often the case that, when applying graph theoretical methods, the graph is transformed to an undirected graph. This may make the analysis simpler, and the mathematics easier to work with, but much of the key information is lost by ignoring the imbalance in the outdegree to indegree ratio. Indeed the Laplacian matrix, used throughout this dissertation, emphasises this imbalance further with each diagonal element equal to the sum of the non-diagonal elements in its row i.e. the indegree of a node.

The primary method of analysis in this chapter is based on the first three eigenvectors of the Laplacian matrix, in a similar manner to Section 3.4 where the Communities of Dynamic Response was introduced for directed graphs. For the cases where directed graphs have not been produced, the Communities of Dynamic Response method shall be employed to show that the method is still effective in undirected cases. The eigenvectors capture the dynamics of information flow from only the structural information given by the graph topology. This approach is analytical, which can be of benefit if considering possible datasets containing billions of neurons where any data processing algorithms would have to be computationally light. It is, therefore, with an eye towards the future of neuroscience that the following work is presented; efforts have already been made to analyse many of the existing, smaller, brain connectomes to show how these graphs correspond to existing intuitions and knowledge gained through experimentation. These previous studies have identified important brain regions by performing sophisticated numerical simulations of information flow throughout the brain. But such algorithms would be unlikely to perform as well as an analytical approach on massive graphs. These previous analyses do provide a useful comparison for the methods developed herein with the majority of this chapter devoted to such comparisons.

Understanding the brain's function is a major pursuit of humankind, with mapping and comprehending the human connectome a critical step. This work takes a step towards achieving such a feat with its potential application on vast and, on the surface, incomprehensible network.

4.1 *Caenorhabditis Elegans*

This chapter shall begin with a relatively simple subject, the *caenorhabditis elegans*; a non-parasitic nematode (worm) that is transparent and unsegmented with a long cylindrical body shape that is tapered at both ends. They grow to about 1 mm in length and have been a hugely popular subject for study, ever since Sydney Brenner introduced the *C. elegans* as a model organism for developmental biology and neurology research, as documented by Laboratory for Optical and Computational Instrumentation, University of Wisconsin-Madison (2009).

The neuronal network of the *C. elegans* was originally charted by White et al. (1986) in 1986 and is shown in Fig. 4.1 (image taken from WormAtlas (2013)). The 279 neuron (node) network, used throughout this section, is an updated version of the original connectome that was produced after further investigation by Varshney et al. (2011). The neurons in this connectome belong, primarily, to four functional categories: motor neurons - control movement through synaptic contacts onto muscle cells; sensory neurons - receive different forms of sensory input; interneurons - relays signals onto other neurons; and polymodal neurons - poses multiple of these functions.

This connectome is composed of two topologies that can also be collated into a single directed graph. The topologies are an undirected electrical junctions network and a directed network of the chemical connections. When considering only the electrical junction, the giant component of the graph is 248 nodes as detailed by Varshney et al. (2011). When the graph becomes directed with the addition of the chemical connections the giant component grows to 274 nodes.

4.1.1 Varshney's Intuitions

In the previous chapter, the networks were analysed by examining the system's modes of dynamic response (eigenvectors). A similar approach has already been taken for the undirected electrical gap junction network of the *C. elegans* but this chapter expands upon the intuitions detailed by Varshney et al. (2011) on the application of spectral

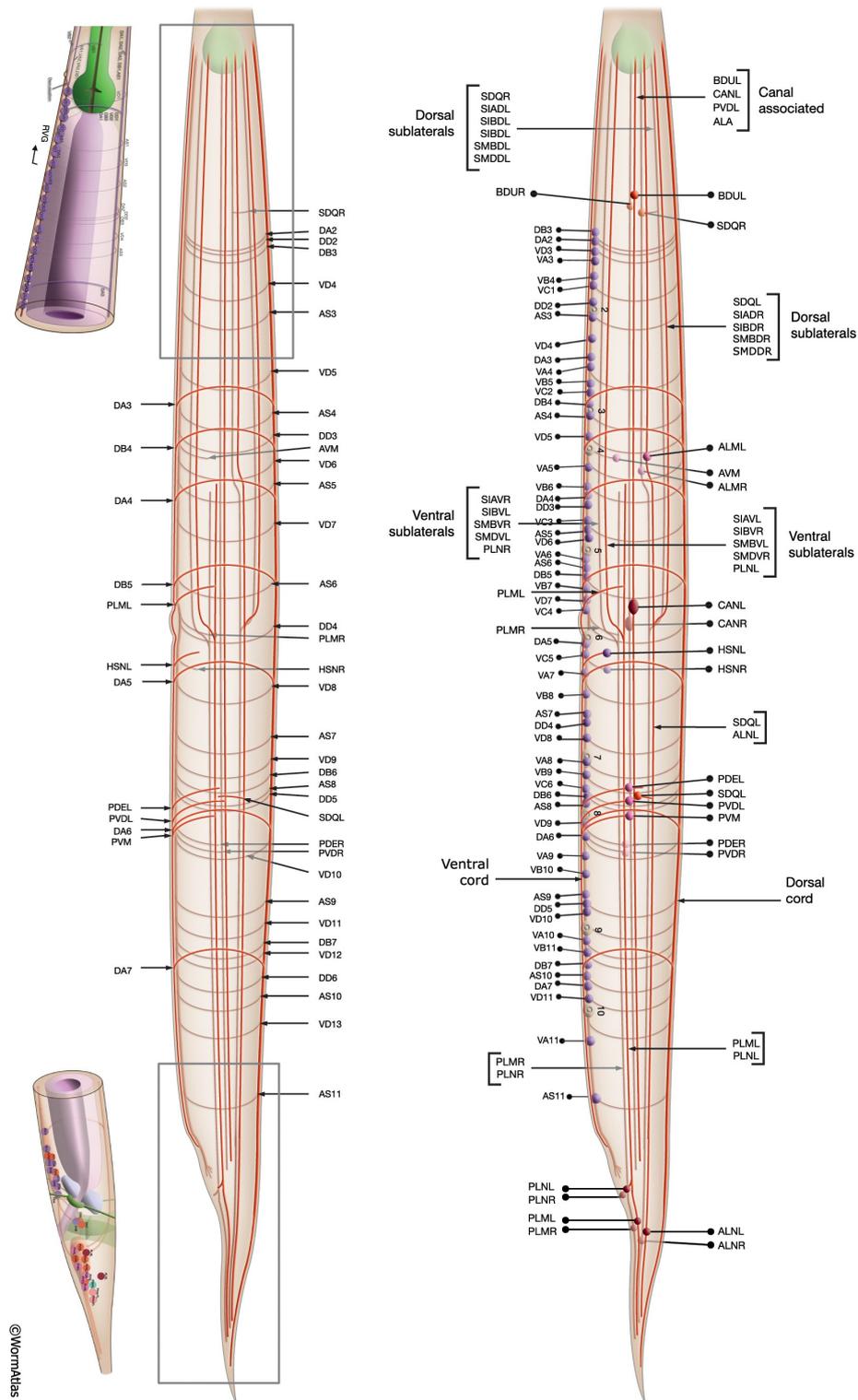


Figure 4.1: Map of neurons with longitudinal nerve tracts and commissures in the body. Image taken from WormAtlas (2013).

analysis to brain connectomes. Varshney et al. highlighted the similarity between the edges of a brain connectome and a network of resistors where a charge could be applied to certain nodes and allowed to dissipate. For 'circuits' represented by small eigenvalues of the Laplacian matrix, the charge will decay slowly after application and, thus, these eigenmodes correspond to long-lived excitations. These long lived excitations would be equivalent to the fastest modes of response in the starling networks, as described in the previous chapter, where the direction of an edge represents the opposite direction of information flow when compared with a connectome.

The eigenvectors of the Laplacian for a strongly connected graph (except for the one associated with λ_1) split the nodes according to the sign of their eigenvector entry. This property of the eigenvectors is what produces the, previously presented, Communities of Dynamic Response (Algorithm 3.3). Varshney et al. (2011) notes that circuits found through experimental studies can be revealed by certain eigenvectors. One such example is displayed in Fig. 4.2 (taken from the paper by Varshney et al. (2011)) where the eigenvector, v_3 , associated with λ_3 highlights the nodes belonging to circuits by their sign and the large magnitude of their v_3 entry. For positive entries, a coupling of chemosensory neurons in the tail (PHBL/R) is found along with interneurons (AVH-L/R, AVFL/R) and motor neurons (VC01-05) involved in egg laying behaviour. These neurons are all weakly coupled to the circuits of chemosensory neurons in the head

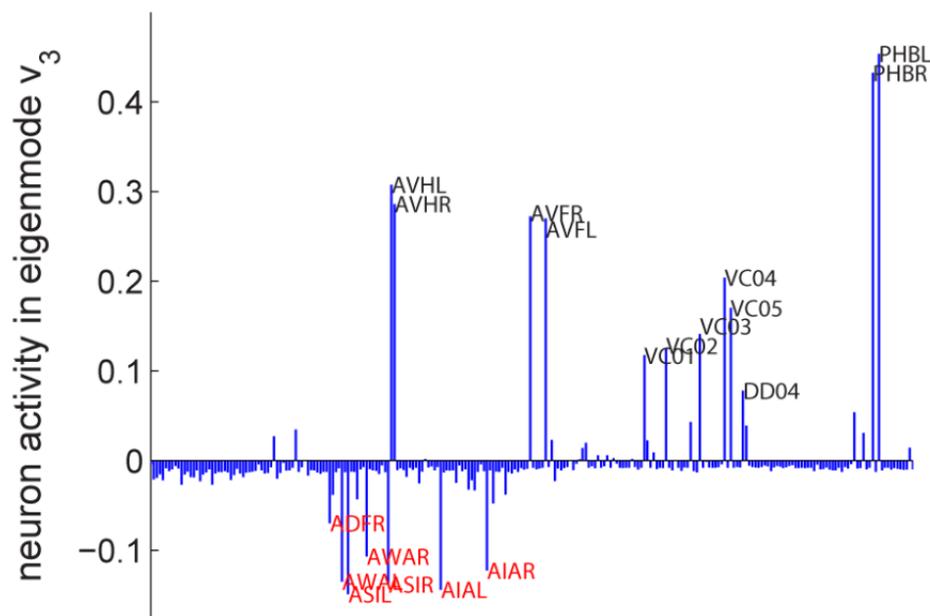


Figure 4.2: Eigenmode (eigenvector) v_3 corresponding to λ_3 . Image taken from Varshney et al. (2011).

(ADFR, ASIL/R, AWAL/R) and other interneurons (AIAL/R) that make a prominent appearance on the negative side.

By employing the Communities of Dynamic Response using three consecutive eigenvectors, as shown in Fig. 4.3, the circuits identified by λ_3 can be clearly defined as belonging to the same circuit/community. The nodes highlighted in Fig. 4.2 have labels attached to them in Fig. 4.3 for clarity.

In both Fig. 4.2 and 4.3 the circuits were clearly identifiable. However, this will not always be the case. Consider λ_6 , see Fig. 4.4, where there appears to be two circuits present. By employing the Communities of Dynamic Response (using λ_5 , λ_6 and λ_7) a more nuanced picture emerges, in Fig. 4.5, where there are over 15 communi-

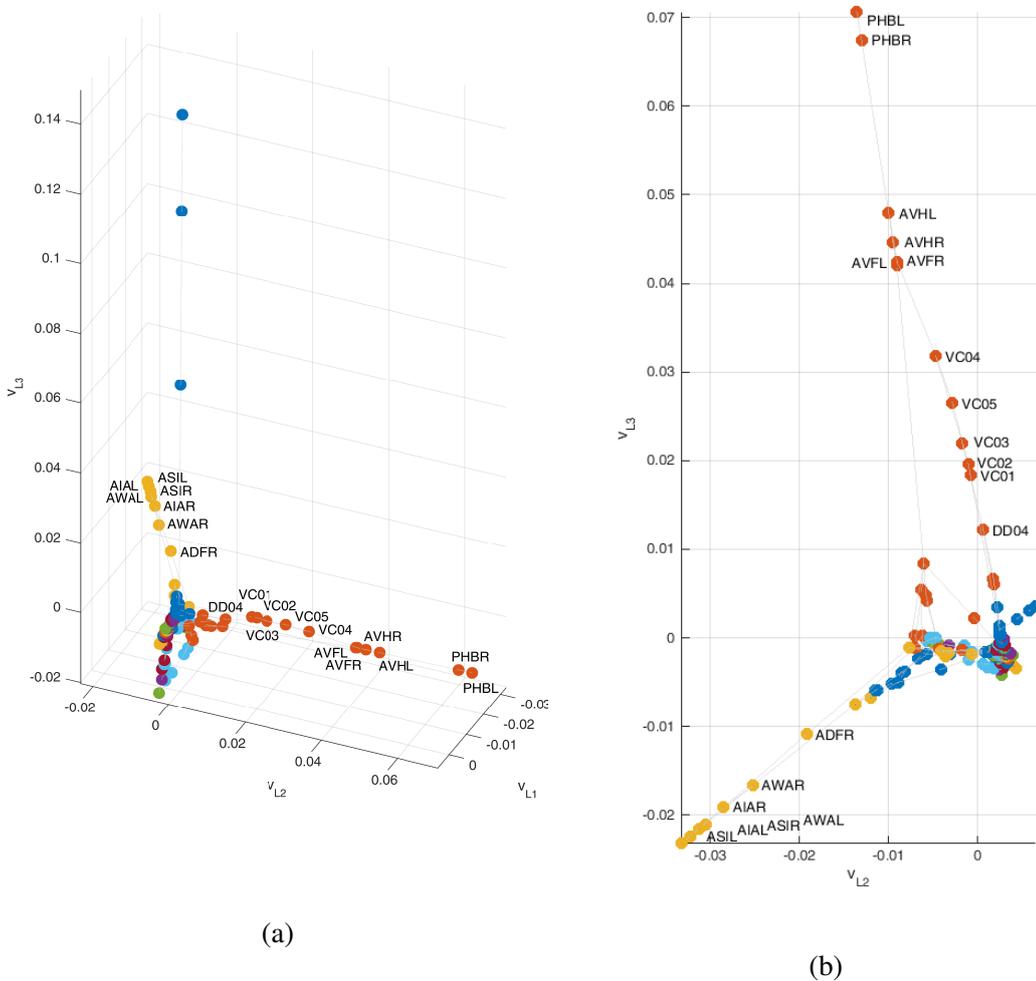


Figure 4.3: Visualisation of node placement in eigenvector space, where v_{L1} , v_{L2} and v_{L3} are associated with λ_1 , λ_2 and λ_3 of the Laplacian matrix for the electrical junction network of the *C. elegans*. Community designation (see Algorithm 3.3) is noted using node colour.

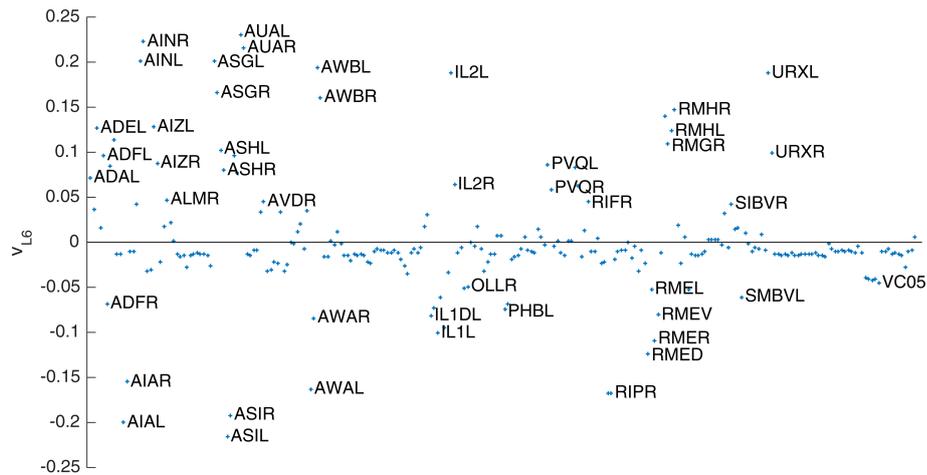


Figure 4.4: Eigenmode (eigenvector) v_6 corresponding to λ_6 .

ties detected. The circuit with chemosensory neurons and interneurons highlighted by λ_3 (ASIL, AIAL, ASIR, AWAL, AIAR, AWAR, ADFR) is still clearly identified in Fig. 4.5.

4.1.2 Bacik’s Flow Model

This section shall compare the results gained from perturbation driven consensus and dynamic response of the Laplacian matrix with that of numerical flow models conducted on the same network. The paper by Bacik et al. (2016) is used for the comparison. Their work took a “dynamics-based (more specifically, flow based) perspective” on the *C. elegans* connectome “rather than focusing on structural features of the network”. This chapter aims to demonstrate that structural information can provide accurate insights into the dynamical behaviour of a network and does, in fact, provide a dynamics-based perspective without the needs for simulation. The aim is to lessen the dependence on numerical simulations that could become computationally exhaustive if considering connectomes where the neurons number in the millions (if not billions). The key to these dynamical insights is in calculating the eigenvectors for a graph, where information about node influence and response can be found.

The model presented by Bacik et al. (2016) uses a continuous-time diffusion process to model the spread of information in the *C. elegans* neuronal network. This approach captures signal diffusion along the directed edges of the connectome with a signal introduced continually to every node in the graph. Each node transmits this signal

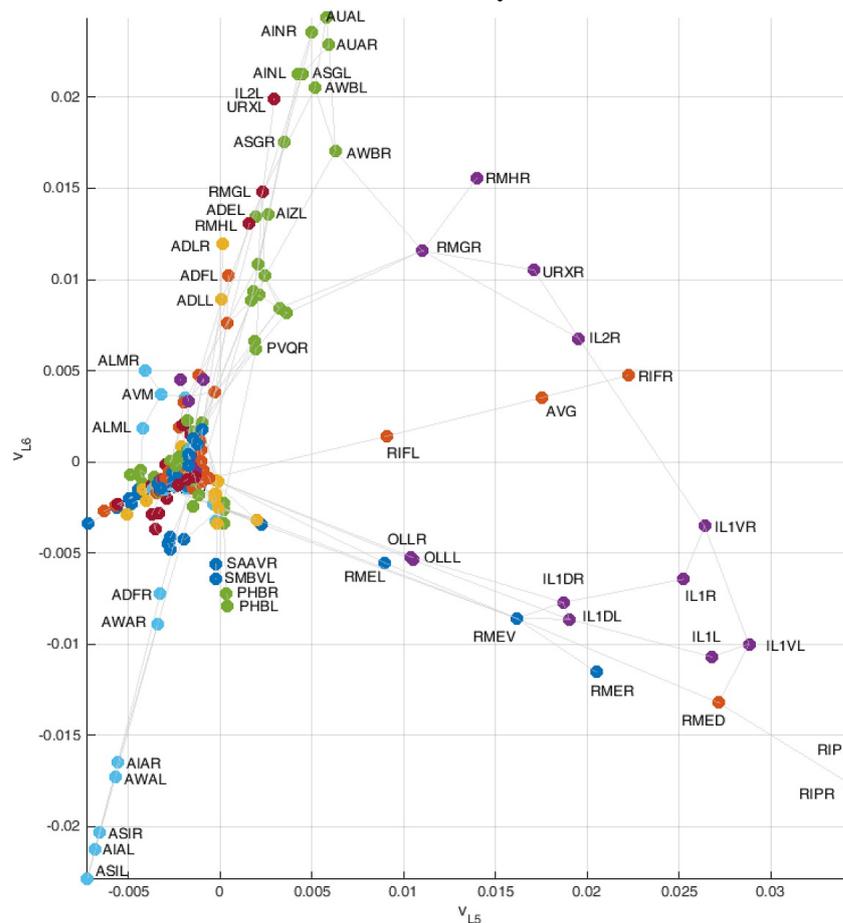


Figure 4.5: Visualisation of node placement in eigenvector space, where v_{L5} and v_{L6} are associated with λ_5 and λ_6 of the Laplacian matrix for the electrical junction network of the *C. elegans*. Community designation (see Algorithm 3.3) is noted using node colour.

along its outgoing edges according to the relative weight of those edges. Signals can escape the network by arriving at sink nodes, which have no outgoing edges only incoming edges. The re-introduction of signals can be understood to represent external stimulus to neurons in the brain.

Bacik et al. investigated flow-based partitions of the connectome using the Markov Stability (MS) framework for community detection. This framework, as well as the diffusion process, shall be described here but for more detail see Bacik et al. (2016).

MS Framework

The MS framework is used to define communities in the graph by examining the diffusion of information based on the framework provided by Bacik et al. (2016). This could be thought of as ink diffusing through the graph; for a completely random graph, where no distinct communities form, the ink would diffuse isotropically and rapidly to an even distribution across all nodes. For a graph, with prominent communities, the ink would be trapped in the communities for longer than would be expected in the random graph. Therefore, the communities, detected in the graph, would depend on time allowed for diffusion. After a short amount of time, small communities will be detected. Given a longer period of diffusion, larger community structures would be revealed. The first analysis presented by Bacik et al. (2016) shall consider five time steps during the diffusion process and, therefore, five different community structures, see Fig. 4.6 where they are referred to as partitions \mathcal{A} to \mathcal{E} .

4.1.3 Single Ablations

The first comparison looks at predicting which nodes are most disruptive when removed from the network. Bacik et al. (2016) revealed the partitions of the *C. elegans* connectome at five different time steps, using the MS framework, as well as uncovering which nodes would be most disruptive to the partition structure at each step.

Fig. 4.6 from Bacik et al. (2016) highlights the nodes that, if deleted/ablated, will have the largest effect on the community (partition) structure present. Since the partitions vary for each time step (\mathcal{A} to \mathcal{E}) the nodes that will be most disruptive, if deleted, vary for each partition structure. The $CV_{[i]}(P)$, used to rank the nodes in Fig. 4.6, represents *community variation* (CV) in the partition structure (P) for a deleted node (i), which was calculated by running the MS framework for all possible single node deletions.

Community Variation

The change induced in the partitions as a result of a node's deletion is defined by Bacik et al. (2016) as the community variation $CV_{[i]}(\hat{P})$. To calculate this, the variation of information, $VI(\hat{P}_a, \hat{P}_b)$, between two partitions, \hat{P}_a and \hat{P}_b is defined by Meilă (2007) as

$$VI(\hat{P}_a, \hat{P}_b) = \frac{2\Omega(\hat{P}_a, \hat{P}_b) - \Omega(\hat{P}_a)\Omega(\hat{P}_b)}{\log(n)}$$

where $\Omega(\hat{P}_a) = -\sum_{\mathcal{C}} q(\mathcal{C}) \log q(\mathcal{C})$ is a Shannon entropy, with $q(\mathcal{C})$ given by the relative frequency of finding a node in community \mathcal{C} in partition \hat{P}_a ; $\Omega(\hat{P}_a, \hat{P}_b)$ is the Shannon entropy of the joint probability; and the factor $\log(N)$ ensures that the measure is normalised between $[0, 1]$. The community variation is then defined as

$$CV_{[i]}(P) = \min_{\tau} VI(\hat{P}, \hat{P}_{[i]}(\tau))$$

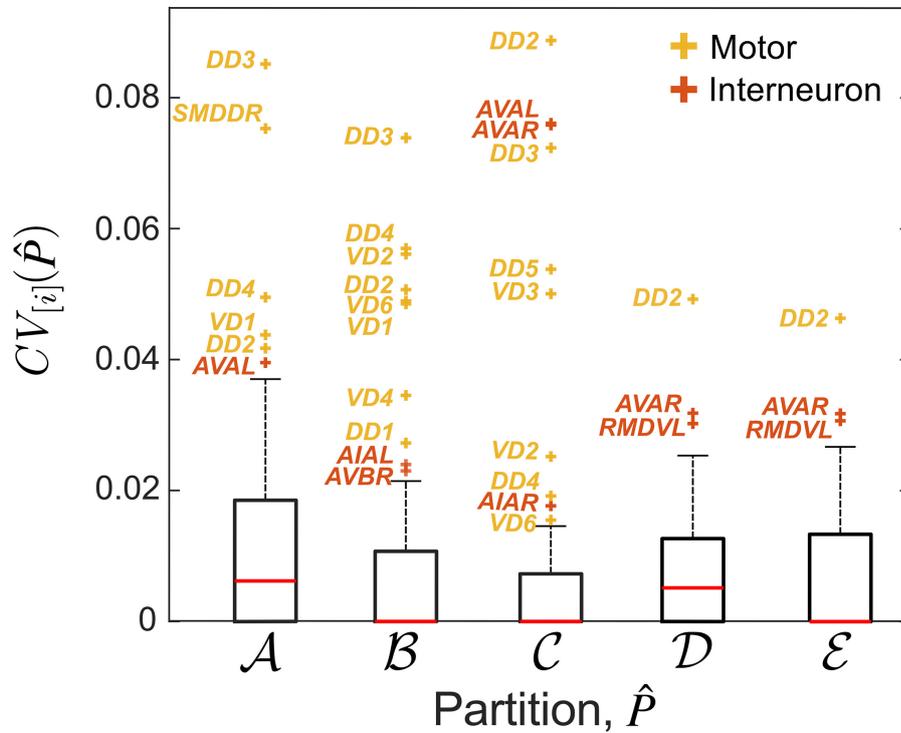


Figure 4.6: Effect of single ablations on the make-up of partitions \mathcal{A} through to \mathcal{E} . Figure taken from Bacik et al. (2016).

where \hat{P} are the partitions found for the unaltered graph and $\hat{P}_{[i]}(\tau)$ is the most similar partition achieved for the graph with node i removed. The community variation values are shown in Fig. 4.6 for the nodes that cause the most disruption.

This community variation analysis focused on the robustness of the graph to node deletion, which is a different concept from that of detecting main bottleneck nodes for information flow in a graph. However, the comparison is still useful for validation as these bottleneck nodes are well connected to the majority of the graph and, therefore, will be the nodes that cause the largest disruption to the partitions in the graph if removed. This claim is supported by Fig. 4.7 where all of the nodes that had the largest effect on partitions \mathcal{A} to \mathcal{E} , as detailed in Fig.4.6, are highlighted in red. Fig. 4.7 (b) shows that the red '+'s are mainly those with the highest v_{L1} values when compared with nodes of the same v_{L2} value. The prominent red '+' nodes in Fig. 4.7 (a) and (b) are all, except one, motor nodes that have the largest effect on partitions \mathcal{A} to \mathcal{E} . Looking at the detail on Fig. 4.7 (c) it can be seen that SMDDR is connected directly to the most prominent dynamic response community, shown in dark blue in Fig. 4.7 (d). This may explain why SMDDR was found to be more disruptive than DA08 (see Fig. 4.7 (c)), despite DA08 possessing a higher v_{L1} value.

The only interneuron that displays prominently, according to v_{L1} , is RMDVL. The other nodes are located near the origin of the plot. The role of the interneurons are to pass information on, therefore, it is not surprising that they do not appear prominently according to v_{L1} . The results from Bacik et al. were for single ablations that cause partition disruption with interneurons likely to disrupt partitions as they act as the bridge between different types of neurons. In fact, the disruptive interneuron nodes AVBR, AVAR and AVAL are found to be the 2nd, 3rd and 4th highest ranked according to the betweenness centrality metric (that quantifies how often a vertex/node acts as a bridge along the shortest path between two other nodes). AIAL and AIAR are disruptive interneurons that neither have high betweenness scores or v_{L1} values. It is proposed that AIAL and AIAR are key nodes in a small but prominent network community, that displayed prominently in Fig. 4.3 and 4.5 when looking at only the electrical junction network. Therefore, their removal is likely to cause significant localised disruption.

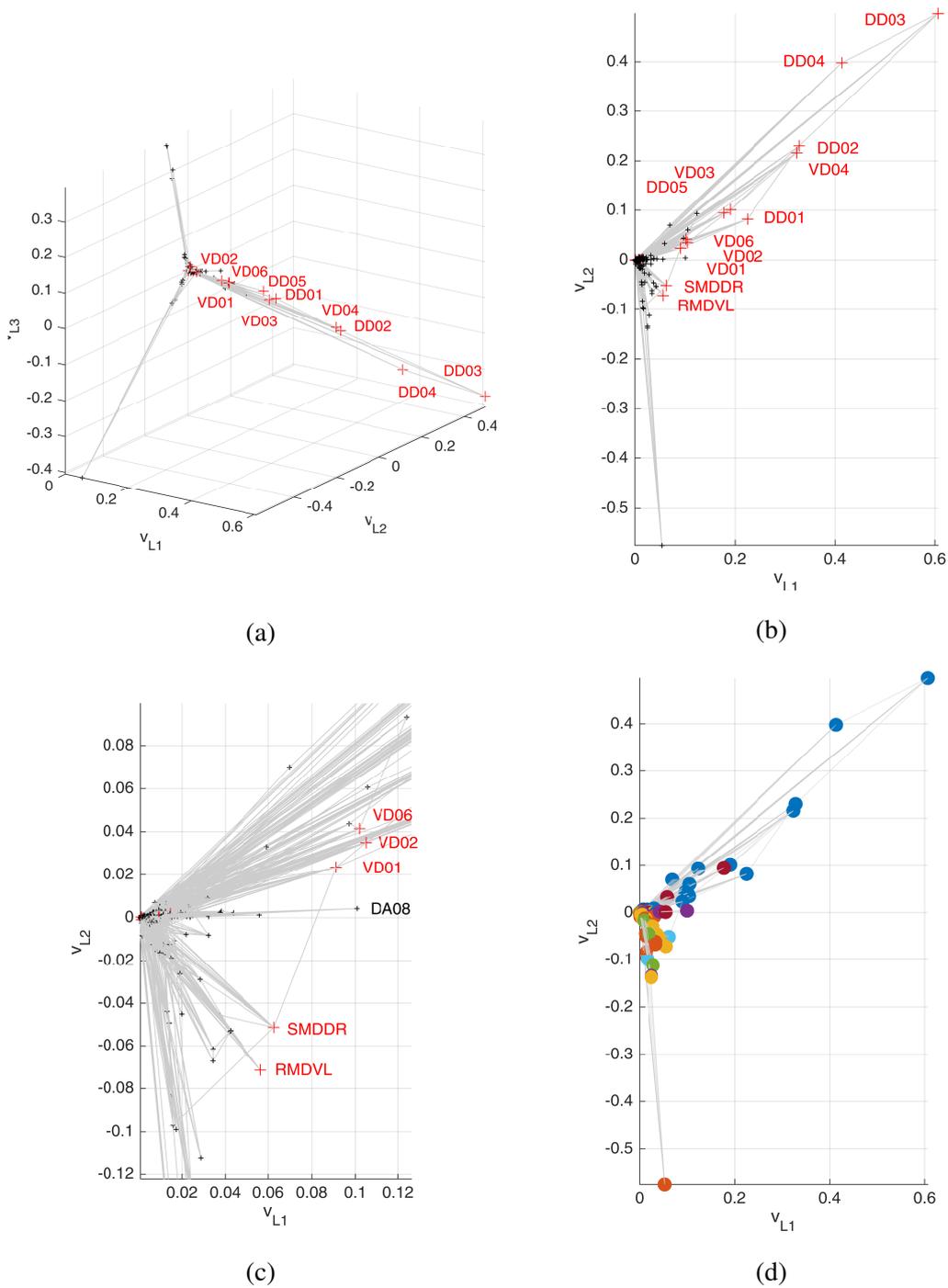


Figure 4.7: Red '+'s and labels mark the nodes, in Fig. 4.6, that caused major disruption to the partitions \mathcal{A} to \mathcal{E} . Black '+'s mark the other nodes in the network. (a) v_{L1} , v_{L2} and v_{L3} are the first left eigenvectors. (b) v_{L1} against v_{L2} (c) zoomed in section (d) the Communities of Dynamic Response that a node belongs to is denoted with the colour of its dot.

4.1.4 Signal Diffusion

The introduction of a simulated signal to certain nodes in the *C. elegans* graph shall now be considered. Perturbation driven consensus was a major theme of the previous chapter and the introduction of a signal is similar to applying a perturbation to a graph, therefore similar methods can be called upon again. Fig. 4.8 was taken from the same Bacik et al. (2016) paper and shows the signal spreading from different sets of starting nodes. These starting nodes are as follows:

- (i1) PDEL, PDER, PLML, PLMR, PVDL, PVDR
- (i2) ADEL, ADER, ALMR, AQR, AVM, BDUL, BDUR, FLPL, FLPR, SDQL, SIADL, SIADR
- (i3) PHAL, PHAR, PHBL, PHBR
- (i4) ADLL, ADLR, ASHL, ASHR, ASKL, ASKR

with the nodes showing the strongest response to the signal highlighted and labelled.

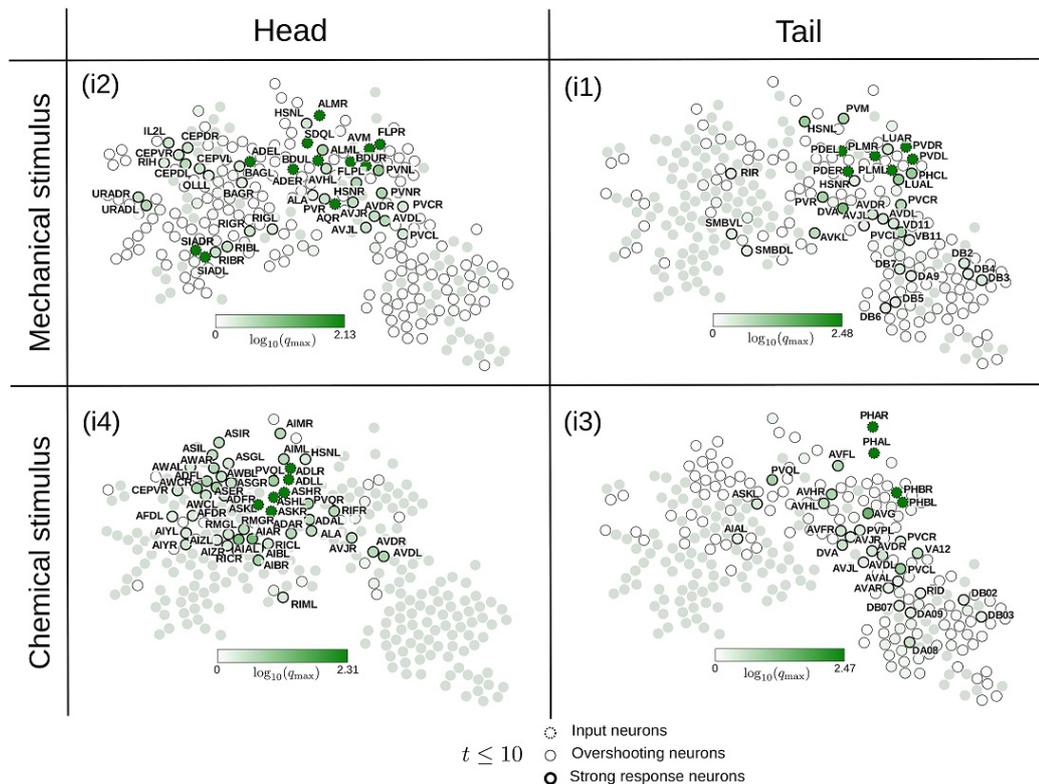


Figure 4.8: Depiction of a simulated signal propagating through the *C. elegans* connectome from differing source neurons. Figure taken from Bacik et al. (2016).

The approach taken throughout this dissertation has to be tweaked for this scenario. Previously the eigenvectors were used to detect the most effective influencers in the network, but now they are being employed to find those who are most influenced by the nodes receiving a perturbation. Therefore, it is the eigenvectors of an altered Laplacian (L), where the diagonal matrix D in $L = D - A$ is created by summing the columns (not the rows as previously seen) as $d_i = \sum_j a_{ij}$. Another alteration of the Laplacian is to add a perturbation (p) to certain elements of D , therefore $d_p = \sum_j a_{ij} + p$ where d_p represents a perturbed element.

The effect of adding a perturbation to a diagonal element is to reduce the value of that entry according to the first left eigenvector (FLE), for an explanation of how p influences the FLE see Theorem 3.3. The FLE shall be referred to as v_{IL1} here as the elements of the diagonal matrix are equal to the sum of the indegrees for each element (plus any perturbations applied). This reduction in v_{IL1} entries will have a knock-on effect for any node connected to a perturbed node, with the largest reductions seen for those nodes that are most heavily influenced by the perturbed nodes. This is true for the FLE of this perturbed Laplacian when the degree matrix depends on the outdegree or indegree. But in the case of indegree, the FLE of the unperturbed Laplacian is a uniform vector. Therefore, any loss in eigenvector entry value for either the perturbed nodes or nodes strongly influenced by those perturbed nodes shall be clearly shown. This is displayed in Figure 4.9 where the red 'x's mark nodes, highlighted in Figure 4.8, that expressed the strongest response to the introduced signal.

In Fig. 4.10 and 4.11 a magnified section from the plots in Fig. 4.9 are displayed. From these figures it can be seen that Bacik et al.'s results do not match completely with those produced by the eigenvector approach. Most of the strongest responders are identified correctly, but there are some black 'x's that are more prominent than would be expected and a few red 'x's that have not seen much of a reduction in their v_{IL1} value. Instead the eigenvector approach builds on Bacik et al.'s result to provide further insights. In particular, it identifies whether strongly responding nodes are part of local circuitry or well connected to the rest of the network. The red 'x's that have not deviated far from their uniform eigenvector value are those that respond strongly to the signal introduced but are also strongly connected to other pathways in the network. Whereas the red 'x's with clearly reduced v_{IL1} values must be heavily influenced by the signal source nodes and so be part of an information pathway with these source nodes at the start.

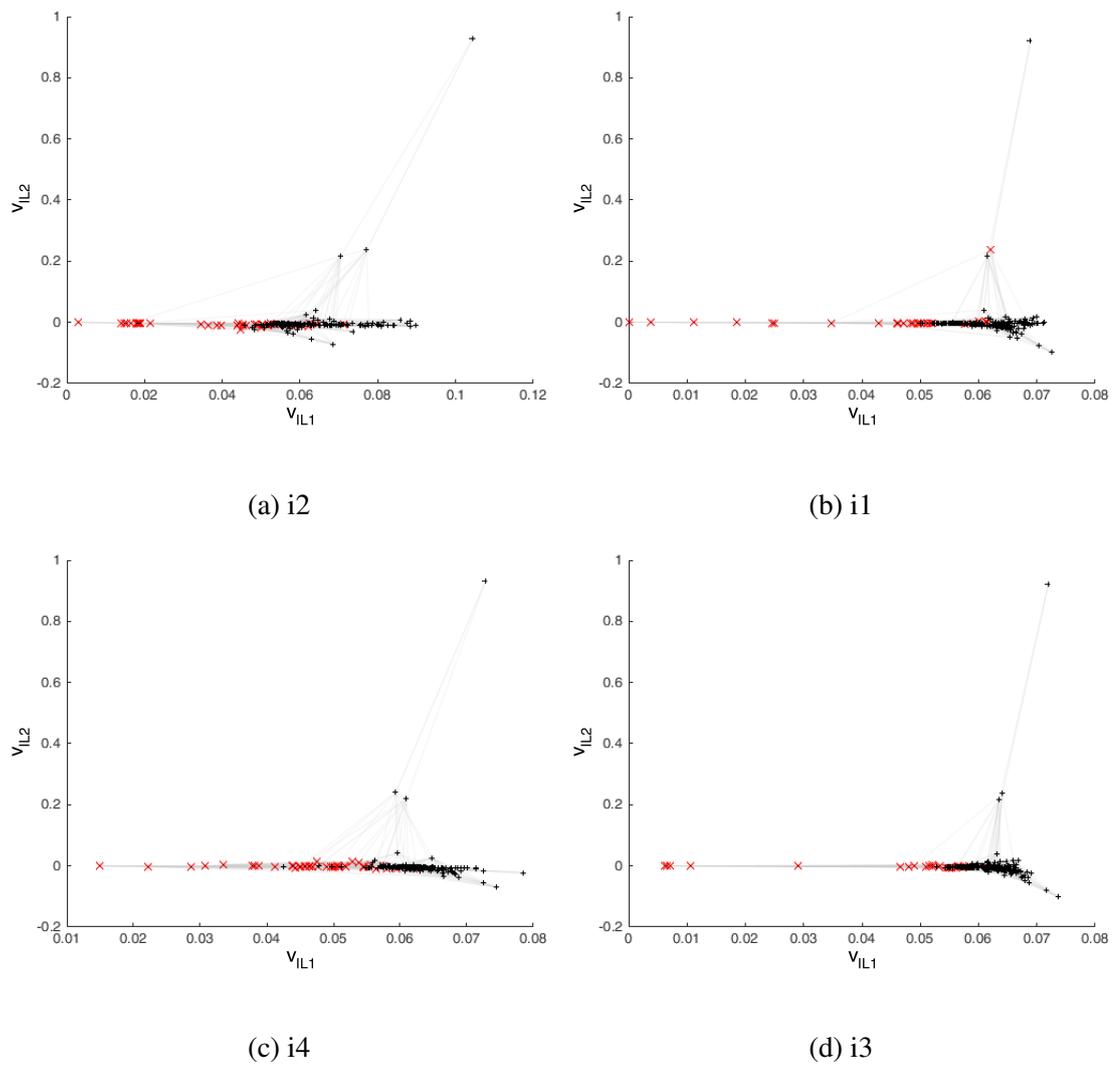
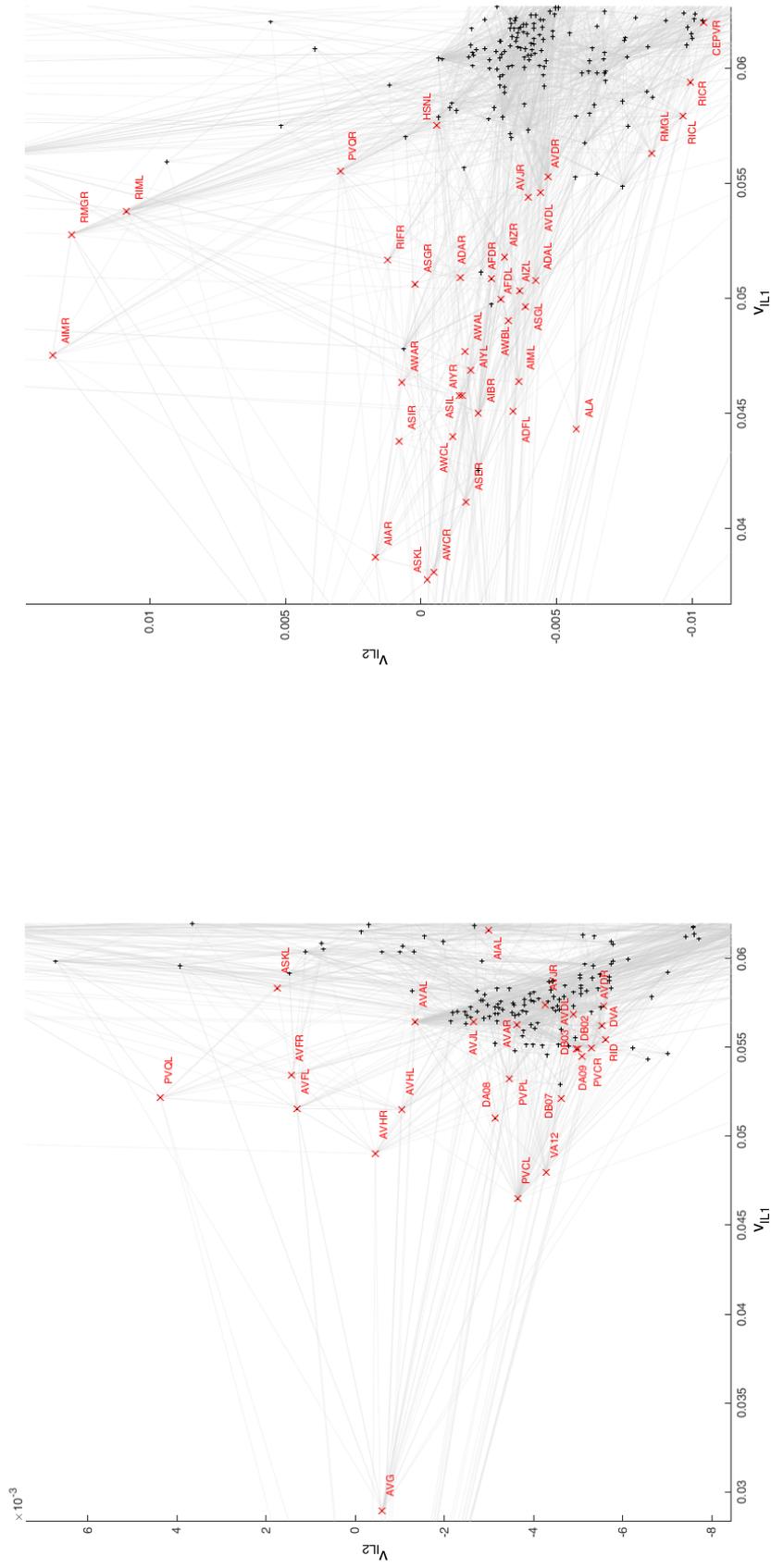


Figure 4.9: Response of *c. elegans* connectome to a perturbation where red '+'s mark nodes highlighted in Fig. 4.8. Black '+'s mark the other nodes in the network.



(a) i4

(b) i3

Figure 4.11: Response of *c. elegans* connectome to a perturbation where red ‘x’ mark nodes highlighted in Fig. 4.8. Black ‘+’ mark the other nodes in the network.

4.2 Macaque

The next subject of study, the macaque, has a substantially larger connectome than the *C. elegans* and is a stepping stone towards considering a human connectome. Macaque brain networks are readily available, which has led researchers to take up the task of applying graph theory to these connectomes in an attempt to better understand the brain's function. This section will continue to demonstrate the efficacy of the Communities of Dynamic Response method, in response to the claimed limitations of using a graph theoretic approach to investigate information flow.

The paper by Mišić et al. (2014) stated that the hippocampus (CA1) of the macaque had long been known to neuroscientists to hold an influential role in the brain's decision making architecture. This region has been found to play an important role in memory function as found by Squire (1992) and Moscovitch et al. (2005) where it is most influential during the initial phase of memory creation. Mišić et al. notes that the hippocampus is particularly noteworthy for its ability to form connections between arbitrarily different external events, which highlights its role as a hub for information from around the brain. Mišić et al. cites a number of graph theory based studies that have failed to identify the hippocampus as an important hub, see Gong et al. (2008); Hagmann et al. (2008); van den Heuvel et al. (2012) for humans and Harriger et al. (2012); Honey et al. (2007); Modha and Singh (2010); Sporns et al. (2007) for the macaque. Stating that these “*analyses of anatomical and functional whole-brain networks have largely failed to demonstrate the topological centrality of the hippocampus.*” The conclusion of this study was that “*the functional capacity of a given region or subnetwork cannot be fully discerned by only analyzing the static structural connectivity of the brain*” *. Mišić et al. argues that the hippocampus is critical as it is a bottleneck in the network through which information is funnelled. A definition that is similar to the description, earlier in this chapter, for what the most prominent nodes according to the FLE represent.

4.2.1 Hippocampus in the Queuing Network

Kötter (2004) created the CoCoMac database, which included a macro level map of the macaque cortical (outer layer of the cerebrum) connectivity. This connectome was an

*Functional capacity is understood to be an assessment of how effectively a region can carry out a given function. In the case of the hippocampus, this would be an assessment of its ability to receive information from across the whole network.

aggregated network created from several hundred published axonal tract-tracing studies. Tract-tracing is the most popular method for mapping connections in the brain, where a fluid (such as silver) is carefully observed as it traverses the connections between neurons to create a map of connections. The network nodes are defined as neuronal areas with the connections between neurons assimilated to provide a macroscopic picture of information flow. The version of this connectome used here and by Mišić et al. contains 242 nodes where each edge has the same weighting.

The information flow was modelled as a discrete-event queueing network by Mišić et al. (2014). This queueing network functions by continually generating signal units, at randomly-selected grey matter nodes in the network, and assigning these units with randomly-selected destination nodes. The signals then diffuse through the network via white matter projections (edges). Grey matter nodes are modelled as servers with a finite buffer capacity, such that if a signal unit arrives at an occupied node, a queue will form. Upon reaching its destination node, the signal unit is then removed from the network. Mišić et al. used this numerical model to show that the hippocampus (CA1) is a central hub according to this aggregated network. Demonstrating that the CA1 experiences a high throughput of signal traffic that places it in the top 3% for the total number of signal units that arrive at a node, the mean number of signal units at a node and the proportion of time a node is occupied by signals.

4.2.2 An Eigenvector Perspective

By examining the eigenvectors of the Laplacian matrix for the CoCoMac connectome, the influence of the hippocampus (CA1) can also be revealed. First, the influential points of information convergence in the CoCoMac network are identified in Fig. 4.12. The most prominent areas are the TFM and TFL with CA1 appearing, on the surface at least, to be unremarkable with a relatively small v_{L1} . However, a graphic from Mišić et al. (2014) demonstrates that this finding is in fact a confirmation of CA1's importance, see Fig. 4.13. Of the three outgoing connections, from TFL and TFM, two of them connect to CA1. This is an important finding as this graph theoretical approach has identified not only the importance of TFL and TFM as collators of information but also CA1, which receives the output of this collation. The eigenvector approach reveals that CA1 does not, in possible contradiction to Mišić et al. (2014), act as a bottleneck but rather processes the collated data from TFM and TFL that appear to be the true bottleneck nodes. This finding is significant as it demonstrates that it is possible to

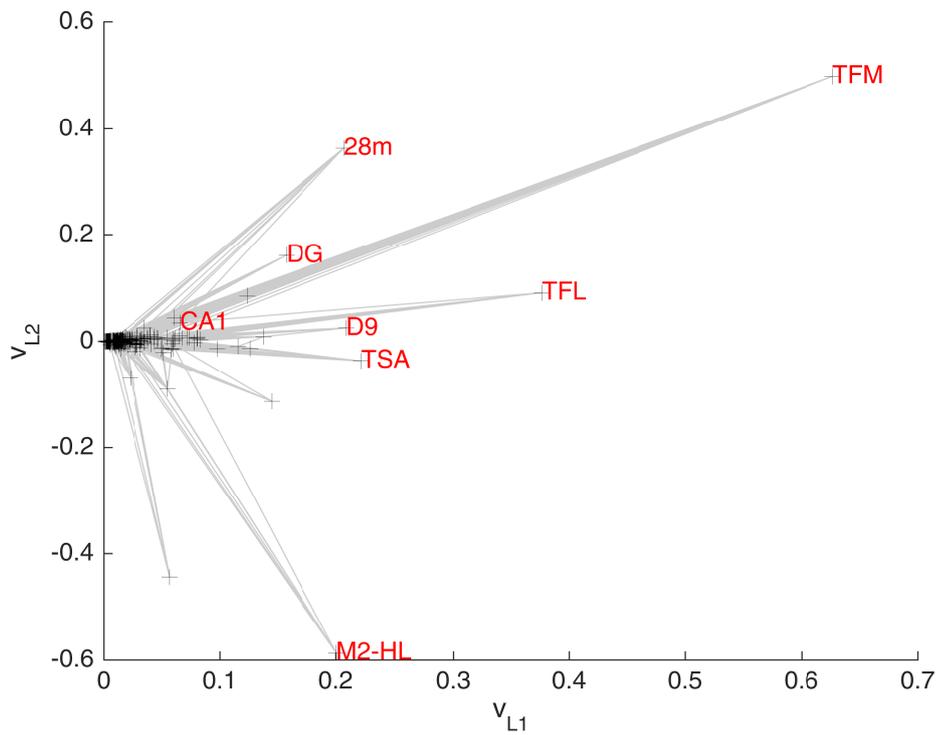


Figure 4.12: v_{L1} and v_{L2} are the first left eigenvectors. Each '+' represents a neuronal area with some prominent areas marked on the plot.

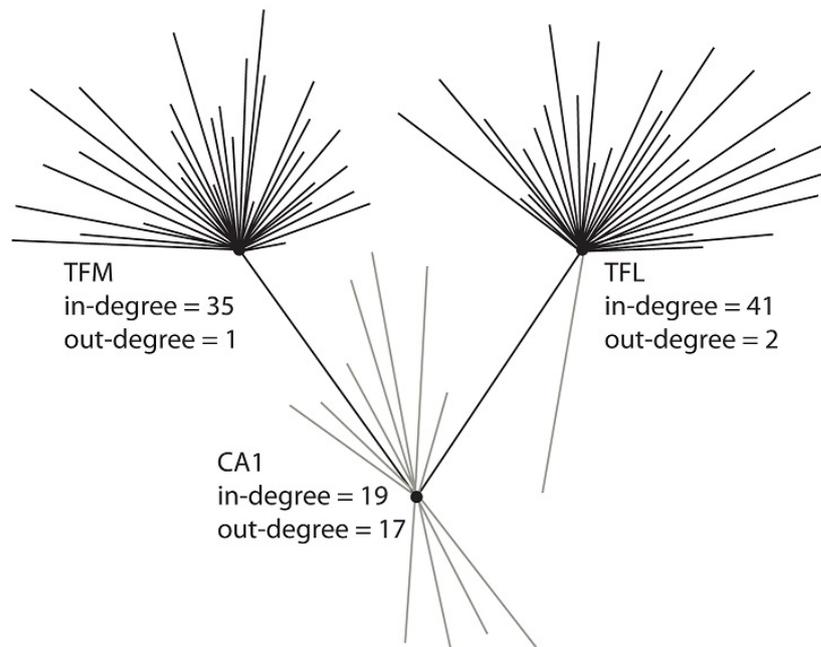


Figure 4.13: Outgoing and incoming connections from the TFL, TFM and CA1 neuronal areas. Image taken from Mišić et al. (2014).

detect CA1's influence and importance by considering only structural connectivity of the brain.

Detecting CA1's influence using eigenvectors has been shown to be possible, but it required some investigation to look beyond TFM and TFL. However, this approach can be extrapolated to other nodes and connectomes reducing the need to numerically investigate information flow. Extending the intuitions gained to the other prominent nodes in Fig. 4.12 results in Table 4.1 where each highlighted prominent node (PN) is detailed alongside an associated outgoing connection node (OCN) or nodes that lie at the end of the PN's outgoing edge(s). These OCNs, like the CA1, have an important role in the macaque brain and are likely to have more influence than the PNs (the bottleneck nodes). One clear example is the M2-HL as the PN that is connected to the M1-hl as the OCN. In this case the supplementary motor cortex is acting as the bottleneck that then provides information to the primary motor cortex, which is the most influential region for motor control.

Table 4.1: Prominent nodes from Fig. 4.12.

Prominent Node (PN)		Outgoing Connection Node (OCN)	
Acronym	Merged Brain Region	Acronym	Merged Brain Region
TFM	Temporal area TF (medial part)	CA1	Hippocampus
TFL	Temporal area TF (lateral part)	CA1, Pros.	Hippocampus, Prosubiculum
TSA	Transitional sensory area	23c, 31, PECg	Area 23c, Area 31, Parietal area PE (cingulate part)
D9	Dorsal area 9	32, 14, M9	Area 32, Orbitofrontal area 14, Medial area 9
28m	Medial entorhinal cortex	TG	Temporopolar area TG
M2-HL	Supplementary motor cortex M2, hindlimb area	M1-HL	Primary motor cortex M1, hindlimb area
DG	Dentate gyrus	ENT	Entorhinal cortex

By representing the nodes with the first two, left, eigenvectors the prominent nodes are identified in Fig. 4.12. A large indegree would be one intuitive identifier of prominent nodes, however of the highlighted nodes in Fig. 4.12 the TFL node has the highest indegree but is only 20th highest when considering the whole network. The outdegree

is also a critical factor, as a node with a high outdegree and indegree will pass on much of the information it receives and not act as a prominent node bottleneck. This claim is supported by the ratio of outdegree to indegree ($O : I$) being significantly higher for TFM and TFL than any other node, where the ratio is 34 and 20.5 respectively. The next closest nodes, when sorted by $O : I$ ratio, are as follows: DG, 28m, D9, TSA, and M2-HL. These nodes are clearly prominent in Fig. 4.13 but the order according to the $O : I$ ratio differs from v_{L1} . In summary, the PNs are likely to have a high indegree and low outdegree, which is why there are few OCNs per PN in Table 4.1.

The paper by Mišić et al. (2014) also lists the most traversed connections in the graph, which are detailed in Table 4.2 with the v_{L1} ranking of the nodes, at either end of the connection, included. It is evident from the table that the outgoing connection node is ranked highly, and for the most part in order, according to v_{L1} . This change in order, according to v_{L1} , for the outgoing nodes could be attributed to node D9 having two highly traversed edges. Whilst the lower ranked neuronal area 28m only has the one and so more traffic accumulates on that edge. Table 4.2 lends further support to the claim that the nodes ranked by v_{L1} can be viewed as bottlenecks for information from across the whole network. The PN nodes are at the source of all of the largest information carrying edges in the network, indicating that the majority of the network's information passes through these PN nodes.

The limitation of both the analysis conducted by Mišić et al. (2014) and that done in this section is that the connectome used does not contain any edge weighting information. It is likely that the connections between regions in the Macaque brain have

Table 4.2: Most traversed edges with node v_{L1} rankings.

Traversed edge ranking	Edge Description	Outgoing node v_{L1} ranking	Incoming node v_{L1} ranking
1	TFM to CA1	1	22
2	TFL to CA1	2	22
3	TFL to Pros	2	50
4	TSA to 31	3	12
5	TSA to 23c	3	29
6	TSA to PECg	3	13
7	28m to TG	5	49
8	D9 to 14	4	47
9	D9 to 32	4	41
10	M2-HL to M1-HL	6	33

differing weights and including this information may alter the results. For example, if the edges listed in Table 4.2 had a larger weight and could carry more information then the PNs would no longer be bottlenecks for information in the network and it is likely that the most prominent according to the first eigenvector would be the OCNs. Since the OCNs would now become information bottleneck points.

4.3 Human

The final subject of study is the human brain where vast scale does not inhibit the brain from responding rapidly to stimulus that enables us, as humans, to react to the world around us. The effectiveness of analysing brain connectomes, with the first eigenvectors of the network, has been presented in the previous sections. The first eigenvector was shown to identify the main information collating nodes - for the *C. elegans* and the macaque - with these claims validated through comparison with previously conducted numerical flow-based analyses. The development of this analytical approach enables analysis of huge networks where numerical flow models would require far more processing power to function.

The first brain considered is that of a 28 year old, right-handed, female (subject ID 113 from Landman et al. (2011)). The connectomes of her brain were generated by Roncal et al. (2013) who produced a series of undirected, 1.8 million node, networks of the human brain from the Magnetic Resonance Imaging (MRI) scans carried out by Landman et al. (2011). The nodes/vertices are defined as the intersection points on a three dimensional grid where each point is 1 mm apart from its neighbours. The edges of the network are defined as any two vertices that are connected by at least a single fibre. The greater the number of fibres between any two vertices, the larger the weight of that edge. The results from two brain scans on subject 113 are considered here where an edge weight of 1 represents a single fibre connection and the highest edge weighting observed was 1749 in scan 2.

Directionality of edge connections has been key to the previous network analyses, conducted in this dissertation, where the left eigenvector has been effective in highlighting influential nodes and pathways. For example, in the macaque case, Mišić et al. (2014) shows that if the directionality of edges are reversed the hippocampus - shown to be a hub for information - is no longer congested with information traffic. For human connectomes, there are few sources of directed graphs and none found on the scale

of millions of nodes. Therefore, this section performs analysis on an undirected connectome. This may not produce as accurate an insight into brain functionality, which is known to use directed transfer of information, as it would using a directed connectome. But, as was displayed previously with the investigation of *C. elegans* and their electrical junction network (see Section 4.1), insights can still be gained including the detection of influential circuits in the brain.

4.3.1 Spectral Analysis

Although there were 1.8 million nodes in the network, not all of these are part of the largest connected component, with approximately 0.9 million nodes for scan 1 and 0.65 million for scan 2 being part of this connected component. Spectral analysis on the Laplacian matrix of such a network would have produced multiple zero eigenvalues, and associated first eigenvectors. One for each connected component in the graph. Hence, it would have been easier, when using the Laplacian for this analysis, to remove the nodes not involved in the largest connected component before calculating the eigenvectors. But by performing the spectral analysis on the adjacency matrix instead, no nodes have to be removed since the prominent eigenvectors of the adjacency and Laplacian matrices are the same when dealing with an undirected network and the first eigenvector is clearly distinguishable.

For the adjacency matrix, the first eigenvector is associated with the eigenvalue of largest magnitude. This first eigenvector has the notable characteristic of only containing positive entries. The following eigenvectors usually contain a fairly even distribution of positive and negative values as has been demonstrated through eigenvector-based plots through out this dissertation (see Fig. 4.2 and 4.4 as an example). However, in the case of the human connectome the results presented differently from previous analyses in Chapter 3. As some of the eigenvectors, associated with large eigenvalues but not the largest, have structures that closely resemble the first eigenvector. These *false* first eigenvectors have large entries that are primarily either positive or negative with the entries of the opposite sign having far smaller values. It is possible that these *false* first eigenvectors identify regions that are well connected within the connectome but are poorly connected to the actual first eigenvector. The following sections shall support the assertion that these *false* first eigenvectors are regions of high influence, which correspond to areas of high brain activity.

To determine the prominent communities, present in the first eigenvectors, the Communities of Dynamic Response method is employed (Algorithm 3.3). Examples of this are displayed in Fig. 4.14 where the majority of first eigenvectors appear like Fig. 4.14 (b) with the orange nodes selected as members of the prominent community. In Fig. 4.14 (a) the most prominent community is less clear cut, therefore nodes from the green, maroon and purple nodes are selected as they contain nodes with the highest v_1 values.

4.3.2 Subject 113

4.3.2.1 Scan 1

Members of the most prominent communities according to each of the first eigenvector and the *false* first eigenvectors are presented by mapping them onto a 3D representation of the brain in Figure 4.15. The top ten eigenvectors were considered from brain scan 1 with the first, second, fourth, fifth, eighth and ninth shown in the figure. The third, fourth, seventh and tenth are not displayed as they had large entries of both signs and, hence, are not *false* first eigenvector but can be used to highlight the community structure of the first eigenvectors, as in Fig. 4.14.

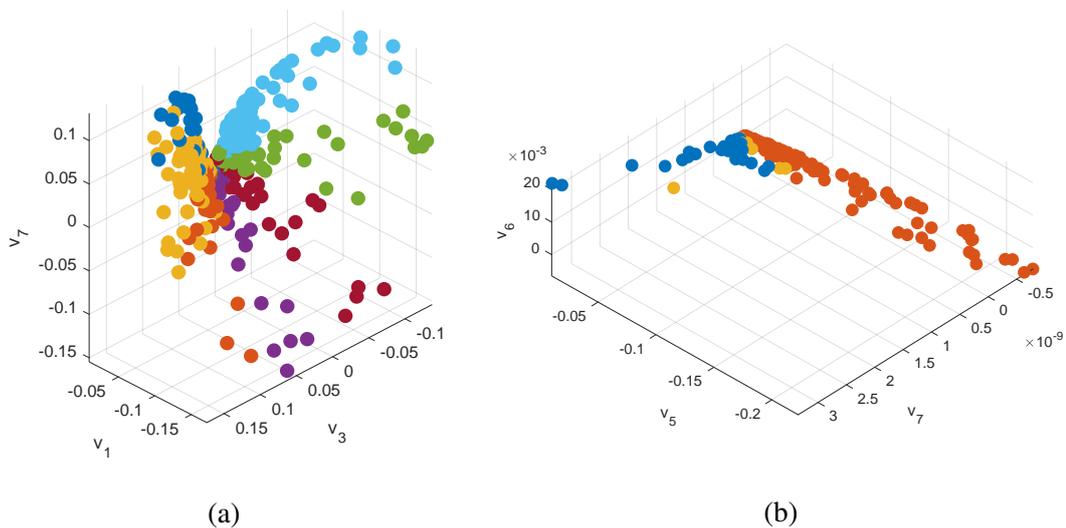


Figure 4.14: Eigenvector plots for subject 113 where (a) uses the first, third and seventh eigenvector to highlight the community structure of the first eigenvector and (b) employs the fifth, sixth and seventh eigenvector to highlight the community of the fifth eigenvector that is a *false* first eigenvector. The Communities of Dynamic Response are denoted by the colour of the node's dot.

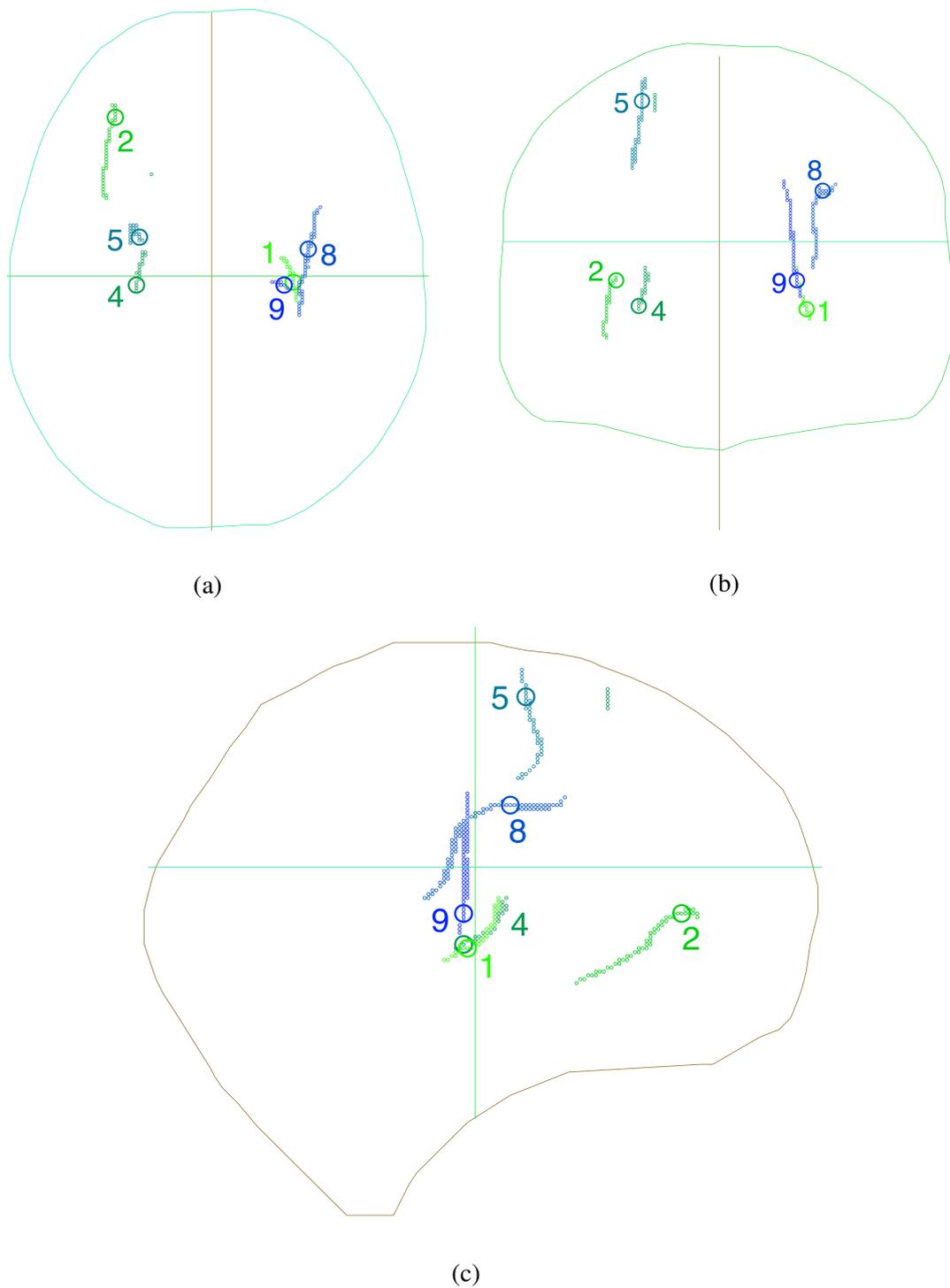


Figure 4.15: Subject 113 - scan 1: brain represented by x,y,z outlines from a surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,4,5,8,9) are displayed with a spectrum of colours from green (1) to blue (9). The most prominent node for each eigenvector is marked with a circle. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled.

It is worth considering what these *false* first eigenvectors represent. Due to their large associated eigenvalues they must indicate pathways in the brain where a large amount of information is passing through. This hypothesis is supported by Figure 4.16 where the 3500 largest weighted edges from brain scan 1 are overlaid on the first eigenvector and *false* eigenvector nodes. The eigenvectors all overlap with the highest weighted edges, although this overlap is reduced for the less influential eigenvectors 8 and 9. In 8 there is still overlap with large edges but it is minimal and mainly centred on the location of the node with the highest valued entry for v_8 . Figure 4.16 appears to indicate that clusters of high weight edges are markers for influential regions with the eigenvectors providing further, more quantitative, insights by ranking these regions in terms of influence. It is also worth noting that for a directed connectome the edge weight is only half the picture as some of the influential regions might be sources or sinks for information but this can only be determined with directed edge data.

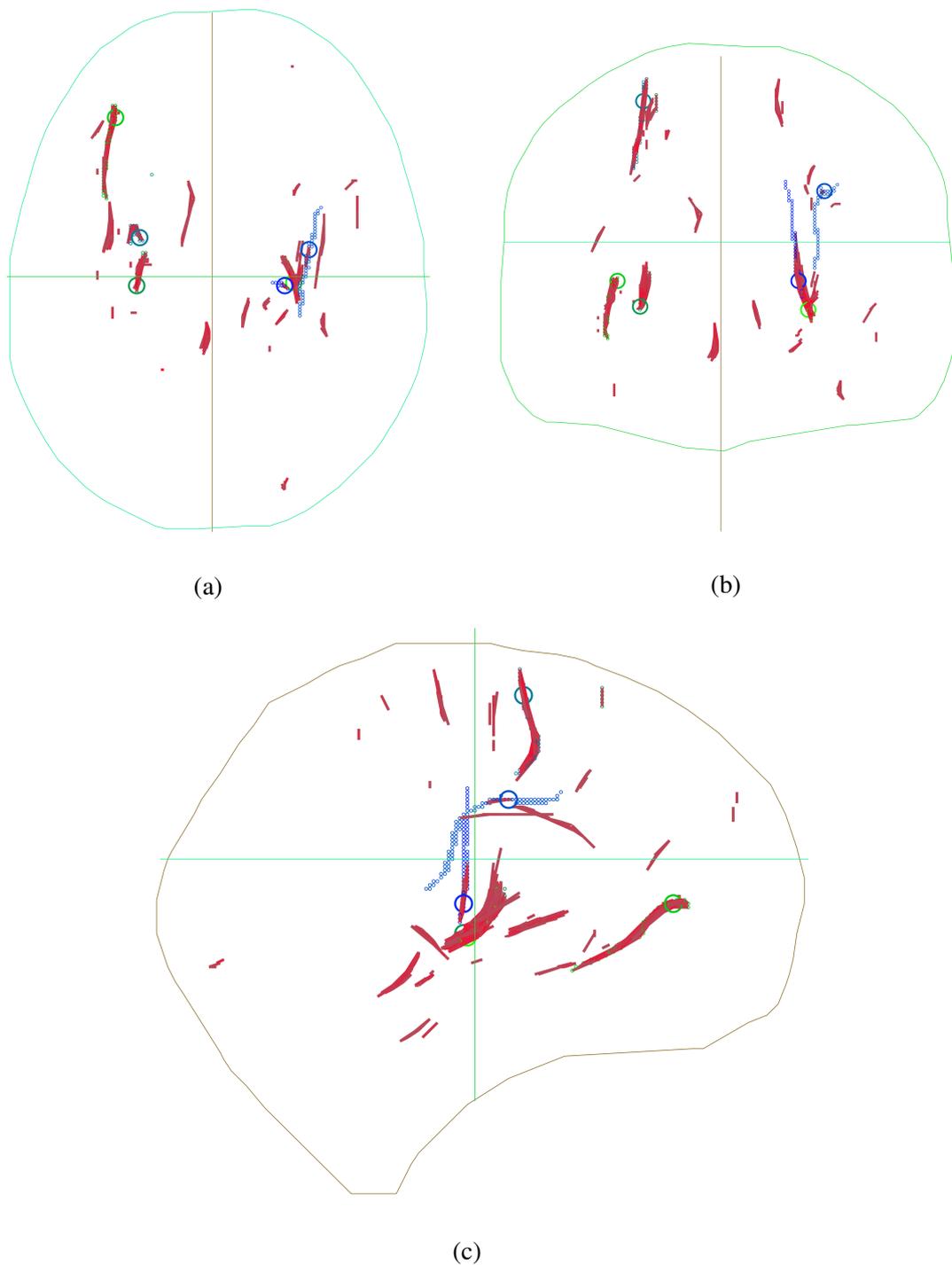


Figure 4.16: Subject 113 - scan 1: brain represented by x,y,z outlines from a surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,4,5,8,9) are displayed with a spectrum of colours from green (1) to blue (9). The 3500 highest traffic edges are then overlaid in red (a) View from above; (b) View from behind; (c) View from the side.

Insights into brain activity

The Montreal Neurological (MNI) Institute created a standard brain atlas by using a large series of MRI scans on normal control subjects. The eigenvectors in Figure 4.15 were manually mapped to the MNI standard model. The Regions of Interest (ROI) defined by Power et al. (2011) as putative functional areas, using the MNI coordinate frame, can now be compared with the most prominent node from each first eigenvector as detailed in Table 4.3 for subject 113 brain scan 1.

For eigenvector 4 (v_4) the nearest region is undefined but it is also close to the left Putamen with v_1 closest to the right Putamen. The Putamen is known to have a prominent role in a range of motor controlling functions, see DeLong et al. (1984); Marchand et al. (2008). These nodes are also in close proximity to the thalamus, which is a known hub for information in the brain, relaying information between subcortical areas and the cerebral cortex, see Hwang et al. (2017). v_2 is beside the left Frontal Mid Orbital and Frontal Mid region. The latter is part of the Brodmann area 46 that plays a central role in managing working memory, as well as memory control and organisation, see Bernal and Perdomo (2008a). v_5 and v_8 are in different parts of the Precentral region on the left and right respectively. The precentral gyrus is also known as the *motor strip* as it is involved in voluntary movements of skeletal muscles, see McCaffrey (2014). v_5 and v_8 are also in, or close to, the Brodmann area 6 - a region that includes part of the precentral gyrus - where the basic functions of this area are thought to be motor sequencing and planning movements, see Bernal and Perdomo (2008b). v_9 is in the Thalamus, which has been described previously as a major information hub. In conclusion, the regions highlighted appear to primarily deal with motor control, information distribution, sustaining attention and managing working memory.

\mathcal{V}	MNI Coordinates	Closest Power et al. (2011) ROI
1	[25,-19,-3]	Putamen_R (aal) - Right Cereberum, Sub-lobar, Extra-Nuclear, White Matter
2	[-34,36,6]	Frontal_Mid_Orb_L (aal) - Left Cerebrum, Frontal Lobe, Middle Frontal Gyrus, White Matter
4	[-27,-20,-2]	Undefined - Left Cerebrum, Sub-lobar, Extra-Nuclear, White Matter
5	[-26,-4,62]	Precentral_L (aal) - Left Cerebrum, Frontal Lobe, Middle Frontal Gyrus, Gray Matter, brodmann area 6
8	[30,-8,34]	Precentral_R (aal)- Right Cerebrum, Frontal Lobe, Precentral Gyrus, White Matter
9	[22,-20,6]	Thalamus_R (aal) - Right Cerebrum, Sub-lobar, Thalamus, Gray Matter

Table 4.3: Closest Power et al. (2011) ROI for each most prominent eigenvector node

4.3.2.2 Scan 2

A connectome was generated for the same subject (ID 113) after a different scan was conducted in the same study by Landman et al. (2011). This connectome was analysed and found to display *false* first eigenvectors again, though there were fewer present than in brain scan 1. The results are displayed in Fig. 4.17 where only 3 *false* first eigenvectors are found this time (after investigating the top ten eigenvectors). Fewer eigenvectors, and fewer prominent nodes belonging to each vector, are clear indicators of a change in the functional network that, unlike the underpinning structural network, can change depending on the task undertaken by the brain. What is also evident is the similarity of the four eigenvectors present in Figure 4.17 to the first four seen in Figure 4.15. In fact, Figure 4.18 overlays the first four eigenvector from brain scan 1 and 2 to show that they are part of the same pathways. Table 4.4 confirms that the prominent eigenvector pathways overlap and lie in close proximity. The pathways consist of nodes that are members of the most prominent communities and also have a large enough entry for the first eigenvector in question, i.e. $|v_i| > 0.05 \forall i \in \mathcal{V}$. Despite the similarities in pathways, there are significantly fewer prominent nodes belonging to each pathway in brain scan 2 than scan 1. The order of the overlapping pathways is also slightly altered with the 3rd and 4th ranked regions, according to the eigenvectors, swapping place.

Figure 4.19 continues to emphasise this detected difference in functional activity, with

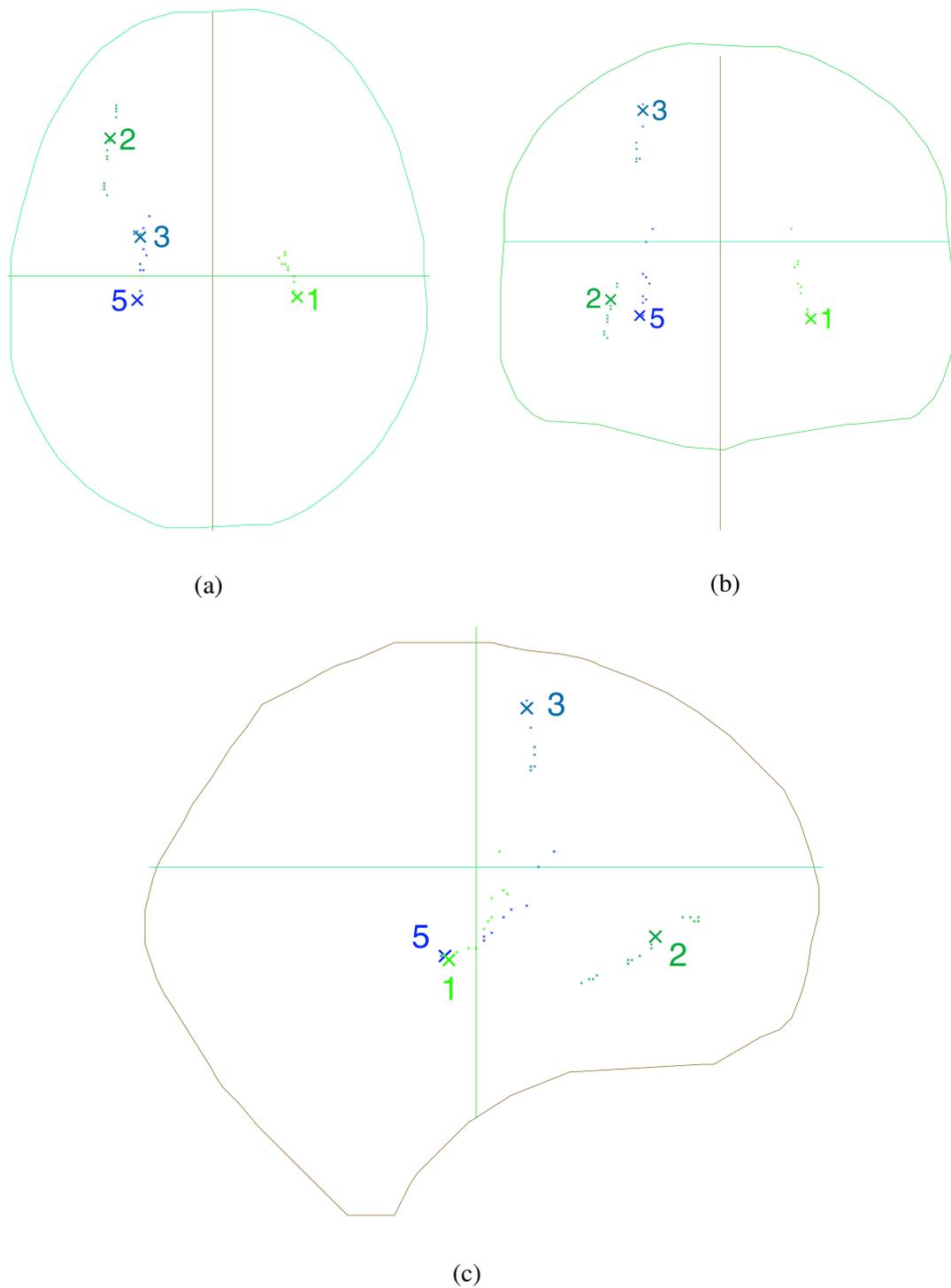


Figure 4.17: Subject 113 - scan 2: brain represented by x,y,z outlines from a surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,3,5) are displayed with a spectrum of colours from green (1) to blue (5). The most prominent node for each eigenvector is marked with a cross. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled.

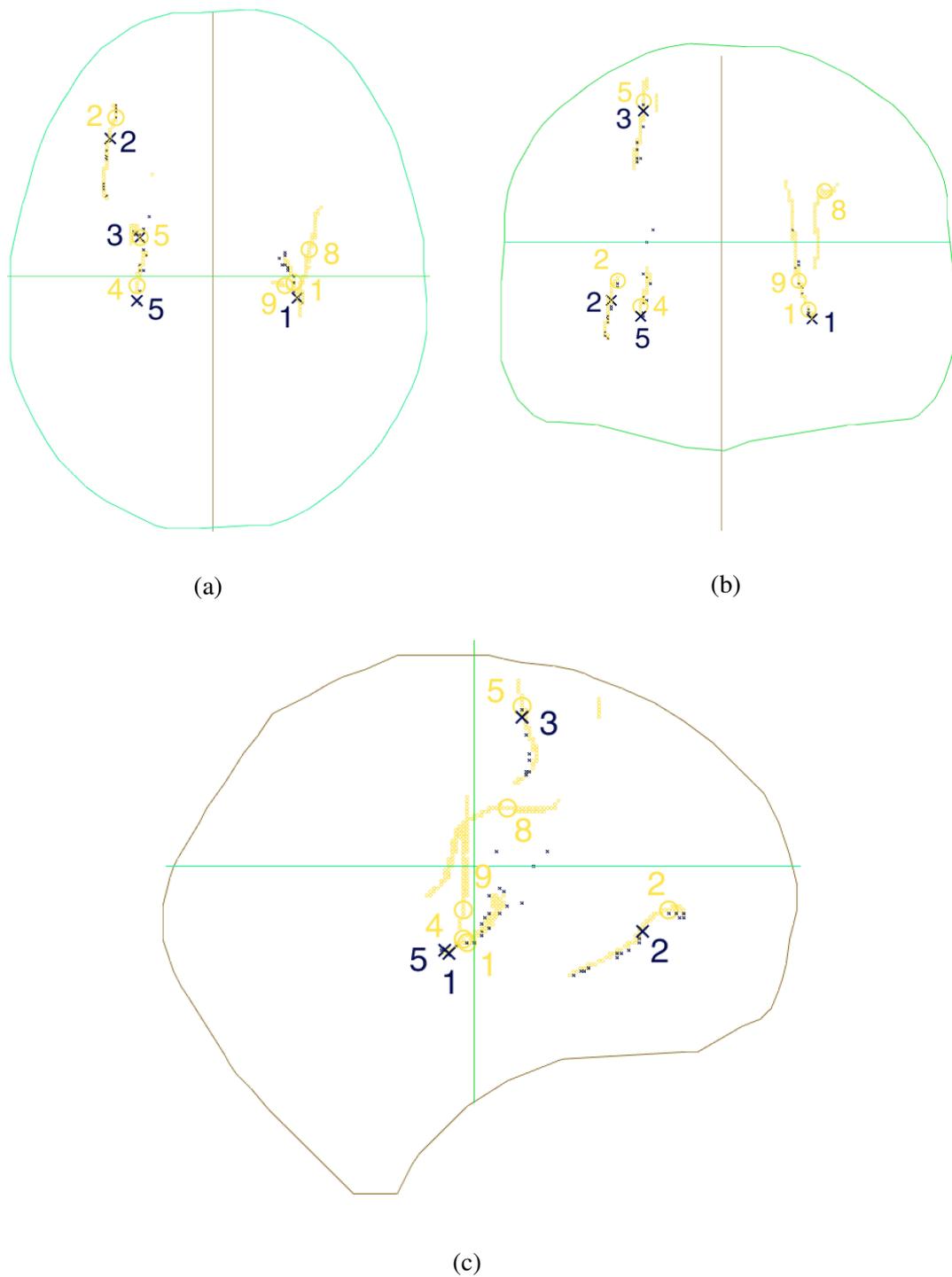


Figure 4.18: Subject 113: brain represented by x,y,z outlines from a surface model. Prominent nodes from scan 1 are displayed in yellow o's with scan 2 marked in black x's. The most prominent node for each eigenvector is marked with a circle or cross. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled.

Table 4.4: Scan comparison of prominent nodes for subject 113.

Eigenvectors		No. of Nodes		Nearest node comparison		
Scan 1	Scan 2	Scan 1	Scan 2	Min.	Max.	Mean
1	1	34	11	0	12.04	2.18
2	2	57	16	0	1.73	0.97
4	5	41	11	0	17.09	4.34
5	3	51	8	0	1.73	0.77

the top 3500 edges far less clustered than before.

Figure 4.20 provides further insight into the situation, revealing that the highest weighted edges belong to brain scan 2, but that the edge weights rapidly decline to be consistently lower than the same rank of edge in brain scan 1. This is interesting as it supports the finding that fewer *false* first eigenvectors, with fewer prominent members in each, were found for scan 2. Since a large amount of information traffic, present in a few pathways, combined with a relatively even spread of edge weights throughout the rest of the network, could result in information being channelled towards these few influential pathways. This would mark these influential pathways as far more prominent than other pathways in the brain. In scan 1, where there is a more even spread of edge weights, it is not surprising that more pathways will rise to prominence with the most influential pathways less prominent than they are in scan 2.

By comparing the top 3500 edges, according to weight, in Fig. 4.16 & 4.19 it would be difficult to identify this brain as belonging to the same subject. Whereas, Fig. 4.18 and Table 4.4 reveal that the prominent pathways from both scans are very similar and overlapping. Subject 679 shall now be presented to emphasise that the pathways found are specific to a person with a different brain presenting notably different pathways.

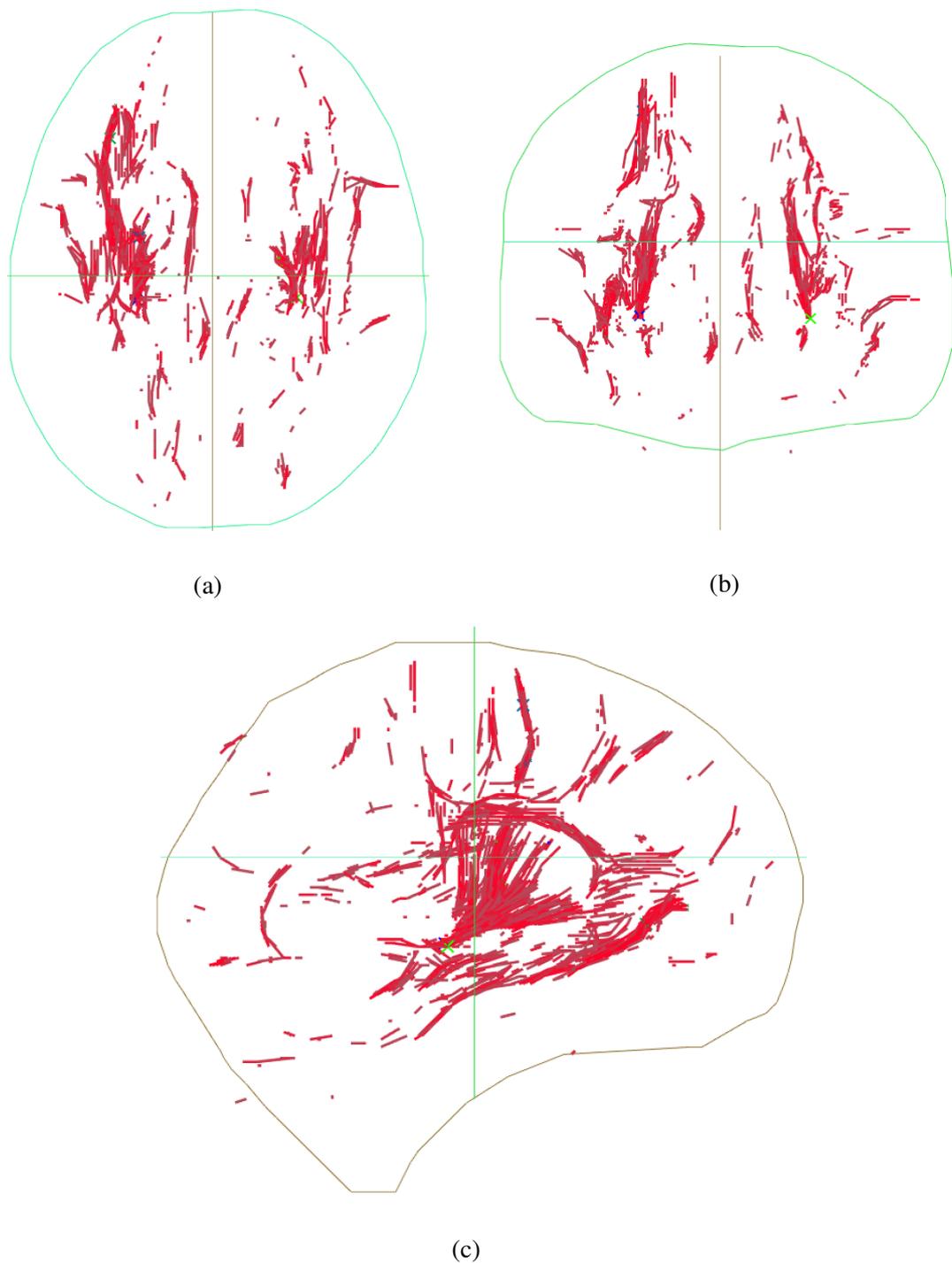


Figure 4.19: Subject 113 - scan 2: brain represented by x,y,z outlines from a brain surface model. Prominent nodes from the first eigenvector (1) and *false* first eigenvectors (2,3,5) are displayed with a spectrum of colours from green (1) to blue (9). The 3500 highest traffic edges are then overlaid in red (a) View from above; (b) View from behind; (c) View from the side.

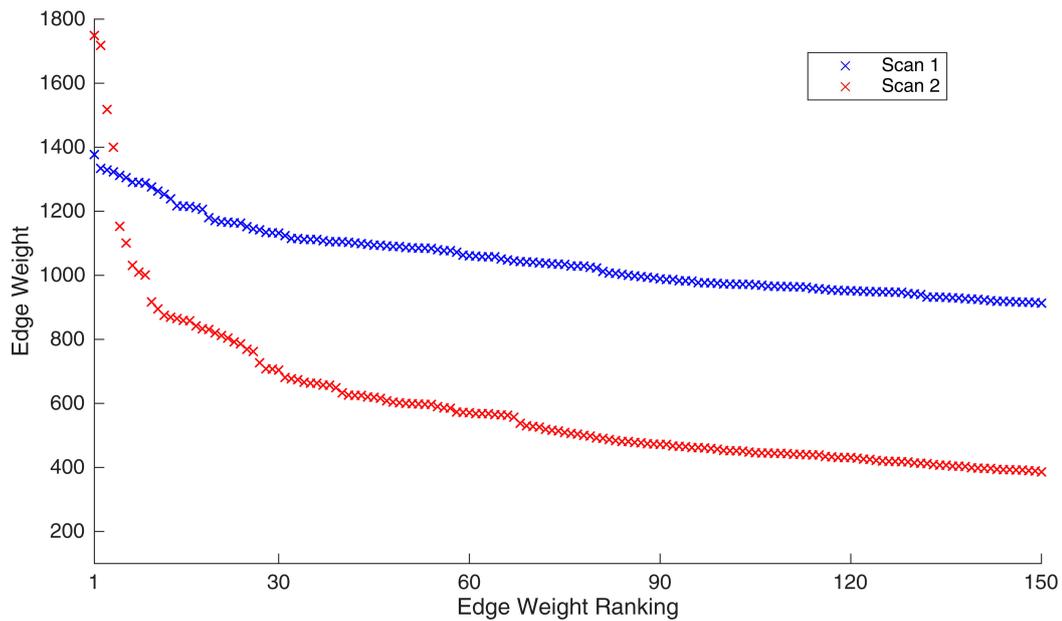


Figure 4.20: Edge weight comparison between brain scan 1 and 2 for subject 113 from Landman et al. (2011).

4.3.3 Subject 679

Subject 679 is similar to subject 113 in both gender, age and handedness as a female of 30 years old with a dominant right hand. Her connectome was created in the same study as subject 113 (see Roncal et al. (2013)) but her brain prominent pathways proved to be quite distinct. Again two separate scans were compared in Fig. 4.21 that show a clear correlation in prominent pathways between these scans.

For subject 679 there are four overlapping pathways with Table 4.5 detailing that close proximity and overlapping nature of these paths. When performing the same comparison between the pathways of subject 113 and 679 the closest match is for v_8 of subject 113 scan 1 with v_4 of subject 679 scan 2. The mean distance between nodes is 4.43 mm with a minimum distance of 1 mm and a maximum of 7.55 mm. This closest match fails to have a single overlapping node but does equal the highest mean observed in the matching pathways from 113 and 679. The most prominent node from each pathway in this closest match is found in the right side of the Putamen. The right side of the Putamen appears as a prominent node in all four scans, probably indicating its relevance to the situation each subject is in when undergoing a scan. The second most common region is the Precentral gyrus with the left and right side turning up in three scans each. The Putamen and Precentral gyrus are recognised for their involve-

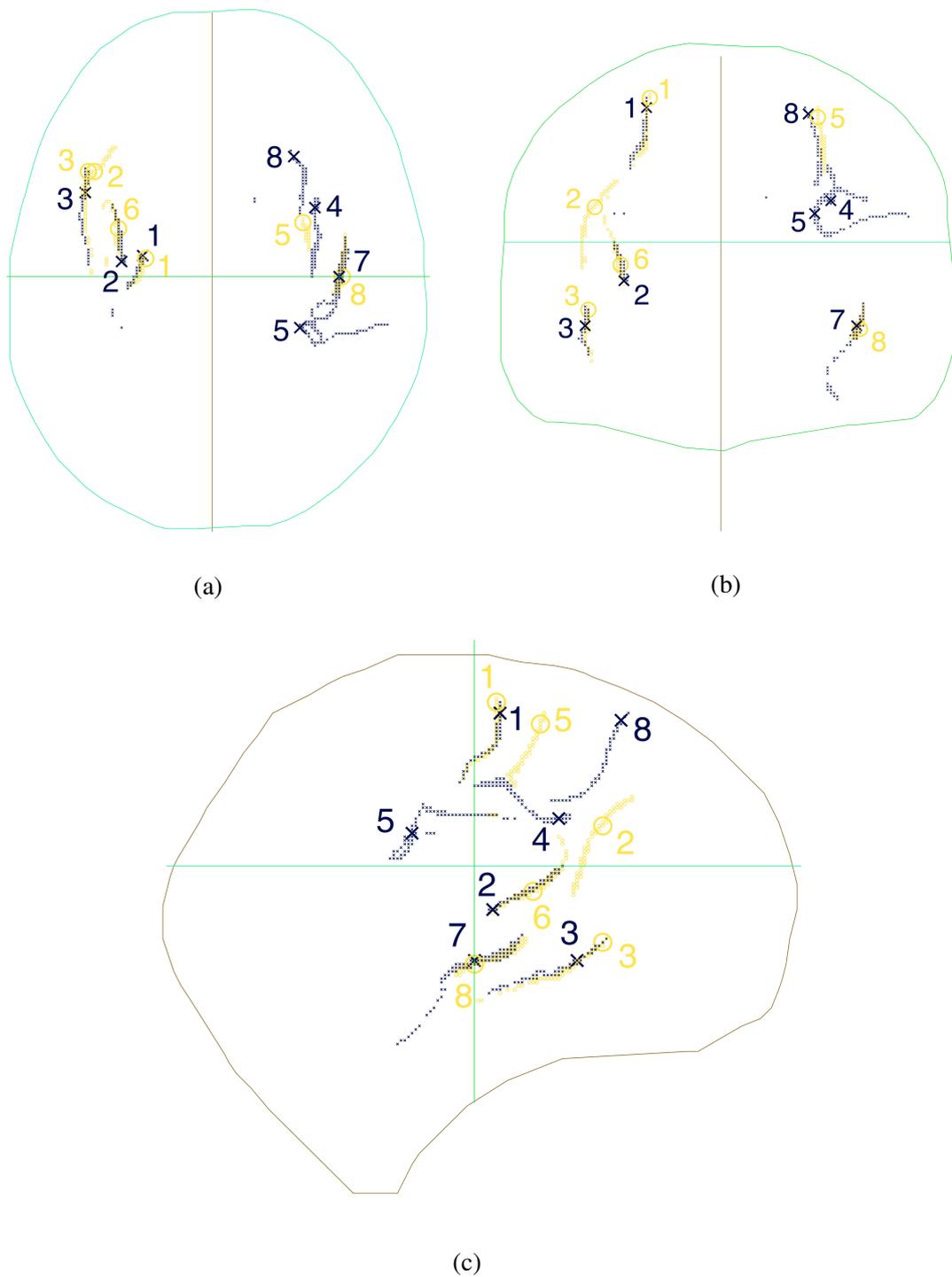


Figure 4.21: Subject 679: brain represented by x,y,z outlines from a brain surface model. Prominent nodes from scan 1 are displayed with yellow o's with scan 2 marked with black x's. The most prominent node for each eigenvector is marked with a circle or cross. (a) View from above; (b) View from behind; (c) View from the side with eigenvectors labelled.

Table 4.5: Scan comparison of prominent nodes for subject 679.

Eigenvectors		No. of Nodes		Nearest node comparison		
Scan 1	Scan 2	Scan 1	Scan 2	Min.	Max.	Mean
1	1	39	46	0	1.73	0.67
3	3	45	55	0	3.74	1.08
6	2	56	52	0	22.16	1.86
8	7	56	90	0	2	0.48

ment in motor control, therefore it is likely that the subjects were performing some form of motor control whilst undergoing the scan.

The second closest match sees the mean distance jump up to 9.02 mm, which is significant as it demonstrates that eigenvectors can be used to effectively identify a subject's brain. It is possible to speculate as to how such analysis could be used. By being able to compare the prominence of pathways in the brain a patient could be monitored to quantitatively assess changes in their brain's performance. For example, a stroke victim retraining neural pathways to regain speech could be monitored with scans taken during speech based tasks. The prominence of the relevant neural pathways (i.e. those in speech related regions of the brain) could be monitored to observe their growing prominence as the pathways are retrained. This could provide a metric with which to measure progress and provide further insight into the process of brain plasticity. This analysis could even form the basis of the treatment itself, as noted by Fornito et al. (2017). Fornito et al. stated that improved understanding of pathological circuitry had guided deep brain stimulation used in the treatment of Parkinson's disease, depression, and obsessive compulsive disorder. The potential of employing noninvasive brain stimulation techniques, such as transcranial magnetic stimulation, was also discussed where the challenge lay in identifying specific neural systems for targeted intervention. The work, presented in this chapter, has introduced a potential solution to this challenge as eigenvectors can identify specific neural pathways and could also be part of the treatment feedback loop by determining if the targeted pathways are becoming more or less prominent over time.

4.4 Summary

Some of the intuitions gained from the previous chapter and the development of Algorithm 3.3 were applied here to analyse brain connectomes. First the *C. elegans* - a

small, transparent, flat worm - was considered. Previous intuitions from Varshney et al. (2011) were expanded upon by employing a combination of eigenvectors to reveal the network communities that correspond to circuits in the brain. Investigations into the robustness of the *C. elegans* connectome by Bacik et al. (2016) found that the most influential nodes were also among those that had the greatest impact to partitions in the network when removed. For example the motor neurons, found by Bacik et al. to cause the greatest disruption when removed, were clearly displayed by the system's first eigenvectors as key members of the most influential network community. Bacik et al. (2016) also considered signal diffusion with eigenvectors shown to reveal if a strong response node was in the local dynamic community of the signal source nodes. Next, an adaptation of the eigenvector method used thus far was employed to highlight the strongest responders when a signal was introduced in the network. This method did not precisely replicate the results of Bacik et al., but rather complimented them. It highlighted strong response nodes but also revealed that some strong responders are in fact not under the dynamic influence of the signal introducing nodes. These nodes, not under the influence of the signal receivers, appeared to be well connected to other regions of the connectome and, as a result, would be difficult to influence without contributions from other regions in the network.

The macaque's well studied connectome presented another opportunity to contrast analytical results with those of numerical simulation. Specifically, the influence of the hippocampus was stated to be undetectable using graph theoretic methods but the eigenvectors were able to detect its importance and corroborate the findings of the numerical flow model. The eigenvectors were found to identify the nodes that collate information from across the brain before passing this processed information onto a key region, such as the hippocampus.

The final connectome considered was that of the human brain, with 1.8 million nodes included in the functional connectome. The first eigenvector and *false* first eigenvectors were able to present influential pathways in the brain. Many of these pathways were found to remain when the same subject undertook separate scans. The similarity of pathways could be used to differentiate between the two subjects, but at the same time differences in the ranking order of these pathways, where the most influential node was placed and how many pathway nodes presented as prominent suggested that there was a change in brain function between the scans.

Chapter 5

Discussion

The main objectives of this work were determined by three observations on naturally existing swarms. The first is that natural swarms display a remarkable scalability; a school of fish is a prime example of a swarm that remains responsive and cohesive regardless of whether there are a hundred members or a thousand. The aim was to produce a framework that could support a large, autonomous, engineered swarm that could emulate the scalability witnessed in nature. By creating a novel control scheme that defined an artificial kinematic field for the environment there is, in theory, no limit to the scalability of such a system. In practise, the success of the method relies on the ability of agents to track and implement the vehicle avoidance mechanism developed for this control scheme. This scheme has been demonstrated on aerial vehicles in a laboratory environment, but further work is required to progress beyond the lab and to incorporate a larger number of agents. The limited demonstration was still enough to show the validity of the control scheme that could one day support a large, autonomous, swarm.

The second objective was motivated by the ability of large swarms to rapidly respond to predator attack; starlings flock in huge numbers and can evade predators effectively by increasing their observational capacity but without significantly sacrificing their response speed and manoeuvrability. The example of starlings focused the initial investigations to consider only networks with constant outdegree constraints. The first aim in this case was to understand what makes a node more influential and how topology affects optimal network leaders. This was primarily addressed by the development of semi-analytical optimisers that uncover the optimal leadership for achieving fast convergence to consensus. This ability fed into achieving another aim, that of uncovering what topologies are most effective for producing highly responsive systems. This

investigation revealed how the optimal leadership differed for a number of constant outdegree topologies; including random graph, nearest neighbour and small world networks. This work also considered the effect of increasing or decreasing the outdegree of all nodes and how autocratic leaders could improve network responsiveness. When analysing variable outdegree networks it became clear that these produce a more complicated optimal leadership allocation than constant outdegree networks. But the most effective leaders can still be identified by analysing the top three eigenvectors, which in turn achieved the aim of being able to detect effective leaders in any given network. Finally, the inspiration for this objective was examined with a model starling flock analysed to show that topology plays a critical role in producing an effective predator avoiding flock.

The last objective was inspired by the ubiquity of networked systems and the potential for applying the lessons learnt from swarms on systems. The aim specifically was to apply the tools developed on swarming systems on analysing other highly responsive networked systems. This led to the identification of the brain as an excellent target for eigenvector based analysis. Analysis that could cope with the vast scale of such networks and achieve the aim by providing meaningful results on a system that is not a swarm but does rely on consensus and rapid information transfer. The development of methods for handling variable outdegree networks were especially key in dealing with the far from constant outdegree nature of brain connectomes.

Chapter 6

Future Work

Creating an engineered swarm was a motivation for this dissertation, where the technological capability is no longer the limiting factor but rather it is our ability to control and fully harness the potential of networked systems that holds us back. Chapter 2 was devoted to laying the foundations for a scalable swarm with the control approaches demonstrated, in a laboratory environment and through a remote inspection case study. This work was closer to a proof of concept with limited applications, than a finalised and adaptable system. There is plenty of scope for advancement, firstly it was mentioned that the current proportional controller could be improved by using a proportional and derivative controller, which would tackle the main drawback of the current scheme where the vehicles are prone to overshooting their desired path. An area of further promise would be to develop kinematic fields that are not constrained to circular motion; Lawrence et al. (2008), Frew et al. (2008) and Chen et al. (2013) have already demonstrated the beginnings of such control where their methods focused on placing multiple points in the environment with a limit cycle attached to each. This enabled, in the case of Lawrence et al. (2008) and Frew et al. (2008), the development of more complex loiter manoeuvres for unmanned aerial vehicles (UAVs). While Chen et al. (2013) employed a hybrid approach involving limit cycles to create obstacle avoiding trajectories. The next step for the work presented in this dissertation would be to vary the field to achieve complex trajectories, but this could be done by altering the vector field definition rather than just adding multiple limit cycles and using their interactions to achieve more complex behaviour. Bennet and McInnes (2009), as well as proposing limit cycle behaviour, also introduced bifurcation behaviour that converges to a point that could be used to transport to locations while the limit cycle enables loitering. As in the limit cycle case (see Fig. 2.3 & 2.4), the actual implementation of such

vector fields may require adjustment to deal with the reality of turbulent flight environments and smoother trajectories. There would also be a challenge in how such a system would decide when to alter the field - with a consensus process likely to take place to reach such an agreement - and how a smooth transition between fields could be accomplished.

The focus of much of this dissertation has been on networked system and understanding how influence can be wielded in them. The networks were initially constrained but this did not make them unapplicable, as starlings have similar network constraints when operating as a flock. It was shown, in chapter 3, that starlings employ a low enough outdegree for a nearest neighbour topology to produce multiple distinct communities with these graphs having a high modularity. It could be argued that consensus does not give the whole picture when considering a starling flock, where such a system is in a constant state of flux. If anything it could be argued that the work developed in this dissertation shows that the starling topology has evolved to avoid reaching consensus, where they form communities of influenced nodes that are strongly influenced by local nodes and only weakly affected by the rest of the network. Starlings are seen to move as separate groups, within the flock, that gives the flock its undulating and constantly shifting appearance. Therefore, there is much to investigate as to how these communities of influenced nodes tie up with what is observed from starlings, and whether these undulations are the result of perturbations influencing the local community far faster than the rest of the flock.

The research on leadership in responsive networked systems led to an investigation of brain connectomes from small flat worm's nervous systems to human brains represented by millions of nodes. The macaque connectome was the largest directed connectome considered but it only contained 242 nodes. One of the major advantages of using analytical approaches, like those developed herein, is that they can handle large networks with far less computational resources than numerical analysis requires. Unfortunately, there are only a small number of sources for directed connectomes and these are especially limited for large human connectomes. The brain is known to be a directed network, the accuracy of the results would therefore be enhanced by the use of directed connectomes. Assuming that the availability of data sets will only increase the next step for this work is to take it beyond the validation stage. Applying these eigenvector approaches to draw conclusion from large datasets where common analysis tools would struggle and to realise some of the proposed applications such as constructing a treatment feedback loop for retraining the brain after a stroke.

Chapter 7

Conclusions

The beginnings of this dissertation were firmly rooted in the potential of swarm engineering. In developing the ability to grow and control swarms, a toolset has been produced that is applicable far beyond this niche for networks where responsiveness is a desirable or already existing asset.

By emulating the scalability of bird flocks and fish schools with an artificial kinematic field - defined for the environment around the swarm - a large scale swarm can be created. The real-world validity of this approach was demonstrated through a remote inspection where agents circled, in-plane, around a central object whilst the novel vehicle avoidance mechanism adapted the field for only the trailing vehicle. To progress from this foundation - and onto a fully scalable, flexible and autonomous swarm - the ability to dynamically alter the kinematic field is key, otherwise the scheme is constrained to specific use cases such as the external inspection of large or inaccessible objects.

It is possible to manage swarms, without needing to influence every node, by understanding what makes a node influential. The eigenvector based methods, developed herein, can uncover influential leaders and, as a by-product, discover what topologies and leadership styles are most effective for creating highly responsive systems. Eigenvector centrality is often used in network analysis but, by considering perturbation driven consensus on a directed graph, the first left eigenvector proved to be an invaluable tool that had not previously been fully exploited. By considering multiple left eigenvectors and how these relate to communities that define dynamic response; the nature of a system's response to perturbations can be revealed. This capability can provide insight into nature's effectiveness; in this case, how localised connection con-

straints can construct a topology that places the most effective leaders in the positions from which external perturbations are expected.

The application of eigenvector-based approaches on brain networks proved effective; finding that previous assumptions on the limitations of graph theory's insights, into dynamic processes, are incorrect. Eigenvectors were demonstrated to be an effective bridge from the structural information of the topology to the dynamic performance of the network as information flows through it. The analysis of a, 1.8 million node, human functional connectome was made possible by this analytical approach, which could easily identify prominent neuronal pathways. In fact, the adjacency matrix created from separate MRI scans can be shown to be from the same subject when there is a high similarity in prominent eigenvector pathways. In this case of similar pathways, changes in the subject's functional state during each scan are also evident. This could pave the way to providing quantitative data on the change in mental state and performance of the same subject over time.

This dissertation has contributed to the development of autonomous robotic swarms but, at the time of writing, useful and effective versions of these are not an immediate prospect. Instead, it is the lessons we can take from developing such systems that are timely; benefitting our understanding of complex systems and how to operate more effective networks.

Bibliography

- Albert, R., Jeong, H. and Barabási, A.-L. (2000). Error and attack tolerance of complex networks, *nature* **406**(6794): 378–382.
- Alexanderson, G. (2006). About the cover: Euler and Königsberg's bridges: A historical view, *Bulletin of the American Mathematical Society* **43**(4): 567–573.
- Ascending Technologies* (2015). <http://www.asctec.de/>. [Online; accessed February-2015].
- Attanasi, A. et al. (2015). Emergence of collective changes in travel direction of starling flocks from individual birds' fluctuations, *Journal of The Royal Society Interface* **12**(108): 20150319.
- Autodesk (2015). Autodesk 123d catch, <http://www.123dapp.com/catch>. [Online; accessed February-2015].
- Azad, P., Gockel, T. and Dillmann, R. (2008). Computer vision: principles and practice.
- Bacik, K. A., Schaub, M. T., Beguerisse-Díaz, M., Billeh, Y. N. and Barahona, M. (2016). Flow-based network analysis of the *Caenorhabditis elegans* connectome, *PLoS computational biology* **12**(8): e1005055.
- Balanukhin, R. (2013). Ar.drone, <https://github.com/Ruslan-B/AR.Drone>. [Source Code].
- Balister, P., Bollobás, B., Sarkar, A. and Walters, M. (2005). Connectivity of random k-nearest-neighbour graphs, *Advances in Applied Probability* **37**(01): 1–24.
- Ballerini, M., Cabibbo, N., Candelier, R., Cavagna, A., Cisbani, E., Giardina, I., Lecomte, V., Orlandi, A., Parisi, G., Procaccini, A. et al. (2008). Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study, *Proceedings of the National Academy of Sciences* **105**(4): 1232–1237.
- Barabási, A.-L. (2009). Scale-free networks: a decade and beyond, *science* **325**(5939): 412–413.

- Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks, *science* **286**(5439): 509–512.
- Bassett, D. S. and Bullmore, E. (2006). Small-world brain networks, *The neuroscientist* **12**(6): 512–523.
- Beebe, W. (1921). Edge of the jungle, pp. 291–294.
- Bennet, D. J. and McInnes, C. (2009). Verifiable control of a swarm of unmanned aerial vehicles, *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **223**(7): 939–953.
- Bernal, B. and Perdomo, J. (2008a). Brodmann’s interactive atlas - area 46, <http://www.fmriconsulting.com/brodman/BA46.html>. [Online; accessed December-2017].
- Bernal, B. and Perdomo, J. (2008b). Brodmann’s interactive atlas - area 6, <http://www.fmriconsulting.com/brodman/BA6.html>. [Online; accessed December-2017].
- Brambilla, M., Ferrante, E., Birattari, M. and Dorigo, M. (2013). Swarm robotics: a review from the swarm engineering perspective, *Swarm Intelligence* **7**(1): 1–41.
- Braun, U., Muldoon, S. F. and Bassett, D. S. (2015). On human brain networks in health and disease, *eLS*.
- Bruemmer, D. J., Dudenhoeffer, D. D., McKay, M. D. and Anderson, M. O. (2002). A robotic swarm for spill finding and perimeter formation, *Technical report*, DTIC Document.
- Brutto, M. L. and Meli, P. (2012). Computer vision tools for 3d modelling in archaeology, *International Journal of Heritage in the Digital Era* **1**(1 suppl): 1–6.
- Bumann, D., Krause, J. and Rubenstein, D. (1997). Mortality risk of spatial positions in animal groups: the danger of being in the front, *Behaviour* **134**(13): 1063–1076.
- Cavagna, A., Cimorelli, A., Giardina, I., Parisi, G., Santagati, R., Stefanini, F. and Viale, M. (2010). Scale-free correlations in starling flocks, *Proceedings of the National Academy of Sciences* **107**(26): 11865–11870.
- Chen, H., Chang, K. and Agate, C. S. (2013). Uav path planning with tangent-plus-lyapunov vector field guidance and obstacle avoidance, *IEEE Transactions on Aerospace and Electronic Systems* **49**(2): 840–856.
- Clarke, A. C. (1973). *Profiles of the future: An inquiry into the limits of the possible*. Rev. ed, Harper & Row.
- Cohen, R., Erez, K., Ben-Avraham, D. and Havlin, S. (2000). Resilience of the internet to random breakdowns, *Physical review letters* **85**(21): 4626.

- Czirók, A., Ben-Jacob, E., Cohen, I. and Vicsek, T. (1996). Formation of complex bacterial colonies via self-generated vortices, *Physical Review E* **54**(2): 1791.
- de Lope, J., Maravall, D. and Quiñonez, Y. (2013). Response threshold models and stochastic learning automata for self-coordination of heterogeneous multi-task distribution in multi-robot systems, *Robotics and Autonomous Systems* **61**(7): 714–720.
- Deif, A. (1995). Rigorous perturbation bounds for eigenvalues and eigenvectors of a matrix, *Journal of computational and applied mathematics* **57**(3): 403–412.
- DeLong, M., Alexander, G., Georgopoulos, A., Crutcher, M., Mitchell, S. and Richardson, R. (1984). Role of basal ganglia in limb movements., *Human neurobiology* **2**(4): 235–244.
- Dhal, R., Roy, S., Wan, Y. and Saberi, A. (2014). Majorisations for the eigenvectors of graph-adjacency matrices, *International Journal of Control* **87**(12): 2604–2614.
- Dobie, G., Summan, R., MacLeod, C. and Pierce, S. G. (2013). Visual odometry and image mosaicing for nde, *NDT & E International* **57**: 17–25.
- Edunov, S., Diuk, C., Filiz, I. O., Bhagat, S. and Burke, M. (2016). Three and a half degrees of separation, *Research at Facebook* .
- El Kahi, S., Asmar, D., Fakih, A., Nieto, J. and Nebot, E. (2011). A vision-based system for mapping the inside of a pipe, *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, IEEE, pp. 2605–2611.
- Emmerton, J. and Delius, J. D. (1993). 21 beyond sensation: Visual cognition in pigeons, *Vision, brain, and behavior in birds* p. 377.
- Firoiu, V., Whitney, W. F. and Tenenbaum, J. B. (2017). Beating the world’s best at super smash bros. with deep reinforcement learning, *arXiv preprint arXiv:1702.06230* .
- Fornito, A., Bullmore, E. T. and Zalesky, A. (2017). Opportunities and challenges for psychiatry in the connectomic era, *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging* **2**(1): 9–19.
- Fraundorfer, F. (2015). Building and site reconstruction from small scale unmanned aerial vehicles (uav’s), *Urban Remote Sensing Event (JURSE), 2015 Joint*, IEEE, pp. 1–4.
- Frew, E. W., Lawrence, D. A. and Morris, S. (2008). Coordinated standoff tracking of moving targets using lyapunov guidance vector fields, *Journal of guidance, control, and dynamics* **31**(2): 290–306.
- Gong, G., He, Y., Concha, L., Lebel, C., Gross, D. W., Evans, A. C. and Beaulieu, C. (2008). Mapping anatomical connectivity patterns of human cerebral cortex using in vivo diffusion tensor imaging tractography, *Cerebral cortex* **19**(3): 524–536.

- Granovetter, M. (1978). Threshold models of collective behavior, *American journal of sociology* **83**(6): 1420–1443.
- Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Wedeen, V. J. and Sporns, O. (2008). Mapping the structural core of human cerebral cortex, *PLoS biology* **6**(7): e159.
- Hammack, R. H., Imrich, W., Klavžar, S., Imrich, W. and Klavžar, S. (2011). *Handbook of product graphs*, CRC press Boca Raton.
- Hansen, P., Alismail, H., Rander, P. and Browning, B. (2013). Pipe mapping with monocular fisheye imagery, *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, pp. 5180–5185.
- Harriger, L., Van Den Heuvel, M. P. and Sporns, O. (2012). Rich club organization of macaque cerebral cortex and its role in network communication, *PloS one* **7**(9): e46497.
- Harris, C. and Stephens, M. (1988). A combined corner and edge detector., *Alvey vision conference*, Vol. 15, Citeseer, p. 50.
- Heimer, L. and Robards, M. J. (2013). *Neuroanatomical tract-tracing methods*, Springer Science & Business Media.
- Herbert-Read, J. E., Buhl, J., Hu, F., Ward, A. J. and Sumpter, D. J. (2015). Initiation and spread of escape waves within animal groups, *Royal Society open science* **2**(4): 140355.
- Honey, C. J., Kötter, R., Breakspear, M. and Sporns, O. (2007). Network structure of cerebral cortex shapes functional connectivity on multiple time scales, *Proceedings of the National Academy of Sciences* **104**(24): 10240–10245.
- Hwang, K., Bertolero, M. A., Liu, W. B. and D’Esposito, M. (2017). The human thalamus is an integrative hub for functional brain networks, *Journal of Neuroscience* **37**(23): 5594–5607.
- Jadbabaie, A., Lin, J. and Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules, *Automatic Control, IEEE Transactions on* **48**(6): 988–1001.
- Jones, J. E. (1924). On the determination of molecular fields. ii. from the equation of state of a gas, *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 106, The Royal Society, pp. 463–477.
- Kaw, A., Kalu, E. and Ngyen, D. (2008). Numerical methods with applications.
- Kazadi, S. T. (2000). *Swarm engineering*, PhD thesis, California Institute of Technology.

- Kersten, T. P. and Lindstaedt, M. (2012). Image-based low-cost systems for automatic 3d recording and modelling of archaeological finds and objects, *Progress in cultural heritage preservation*, Springer, pp. 1–10.
- Khanna, R., Möller, M., Pfeifer, J., Liebisch, F., Walter, A. and Siegwart, R. (2015). Beyond point clouds-3d mapping and field parameter measurements using uavs, *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, IEEE, pp. 1–4.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *Autonomous robot vehicles*, Springer, pp. 396–404.
- KMel Robotics (2012). A swarm of nano quadrotors, <https://www.youtube.com/watch?v=YQIMGV5vt4>. [Online; accessed February-2017].
- Kötter, R. (2004). Online retrieval, processing, and visualization of primate connectivity data from the cocomac database, *Neuroinformatics* **2**(2): 127–144.
- Kumar, V. and Sahin, F. (2003). Cognitive maps in swarm robots for the mine detection application, *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, Vol. 4, IEEE, pp. 3364–3369.
- Kunegis, J. (2016). KONECT – network visualization. [Online; accessed 17-10-2016].
URL: <http://konect.uni-koblenz.de/plots/map.ax>
- Kushleyev, A., Mellinger, D., Powers, C. and Kumar, V. (2013). Towards a swarm of agile micro quadrotors, *Autonomous Robots* **35**(4): 287–300.
- Laboratory for Optical and Computational Instrumentation, University of Wisconsin-Madison (2009). A short history of c. elegans research, <http://wormclassroom.org/short-history-c-elegans-research>. [Online; accessed August-2017].
- Landman, B. A., Huang, A. J., Gifford, A., Vikram, D. S., Lim, I. A. L., Farrell, J. A., Bogovic, J. A., Hua, J., Chen, M., Jarso, S. et al. (2011). Multi-parametric neuroimaging reproducibility: a 3-t resource study, *Neuroimage* **54**(4): 2854–2866.
- Lawrence, D. A., Frew, E. W. and Pisano, W. J. (2008). Lyapunov vector fields for autonomous unmanned aircraft flight control, *Journal of Guidance, Control, and Dynamics* **31**(5): 1220–1229.
- Leicht, E. A. and Newman, M. E. (2008). Community structure in directed networks, *Physical review letters* **100**(11): 118703.
- Leonard, N. E. and Fiorelli, E. (2001). Virtual leaders, artificial potentials and coordinated control of groups, *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, Vol. 3, IEEE, pp. 2968–2973.

- Leonard, N. E., Paley, D. A., Lekien, F., Sepulchre, R., Fratantoni, D. M. and Davis, R. E. (2007). Collective motion, sensor networks, and ocean sampling, *Proceedings of the IEEE* **95**(1): 48–74.
- Lewis, T. G. (2011). *Network science: Theory and applications*, John Wiley & Sons.
- Liu, Y.-Y., Slotine, J.-J. and Barabási, A.-L. (2011). Controllability of complex networks, *Nature* **473**(7346): 167–173.
- Lozano, R., Spong, M. W., Guerrero, J. A. and Chopra, N. (2008). Controllability and observability of leader-based multi-agent systems, *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, IEEE, pp. 3713–3718.
- Marchand, W. R., Lee, J. N., Thatcher, J. W., Hsu, E. W., Rashkin, E., Suchy, Y., Chelune, G., Starr, J. and Barbera, S. S. (2008). Putamen coactivation during motor task execution, *Neuroreport* **19**(9): 957–960.
- Marjovi, A., Nunes, J. G., Marques, L. and de Almeida, A. (2010). Multi-robot fire searching in unknown environment, *Field and Service Robotics*, Springer, pp. 341–351.
- MathWorks (2015a). Constrained nonlinear optimization algorithms, <http://www.mathworks.se/help/optim/ug/constrained-nonlinear-optimization-algorithms.html>. [Online; accessed 2-March-2015].
- MathWorks (2015b). fminunc unconstrained minimization, <http://uk.mathworks.com/help/optim/ug/fminunc-unconstrained-minimization.html>. [Online; accessed 2-March-2015].
- McCaffrey, P. (2014). The neuroscience on the web series - chapter 4 cerebral lobes, cerebral cortex, and brodmann’s areas, <https://www.csuchico.edu/~pmccaffrey/syllabi/CMSD%20320/362unit4.html>. [Online; accessed December-2017].
- McCamish, S. B., Romano, M. and Yun, X. (2010). Autonomous distributed control of simultaneous multiple spacecraft proximity maneuvers, *Automation Science and Engineering, IEEE Transactions on* **7**(3): 630–644.
- Meilä, M. (2007). Comparing clusterings an information based distance, *Journal of multivariate analysis* **98**(5): 873–895.
- Merikoski, J. K. and Virtanen, A. (1997). Bounds for eigenvalues using the trace and determinant, *Linear algebra and its applications* **264**: 101–108.
- Mesbahi, M. and Egerstedt, M. (2010). *Graph theoretic methods in multiagent networks*, Princeton University Press.

- Miller, P. (2010). *The smart swarm: How understanding flocks, schools, and colonies can make us better at communicating, decision making, and getting things done*, Avery Publishing Group, Inc.
- Minsky, M. L. (1967). *Computation: finite and infinite machines*, Prentice-Hall, Inc.
- Mišić, B., Goñi, J., Betzel, R. F., Sporns, O. and McIntosh, A. R. (2014). A network convergence zone in the hippocampus, *PLoS computational biology* **10**(12): e1003982.
- Misra, P. and Enge, P. (2006). *Global Positioning System: Signals, Measurements and Performance Second Edition*, Lincoln, MA: Ganga-Jamuna Press.
- Modha, D. S. and Singh, R. (2010). Network architecture of the long-distance pathways in the macaque brain, *Proceedings of the National Academy of Sciences* **107**(30): 13485–13490.
- Moscovitch, M., Rosenbaum, R. S., Gilboa, A., Addis, D. R., Westmacott, R., Grady, C., McAndrews, M. P., Levine, B., Black, S., Winocur, G. et al. (2005). Functional neuroanatomy of remote episodic, semantic and spatial memory: a unified account based on multiple trace theory, *Journal of anatomy* **207**(1): 35–66.
- Murray, R. (2003). Consensus protocols for networks of dynamic agents, *Proceedings of the 2003 American Controls Conference*.
- Nagy, M., Ákos, Z., Biro, D. and Vicsek, T. (2010). Hierarchical group dynamics in pigeon flocks, *Nature* **464**(7290): 890–893.
- Newman, M. E. (2000). Models of the small world, *Journal of Statistical Physics* **101**(3-4): 819–841.
- Ni, W., Wang, X. and Xiong, C. (2013). Consensus controllability, observability and robust design for leader-following linear multi-agent systems, *Automatica* **49**(7): 2199–2205.
- Olfati-Saber, R. (2005). Ultrafast consensus in small-world networks, *American Control Conference, 2005. Proceedings of the 2005*, IEEE, pp. 2371–2378.
- Olfati-Saber, R. and Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays, *Automatic Control, IEEE Transactions on* **49**(9): 1520–1533.
- Omidi, E. and Abdollahi, F. (2013). A distributed adaptive algorithm for speeding up consensus in networked multi-agent systems, *Control, Instrumentation, and Automation (ICCIA), 2013 3rd International Conference on*, IEEE, pp. 323–327.
- Ortiz, A., Bonnin-Pascual, F. and Garcia-Fidalgo, E. (2014). Vessel inspection: A micro-aerial vehicle-based approach, *Journal of Intelligent & Robotic Systems* **76**(1): 151–167.

- Page, L., Brin, S., Motwani, R. and Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.
- Parrot (2015). Parrot ar.drone 2.0 technical specifications, <http://ardrone2.parrot.com/ardrone-2/specifications/>. [Online; accessed January-2015].
- Pastor-Satorras, R. and Vespignani, A. (2001). Epidemic spreading in scale-free networks, *Physical review letters* **86**(14): 3200.
- Penders, J., Alboul, L., Witkowski, U., Naghsh, A., Saez-Pons, J., Herbrechtsmeier, S. and El-Habbal, M. (2011). A robot swarm assisting a human fire-fighter, *Advanced Robotics* **25**(1-2): 93–117.
- Piskorski, S., Brulez, N., Eline, P. and DHaeyer, F. (2012). Ar.drone developer guide sdk 2.0.
- Potts, W. K. (1984). The chorus-line hypothesis of manoeuvre coordination in avian flocks, *Nature* **309**(5966): 344.
- Power, J. D., Cohen, A. L., Nelson, S. M., Wig, G. S., Barnes, K. A., Church, J. A., Vogel, A. C., Laumann, T. O., Miezin, F. M., Schlaggar, B. L. et al. (2011). Functional network organization of the human brain, *Neuron* **72**(4): 665–678.
- Prettejohn, B., Berryman, M. and McDonnell, M. (2011). Methods for generating complex networks with selected structural properties for simulations: a review and tutorial for neuroscientists.
- Punzo, G. (2013). *Verifiable Swarm Engineering with Limited Communication*, PhD thesis, University of Strathclyde.
- Punzo, G., Young, G. F., Macdonald, M. and Leonard, N. E. (2016). Using network dynamical influence to drive consensus, *Scientific reports* **6**(26318).
- Rahmani, A., Ji, M., Mesbahi, M. and Egerstedt, M. (2009). Controllability of multi-agent systems from a graph-theoretic perspective, *SIAM Journal on Control and Optimization* **48**(1): 162–186.
- Remondino, F., Barazzetti, L., Nex, F., Scaioni, M. and Sarazzi, D. (2011). Uav photogrammetry for mapping and 3d modeling—current status and future perspectives, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **38**(1): C22.
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH computer graphics* **21**(4): 25–34.

- Roncal, W. G., Koterba, Z. H., Mhembere, D., Kleissas, D. M., Vogelstein, J. T., Burns, R., Bowles, A. R., Donavos, D. K., Ryman, S., Jung, R. E. et al. (2013). Migraine: Mri graph reliability analysis and inference for connectomics, *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, IEEE, pp. 313–316.
- Şahin, E. (2004). Swarm robotics: From sources of inspiration to domains of application, *International workshop on swarm robotics*, Springer, pp. 10–20.
- Scheffer, V. B. (1983). *Spires of Form: Glimpses of Evolution*, University of Washington Press.
- Selous, E. (1931). Thought transference (or what?) in birds.
- Shang, Y. and Bouffanais, R. (2014). Influence of the number of topologically interacting neighbors on swarm dynamics, *Scientific reports* **4**: 4184.
- Shanmugavel, M., Tsourdos, A., White, B. and Żbikowski, R. (2010). Co-operative path planning of multiple uavs using dubins paths with clothoid arcs, *Control Engineering Practice* **18**(9): 1084–1092.
- Shao, H., Mesbahi, M. and Xi, Y. (2015). The relative tempo of discrete-time consensus networks, *Control Conference (CCC), 2015 34th Chinese*, IEEE, pp. 7362–7367.
- Shi, J. and Tomasi, C. (1994). Good features to track, *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, IEEE, pp. 593–600.
- Six Degrees of Separation* (1993).
- Sporns, O., Honey, C. J. and Kötter, R. (2007). Identification and classification of hubs in brain networks, *PloS one* **2**(10): e1049.
- Squire, L. R. (1992). Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans., *Psychological review* **99**(2): 195.
- Stewart, G. (2000). Matrix algorithms volume I: Basic decompositions.
- Travers, J. and Milgram, S. (1967). The small world problem, *Psychology Today* **1**: 61–67.
- Treherne, J. and Foster, W. (1981). Group transmission of predator avoidance behaviour in a marine insect: the trafilgar effect, *Animal Behaviour* **29**(3): 911–917.
- van den Heuvel, M. P., Kahn, R. S., Goñi, J. and Sporns, O. (2012). High-cost, high-capacity backbone for global brain communication, *Proceedings of the National Academy of Sciences* **109**(28): 11372–11377.

- Van Den Heuvel, M. P. and Pol, H. E. H. (2010). Exploring the brain network: a review on resting-state fmri functional connectivity, *European neuropsychopharmacology* **20**(8): 519–534.
- Varshney, L. R., Chen, B. L., Paniagua, E., Hall, D. H. and Chklovskii, D. B. (2011). Structural properties of the caenorhabditis elegans neuronal network, *PLoS computational biology* **7**(2): e1001066.
- Vasile, C., Pavel, A. and Buiu, C. (2011). Integrating human swarm interaction in a distributed robotic control system, *Automation Science and Engineering (CASE), 2011 IEEE Conference on*, IEEE, pp. 743–748.
- Vicon (2015). Vicon motion tracking system, <http://www.vicon.com/Software/Tracker>. [Online; accessed January-2015].
- Vicsek, T. (2001). *Fluctuations and scaling in biology*, Oxford University Press New York.
- Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I. and Shochet, O. (1995). Novel type of phase transition in a system of self-driven particles, *Physical review letters* **75**(6): 1226.
- Vicsek, T. and Zafeiris, A. (2012). Collective motion, *Physics Reports* **517**(3): 71–140.
- White, J. G., Southgate, E., Thomson, J. N. and Brenner, S. (1986). The structure of the nervous system of the nematode caenorhabditis elegans: the mind of a worm, *Phil. Trans. R. Soc. Lond* **314**: 1–340.
- Winfield, A. F., Harper, C. J. and Nembrini, J. (2004). Towards dependable swarms and a new discipline of swarm engineering, *International Workshop on Swarm Robotics*, Springer, pp. 126–142.
- WormAtlas, Department of Neuroscience, A. E. C. o. M. (2013). Nervous system general description, <http://www.wormatlas.org/hermaphrodite/nervous/Neuroframeset.html>. [Online; accessed August-2017].
- Xiao, L. and Boyd, S. (2004). Fast linear iterations for distributed averaging, *Systems & Control Letters* **53**(1): 65–78.
- Ying, W., Wen, Y. and Xiaofan, W. (2014). Optimal leader selection for fast consensus via consensus centrality, *Control Conference (CCC), 2014 33rd Chinese*, IEEE, pp. 1482–1487.
- Young, G. F., Scardovi, L., Cavagna, A., Giardina, I. and Leonard, N. E. (2013). Starling flock networks manage uncertainty in consensus at low cost, *PLoS Comput Biol* **9**(1): e1002894.