

Combining Event Cameras with Spiking Neural Networks for Bio-Inspired Event-Based Visual Inertial Odometry

GREGOR SEAN MACKENZIE

PHD

UNIVERSITY OF YORK
Physics, Engineering and Technology
September, 2025

Combining Event Cameras with Spiking Neural Networks for Bio-Inspired Event-Based Visual Inertial Odometry

GREGOR SEAN MACKENZIE

*Physics, Engineering and Technology
University of York*

Abstract

The domain of Neuromorphic Computing applies models and concepts from neuroscience, emulating the computational behaviours observed in biological neurons through the medium of spiking neural networks, and applying them to engineering tasks. In recent years, Neuromorphic Computing has seen rapid growth, with applications that have transitioned from lab-based testing and simulation, to real-world scenarios. Navigation tasks are the subject of extensive and varied research due to the variety of implementations afforded by sensor fusion and the range of possible solutions, e.g. Simultaneous Localisation and Mapping (SLAM), Visual Inertial Odometry (VIO), visual place recognition. An entirely event-based system capable of performing VIO on real-world data has yet to be proposed. Here, we combine event cameras with spiking neural networks to perform localisation tasks on complex real-world driving data, creating a spike-based representation of position and orientation. We show how navigation-oriented neural networks that have been observed in the mammalian brain, such as head direction, grid, and place cells, can be adapted to track both orientation and position in real-world driving scenarios through inertial signals. It is also shown that stereo event-based visual place recognition can be combined with inertial integration for loop closure, providing a robust solution that corrects accumulated errors. These results show the viability of a system that can take advantage of the asynchronous, low-power benefits of neuromorphic processing in an end-to-end fashion by maintaining sparse event encoding from the event camera input to the spiking neural network outputs. It is anticipated that neuromorphic navigation systems can provide an optimal solution for applications where low latency and low power are constraints, such as autonomous robotics. This work should be expanded upon with the development of more complex visual place recognition spiking neural networks, refinement of the spatial network behaviours, and with increased dimensionality to allow flying agents to also be tracked accurately through 3D space.

Acknowledgment

First and foremost, I would like to express my appreciation to my supervisor Prof. David Halliday, whose support and guidance have been invaluable throughout the duration of my studies. I also extend my gratitude to Dr. Mark Post and Prof. Martin Trefzer for acting as my panel members, providing insight and expertise beyond my own; revealing new ideas and paths for my research.

Within the wonderful community at the University of York, there are too many people to individually thank for their support and friendship, so instead I will express my gratitude to everyone involved in the ECRF, YUMC, and Ben Lairig. To those involved in the ECRF, I thank you for fostering a community amongst the researchers and facilitating friendships and social outings. I thank those involved with YUMC for their welcoming nature and offering a break from the stresses of academia. Finally, I thank everyone in Ben Lairig for providing a sense of adventure, broadening my horizons, and offering a true getaway from the day-to-day.

Finally, but without a doubt, most importantly, I would like to thank my family. I thank my sister, Isla, for serving as inspiration to continue to pursue my studies, putting me on the path to where I am today. I thank my Dad, Ewan, and my Mum, Sue, whose love and support has facilitated all that I have achieved both during my PhD and in life.

It brings me great joy to acknowledge all those who have shaped this journey for me, so to conclude, I thank you all.

Author's Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for a degree or other qualification at this University or elsewhere. All sources are acknowledged as references.

The following manuscripts were prepared and submitted for publication during the development of this thesis. Full copies of these papers are included in Appendix B.

- Event-Based Direction Tracking Using Neuromorphic Methods, submitted to NeurIPS 2024, Rejected.
- Event-Based Orientation Tracking Using Neuromorphic Methods, submitted to WACV 2025, Rejected.

Contents

Abstract	ii
Acknowledgement	iii
Author’s Declaration	v
Glossary	xxii
1 Introduction	1
1.1 (Spiking) Neural Networks	1
1.1.1 The Neuron	3
1.1.1.1 Synapses	4
1.1.1.2 Neuronal Dynamics	5
1.1.2 Neuron Models	7
1.1.2.1 Hodgkin Huxley Model	9
1.1.2.2 Leaky Integrate-and-Fire (LIF) Model	12
1.1.3 Topology	15
1.1.3.1 Feedforward Networks	15
1.1.3.2 Recurrent Networks	16
1.1.4 Neural Coding	17
1.1.4.1 Rate Coding	17
1.1.4.2 Temporal Coding	17
1.1.5 Network Training	19
1.1.5.1 Supervised Learning	20
1.1.5.2 Unsupervised Learning	21
1.1.5.3 Reinforcement Learning	21
1.1.5.4 Conversion Methods	22
1.1.5.5 Delay Learning	23
1.2 Neuromorphic Computing and Event-Based Sensing	24
1.2.1 Neuromorphic Platforms	24
1.2.2 Event-Based Sensors	25
1.3 Navigation	27
1.3.1 Simultaneous Localisation and Mapping	27
1.3.2 Visual Odometry	28
1.3.3 Visual Inertial Odometry	28
1.3.4 Related Tasks	28

1.4	System	28
1.4.1	System Limitations	30
1.4.2	System Contributions	30
1.5	Thesis Structure	30
2	Background	33
2.1	Biological Representations of Space	33
2.1.1	Head Direction Cells	34
2.1.2	Grid Cells	35
2.1.3	Place Cells	36
2.2	Neuromorphic Vision Systems	37
2.3	Navigation Systems	39
2.3.1	Event Camera based Navigation	40
2.3.2	Neuromorphic Computation in Navigation	40
2.3.3	Combining Event Cameras and Neuromorphic Comput- ing for Navigation	41
2.4	Event Camera Data	42
2.5	Challenges	46
3	Direction Tracking	49
3.1	Introduction	49
3.2	Network Connectivity	49
3.3	Testing	56
3.3.1	Angular Velocity Analysis	56
3.3.2	Network Tuning	57
3.4	Results	61
3.5	Discussion	68
4	Velocity Integration	71
4.1	Introduction	71
4.2	Network Connectivity	73
4.3	Testing	76
4.3.1	Input Signals	76
4.3.2	Speed Analysis	77
4.3.3	Network Tuning	80
4.4	Discussion	83
5	Position Tracking	85
5.1	Introduction	85
5.2	Network and Training	85
5.2.1	Distance Analysis	86
5.2.2	Training Specification	89
5.2.3	Input Signals	90
5.3	Results	95
5.4	Discussion	98

6	Visual Place Recognition and Reset Mechanism	101
6.1	Introduction	101
6.2	Network Structure	102
6.3	Network Operation	104
6.4	Results	106
6.5	Discussion	112
7	Conclusion	115
7.1	Contributions	116
7.2	Future Work	117
	Bibliography	121
	Appendix	145
A	Additional Figures	145
A.1	Head Direction Network Shift Response	146
A.2	Head Direction Network Activity	152
A.3	Head Direction Network Errors Against Time	160
A.4	Artificial Grid Signals	168
A.5	Place Cell Network Activity	176
A.6	Visual Place Recognition Network Reset Signals	192
A.7	Head Direction Network Activity with a Reset Signal	208
A.8	Head Direction Network Errors Against Time with a Reset Signal	216
B	Submitted Manuscripts	225
B.1	Manuscript 1: Event-Based Direction Tracking Using Neur- omorphic Methods	226
B.2	Manuscript 2: Event-Based Orientation Tracking Using Neur- omorphic Methods	234

List of Figures

1.1	A comparison of Artificial Neural Network (a) and Spiking Neural Network (b) processing from input to output at the neuron level.	3
1.2	The regions and direction of travel for activity in a biological neuron.	4
1.3	The Synapse. The point where neurons transmit data.	5
1.4	The order of processes that a biological neuron undergoes in triggering an outgoing spike.	6
1.5	The resting, sub-threshold, spiking, and repolarization behaviour seen in a spiking neurons response.	7
1.6	Examples of analogue neuron activation functions	8
1.7	Given that the neuronal dynamics observed by Hodgkin and Huxley are changes in electrical potentials, the model can also be represented by the circuit diagram shown here.	12
1.8	An equivalent circuit diagram to the LIF neuron model.	15
1.9	A basic example of feedforward and recurrent neural network connectivity.	16
1.10	How different rate codes respond to a stimulus, across runs or population where applicable.	18
1.11	How different temporal codes respond to a stimulus.	19
1.12	The processes that take place when training a neural network using a supervised learning approach.	21
1.13	Long term potentiation and long term depression caused by relative spike timing when using STDP.	22
1.14	The processes involved in training an SNN via conversion methods.	23
1.15	The effect of modulating delay on a Postsynaptic neurons response.	23
1.16	A comparison between the Von Neumann architecture and a Neuromorphic Computing architecture.	25
1.17	The SpiNNaker Project	26
1.18	A Davis346 Event Camera	26
1.19	The outputs of an RGB camera, and event camera in low light scenarios, with negative events shown in black and positive events shown in white.	27
1.20	The design of a neuromorphic navigation system showing how the different spatial networks interact to create localisation signals.	29

2.1	The firing rate of head direction cells in response to different head directions [110]	34
2.2	The hexagonal firing pattern of grid cells occurring at different scales, for different cells, over the same environment [122].	35
2.3	A place cell firing in a single location within an environment [55].	36
2.4	An overview of known spatial cells showing the different firing patterns, their interconnectedness, and how they are distributed through different brain regions [55].	37
2.5	Object recognition performed on event camera data.	38
2.6	The areas of a visual SLAM pipeline specifically identified for improvement through deep learning by [157].	39
2.7	An example of Neuromorphic Computing being applied to uni-dimensional SLAM, showing (a) a breakdown of the system; (b) the connectivity within networks; and (c) the resulting outputs [167].	41
2.8	Combining an event camera with Neuromorphic Computing to achieve control of a vehicle. (a) a breakdown of the system used for simulation; (b) the connectivity within network; and (c) the results of the lane keeping task [176].	42
2.9	The sensor configuration used in the ECMD dataset, and a sample of the resulting recordings [181].	43
2.10	The sensor mount used to record the MVSEC dataset, mounted on a car, motorbike, and UAV [186].	44
3.1	An overview of the layers in the Head Direction network, and how they are connected. For a detailed, neuron level connectivity mapping see Figure 3.5	50
3.2	The synaptic weights between neurons in the head direction layer.	53
3.3	The synaptic weights between the head direction and shift layers.	53
3.4	The synaptic weights between each of the shift layers and the head direction layer.	54
3.5	The connectivity within the Head Direction network at the neuron level with connections focused around the neuron at index 0. Neuron 0 of the Head Direction layer is connected via excitatory connections to its nearest neighbours and inhibits distant head direction neurons. It also inhibits distant shift neurons. The angular velocity input is fed as an excitatory signal directly into the shift layer, but is counteracted in distant neurons by the inhibition. Activity stimulated in Neuron 0 of the Shift layer connects via excitatory connections to Neuron 1 of the Head Direction layer, meaning activity shifts in the Head Direction layer.	55
3.6	The distribution of angular velocity recordings from the ECMD dataset.	57
3.7	The angular velocity distributions for the ECMD dataset broken down by scene.	58

3.8	The effect of increasing the resting potential of a LIF neuron on its f-I response.	59
3.9	The shift rate of the Head Direction network when exposed to constant angular velocity inputs. (a) Inputs from 0 to 1 rad/s at 0.05 rad/s intervals to cover the entire range of angular velocities recorded in the ECMD dataset. (b) Inputs from 0 to 0.1 rad/s to cover the majority of the angular velocities recorded in the ECMD dataset at a 0.005 rad/s increments.	60
3.10	A subset of Head Direction network activity and ground truth data for the ECMD dataset. (a) The sequence resulting in the lowest MAE. (b) The sequence resulting in the highest MAE. (c)-(d) Additional sequences showing accurate direction tracking behaviour.	64
3.11	The error in the Head Direction network where drift over time is observed.	65
3.12	The error in the Head Direction network where a large error appears rapidly and persists, resulting in a continued offset between the ground truth direction and network direction. The beginning of the offset is circled in orange, after which the persistence of the error is visible in the blue error signal.	66
3.13	The mean absolute error plotted against the duration for each sequence of the ECMD dataset.	67
3.14	The distribution of the MAE when grouped by scenario.	68
4.1	An overview of the layers in the Grid network, and how they are connected. For detailed, neuron level connectivity mapping see Figures 4.2, 4.3, and 4.4. If the Grid Position neuron at (5,5) is active then it excites its neighbours and inhibits distant Grid Position neurons. It also inhibits distant neurons in the six Shift layers. The Head Direction signal inhibits the Shift layers that don't match the current Head Direction, meaning, only the Shift neurons at similar (x, y) coordinates, in the Shift layer representing the current Head Direction aren't inhibited. Linear Speed attempts to excite all of the Shift neurons, but only those that aren't inhibited spike. These Shift neurons connect back to the Grid Position layer one step in their designated direction, meaning activity in the Grid Position layer shifts.	72
4.2	The connection polarity between neurons in the grid position layer.	73
4.3	The regions where grid position neurons make inhibitory connections to neurons in the shift layers.	74
4.4	The regions where the neurons in the shift layers make excitatory connections to neurons in the grid position layer.	75
4.5	Estimations of the vehicle's speed based on acceleration recordings from the IMU, and using the position recorded in the ground truth data from the ECMD dataset.	78

4.6	The distribution of speeds estimated from the ground truth data from the ECMD dataset.	79
4.7	The distributions of speeds for the ECMD dataset, grouped by scene.	80
4.8	The HOA in the grid position layer of the Grid network, tested at the centre and fringe of the network.	81
4.9	The Grid network activity shifting as expected along the axis of one of the shift layers to verify the functionality of the shift mechanism.	82
5.1	The distributions of the distances covered in the ECMD dataset, grouped by scene.	88
5.2	The ground truth positions for the ECMD sequences that covered the largest, and smallest amount of distance along each axis.	89
5.3	An overview of the layers in the Place Cell network and how they are connected in the training and deployment phases. The Grid Cells are connected to the place cell network via all to all STDP Synapses, totalling 140,000 Synapses at the start of training. This is a for a network consisting of 200 Place Cells, 100 Grid Position neurons per scale, and 7 scales of Grid. Following training all of these connections are maintained, however many will be 0 or close to 0, and could be pruned for computational efficiency.	90
5.4	The movement of the artificial grid signal as the ground truth position is updated. (a) The ground truth signal is shown on the left and the right shows the artificial grid layer activity. The two areas of activity over each period on the right are a result of how the neurons are indexed and represent a cluster when plotted spatially (see (b)). (b) The network activity plotted spatially from the start to the first ground truth update; between the two ground truth updates; and from the second ground truth update until the end of the simulation, from left to right.	91
5.5	The artificial grid activity showing different behaviour based on the vehicle path.	93
5.6	The effects of the inhibitory, training, and grid signals, that are used during the training phase, on a place cell.	94
5.7	Place Cell network connectivity at the neuronal level during training.	95
5.8	Place cell firing activity and the corresponding ground truth trajectories. (a) The place cell activity when the vehicle exclusively enters new Place Fields. (b) When the vehicle regularly stops along its journey this is also reflected in the place cells. This is circled in purple. (c) When a Place Field is revisited by the vehicle returning along the road in the opposite direction the place cell becomes active again as circled in green, but this behaviour wasn't always shown, see the orange circle in (d).	97

6.1	The structure and connectivity of the visual processing part of the stereo vision network. Showing the cuboid shape of the network and how the two sets of event camera data are passed into the network perpendicularly. Also shown is how the event cameras connect along an entire row.	103
6.2	An overview of the areas of the vision network and how they are connected in the training and deployment phases.	105
6.3	Reset signals from the vision network. (a) The head direction reset signal. (b) The grid cell reset signal with a step size of 10m.	106
6.4	It was seen that when training was run beyond the desired reset signal being achieved that the neuron would quickly saturate, no longer providing a useful reset signal.	108
6.5	The activity in the Head Direction network with and without a reset being induced at the midpoint of the simulation. (a) and (b) show the effective use of a reset to reduce the recorded error. (c) and (d) show where a reset increased the recorded error.	109
6.6	The error distributions by scenario with and without the visual reset signal.	111
A.1	Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.	146
A.2	Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.	147
A.3	Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot. (d) shows all of the measured rotations as angular velocities plotted against the input angular velocity.	148
A.4	Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.	149
A.5	Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.	150
A.6	Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot. (h) shows all of the measured rotations as angular velocities plotted against the input angular velocity.	151
A.7	Head Direction network activity for the ECMD dataset.	152
A.8	Head Direction network activity for the ECMD dataset.	153
A.9	Head Direction network activity for the ECMD dataset.	154
A.10	Head Direction network activity for the ECMD dataset.	155
A.11	Head Direction network activity for the ECMD dataset.	156
A.12	Head Direction network activity for the ECMD dataset.	157
A.13	Head Direction network activity for the ECMD dataset.	158
A.14	Head Direction network activity for the ECMD dataset.	159
A.15	Direction errors recorded in the Head Direction network when run on the ECMD dataset.	160
A.16	Direction errors recorded in the Head Direction network when run on the ECMD dataset.	161

A.17 Direction errors recorded in the Head Direction network when run on the ECMD dataset.	162
A.18 Direction errors recorded in the Head Direction network when run on the ECMD dataset.	163
A.19 Direction errors recorded in the Head Direction network when run on the ECMD dataset.	164
A.20 Direction errors recorded in the Head Direction network when run on the ECMD dataset.	165
A.21 Direction errors recorded in the Head Direction network when run on the ECMD dataset.	166
A.22 Direction errors recorded in the Head Direction network when run on the ECMD dataset.	167
A.23 The artificial grid signals developed from ground truth positions.	168
A.24 The artificial grid signals developed from ground truth positions.	169
A.25 The artificial grid signals developed from ground truth positions.	170
A.26 The artificial grid signals developed from ground truth positions.	171
A.27 The artificial grid signals developed from ground truth positions.	172
A.28 The artificial grid signals developed from ground truth positions.	173
A.29 The artificial grid signals developed from ground truth positions.	174
A.30 The artificial grid signals developed from ground truth positions.	175
A.31 Place Cell network activity accompanied by the corresponding ground truth trajectory.	176
A.32 Place Cell network activity accompanied by the corresponding ground truth trajectory.	177
A.33 Place Cell network activity accompanied by the corresponding ground truth trajectory.	178
A.34 Place Cell network activity accompanied by the corresponding ground truth trajectory.	179
A.35 Place Cell network activity accompanied by the corresponding ground truth trajectory.	180
A.36 Place Cell network activity accompanied by the corresponding ground truth trajectory.	181
A.37 Place Cell network activity accompanied by the corresponding ground truth trajectory.	182
A.38 Place Cell network activity accompanied by the corresponding ground truth trajectory.	183
A.39 Place Cell network activity accompanied by the corresponding ground truth trajectory.	184
A.40 Place Cell network activity accompanied by the corresponding ground truth trajectory.	185
A.41 Place Cell network activity accompanied by the corresponding ground truth trajectory.	186
A.42 Place Cell network activity accompanied by the corresponding ground truth trajectory.	187
A.43 Place Cell network activity accompanied by the corresponding ground truth trajectory.	188

A.44 Place Cell network activity accompanied by the corresponding ground truth trajectory.	189
A.45 Place Cell network activity accompanied by the corresponding ground truth trajectory.	190
A.46 Place Cell network activity accompanied by the corresponding ground truth trajectory.	191
A.47 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	192
A.48 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	193
A.49 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	194
A.50 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	195
A.51 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	196
A.52 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	197
A.53 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	198
A.54 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	199
A.55 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	200
A.56 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	201
A.57 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	202
A.58 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	203
A.59 The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	204

A.60	The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	205
A.61	The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	206
A.62	The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.	207
A.63	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	208
A.64	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	209
A.65	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	210
A.66	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	211
A.67	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	212
A.68	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	213
A.69	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	214
A.70	Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.	215
A.71	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	216
A.72	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	217
A.73	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	218
A.74	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	219
A.75	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	220
A.76	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	221
A.77	Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	222

A.78 Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.	223
--	-----

List of Tables

1.1	This table gives the variables and constants adjusted by Hodgkin and Huxley for the data which they observed from the giant axon of the squid using a resting potential of 0V; the resting potential is now accepted to be -0.065V . To offset the resting potential the reversal potentials must also offset.	9
1.2	This table gives the variables and constants calculated to implement the Leaky Integrate and Fire (LIF) neuron model.	13
2.1	The primary navigation related event camera datasets alongside some non-event camera equivalents that can be used in conjunction with a video to event camera converter.	45
3.1	The connectivity and synaptic weights found within the Head Direction network.	54
3.2	Neuron model parameters used in the Head Direction network.	55
3.3	A summary of descriptive statistics for the angular velocities recorded in the ECMD dataset.	57
3.4	The Mean Absolute Error (MAE) recorded between the Head Direction network and the ground truth readings when applied to the ECMD dataset.	61
3.5	The average MAE when grouping the sequences of the ECMD dataset by time of day, difficulty, and scenario.	67
4.1	The connectivity used in the Grid network	74
4.2	A summary of descriptive metrics for the speeds estimated from the ground truth data from the ECMD dataset.	79
5.1	Key neuron and synapse parameters used in the Place Cell network.	86
5.2	The minimum and maximum distances covered by a single sequence in the ECMD dataset along each axis.	87
6.1	Key neuron and synapse parameters used in the Visual Reset network.	104
6.2	Errors observed in the Head direction network for the ECMD dataset with and without the reset mechanism. Recorded using the Mean Absolute Error (MAE).	109

Acronyms

- ANN** Artificial Neural Network. xi, 1–3, 8, 16, 19, 20, 22, 24, 25, 37–39
- ECMD** Event-Centric Multisensory Driving. xii–xix, xxi, 43, 45, 49, 55–58, 60, 61, 64, 67, 68, 78–80, 82, 83, 85, 87–89, 92, 93, 95, 98, 102, 103, 105–107, 109, 112, 117, 152–167, 192–223
- HOA** Hill of Activity. xiv, 49, 50, 52–55, 58, 61, 63, 73–76, 80–83, 90, 92
- IMU** Inertial Measurement Unit. 40, 43, 44, 76, 77, 115
- LIF** Leaky Integrate and Fire. xi, 8, 14, 15, 23, 50, 55, 69, 84, 116, 117
- MAE** Mean Absolute Error. xiii, xxi, 61, 63, 64, 67, 68, 107, 109, 112, 116
- MEC** Medial Entorhinal Cortex. 16, 35
- SLAM** Simultaneous Localisation and Mapping. ii, xii, 27, 38, 39, 41
- SNN** Spiking Neural Network. xi, 1–3, 8, 12, 16, 17, 20–25, 28–30, 33, 37–40, 46, 115, 116
- STDP** Synaptic Time Dependent Plasticity. xi, xiv, 20, 22, 89, 90, 93–95, 98, 104, 115
- UAV** Unmanned Aerial Vehicle. xii, 27, 44, 118
- VIO** Visual Inertial Odometry. ii, 27, 28, 30, 38, 39, 43, 47, 117
- VO** Visual Odometry. 27, 28, 38, 39, 43

Glossary

Analogue Neuron Used here to refer to a non-spiking artificial neuron model.

8

Cognitive Map The internal map stored by the brain. 36, 37

Membrane Potential The potential difference across the cell wall (membrane) of a neuron. 5–7, 13, 14, 51

Neuromorphic Computing An alternative approach to computing, mimicking information processing in the brain. ii, vii, xi, xii, 1, 2, 24, 25, 33, 41, 42, 49, 116, 117

Place Field The distinct location within an environment represented by a place cell. xiv, 29, 30, 36, 37, 85–87, 90, 92–99, 115–117

Postsynaptic Coming after a synapse. xi, 5, 6, 21, 23, 54, 74

Presynaptic Coming before a synapse. 4, 21, 54, 74, 83

Synapse The connection point between neurons. xi, xiv, 4–6, 21, 22, 24, 25, 52, 73, 77, 86, 89, 90, 94, 98, 104, 113, 116–118

Chapter 1

Introduction

This thesis investigates a neuromorphic approach to localisation that operates entirely on events, from the sensing inputs to the localisation signals at the output. The system operates using a series of networks that take inspiration from the mammalian navigation system, emulating head direction, grid cell, and place cell activity to integrate inertial cues to track direction and position. Additionally, these networks are paired with a stereo event-based visual place recognition network that provides loop closure by triggering reset signals when known scenes are encountered. The testing and development of the system has been done using a real-world driving dataset recorded with event cameras, providing insight into whether such a system is viable. By maintaining the asynchronous, sparse, event-based representation throughout, the work demonstrates how bio-inspired networks can be translated into practical localisation systems driving tasks and this can be extended to robotics and other navigation scenarios.

This chapter serves to introduce the technologies and concepts that contribute to the system, without reviewing specific implementations (for this, see Chapter 2). Initially Spiking Neural Network (SNN)s are introduced, with some crossover to the more widely known Artificial Neural Network (ANN)s. Next, this is built on by introducing the subject of Neuromorphic Computing. Following this, different approaches to navigation tasks are discussed, based on what is tracked, recorded, and mapped. Finally, we introduce the system proposed in this work and detail the structure of the remainder of this thesis.

1.1 (Spiking) Neural Networks

As technology has advanced, the terms artificial intelligence and machine learning have become prevalent throughout everyday life, with all the largest companies utilising these technologies in some manner. However, they are extremely general catch-all terms that often refer to the implementation of Artificial Neural Network (ANN). These are a form of neural network that has seen broad use due to the development of effective learning methods but operates in a way that is greatly abstracted from biological neural networks

[1], [2]. A form of neural network which is currently less commonly used, but is truer to its biological counterpart is Spiking Neural Network (SNN) [3]. In SNNs, the data is encoded and transmitted in the form of spikes, also known as events or firing, in a way that emulates the behaviours of biological neurons [4].

The lack of abundance of SNNs can be attributed to the challenge of implementing effective learning algorithms for these models, due to the nature of their information encoding and operation [5]. Despite this, SNNs have been described as the third generation of neural networks [6], with ANNs being their predecessor. This is due to the beneficial properties of spike encoding and spiking neuron operation [7], which has led to their ubiquity for computation in biology, in the form of the nervous system.

The primary advantages of SNNs are low power operation [8] and minimal latency [9] due to sparse, asynchronous representation [10]. Spiking neurons are extremely power efficient relative to artificial neurons as power must only be drawn for the duration of a spike, resulting in a finite number of short power draws compared to the continuous load required for an artificial neuron's output to be maintained; see Figure 1.1. This occurs because the neuron performs temporal integration on incoming spikes, resulting in asynchronous, sparse outputs, unlike the continuous output of an artificial neuron. Latency is also minimised in the response time of SNNs due to the asynchronous operation of spiking neuron models. Beyond the intrinsic properties of SNNs, spike coding schemes can also reduce the delays caused by information propagating through a network, certain schemes able to operate on the timing of a single spike [11]; a biological example is reflex reactions. Finally, the density of information conveyed through spikes can exceed that of an analogue signal due to the variety of encoding mechanisms, with rate codes (see Section 1.1.4.1) analogous to analogue values. These advantages continue to be exploited as more efficient computing platforms are introduced [12]; new coding schemes are implemented [7], [13]; and more effective learning algorithms are developed [14], [15].

Initially SNNs were used as a tool to model biological neural networks by the computational neuroscience community [16]. More recently, the Neuromorphic Computing community has developed from the initial work of Carver Mead [17], employing SNNs as a computing paradigm for engineering tasks.

With an application-driven approach, the neuromorphic community has progressed by producing an alternate computing paradigm [18], [19] by mimicking biological neuron mechanisms through hardware, with prominent examples being Loihi [20], TrueNorth [21], and SpiNNaker [22]. In addition to developing computing platforms, there have also been many contributions to improving learning algorithms for SNNs [14], [15], [23], both biologically plausible [24] and implausible [25]. Frequently, the ability of biological neural networks surpasses that of their neuromorphic counterpart [26], [27], [28] and, as such, offers a strong basis for inspiration. In addition to improving the capabilities of networks, there has also been a continued growth of applications for SNNs [29].

Given the derivation of SNNs from biological neurons, the basis of biological computation is introduced below. This is followed by a more in depth introduction to Neuromorphic Computing, and Navigation, before they are

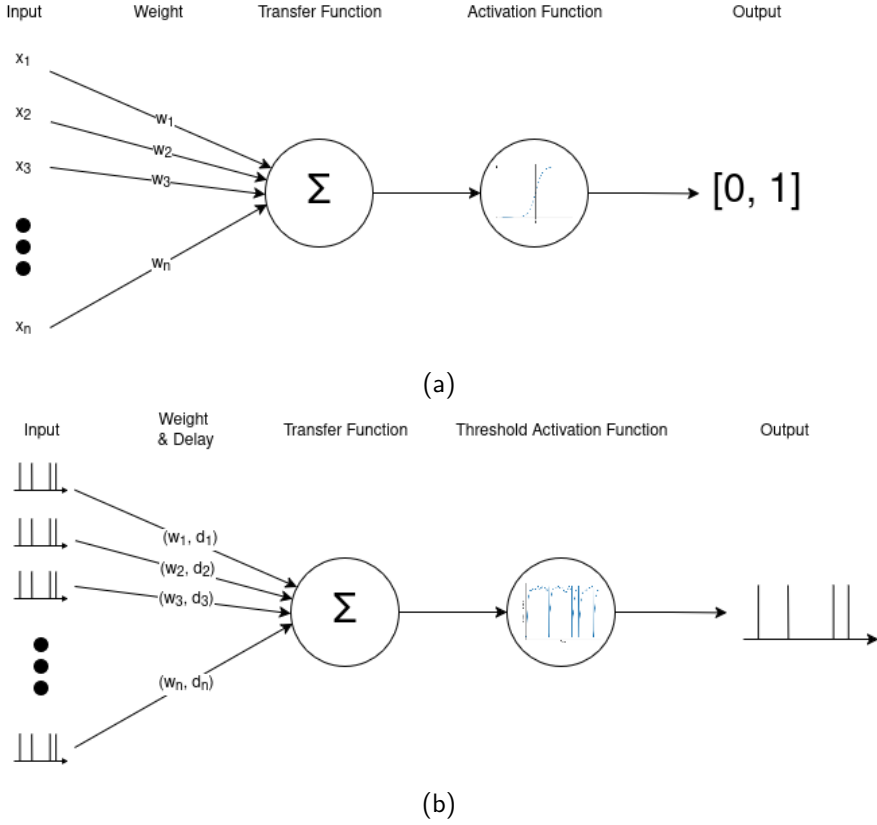


Figure 1.1: A comparison of Artificial Neural Network (a) and Spiking Neural Network (b) processing from input to output at the neuron level.

expanded upon in the next Chapter 2. Due to the complexity of the central nervous system and its subsystems, there is considerably more depth than can be detailed here, and indeed than we are aware of. As such, further reading from a cognitive neuroscience perspective can be obtained from [1], [30], and, with a more relevant focus, for this project, on computational neuroscience from [31].

1.1.1 The Neuron

Neurons are a class of cell that are specialised for signal processing [7], [32], see Figure 1.2. At an individual level neurons operate by receiving an input, performing basic processing, and potentially giving an output based on its state and the input. Despite the simplicity of a single neuron's operation, just a small number of interconnected neurons can exhibit complex behaviour due to the abstraction that occurs as a signal propagates through a network. Information can be represented across a network with different areas responding

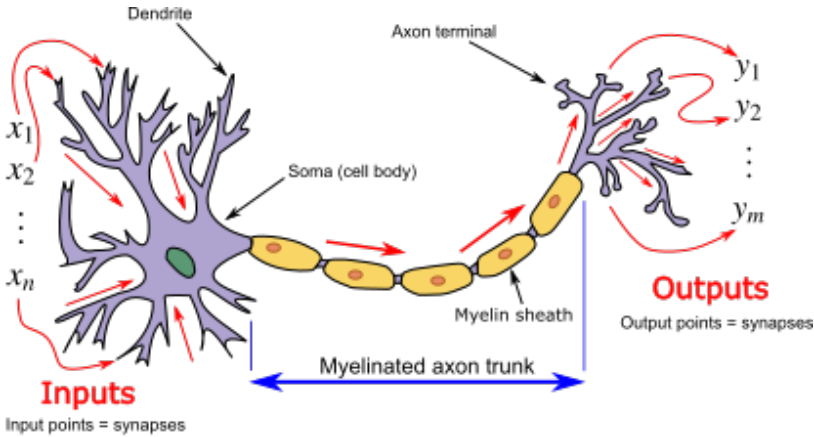


Figure 1.2: The regions and direction of travel for activity in a biological neuron.

to different stimuli, creating a low-level representation of the input. Together, multiple low-level representations of information can conjunctively stimulate other areas of the network, creating a higher-level abstraction resulting from the information propagating from multiple sources through specialist networks.

A neuron consists primarily of three distinct regions, dendrites, soma, and axon; the dendrites and axon together are called neurites. When a neuron receives incoming spikes and is stimulated sufficiently to produce a spike itself, the incoming spikes are received in the dendrites, pass through the soma, and the outgoing spike is transmitted along the axon, to be output to connected neurons.

Dendrites form tree-like structures that extend from the soma to make incoming connections with a Presynaptic neuron. The sites at which dendrites receive an input are spines and are located across the length of the dendrite trees. The soma is an alternative name for the cell body of a neuron and therefore contains the typical organelles of a cell, such as the nucleus and mitochondria, suspended in cytoplasm consisting predominantly of potassium (K^+), sodium (Na^+), chlorine (Cl^-), and calcium (Ca^{2+}) ions. In contrast to the many dendrites that branch from the soma to receive inputs, each neuron only has a single axon to transmit outgoing signals; however, multiple pathways branch from the axon, known as axon collaterals.

1.1.1.1 Synapses

The Synapse is the point at which an axon collateral interacts with a dendritic spine for the passing of signals either between neurons, see Figure 1.3, or in the case of a cyclical connection, from a neuron's output to its own input. As a result of this naming, the neuron giving the output, i.e. coming before the Synapse in the network, is called the Presynaptic neuron, and the neuron receiving the input, i.e. following the Synapse in the network, is called the

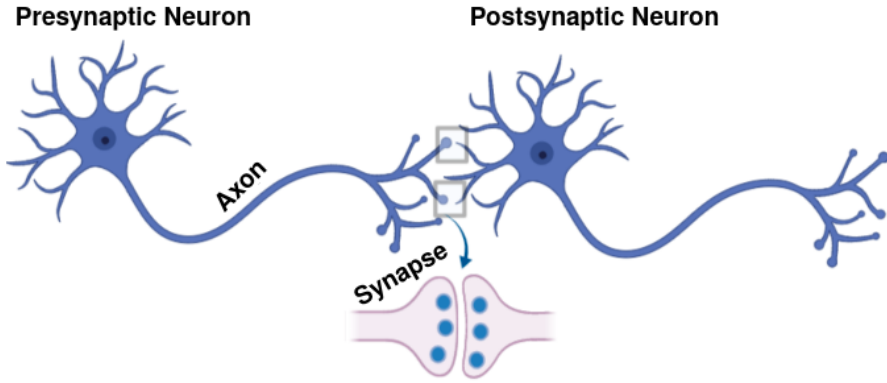


Figure 1.3: The Synapse. The point where neurons transmit data.

Postsynaptic neuron.

Despite the simplicity of the concept of Synapses, they form an important part of the complexity of a neural network. Not only can the signal be passed via different methods, either electrically or chemically, depending upon the Synapse, but they also give the polarity of a connection between neurons. Connections within a neural network are predominantly excitatory, causing an increase in Postsynaptic Membrane Potential, but some can be inhibitory, causing a decrease in Postsynaptic Membrane Potential. This increases the potential complexity of the networks, allowing more complex dynamics to occur. The distribution of excitatory to inhibitory synaptic connections can vary depending on the type of neuron, which dictates its function, with values ranging from approximately 5-30% of the connections being inhibitory [33], [34].

1.1.1.2 Neuronal Dynamics

Information propagates through biological neural networks in the form of spikes; however, neurons do not simply repeat incoming spikes, and spikes travelling within a neuron experience a loss in strength. Therefore, there are internal processes performed by the neuron that create spikes and maintain their signal strength through regenerative mechanisms as they propagate along the axon. These signals are measured as an electrical potential across the neuronal membrane caused by differences in concentrations of the ions held in the intracellular cytoplasm and the ions found in the extracellular fluid. The processes involved in a biological neuron spike are described below and show in Figure 1.4.

Although most of the neuronal membrane is impermeable to the cytoplasm and extracellular fluid, to enable changes in the Membrane Potential, the membrane is bridged by many transmembrane proteins. The transmembrane proteins perform one of two roles, acting as an ion pump or as an ion channel. Ion pumps actively and selectively transfer ions against the concentration

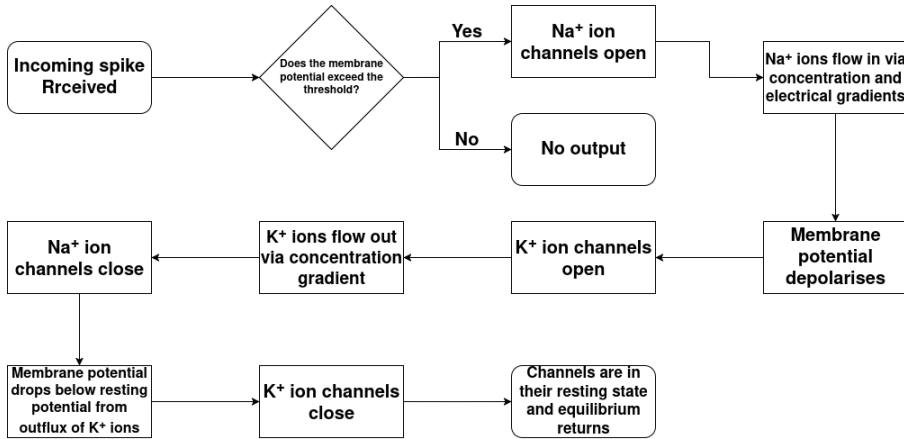


Figure 1.4: The order of processes that a biological neuron undergoes in triggering an outgoing spike.

gradient and consume energy to do so. Ion channels allow the flow of select ions through the neuronal membrane at a rate dictated by the permeability of the channel. The permeability of the ion channels throughout the neuronal membrane can be changed on the basis of factors such as transmembrane voltage and chemical or physical stimuli, making them gated ion channels. The neuronal membrane offers the highest permeability to K^+ ions because the membrane features more K^+ selective channels, relative to the other ion channels. With the neuron in its resting state, the concentration of K^+ is higher within the neuron, while the concentrations of Na^+ and Cl^- are higher externally.

A negative resting Membrane Potential occurs because the membrane is more permeable to K^+ than to Na^+ . When combined with the higher internal concentration of K^+ , the result is a flow of positive ions that leave the neuron creating a negative Membrane Potential. This negative Membrane Potential gives an electrical gradient in the opposite direction to the concentration gradient, allowing an equilibrium to form, which results in a resting Membrane Potential of approximately -70mV for a typical neuron.

For a neuron to transmit a spike, its Membrane Potential must exceed a threshold. For this to occur, the neuron must first be stimulated at its incoming Synapses, resulting in the opening of the ion channels in the dendrite. This opening causes an influx of ions that will be passively conducted through the neuron, known as an excitatory Postsynaptic potential. Following the excitatory Postsynaptic potential being conducted to the spike-triggering zone in the axon hillock (where the axon emerges from the soma), if the Membrane Potential exceeds the threshold, the Na^+ gated ion channels will open, causing an influx of Na^+ ions. Na^+ ions will flow rapidly through the channel because they travel down both the electrical and concentration gradients. Once the Membrane Potential has undergone depolarization due to the influx of positive Na^+ ions, a threshold will be reached causing the K^+ gated ion channels to open, allowing

K^+ ions to flow out of the neuron down the concentration gradient. The outflow of K^+ ions reduces the Membrane Potential and, combined with the closing of Na^+ channels, causes the Membrane Potential to repolarize and drop below the resting Membrane Potential. This decreased Membrane Potential triggers the closure of the K^+ channels, at which point the channels have returned to their initial states. As a result, the neuron will slowly return to its resting Membrane Potential with the ions returning to their original concentrations. This has also been explained diagrammatically in Figure 1.4

As the Membrane Potential increases with the incoming spikes, the ions react linearly to their respective concentration, and electrical gradients, resulting in the response to a spike being a logarithmic increase in the Membrane Potential. Due to the same passive mechanisms, when the Membrane Potential is above the resting potential but is not actively stimulated, it undergoes an exponential decay. In contrast to this passive sub-threshold behaviour, once the threshold is reached and triggers the active mechanism of opening the ion channels, the Membrane Potential will always produce a spiking output, giving an all-or-nothing behaviour [35]. These behaviours are shown in Figure 1.5

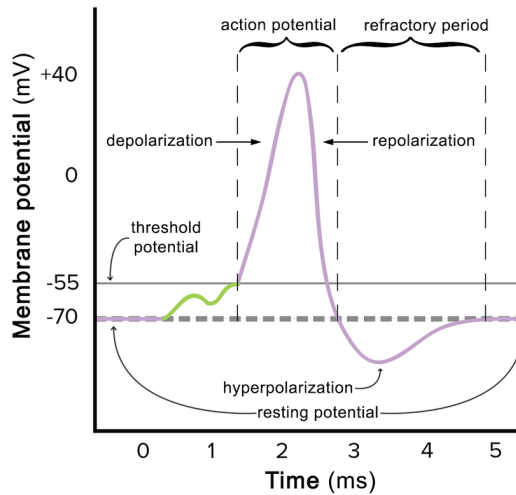


Figure 1.5: The resting, sub-threshold, spiking, and repolarization behaviour seen in a spiking neurons response.

1.1.2 Neuron Models

Mathematical models for neurons have been present in the literature for many decades, with early prominent work by Hodgkin and Huxley [35], McCulloch and Pitts [36], and Lapique [37]. The first model of a biological neuron is attributed to McCulloch and Pitts [36], [38].

The McCulloch and Pitts neuron operates by summing all input signals with the Membrane Potential until a threshold was reached, at which point an all-or-nothing binary output was given and the membrane voltage was reset [36].

As a result of the all-or-nothing nature of the output, data cannot be encoded in the form of the outgoing signal; only its timing and a lack of leakage when summing inputs synchronises incoming data, eliminating temporal information stored through asynchronicity. Therefore, in comparison to modern neuron models, the ability of the signal to carry data was severely limited. Due to data transmission limitations, the McCulloch and Pitts neuron was superseded by Analogue Neuron models.

Unlike the McCulloch and Pitts neuron, analogue neurons have been optimised for computation and do not replicate biological signals. As such analogue neurons do not have a standardised output, but instead apply an activation function such as Rectified Linear Unit (ReLU), Gaussian Error Linear Unit, or a Sigmoid function; see Figure 1.6. This results in better data processing, making them a widely used tool in many industries. Due to the focus of this work on SNNs, for a more in depth understanding of ANNs see [39].

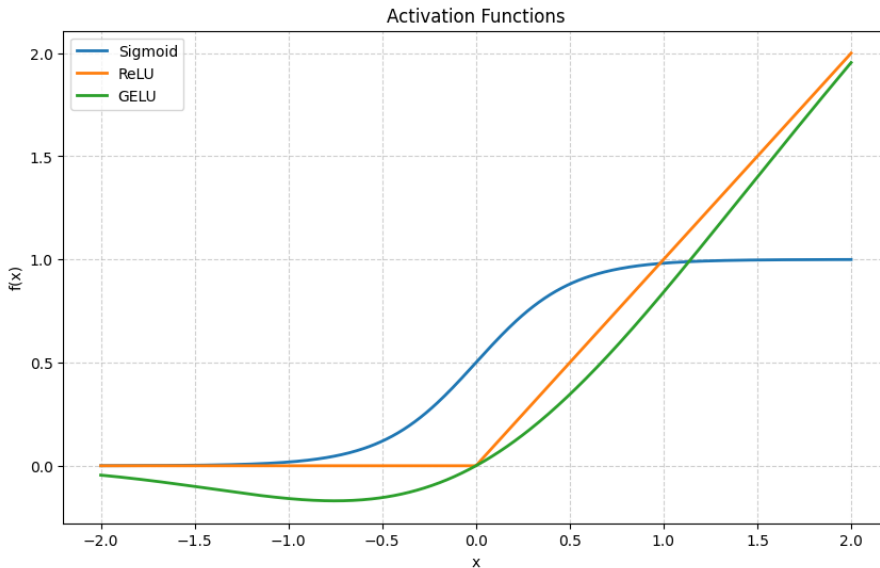


Figure 1.6: Examples of analogue neuron activation functions

Here we introduce the Hodgkin Huxley, and Leaky Integrate and Fire (LIF) spiking neuron models as these are the most relevant to this work, occupying opposite ends of the spectrum. The Hodgkin Huxley model features a set of equations derived empirically to accurately represent biological neuron behaviours, while the LIF model is optimised for computational efficiency, and therefore widely used. For in depth reviews of spiking neuron models that introduce spiking behaviour, and cover a greater range of models, see the works of Gerstner [40] and Izhikevich [41].

1.1.2.1 Hodgkin Huxley Model

The Hodgkin Huxley model aims to accurately replicate the internal mechanisms by which a biological neuron operates, based on experiments performed on a giant axon of a squid [35]. As such, the following equations serve as a mathematical description of the processes described in Section 1.1.1.2; see Table 1.1 for parameter details.

Table 1.1: This table gives the variables and constants adjusted by Hodgkin and Huxley for the data which they observed from the giant axon of the squid using a resting potential of 0V; the resting potential is now accepted to be -0.065V . To offset the resting potential the reversal potentials must also offset.

Description	Symbol	Value	Units
Firing rate	v	n/a	Hz
Time period between spikes	T	n/a	s
Time	t	n/a	s
Input current	I	n/a	mA
Capacitive part of the input current	I_C	n/a	mA
Part of the input current which passes through the ion channels	I_k	n/a	mA
Capacitance of the capacitor (membrane)	C	n/a	F
Charge of the capacitor (membrane)	Q	n/a	C
Voltage across the capacitor (membrane)	u	n/a	mV
Resistance of an unspecified leakage channel	R	n/a	Ω
Sodium ion channel gating variable	m	n/a	n/a
Sodium ion channel gating variable	h	n/a	n/a
Potassium ion channel gating variable	n	n/a	n/a
Derivative of m with respect to t	$\dot{m}$	n/a	n/a
Derivative of h with respect to t	$\dot{h}$	n/a	n/a
Derivative of n with respect to t	$\dot{n}$	n/a	n/a
Asymptotic value given by $\dot{m}$ equation	m_0	n/a	n/a
Asymptotic value given by $\dot{h}$ equation	h_0	n/a	n/a
Asymptotic value given by $\dot{n}$ equation	n_0	n/a	n/a

Time constant given by $\dot{m}$ equation	τ_m	n/a	n/a
Time constant given by $\dot{h}$ equation	τ_h	n/a	n/a
Time constant given by $\dot{n}$ equation	τ_n	n/a	n/a
Sodium ion reversal potential	E_{Na}	115	mV
Potassium ion reversal potential	E_K	-12	mV
Unspecified leakage channel reversal potential	E_L	10.6	mV
Conductance of the Sodium ion channel if open	g_{Na}	120	$\frac{mS}{cm^2}$
Conductance of the Potassium ion channel if open	g_K	36	$\frac{mS}{cm^2}$
Conductance of an unspecific leakage channel	g_L	0.3	$\frac{mS}{cm^2}$
An empirical function of u with respect to m	α_m	$\frac{(2.5-0.1u)}{(e^{(2.5-0.1u)}-1)}$	n/a
An empirical function of u with respect to n	α_n	$\frac{(0.1-0.01u)}{(e^{(1-0.1u)}-1)}$	n/a
An empirical function of u with respect to h	α_h	$0.07e^{(\frac{-u}{20})}$	n/a
An empirical function of u with respect to m	β_m	$0.125e^{(\frac{-u}{80})}$	n/a
An empirical function of u with respect to n	β_n	$4e^{(\frac{-u}{18})}$	n/a
An empirical function of u with respect to h	β_h	$\frac{1}{(e^{(3-0.1u)}+1)}$	n/a

By definition

$$v = \frac{1}{T} \quad (1.1)$$

$$C = \frac{Q}{u} \quad (1.2)$$

$$I_C = C \frac{du}{dt} \quad (1.3)$$

$$g_L = \frac{1}{R} \quad (1.4)$$

The sum of all the input currents is equal to the total input current according to Kirchhoff's first law

$$I(t) = I_C(t) + \sum_k I_k(t) \quad (1.5)$$

By combining (1.3) and (1.5) it is also evident that

$$C \frac{du}{dt} = I(t) - \sum_k I_k(t) \quad (1.6)$$

The conductance of a channel multiplied by the voltage across it gives the current which can flow through it. This must also be combined with the gating variable to determine the probability that it is open to allow the current to flow. This can be summed across the three ion channels to give the total current passing through the ion channels, as described by

$$\sum_k I_k = g_{Na} m^3 h (u - E_{Na}) + g_K n^4 (u - E_K) + g_L (u - E_L) \quad (1.7)$$

The gating variables m , n , and h are described by the differential equations

$$\dot{m} = \alpha_m(u)(1 - m) - \beta_m(u)m = -\frac{1}{\tau_m(u)}[m - m_0(u)] \quad (1.8)$$

$$\dot{n} = \alpha_n(u)(1 - n) - \beta_n(u)n = -\frac{1}{\tau_n(u)}[n - n_0(u)] \quad (1.9)$$

$$\dot{h} = \alpha_h(u)(1 - h) - \beta_h(u)h = -\frac{1}{\tau_h(u)}[h - h_0(u)] \quad (1.10)$$

where

$$\dot{m} = \frac{dm}{dt} \quad (1.11)$$

$$\dot{n} = \frac{dn}{dt} \quad (1.12)$$

$$\dot{h} = \frac{dh}{dt} \quad (1.13)$$

When the voltage u is fixed, m tends towards m_0 according to the time constant τ_m ; n tends towards n_0 according to the time constant τ_n ; and h tends towards h_0 according to the time constant τ_h . Giving

$$m_0(u) = \frac{\alpha_m(u)}{[\alpha_m(u) + \beta_m(u)]} \quad (1.14)$$

$$n_0(u) = \frac{\alpha_n(u)}{[\alpha_n(u) + \beta_n(u)]} \quad (1.15)$$

$$h_0(u) = \frac{\alpha_h(u)}{[\alpha_h(u) + \beta_h(u)]} \quad (1.16)$$

$$\tau_m(u) = [\alpha_m(u) + \beta_m(u)]^{-1} \quad (1.17)$$

$$\tau_n(u) = [\alpha_n(u) + \beta_n(u)]^{-1} \quad (1.18)$$

$$\tau_h(u) = [\alpha_h(u) + \beta_h(u)]^{-1} \quad (1.19)$$

According to eqs. (1.14) to (1.16), as the membrane voltage u increases, so do m_0 and n_0 , while h_0 decreases. An increased membrane voltage causes the sodium channels to open, allowing an influx of positive ions, which if sufficiently large causes an action potential. In the case of an action potential, when the membrane voltage u becomes extremely high, the decrease in h_0 causes the sodium ion channels to close; however, this is delayed due to τ_h

being greater than τ_m meaning a slower reaction. Similarly to τ_h , τ_n is also larger and therefore slower to react, which means that when the sodium ion channels begin to close, the potassium ion channels begin to open, causing an outflow of positive ions to drive down u . This results in an undershoot of the resting potential, with u becoming more negative; however, this causes the ion channels to return to their non-spiking states allowing the neuron to return to its equilibrium, known as its resting state. An equivalent circuit diagram of the Hodgkin-Huxley neuron can be seen in Figure 1.7.

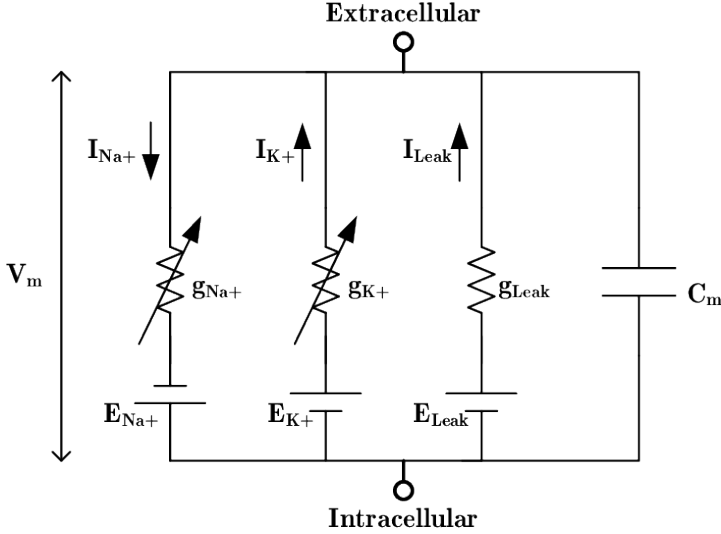


Figure 1.7: Given that the neuronal dynamics observed by Hodgkin and Huxley are changes in electrical potentials, the model can also be represented by the circuit diagram shown here.

1.1.2.2 Leaky Integrate-and-Fire (LIF) Model

Due to the usefulness of SNNs for computation, many simplified mathematical models of neurons have been proposed. They aim to replicate the behaviours exhibited by the Hodgkin Huxley neuron with a reduced number of floating point operations (flops), thus achieving greater computational efficiency. The most commonly used and one of the simplest spiking neuron models is the leaky integrate and fire (LIF) model.

Despite its wide usage for SNNs nowadays, the origins of the equations behind the LIF neuron were first derived from experiments on nerves [37]. The following equations are taken from Gerstner [40], the original work being attributed to Lapique [37] and translated here [42], [43]. For parameter details, see Table 1.2.

Table 1.2: This table gives the variables and constants calculated to implement the Leaky Integrate and Fire (LIF) neuron model.

Description	Symbol
Time	t
Membrane Potential	u
Membrane time constant	τ_m
Spike threshold Membrane Potential	u_{thresh}
Membrane Potential reset value	u_r
Membrane resting potential	u_{rest}
Input current	I
Capacitive part of the input current	I_C
Resistive part of the input current	I_R
Membrane capacitance	C
Charge of the capacitor	Q
Membrane resistance	R
Absolute refractory period	Δ^{abs}
Time period between spikes	T
Spike frequency	v
Time step	k

By definition

$$I_R = \frac{u(t)}{R} \quad (1.20)$$

$$C = \frac{Q}{u} \quad (1.21)$$

$$I_C = C \frac{du}{dt} \quad (1.22)$$

and according to Kirchoff's Law

$$I(t) = I_R + I_C \quad (1.23)$$

We can define

$$\tau_m = RC \quad (1.24)$$

Therefore, by combining (1.20), (1.21), and (1.23)

$$I(t) = \frac{u(t)}{R} + C \frac{du}{dt} \quad (1.25)$$

By multiplying everything in (1.25) by R we get

$$\tau_m \frac{du}{dt} = -u(t) + RI(t) \quad (1.26)$$

Whether a spike is emitted at time $t^* = t_{k+1}$ is then determined according to

$$u(t_k) < u_{thresh} \quad \text{and} \quad u(t_{k+1}) > u_{thresh} \quad (1.27)$$

resulting in

$$u(t) = u_r \quad \text{for} \quad t^* \leq t < t^* + \Delta^{abs} \quad (1.28)$$

Having defined the behaviour of the model and given the conditions which trigger an output spike it is also useful to derive inter-spike interval and therefore the spike rate. According to (1.24), if we integrate (1.26) with respect to t , for a constant input current $I(t) = I_0$ we get

$$u(t) = RI_0 \left[1 - \exp\left(-\frac{t - t^{(1)}}{\tau_m}\right) \right] \quad (1.29)$$

where $t^{(1)}$ is the time of the last spike after which u reset according to $u_r = 0$. According to (1.29) as $t \rightarrow \infty$, $u(\infty) = RI_0$ so $RI_0 > u_{thresh}$ for further spiking to occur. This means

$$u_{thresh} = RI_0 \left[1 - \exp\left(-\frac{t^{(2)} - t^{(1)}}{\tau_m}\right) \right] \quad (1.30)$$

with $t^{(2)}$ being the timing of the second spike. According to this statement $T = t^{(2)} - t^{(1)}$. So when including the absolute refractory period

$$T = \tau_m \ln \frac{RI_0}{RI_0 - u_{thresh}} + \Delta^{abs} \quad (1.31)$$

meaning

$$v = \left[\Delta^{abs} + \tau_m \ln \frac{RI_0}{RI_0 - u_{thresh}} \right]^{-1} \quad (1.32)$$

since

$$v = \frac{1}{T} \quad (1.33)$$

The LIF neuron model, mathematically described above, is equivalent to a simple RC circuit, where the membrane capacitance C represents the ability of the membrane to integrate incoming currents and the membrane resistance R models the leak of charge through ion channels. The Membrane Potential corresponds to the voltage across the capacitor, which rises as current is injected and decays exponentially towards the resting potential when no input is present. This provides an intuitive physical interpretation of the LIF dynamics, where the charging and discharging behaviour of the RC circuit mirrors the integration and leakage processes of a biological neuron; see Figure 1.8.

Given that the purpose of the LIF neuron is to replicate the behaviour of the Hodgkin Huxley neuron, with lower computational overheads, the behaviour should be compared. The LIF model represents neuronal behaviour with a simplified circuit, where the Membrane Potential integrates incoming current until it exceeds a threshold, at which point a spike is emitted and the potential is reset. This captures the essential temporal integration and threshold behaviour with low computational cost. In contrast, the Hodgkin Huxley model provides a detailed description of action potential generation by explicitly modelling the dynamics of ion channels through differential equations. This enables it to reproduce the complete range of spiking behaviours, including refractory

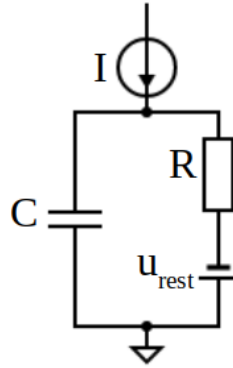


Figure 1.8: An equivalent circuit diagram to the LIF neuron model.

periods, bursting, and frequency adaptation, which the LIF model cannot capture. However, this comes at significant computational expense, making the Hodgkin Huxley model less practical for neuromorphic systems. Given that the LIF neuron offers most spiking neuron behaviours at a low computational cost, and the Hodgkin Huxley model offers biologically accurate behaviour, and the choice between them depends on whether the modelling priority is computational efficiency or physiological accuracy.

1.1.3 Topology

In the most basic sense, structurally, neural networks fall into one of two categories, feedforward and recurrent neural networks; see Figure 1.9. Feedforward neural networks contain synaptic connections which exclusively pass information forward through the network, from the input layer to the output layer. Recurrent neural networks may contain many feedforward connections, but these networks also contain at least one connection not towards the output. Such connections can return information to previous layers or to an input on the same layer, e.g. a neuron may have a recurrent connection that connects its output to its own, or a neighbouring neuron's input. Due to the unlimited range of connections available, recurrent neural networks can produce far more complex dynamics, which means that they are capable of approximating any dynamical system [44]; and are typically what is observed in biology [45]. However, this comes at the cost of much greater complexity when both training and analysing network activity.

1.1.3.1 Feedforward Networks

As stated previously, in a feedforward network, every synaptic connection is directed away from the input layer and towards the output layer. This allows for simplicity of design and analysis, avoiding the risk of a black box, and

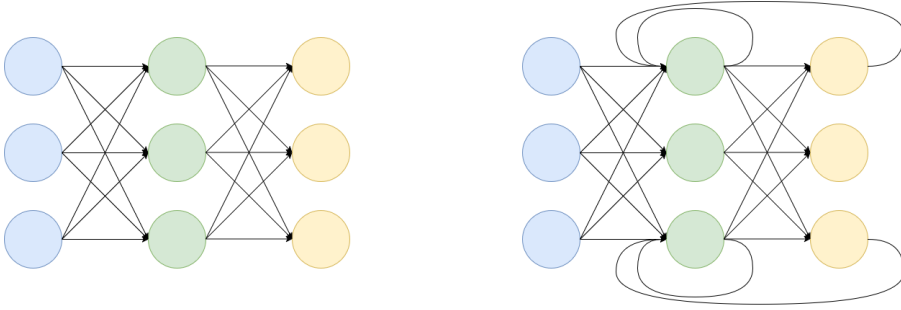


Figure 1.9: A basic example of feedforward and recurrent neural network connectivity.

provides ease of training, which in turn can result in an effective network.

It is through feedforward networks that neural networks have grown to be so widespread. The most common example being the use of convolutional neural networks in computer vision tasks [46], [47]. As with most successful aspects of non-SNNs, this has prompted replication in SNN research [48], [49], [50]

1.1.3.2 Recurrent Networks

Despite the ability to create richer dynamics, recurrent neural networks are less prevalent in engineering applications than feedforward networks. This is because the added complexity of the connections can significantly increase the difficulty of training [51]. For example, if one was trying to implement backpropagation [52], a prominent training algorithm in ANNs, infinite loops would be encountered causing the training to fail. From the perspective of network dynamics, the introduction of recurrent connections allows the retention of past information, to inform decisions rather than solely operating on new information [53]. As a result of the complexity of connections in recurrent neural networks, there are several classifications which can be made within this category; usually forming bio-inspired networks, probability-based connectivity, and reservoir networks.

Due to the derivation of neural networks from biological activity, simplified biological neural network architectures are often used when a task has a known biological equivalent for which the biological neural network architecture is known [54]. This results in many recurrent neural network architectures, as biological neural networks are both large and complex, as to be optimal for their given task and sufficiently general for versatility. An example of this is the use of different specialised neurons and structures for navigation tasks, taking inspiration from activity observed in the hippocampus and Medial Entorhinal Cortex (MEC) [55].

When there isn't a biological inspiration for the neural network, or sparse connectivity is desired, it is typical for SNN connectivity to be modelled using the distance between neurons as a function for the probability of them forming

a synaptic connection [56]. Due to connections being modelled solely as a function of distance, recurrent connections are typically formed within these models. This is intentional and motivated by the capabilities of recurrent neural networks to create more complex dynamics, but also introduces stochasticity, enabling exploratory behaviour through the network design rather than strictly dictating how a network should behave.

For the same reasons as described for probability-based connectivity, SNNs are also used in reservoir computing systems, where there is an input layer, a randomly connected reservoir, and an output layer. The connections within the reservoir are made randomly and training in the network is only performed on connections made to the output layer. The purpose of this, similar to probability-based models, is to take advantage of the complex dynamics that naturally occur in the recurrent neural network formed by the reservoir, while still having a simpler feedforward network to train between the reservoir and the output layer [57], [58].

1.1.4 Neural Coding

Typically, neural coding in SNNs can be considered either rate encoding or temporal coding. Although both neural coding schemes attempt to convey the same information, they do so in different manners. Rate encoding is typically based on the number of spikes over a given duration, whereas temporal codes aim to convey the information in the specific timing of either one or many spikes.

1.1.4.1 Rate Coding

Rate codes are typically implemented using the frequency over a specified duration to denote the information; see Figure 1.10. This form of coding translates well to traditional neural networks because of the inherent ability to translate a spike rate to an analogue value, as well as its inability to express rapid variations. Density codes are also an implementation of rate coding, utilising the same coding scheme, however, also averaging this encoding over multiple runs, thus making it biologically implausible. Population rate coding again uses the frequency of spikes but then averages this frequency across a population of neurons.

1.1.4.2 Temporal Coding

In contrast to rate codes, temporal coding schemes vary significantly in implementation [59]; see Figure 1.11. This is because there are many ways to encode information in the specific timing of spikes, such as delays, intervals, ordering, and timing relative to a global signal. This has resulted in the following general temporal coding schemes: time to first spikes, inter-spike interval, rank order coding, phase coding, binary coding, and temporal contrast.

Time to first spike uses the delay between the first stimulus of the neuron and the generation of an outgoing spike to encode information, with an inverse relationship between input strength and delay, meaning a large (small) input

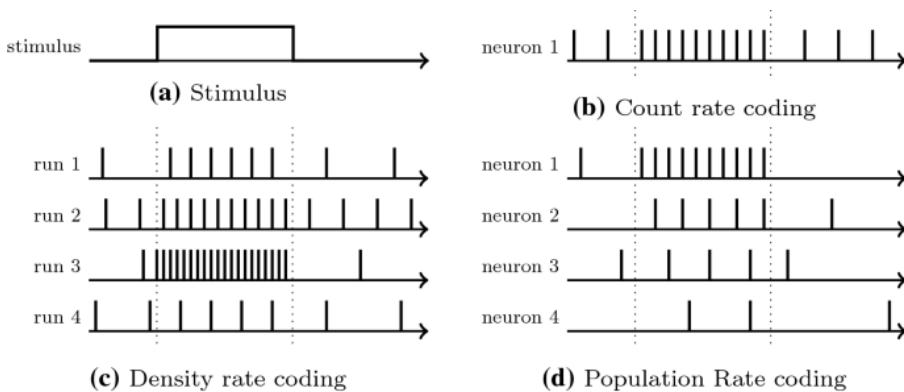


Figure 1.10: How different rate codes respond to a stimulus, across runs or population where applicable.

results in a short (long) delay [60]. This type of encoding has been observed biologically with the encoding of signals in the retina [61]. Time to first spike is limited because the first spikes are the only ones that are considered, and the following spikes are redundant, although for this reason it is also efficient. The efficiency arises because all of the information is encoded in a single spike. Also, in the case of small stimuli, there is also a slow neuronal response, introducing latency to the network.

Inter-spike interval uses a similar encoding to time to first spike; however, in contrast to encoding using the delay between the stimulus and the first spike, it uses the latency between spikes. This method removes the limitation of only encoding information in the first spike, allowing more information to be encoding and thus transmitted through the network [62].

Rank order coding uses a population of neurons with information encoded in the order in which the neurons emit their first spike [63]. With the information encoded in the relative timing of the first spike, this coding scheme has parallels to the encoding of time to first spike although rank order coding encodes information in the relative timing across a population [64], while time to first spike operates on a single neuron irrespective of the larger population.

Phase coding uses the timing of a spike relative to a global signal, such as existing rhythms of network activity (a biological example of this being theta rhythms; for details, see [65]), to denote the value represented a single spike [66], [67]. The information is encoded by the timing of the spike having a corresponding value in the global signal, meaning that the value of the global signal at the time of the spike denotes the output value. Encoding in this manner can introduce latency to the network as the neuron is only able to operate in time with the oscillation of the global signal.

Binary coding uses the timing of spikes to indicate binary bits, and has two common methods of implementation. The first method dictates that a period of time is separated into equal length windows and that the presence (absence) of a spike within a window indicates a one (zero); resulting in a

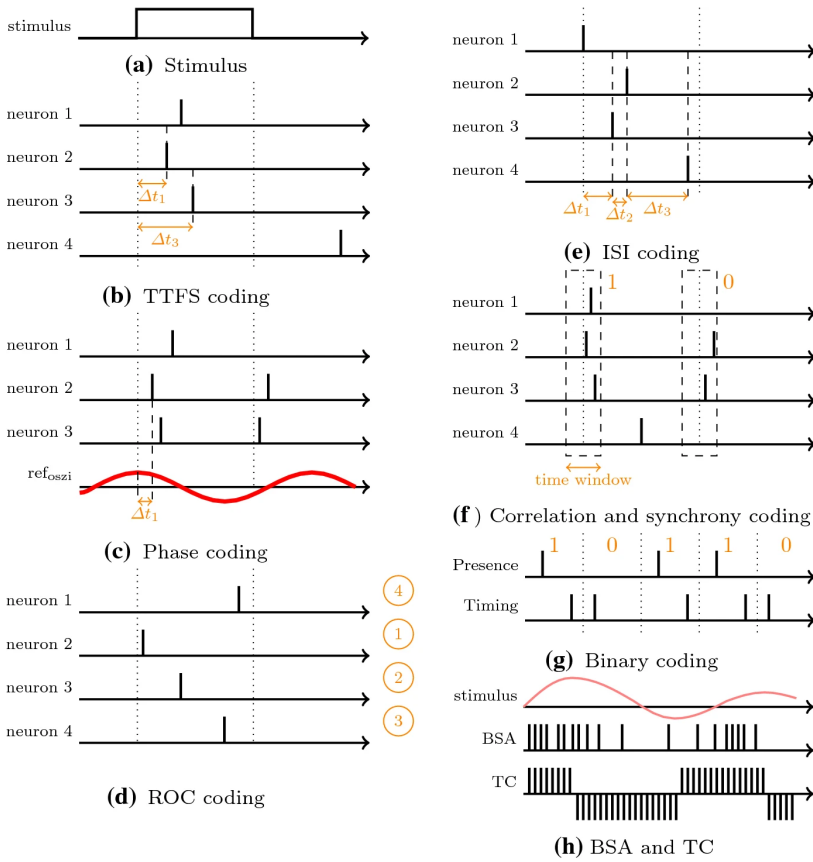


Figure 1.11: How different temporal codes respond to a stimulus.

constant stream of bits. The second implementation also separates the time period into windows; however, now the timing of the spike within the window indicates a one or zero, e.g., a spike in the first (second) half of the window indicates a one (zero) [7]. This second implementation allows for gaps in the bit stream in the absence of any spike within a window. Relative to other coding schemes, binary coding is limited, with the drawbacks being the ability to only transmit binary signals and the introduction of latency, similar to those exhibited by phase coding.

1.1.5 Network Training

Learning occurs in ANNs through the modulation of synaptic characteristics, such as weight and delay. However, this modulation can be influenced in a variety of ways resulting in a plethora of learning algorithms. Due to the range of algorithms they can be separated, with how the learning occurs being the defining characteristic. This gives supervised, unsupervised, and reinforcement learning algorithms. Although these classifications can be used across all ANNs,

SNNs can also be trained using a two step conversion method, with the first step using one of the aforementioned learning classifications to train an ANN which is then converted to an SNN.

Another defining characteristic of a learning algorithm is its ability to perform online learning. An offline learning algorithm undergoes a training phase, following which the network parameters are set. This means that once the network has completed its training phase and the synaptic weights and/or delays have been adjusted, it will be deployed and will not undergo further training. During deployment, all synaptic weights and delays remain constant unless another training phase is initiated. The motivations for employing such a method of training may be reduced complexity and power draw once deployed due to there being no need for reconfiguration; external feedback being unavailable to the network following deployment; or that sufficient accuracy can be achieved through a training phase.

In contrast, a network using online learning can continue training while performing its designated task, but may still undergo a training phase prior to its initial deployment [68]. This continued training allows the network to adapt to potentially changing demands and allows the accuracy to increase because it does not have a limited training dataset. For this to occur, the network must be able to learn in the absence of external aids, thus limiting the available learning techniques available for training. Due to a smaller range of possible algorithms, online learning can be less accurate and therefore less suitable for certain applications.

1.1.5.1 Supervised Learning

Supervised learning takes place when the network is exposed to a training set which has a known set of corresponding outputs. When the network assesses an input from the training set and provides an output, this is compared with the corresponding, correct output, and the network is adjusted to minimise the difference between the network's output and the correct output; see Figure 1.12. In SNNs the synaptic properties of the network can be adjusted by through local learning rules such as Synaptic Time Dependent Plasticity (STDP) [23].

In traditional neural networks backpropagation [52], a specific form of supervised learning, has shown great success and has therefore inspired many attempts at replication for training SNNs [69], [70], [71], [72]. In backpropagation, the comparison of the network output with the correct output provides an error function that is minimised by iterating through the layers via gradient descent to minimise the error at each network layer [72]. However, this can only be done because the activation functions in an ANN are differentiable, which isn't true of spiking neurons. Despite this, there is continued work and success in using alternative error signals that are used as approximates to inform training [73]. Beyond adaptations of backpropagation, a range of supervised learning approaches have been explored in SNNs, each using the temporal dynamics of spikes in different ways. Ponulak and Kasinski [23] build on Hebbian learning [74], combining local timing-based learning rules with a global signal to guide learning through the desired spike patterns.

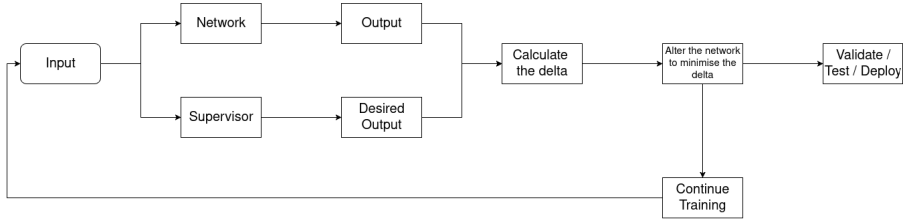


Figure 1.12: The processes that take place when training a neural network using a supervised learning approach.

1.1.5.2 Unsupervised Learning

Unsupervised learning occurs when a neural network is trained without receiving any feedback on the quality of the output or information about the task to be achieved [75], [76]. As such, in SNNs, the modulation of Synapses is entirely based on the occurrence of local events within the network. This results in behaviours of the network being based upon perceived patterns in the features of inputs.

In SNNs, unsupervised learning methods are typically derived from Hebbian learning [74]. Hebbian plasticity is often associated with the phrase "neurons that fire together, wire together"; the meaning of this is that when a Presynaptic neuron is activated, followed closely by a Postsynaptic neuron activation, the Synapse experiences long-term potentiation, which means that the synaptic weight connecting the two neurons is increased. Despite being a long-standing concept, the principles of Hebbian plasticity still form the basis for new algorithms [77], [78].

Based on Hebbian learning, *acrshortstdp* also dictates that should a Postsynaptic neuron fire shortly before a Presynaptic neuron, then the synaptic weight connecting neurons decreases [79]; see Figure 1.13. In an extreme case, *acrshortstdp* can also be used to prune connections, that is, the synaptic weight that connects neurons is reduced to zero [80]. Beyond this, *acrshortstdp* has also evolved to encompass many variations that account for additional complexities such as the use of more than two spikes and modulating delays [81].

1.1.5.3 Reinforcement Learning

Reinforcement learning can be seen to take a middle ground between supervised and unsupervised learning, as the network is not tuned to meet an ideal response, but does still receive external feedback indicating the quality of its output. However, unlike supervised learning the reinforcement signal is sourced organically from the environment. A network implementing a reinforcement learning algorithm receives an input, to which it will respond with an output; the environment will then provide a reinforcement signal, which could be either positive or negative, and the network will adapt in an attempt to maximise the reinforcement signal in the future. The release of dopamine is an example of a biological reward signal that is observed in response to positive actions

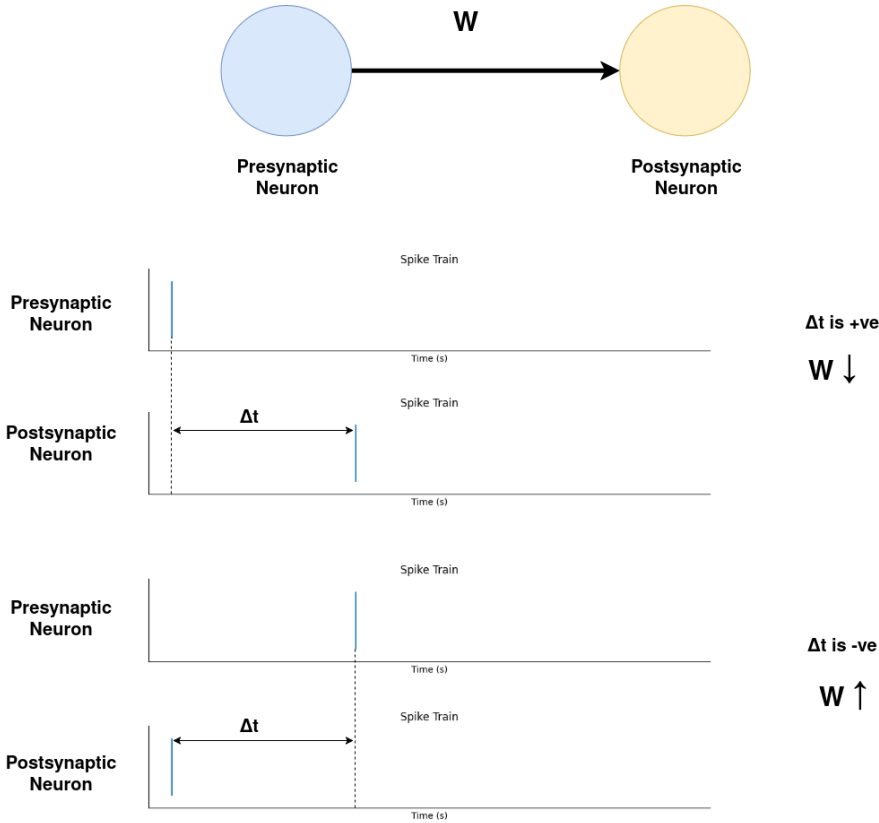


Figure 1.13: Long term potentiation and long term depression caused by relative spike timing when using STDP.

[82]. Reinforcement learning in SNNs is most commonly implemented through reward-modulated extensions of Hebbian plasticity. A widely used approach is reward-modulated acsrshortstdp (R-STDP), where traditional acsrshortstdp weight updates are scaled by a global reward signal, ensuring that only Synapses that contribute to the rewards experience significant potentiation [83], [84].

1.1.5.4 Conversion Methods

Currently, state-of-the-art performance can be achieved by SNNs that are trained using conversion methods [85]. In conversion methods, the SNNs itself do not undergo any direct training. Instead, due to the success of ANNs, an equivalent ANN is trained using one of the methods mentioned previously. The Synapse characteristics resulting from this training are then applied to the equivalent SNN [86]. This process is shown in Figure 1.14. Training SNNs in this manner depends on the success of ANNs and, as such, limits SNNs to that level of capability rather than exploiting behaviours specific to spike-based computation. Therefore, training SNNs via conversion methods is not optimal

and does not provide the behaviours to fully utilise spike-based computation, as seen in biological neural networks. However, this training technique produces state-of-the-art results [87].

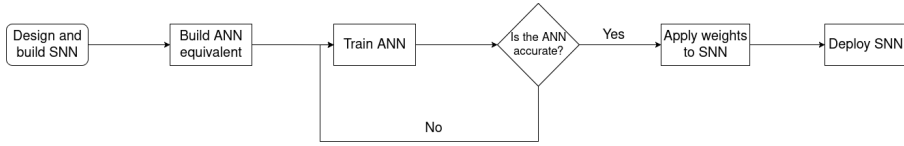


Figure 1.14: The processes involved in training an SNN via conversion methods.

As discussed in Section 1.1.2.1, at the neuronal level, selected behaviours observed in biological neurons can be replicated through spiking neuron models. The Hodgkin-Huxley model provides an example of this, capturing membrane dynamics and spike generation. However, the complexity and performance of biological neural networks are not fully replicated at the network level. Beyond the use of simplified neuron models in most SNNs, biological neural networks also contain diverse cell types and complex synaptic mechanisms, among other processes that are usually simplified or omitted in SNNs. This is reflected in this thesis, where LIF neurons and modulation of synaptic weight are implemented.

1.1.5.5 Delay Learning

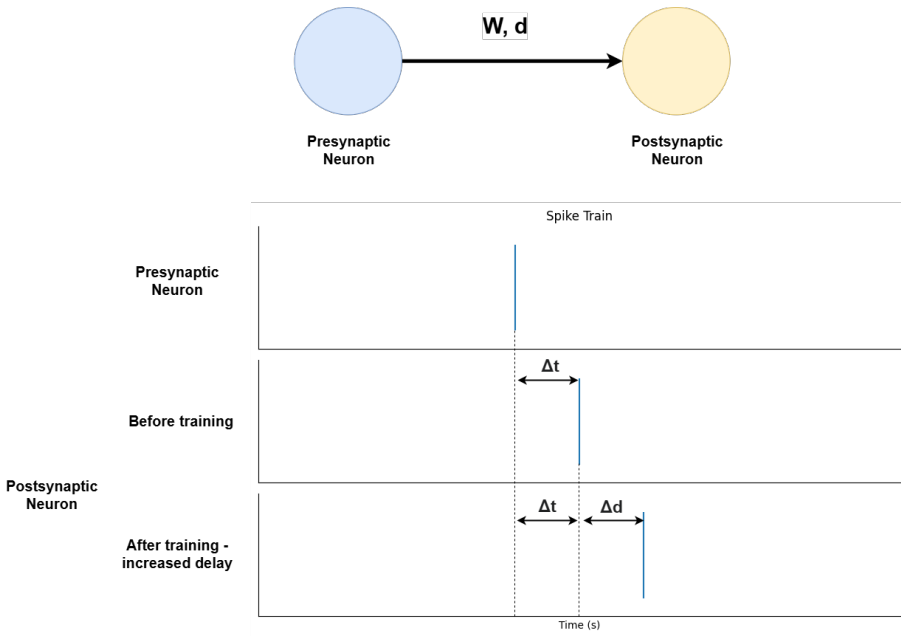


Figure 1.15: The effect of modulating delay on a Postsynaptic neurons response.

Until now, the characteristics of the learning algorithms that have been discussed have focused on modulating synaptic weights. However, biological Synapses and Synapses in SNNs can also apply a delay to the signals they transmit, and this characteristic can be modulated for training purposes [88]; see Figure 1.15. The reason biological neural networks and SNNs can learn by modulating synaptic delay is due to the importance of spike timing in information encoding [89]; see Section 1.1.4.2. As such learning algorithms which modulate delays are becoming more prevalent, but still remain less common, again because of the influence of ANNs for which only synaptic weights can be modulated.

1.2 Neuromorphic Computing and Event-Based Sensing

As computational capacity has grown, the scale of SNN has naturally grown with it. As a result, more efficient purpose built hardware is desirable for effective large-scale implementation; and this is the purpose of Neuromorphic Computing platforms [90]. Neuromorphic platforms aim to avoid bottlenecks, such as those experienced with memory in von Neumann architectures [91], taking inspiration from the brain as a computing paradigm [92]; see Figure 1.16. The concept of Neuromorphic Computing was first developed by Carver Mead [17] with the intention of replicating brain-like computation by adopting the form of a neural system in silicon [93]. Since then, multiple platforms have been developed by large technology companies, such as IBM [20] and Intel [94], as well as through research initiatives such as the Human Brain Project [22]. Beyond developing large-scale hardware platforms for computation, Neuromorphic Computing has grown to include research in materials (not discussed further due to lack of relevance to the project) [95], perception [96], and informatics [97]; bridging the gap between engineering and neuroscience.

1.2.1 Neuromorphic Platforms

Neuromorphic Computing systems have been developed to emulate the biological principles of neural information processing, using spike-based computation. Several neuromorphic platforms have been created, each with unique characteristics and optimisations. These have been developed by both research institutes, in the case of SpiNNaker [22] (see Figure 1.17) and BrainScaleS [98], and private companies, in the case of Loihi [20] and TrueNorth [94].

BrainScaleS is an hybrid, analogue-digital system which is formed by multiple connected wafers, each wafer consisting of several Hi-CANN neurocores, giving analogue operation, but still using discrete time steps. BrainScaleS was created at Universität Heidelberg under the Human Brain Project and is capable of emulating 4 million neurons and 1 billion Synapses. Following this, BrainScaleS-2 uses a scalable architecture, where one core can emulate 512 neurons and can be combined with many other cores to emulate larger networks.

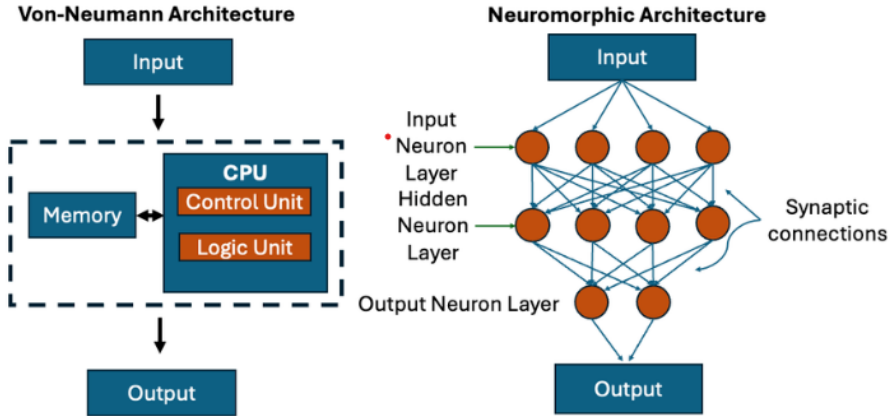


Figure 1.16: A comparison between the Von Neumann architecture and a Neuromorphic Computing architecture.

Loihi is a neuromorphic chip created by Intel in 2018, which is capable of emulating 130,000 neurons and 130,000,000 Synapses [20]. This emulation is achieved through the use of 128 neuromorphic cores which are implemented digitally and operate using discrete time steps. As a successor to Loihi, Intel released Loihi 2 in 2021 significantly increasing the chip’s capacity for emulating neurons [99]. Loihi 2 is capable of emulating approximately 1,000,000 neurons connected via 120,000,000 Synapses, a slight reduction from Loihi.

Similarly to BrainScaleS, SpiNNaker was created as part of the Human Brain Project, but at the University of Manchester in 2014 [22]. SpiNNaker employs a scalable architecture with each core capable of simulating up to 1000 neurons depending upon the complexity of the neuron model, and systems capable of simulating having been made. Following SpiNNaker, SpiNNaker2 has been made under the SpiNNcloud project at the Technische Universität Dresden [100]. SpiNNaker2 integrates ANN capabilities and also aims to significantly increase the core count from the original SpiNNaker project.

IBM’s TrueNorth is one of the most widely recognised neuromorphic chips, created in 2014. It features a highly parallel architecture consisting of 4096 cores, each modelling 256 neurons and 64,000 Synapses, resulting in a chip capable of emulating 1 million neurons and 256 million Synapses [94]. TrueNorth’s operation occurs through discrete, 1 ms time steps, with events being synchronised by a global clock [21].

1.2.2 Event-Based Sensors

Event-based sensors are designed with the purpose of mimicking biological sensory organs and, as such, provide a spiking output that can naturally be interpreted by SNNs. The three most commonly mimicked senses are vision, audition, and olfaction [96], [101]. The concept of event-based vision sensors is introduced here; for wider reading on event-based sensing, see [101]. For an



Figure 1.17: The SpiNNaker Project

example of an event-camera see Figure 1.18



Figure 1.18: A Davis346 Event Camera

Traditional video is captured using frames which consist of an array of pixel values containing RGB data. At regular intervals, the frame is updated with all of the pixel values being updated for every frame. Event-based cameras also use an array of pixels but do not have dedicated pixel values. Instead, each pixel updates independently and asynchronously and only does so when its input exceeds a defined threshold [102], [103]; see Figure 1.19. Rather than storing pixel values, when the positive (negative) threshold is exceeded, the pixel of the event camera outputs a positive (negative) event, which may then be processed. Event cameras can operate both more efficiently and enable faster response

times. Efficiency is improved by minimising redundant updates during still periods, and by removing the delays involved in synchronising updates event cameras enable faster response times [102].



Figure 1.19: The outputs of an RGB camera, and event camera in low light scenarios, with negative events shown in black and positive events shown in white.

1.3 Navigation

Navigation is a fundamental task faced in robotics and engineering. Effective navigation capabilities enable autonomous systems to move and interact with their environment efficiently. Various navigation techniques have been developed depending on the capabilities and sensing modalities of the system; examples being Simultaneous Localisation and Mapping (SLAM), Visual Odometry (VO), and Visual Inertial Odometry (VIO). These methods allow robots, Unmanned Aerial Vehicle (UAV)s, and vehicles to perceive their surroundings and maintain accurate localisation signals accordingly.

1.3.1 Simultaneous Localisation and Mapping

Simultaneous Localisation and Mapping is a technique in robotics and autonomous systems that enables an agent to construct a map of an unknown environment while simultaneously estimating its own position and orientation within that map. Traditional SLAM methodologies relied on geometric feature extraction and probabilistic filtering methods such as Extended Kalman Filters and particle filters. More recently, deep learning has facilitated hybrid approaches that integrate learned representations with classical geometric constraints, enhancing robustness in dynamic environments. Specific variations of SLAM include monocular SLAM, stereo SLAM, and RGB-D SLAM, depending on

sensor configurations, with other variations also existing with the integration of alternative sensors.

1.3.2 Visual Odometry

Visual Odometry (VO) estimates an agent’s motion by analysing consecutive visual data captured by a camera. Changes in position and direction are calculated by tracking movement over time. It can be classified into direct and feature-based methods. Feature-based methods identify key points in the scene, whereas direct methods consider the whole scene. VO can be monocular or stereo, with stereo VO generally providing better depth estimation capabilities. It is widely used in robotics and mobile devices, particularly for navigation in GPS-denied environments; however, it can suffer from drift over time.

1.3.3 Visual Inertial Odometry

Visual Inertial Odometry (VIO) offers an enhancement to visual odometry by integrating IMU data, enabling improved motion estimation, particularly in scenarios where visual features are sparse or non-existent. The fusion of visual and inertial data is typically achieved through filtering frameworks such as the Extended Kalman Filter or factor graph-based optimisation. VIO is commonly used in drones, autonomous robots, and wearable AR/VR devices, where precise localisation is essential for stability and user experience.

1.3.4 Related Tasks

Beyond SLAM and odometry, other navigational tasks play an important role. Optical flow estimation provides an estimate of the movement of features in an image, such as objects, surfaces, and edges. Depth estimation aims to recover the 3D structure of a scene from 2D data, with benefits for mapping purposes. Path planning and control are essential capabilities for an intelligent autonomous system to determine where to navigate through an environment and be able to accurately follow the planned route. All of these tasks are adjacent to those tackled in this work hence warrant mentioning, however, are not fundamental to the project and serve more as areas that may be used for expansion through further work.

1.4 System

In this thesis, we propose a system that performs localisation, exclusively using SNNs for information processing. The networks integrate inertial information through populations of spiking neurons inspired by biological navigation circuits before also combining vision cues for improved accuracy and robustness. Inertial data from an IMU drives two core networks: a Head Direction network, implemented as a ring attractor of spiking neurons, and a Grid Cell network, implemented as a two-dimensional toroidal attractor. The Head Direction network provides a continuous representation of orientation, while the grid cell

network integrates velocity inputs to construct an internal coordinate system that supports path integration through a trained place cell network. Place cells are trained using grid cell activity to provide a single interpretable localisation signal, with each place cell corresponding to a distinct Place Field. Together, these network components enable an agent to maintain an estimate of position and heading in the absence of external cues.

Event camera data, processed by spiking neural networks that are trained for visual place recognition, provide a complementary reset mechanism. This visual input is used to perform loop closure by resetting the Head Direction and Grid Cell Networks when a known scene is recognised, thus correcting accumulated drift and aligning the network’s representation with the known ground truth. In many areas, the system intentionally mirrors the functional organisation of the mammalian navigation networks, where head direction cells, grid cells, and place cells cooperate to support robust spatial navigation. For an overview of the system design, see Figure 1.20.

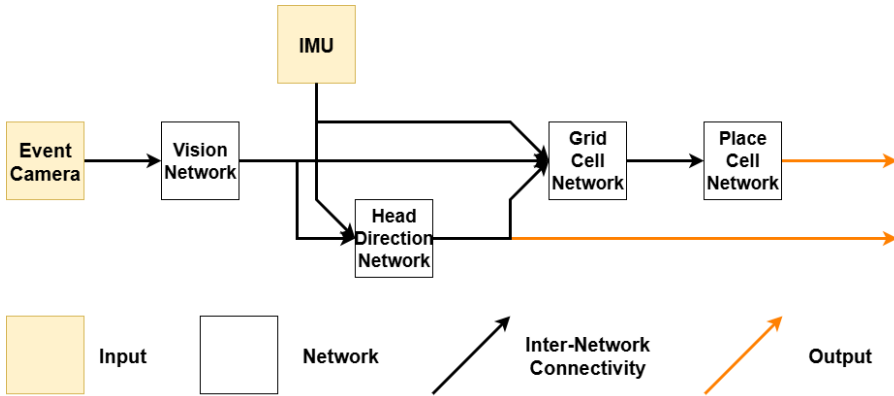


Figure 1.20: The design of a neuromorphic navigation system showing how the different spatial networks interact to create localisation signals.

The system demonstrates how neuromorphic principles can be leveraged to design navigation algorithms that combine the efficiency and temporal precision of event-based vision sensors with SNNs to leverage their inherent sharing of an information encoding paradigm. By basing the design in bio-inspired network structures, the system aims to balance efficiency, with general applicability, creating a system that can easily be adapted to specific environments and offers a novel approach to real-world event-based visual-inertial odometry.

The implementation developed for this thesis was written in Python, with input data loaded and processed using **pandas** [104], [105]. The SNNs were built, trained, and simulated using the NEST spiking neural network simulator [106], which provided the neuron, synapse, and plasticity models used throughout the contribution chapters. These tools were selected because they are widely used, well documented, and provided the functionality and flexibility required to process event-based data, construct the proposed networks, and explore different network configurations. Where applicable, the network parameters,

input encodings, training procedures, and evaluation metrics are described in the relevant chapters to support replication of the results.

1.4.1 System Limitations

Given the nature of the system as a novel implementation it faces limitations in all facets of the network. In the Head Direction network, noticeable drift is observed, but more worrying are the large rapidly occurring errors that were observed in a small number of test sequences. Regarding the Grid Cell network, stable behaviour and the activity shifting mechanism were successfully achieved but the empirical approach to establishing linear shift response didn't produce useable outcomes. The place cell network showed the ability to repeatedly localise to the same position, but this didn't always happen when travelling in opposite directions along the same road, implying that the Place Field size isn't appropriately sized. While the visual place recognition network was able to create a signal to trigger a reset in the spatial networks, the observed behaviour implies that attempting to train multiple resets in a single neuron will lead to saturation with the current combination of network structure and training scheme.

1.4.2 System Contributions

We believe that biological neural networks form highly optimised solutions to fundamental perception and navigation tasks. So, by emulating these networks in artificial systems, it is possible to develop efficient and generalisable solutions to complex real-world problems. This thesis makes advances to this end by extending the application of bio-inspired network architectures to tackle Visual Inertial Odometry for driving scenarios with a neuromorphic approach.

The first key contribution of this work is extending the use of Head Direction ring attractor networks by adopting a spike timing-based approach, unlike former rate-based implementations, to track vehicle orientation. This temporal coding approach enabled the integration of an event-based visual place recognition network that provides loop closure by resetting the head direction at known locations. We also propose the application of toroidal Grid Cell networks for velocity integration that combined with a place cell network provides positional tracking. These systems haven't previously been applied to real-world driving scenarios. In addition, the visual place recognition signals are also shown to reset the positional signal of the Place Cell network to provide consistent loop closure of all localisation signals.

Together, these contributions broaden the application of bio-inspired SNNs beyond the typical neuroscience focused modelling, and show their potential for neuromorphic navigation tasks.

1.5 Thesis Structure

The remainder of this thesis is organised into six chapters that present the background and relevant preliminary information; the research and findings;

and the conclusions, which also includes directions for future work to build upon what has been presented. Chapter 2 provides the necessary background of both concepts and requirements for implementation, including details of neuroscience, datasets, and simulation platforms. Following on from the introductory chapters, Chapter 3 is the first to presents the core work, and contributions of the thesis. Here we present a Head Direction network that operates using spike timing running on real-world inertial data. In Chapter 4, we present a conceptual framework for velocity integration through a Grid Cell network that integrates inertial signals to detect position changes. A method for interpreting the many grid cell signals into a single representation of position, through the use of a trained Place Cell network is implemented in Chapter 5. The practical work contained in this thesis is concluded in Chapter 6 by presenting Monocular and Stereo vision networks, performing visual place recognition to induce a reset signal into the spatial networks. Finally, Chapter 7 summarises the findings and limitations of the contributions presented throughout this thesis and provides ideas for the direction of future work.

Chapter 2

Background

This chapter surveys the foundations underpinning the neuromorphic navigation system developed in this thesis, signposting the origins of different branches of research, and highlighting specific, recent innovations that have served as inspiration to the work that follows in this thesis. We begin with biological representations of space, discussing head direction, grid, and place cells. Here we provide details of each cell's activity; how they interact with the environment and other cells; and theories about network structures that can replicate their observed behaviours. Then, the use of event cameras and Neuromorphic Computing in vision tasks is reviewed. This section primarily focuses on tasks adjacent to navigation, ones that can be incorporated into a larger navigation system to form a complex solution, as well as briefly touching on vision in navigation tasks which is elaborated on in the following section. Next, we survey navigation systems, starting with traditional, non-neuromorphic implementations, followed by reviewing algorithms using Neuromorphic Computing or event cameras, before finally discussing implementations that combine event cameras with neuromorphic computation for an entirely event-based system. The following two sections discuss event cameras, and their datasets, highlighting those that are applicable to driving and navigation tasks. Also in this section, non-event camera datasets are briefly discussed and how they can be used in conjunction with video to event camera conversion. Finally, we survey Spiking Neural Network simulators, discussing their suitability to the different applications of SNN simulation.

2.1 Biological Representations of Space

The representation of space in the brain is fundamental for navigation in animals. This spatial cognition is based on a distributed network that performs path integration and encodes location and direction, allowing effective navigation and memory formation. This is summarised in Figure 2.4, at the end of the Section, once the primary neuron types relating to this work have been introduced. The three main types of spatially related neurons, head direction cells [107], grid cells [108], and place cells [109], work in synergy to provide a detailed

representation of position, based on environmental and egocentric cues [55].

2.1.1 Head Direction Cells

Head direction cells represent an animal's directional heading, acting as an internal compass that signals direction independently of location or movement. Their firing rates are maximised when the animal's head is aligned with a specific preferred direction; see Figure 2.1. These cells are found in several brain regions that form a distributed network that underpins the neural compass.

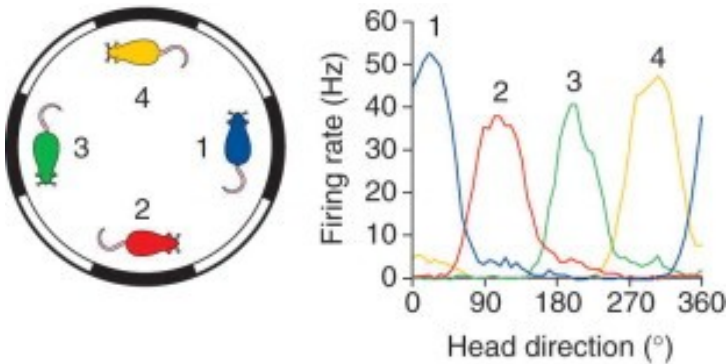


Figure 2.1: The firing rate of head direction cells in response to different head directions [110]

The discovery of head direction cells demonstrated their specific sensitivity to rotational cues [107], suggesting that they contribute to an allocentric reference frame necessary for navigation. The head direction system integrates vestibular and sensory inputs to maintain a stable representation of direction [111], [112]. The neural circuit allows head direction cells to integrate self-motion cues with external sensory inputs to perform one element of path integration.

The vestibular system plays an essential role in head direction signalling. Disruption of vestibular inputs, such as damage to the semicircular canals or otolith organs, results in a severe impairment of the stability of Head Direction Cell activity [113], [114]. This has also been demonstrated across species with semicircular canal occlusion disrupting Head Direction Cell activity in chinchillas [115].

External sensory cues, such as environmental landmarks, also play a critical role in Head Direction Cell activity. Head direction cells maintain stable activity across diverse environments, adjusting their preferred directions when external cues are rotated [116]. Similarly, avian Head Direction Cell activity has been shown to align with key environmental cues [117].

Head Direction Cell tuning is modulated by the brain's architecture. The directional precision represented by head direction cells scales along the dorsoventral axis with the highest precision found dorsally [118].

The effectiveness of head direction cells as a concept is demonstrated by the diversity of species in which they have been observed, including birds [117], rats [107], mice [113], monkeys [116], chinchillas [115], bats [119], and fish [120]. Beyond evidence of their presence in vertebrates, similar activity has also been observed in the brain of *Drosophila* [121].

Head direction cells are essential components of the brain's navigation system, dynamically integrating self motion cues with sensory landmarks. This neural compass provides a stable directional activity which interacts with other spatial cell types such as grid and place cells to support navigation in complex environments.

2.1.2 Grid Cells

Grid cells are neurons that provide a coordinate system for spatial navigation through their neuronal activity, creating a periodic hexagonal firing pattern that tessellates across an environment (see Figure 2.2), allowing a precise representation of travel [108]. Grid cells form a fundamental component of the brain's navigation system, integrating with head direction cells and place cells to create a comprehensive representation of position and direction.

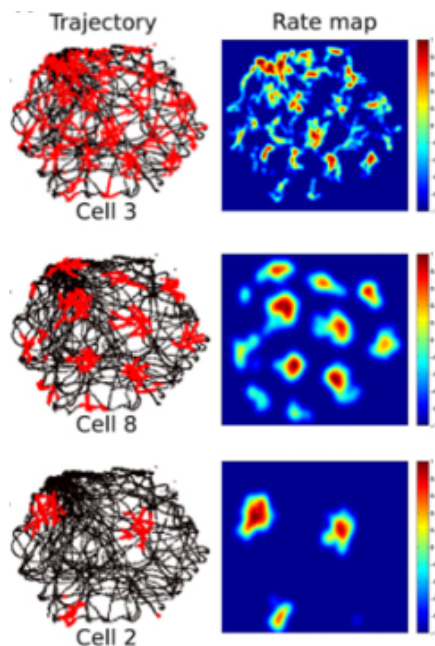


Figure 2.2: The hexagonal firing pattern of grid cells occurring at different scales, for different cells, over the same environment [122].

The hexagonal firing pattern of grid cells is scale invariant in response to the environment, with the spacing of grid cell activity increasing along the dorsoventral axis of the Medial Entorhinal Cortex (MEC) [123], [124].

This hierarchical organisation allows grid cells to represent spaces at different resolutions, with smaller grids supporting highly granular spatial discrimination and larger grids enabling large-scale navigation. In application, this means that grid cell activity is stable across environments and only remaps in response to changes in a known environment.

Grid cell activity has been replicated through the use of attractor dynamics, with it being believed that self motion can be encoded through path integration [125], [126]. In accordance with attractor dynamics, recurrent network connectivity allows grid cells to maintain spatial firing patterns based on internal estimates of movement. Building upon this grid cells may also integrate external sensory cues with self motion cues enabling robust and adaptable spatial representations [127].

In summary, grid cells are central to the brain’s spatial navigation abilities, providing a metric for distance and direction that is both precise and adaptable to the environment. Their interactions with other spatial cell types, combined with their representation across species, highlights their importance in representing space for neuronal computation.

2.1.3 Place Cells

Place cells, first identified in 1971 [109], are essential for encoding specific locations. Place cells fire maximally when an animal occupies a specific location within an environment; a Place Field (see Figure 2.3). By signalling for a unique position within an environment, the place cell provides a localisation signal, which may also be used for mapping. Initially discovered in rats [109], place cells have also been identified in chinchillas [115], bats [128], [129], [130], and various primates [131], [132], [133], [134].

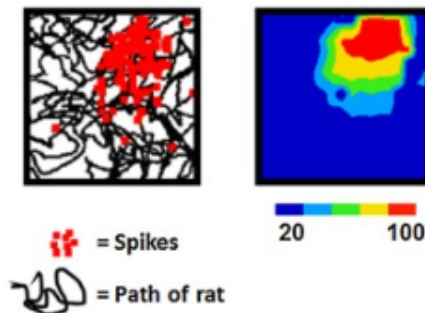


Figure 2.3: A place cell firing in a single location within an environment [55].

O’Keefe and Conway [135] expanded upon this concept by showing that Place Fields could be modified by environmental cues, suggesting that place cell firing adapts to changing surroundings based on the availability of sensory information. This adaptability allows place cells to respond to sensory cues from landmarks, making them integral to the Cognitive Map. A change in

external conditions, such as darkness or rotating landmarks, was also shown to affect spatial learning without altering intrinsic place cell activity [136].

The interaction between place cells and grid cells underscores the distributed system that forms the Cognitive Map. Place cells receive input from grid cells [137], forming a network in which place cells encode location based on self motion and external sensory cues. This interaction highlights the relationship between place cells, which encode specific locations, and grid cells, which provide the path integration required to navigate based on self motion cues.

Moreover, the influence of place cell remapping in response to environmental changes provides insight into spatial memory. Place cells adjust their Place Fields when the spatial context changes [138], allowing multiple environments to be represented within a single neural network.

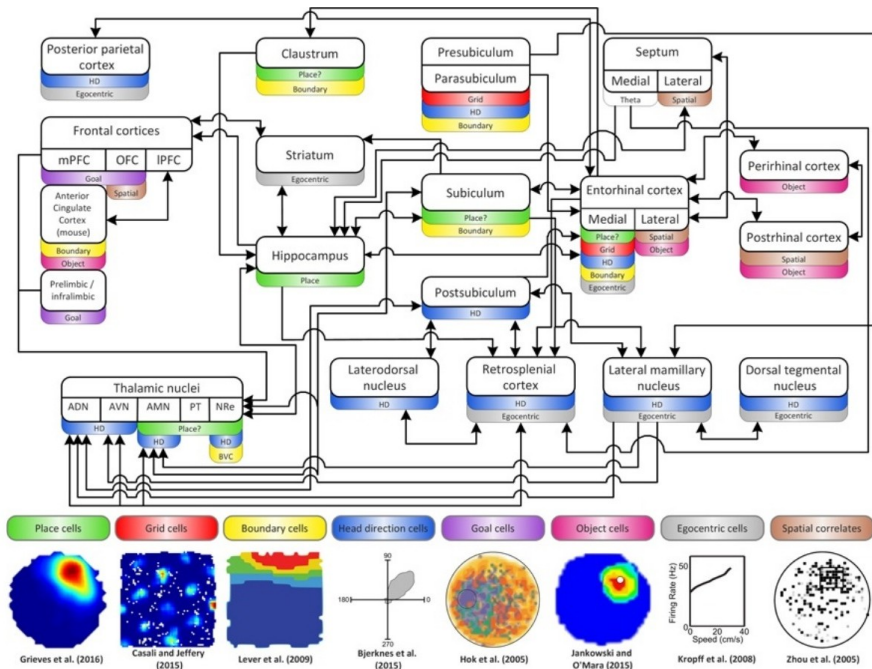


Figure 2.4: An overview of known spatial cells showing the different firing patterns, their interconnectedness, and how they are distributed through different brain regions [55].

2.2 Neuromorphic Vision Systems

The use of ANNs in computer vision is widespread, and they typically provide state of the art solutions. This, and the increase in of availability of event-based data (see Section 1.2.2), has led to the increased use of SNNs for visual processing [139], [140]. But most event-based, or neuromorphic vision systems still combine event cameras with alternate information processing methods

such as ANNs.

Recognition and classification are fundamental tasks in computer vision, with object and digit classification seeing extreme popularity, and visual place recognition being especially applicable here as it can contribute to navigation tasks. Early work integrated event-based sensing and neuromorphic computing to perform object recognition and tracking in real-time [141]; see Figure 2.5. With the introduction of datasets that natively supported neuromorphic processing [140], the use of SNNs in such tasks became more prevalent [139], [142]. With the natural integration of event camera data with SNN processing, relevant datasets shifted to dynamic recognition tasks as they better suited the sensing modality. This saw the development of gesture recognition datasets, and algorithms [143], [144]. Also owing to the dynamic nature of the scenes, there has been a surge of event-based driving datasets (see Section 2.4) providing opportunities for the development of event-based visual place recognition algorithms, however, currently such algorithms have only been applied to RGB data [145].

A common theme in the application of SNNs to vision tasks is that they are typically used in conjunction with event-based sensors due to their inherently similar information encoding. Much of the neuromorphic, and event-based vision literature applies alternative methods, such as ANNs), for information processing to event camera data, leaving many areas of research open for the novel application of SNNs. Particularly the application of SNNs to visual place recognition. However, one area where vision and SNNs are more often combined is in navigation tasks as discussed in the next section. and in contributing to navigation tasks, such as SLAM, VO, and VIO.



Figure 2.5: Object recognition performed on event camera data.

2.3 Navigation Systems

Neuromorphic computing has increasingly been applied to improve efficiency and reduce latency in engineering tasks, and this also applies to navigation algorithms such as visual SLAM, VO, and VIO. Traditionally, these tasks use a feature-based method in which points are identified within an image, and tracks the movement of these features to estimate motion; a method known as optical flow [146], [147]. In the case of visual SLAM, this is extended with the addition of depth estimation or sensing for map formation. Alternatively, the direct method uses pixel intensities, omitting the need for feature identification, and performs optical flow directly on these. This can be done on all of the pixels (dense) or on select pixels (semi-dense or sparse). While it has proven successful, it is a computationally expensive process which is exacerbated by complex, dynamical environments, and increasing camera resolutions. As with many computational tasks the success of ANNs in other areas has led to their application to visual navigation tasks [148], [149] (see Figure 2.6) and naturally SNNs are also applied [150].

The use of events in navigation related tasks is not entirely recent [151], although far more traction has been gained in recent years. Early works typically looked to replicate brain regions with a focus on mimicking networks which had been observed in vivo [151], [152], [153], and while this trend persists [54], [154], [155], this is usually now done in an attempt to optimise for engineering tasks. It can also be observed that event-based is often used to refer to systems using event cameras as an input, with the processing being done with alternative methods [156].

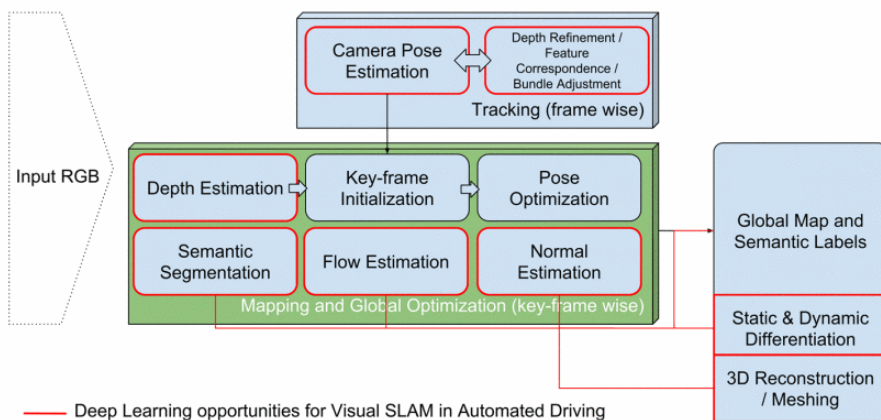


Figure 2.6: The areas of a visual SLAM pipeline specifically identified for improvement through deep learning by [157].

2.3.1 Event Camera based Navigation

Event cameras provide data that is asynchronous and low latency with a high dynamic range with transformative potential as a sensing modality. Given event cameras capture pixel-level brightness changes they are particularly suited for environments with fast motion and varied lighting conditions. These properties can be leveraged for navigation tasks in challenging environments by improving the robustness of data at the point of acquisition [158].

There are a range of implementations using event cameras for navigation related tasks ranging from optical flow estimation [159] to control [160], [161] but the most relevant are the odometry and SLAM implementations. The most prominent work for purely visual SLAM using event cameras is that of Kim [162] who achieved 6 degrees of freedom (DOF) tracking and 3D reconstruction. With regards to SLAM using an event camera as part of a wider sensor fusion system Vidal et al. [163] presented work in which combined frames, events and Inertial Measurement Unit (IMU) data to provide a robust implementation applied to drone flight over a complex environment. In comparison to frame-based odometry systems it was shown that event-based stereo visual odometry can show better performance in dynamic scenarios [164]. An alternate approach is the combination of event data and frames as used in Event-Aided Direct Sparse Odometry [165]. Here event data was combined with sparse frame-based features to enhance tracking accuracy and computational efficiency. By employing a hybrid approach drift is minimised and the system is shown to be more robust in environments where visual features are sparse. Similarly, an event camera approach to simultaneous localisation and mapping has been shown to offer improvement by accumulating events to create event frames which can then be processed using traditional methods [166].

2.3.2 Neuromorphic Computation in Navigation

Initial applications of SNNs to navigation related tasks focused on simple networks where a small number of neurons can achieve an attraction or avoidance signal [168]. This resulted in many systems focused on tasks such as lane keeping, obstacle avoidance, and target acquisition [169]. Beyond this there is a range of similar, but distinct implementations that can be drawn from brain models, found in computational neuroscience, and neuromorphic computing implementations. The majority of these works use RGB or RGB-D cameras as the primary sensor and a bio-inspired network architecture, resulting in a well defined understanding of the different cell types that are used in navigation [55], [170], [171]; see Section 2.1.

One of the earliest bio-inspired navigation systems was RatSLAM [151], which mimicked rodent hippocampal mapping using attractor dynamics. Although RatSLAM didn't use event-based processing, these concepts have since been refined using neuromorphic computing. Systems following this show clear themes using bio-inspired networks and performing 2 dimensional mapping [154], [167], [172], [173] (see Figure 2.7). While there are obvious parallels between the aforementioned work and the proposed framework, undoubtedly the work of greatest relevance are the existing end-to-end event-based systems.

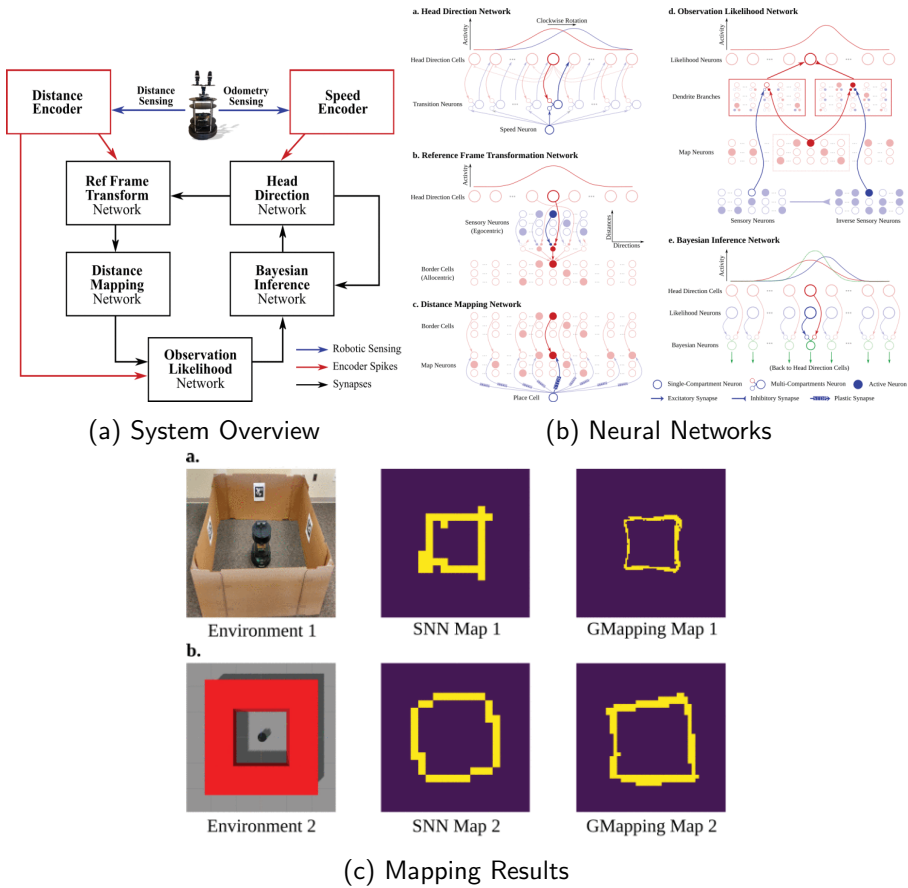


Figure 2.7: An example of Neuromorphic Computing being applied to uni-dimensional SLAM, showing (a) a breakdown of the system; (b) the connectivity within networks; and (c) the resulting outputs [167].

2.3.3 Combining Event Cameras and Neuromorphic Computing for Navigation

Combining event cameras and neuromorphic computation can be used to create an end-to-end event-based navigation system, creating a cohesive framework that utilises both the sensing and processing advantages. The combination of event-based sensing and neuromorphic computation, as seen in these systems, represents a potential advancement in autonomous, high speed, navigation. By leveraging bio-inspired principles, end-to-end systems can provide highly efficient, low latency solutions that are suitable for complex real world environments where conventional frame-based approaches may fall short.

There exists a range of work on control [174] and perception [54], [175] tasks but until recently the most advanced navigational tasks were obstacle avoidance and target acquisition. These consisted of lane keeping robots [168],

[176] (see Figure 2.8) and target acquisition systems [177], [178]. With the view of developing a fully event-base SLAM system Kreiser et al. [155], [172], [179] made strides in this direction. Initially a vision based head direction network was developed allowing the robot to orient itself and correct accumulated errors [155]. Following this 1 dimensional SLAM was achieved [179] including error correction and mapping capabilities. The latest development in event-based SLAM presents a sensor fusion solution, combining an event camera and a radar to achieve a SLAM system capable of learning on the fly while moving within a complex real world environment [180]. While travelling along aisles, a 2D map is created to show obstacles.

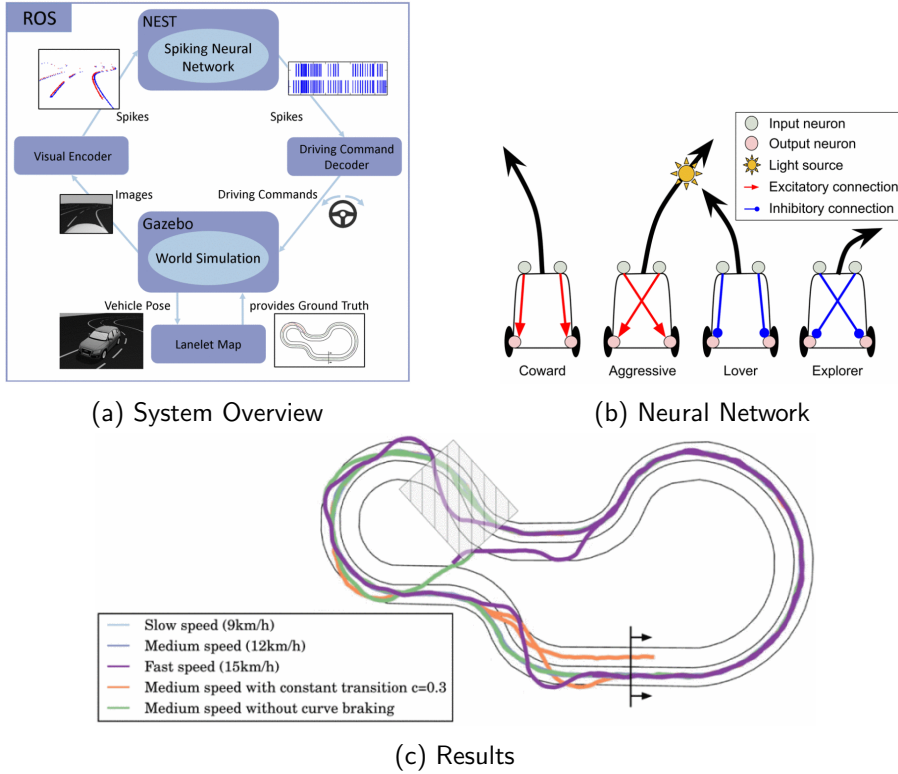


Figure 2.8: Combining an event camera with Neuromorphic Computing to achieve control of a vehicle. (a) a breakdown of the system used for simulation; (b) the connectivity within network; and (c) the results of the lane keeping task [176].

2.4 Event Camera Data

The development and benchmarking of neuromorphic visual-inertial odometry systems depend critically on the availability of datasets that accurately capture the unique properties of event-based vision. Unlike traditional cameras, event

cameras output asynchronous events in response to brightness changes at each pixel, resulting in sparse, temporally precise streams of visual data. To enable algorithmic development and fair comparison of methods, a wide range of datasets have been curated to reflect varying environments, sensor configurations, and application domains. These datasets are broadly categorized into stereo and monocular configurations. This section provides a detailed and critical review of leading datasets in both categories.

Stereo event camera datasets are particularly important for tasks requiring depth perception and 3D spatial reasoning, including stereo VO, depth estimation, and stereo visual SLAM. The availability of synchronized IMU data and precise ground truth enhances their utility for evaluating across navigation methodologies e.g., comparing VO and VIO systems.

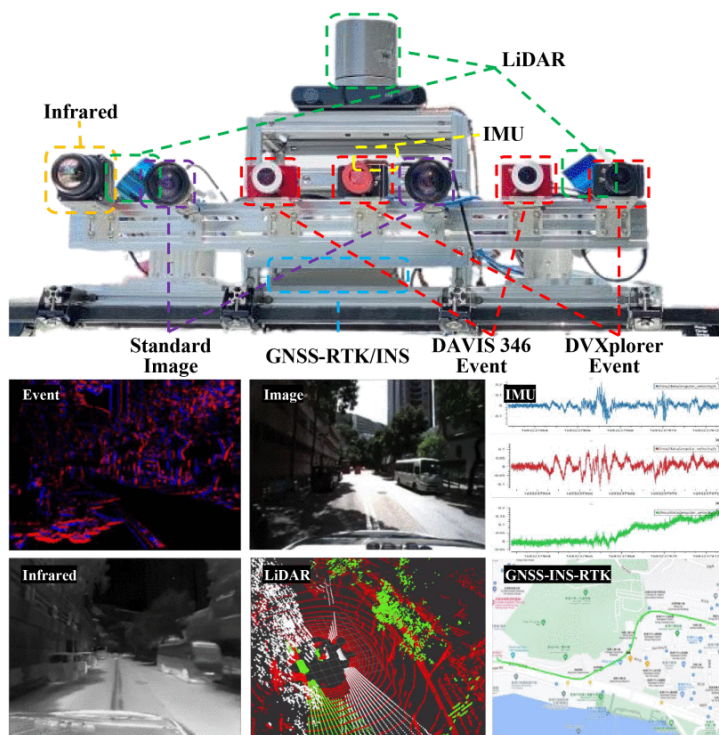


Figure 2.9: The sensor configuration used in the ECMD dataset, and a sample of the resulting recordings [181].

The Event-Centric Multisensory Driving (ECMD) dataset provides recordings from event and RGB cameras, as well as LIDAR and IMU data [181]; see Figure 2.9. These recordings are all taken using a car driving along roads in a range of scenarios. DSEC also consists of similar recordings, featuring stereo event and RGB cameras as well as LIDAR and IMU recordings [182]. With a similar sensor set up FusionPortable has stereo event and RGB cameras, recording alongside LIDAR and an IMU but in a different environment [183].

Here the sensors are mounted to a dolly or quadruped, or are handheld, and feature recordings from pedestrian areas. Also using stereo event and RGB cameras the TUM-VIE dataset features indoor and outdoor recordings, however this is the first dataset to not feature LIDAR recordings [184]. The datasets encompassing the largest range of environments are M3ED and MVSEC. Both include recordings from stereo event and RGB cameras, alongside LIDAR and IMU data [185], [186]. They also both include car and UAV mounted recordings, with M3ED also having quadruped recordings, while MVSEC has motorbike and handheld recordings; see Figure 2.10. In addition to the scenes covered by the other datasets M3ED includes offroad driving in a forest.

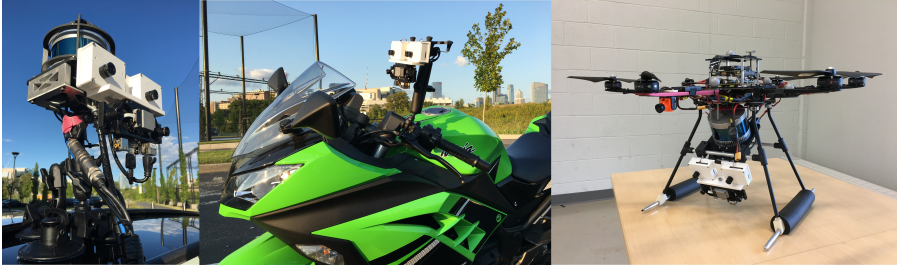


Figure 2.10: The sensor mount used to record the MVSEC dataset, mounted on a car, motorbike, and UAV [186].

Monocular event camera datasets are generally used in scenarios where weight, size, or power constraints prevent stereo configurations. There are a multitude of monocular, car-based datasets featuring event camera data with the most widely used examples being DDD17 [187], DDD20 [188], N-Cars [189], and Brisbane Event VPR [190]. These all feature more lightweight sensor configurations than seen amongst the stereo datasets, omitting LIDAR and IMU data. For a more comprehensive list of monocular datasets as well as details about their sensor configurations, and recording environments, see Table 2.1.

It is also possible to use existing datasets that don't feature event cameras through the conversion of video to event streams, enabling the use of more popular datasets such as KITTI and KITTI-360. This allows comparison with a wider range of algorithms but does so at the cost of introducing a reality gap through the conversion, given that the converted event stream won't perfectly replicate event camera behaviour. Emulating event camera activity, either through conversion or simulation is the subject of large amounts of research, most notably [191], [192], [193]

Table 2.1: The primary navigation related event camera datasets alongside some non-event camera equivalents that can be used in conjunction with a video to event camera converter.

Dataset	Vehicle	Scenario	Event Cameras	RGB Cameras	LIDAR	IMU
ECMD	Car	Road	x4	x2	✓	✓
MVSEC	Car, Motorbike, UAV, Handheld	Road, Indoor, Outdoor	x2	x2	✓	✓
DSEC	Car	Road	x2	x2	✓	✓
M3ED	Car, UAV, Quadruped	Road, Indoor, Outdoor, Forest	x2	x3	✓	✓
FusionPortable	Handheld, Quadruped, Dolly	Indoor, Outdoor	x2	x2	✓	✓
TUM-VIE	Handheld, Helmet	Indoor, Outdoor	x2	x2	✗	✓
DDD17	Car	Road	x1	x1	✗	✗
DDD20	Car	Road	x1	x1	✗	✗
MA-VIED	Car	Road	x1	x1	✗	✓
CED	Car, Handheld, Static	Road, Indoor, Gestures	x1	x1	✗	✗
N-Cars	Car	Road	x1	x1	✗	✗
Brisbane Event VPR	Car	Road	x1	x1	✗	✗
Vivid++	Car, Handheld	Road, Indoor, Outdoor	x1	x1	✓	✓
M2DGR	Dolly	Road, Outdoor, Indoor	x1	x1	✓	✓
DET	Car	Road	x1	✗	✗	✗
TUM-RGBD	Handheld, Dolly	Indoor	✗	x1	✗	
KITTI	Car	Road	✗	x4	✓	✓
KITTI-360	Car	Road	✗	x5	✓	✓
DrivingStereo	Car	Road	✗	x2	✓	✓

2.5 Challenges

The literature reviewed in this chapter shows that event cameras, SNNs, and bio-inspired spatial representations each provide promising routes towards navigation systems, with a wide range of approaches taken. However, it also highlights several challenges that must be addressed before these approaches can be combined into a complete neuromorphic localisation system. These challenges relate to maintaining event-based processing at all stages in the navigation system, adapting biological representations of space to real-world data, integrating inertial and visual information, providing loop closure, and the availability of datasets to evaluate these systems.

The first challenge is utilising event-based representations throughout the system. Event cameras produce sparse, asynchronous outputs that are inherently aligned with the event-based representation used by SNNs. However, many event-based navigation systems use intermediary representations, such as frames, event images, or extracted features, before processing using conventional methods. Although these approaches can be effective, they reduce the extent to which the system benefits from the temporal sparsity of an event-based sensor. This creates an opportunity for systems that retain an event-based representation from the sensor input through to the localisation output.

The second challenge is translating isolated bio-inspired spatial representations into a complete system that operates on real-world data. Head direction cells, grid cells, and place cells provide useful inspiration for tracking orientation and position, but they are typically considered as the sole focus rather than as a component of a larger artificial network. Combining them within a larger SNN requires each representation to produce an output that can be interpreted by the next network, while preserving stable activity within each subnetwork. This is challenging because errors in one representation can propagate into the others; for example, drift in the Head Direction network affects the direction of velocity integration in the Grid network, which then affects the Place Cell representation. Therefore, the system must consider biological inspiration, network size, stability of spiking behaviour, and interpretability across multiple connected subnetworks rather than within a single isolated, homogeneous network.

The third challenge is developing visual place recognition methods that are appropriate for event camera data. Existing event-based vision work shows the usefulness of event cameras for dynamic scenes, but visual place recognition remains difficult because the appearance of a location can change drastically based on conditions such as lighting, weather, traffic, viewing angle, and time. This occurs in driving scenarios, where the same physical location may be encountered from different lanes or directions. A robust neuromorphic navigation system therefore requires a reset mechanism that can recognise stable landmarks while avoiding unreliable or ambiguous visual cues. To address this, an appropriate dataset that can allow the network to account for all of these variations is required.

The final challenge is the suitability of the datasets. Existing event-camera datasets provide valuable recordings across a range of sensor platforms and

environments [181], [182], [185], [186]. However, they typically offer a similar structure, featuring recordings of many distinct routes. Neuromorphic localisation and visual reset mechanisms would benefit from datasets that contain repeated traversal of the same routes under varied conditions. Such datasets would allow systems to be tested not only on whether they can track motion through a single sequence but also on whether they can learn and reuse landmarks across different times of day, traffic densities, weather conditions, and directions of travel.

This thesis aims to address some of these challenges by developing an event-based localisation network that combines bio-inspired subnetworks with event-camera inputs. The Head Direction network investigates spike-based orientation tracking from angular velocity signals, the Grid and Place Cell networks explore spike-based representations of position, and the visual place recognition network provides a reset mechanism for correcting accumulated error. In doing so, the work positions itself between event-based sensing, neuromorphic computation, and bio-inspired navigation, with the aim of demonstrating how these areas can be combined towards an end-to-end event-based VIO system.

Chapter 3

Direction Tracking

3.1 Introduction

A crucial component of any system aiming to tackle the task of odometry is the ability to accurately and reliably track the direction of the agent at any given time. Although there are many approaches for this, the emerging field of Neuromorphic Computing is one that is largely unexplored for engineering applications despite the fact that biological models are available [194]. By taking inspiration from biological systems for tracking direction, it is possible to utilise the intrinsic event-based operation of spiking neural networks to create a biologically plausible network capable of performing well in real world engineering scenarios.

Initially, it was attempted to implement this network by converting a rate-based model to use LIF neurons and then derive network parameters. This was unsuccessful, so then using an empirical approach the Head Direction network was implemented and using this approach was found to be capable of accurately tracking direction across the range of angular velocities found in driving scenarios. These have been collected from the IMU data that is integrated into the left event camera in the ECMD dataset [181].

3.2 Network Connectivity

The Head Direction network is formed using a ring network structure consisting of three layers, with each layer being made up of 360 neurons resulting in 1° increments. These layers are the head direction layer and two shift layers; see Figure 3.1. The head direction layer uses a self-sustaining Hill of Activity (HOA) [153] to represent the current direction of the vehicle. The HOA refers to a bell-shaped firing pattern within a network layer, where the centre, or peak, of activity corresponds to the direction or position represented by that layer. When the network integrates motion, the HOA shifts through the layer, with movement of the peak representing a change in the encoded direction or position. This motion integration occurs using two shift layers that act to shift

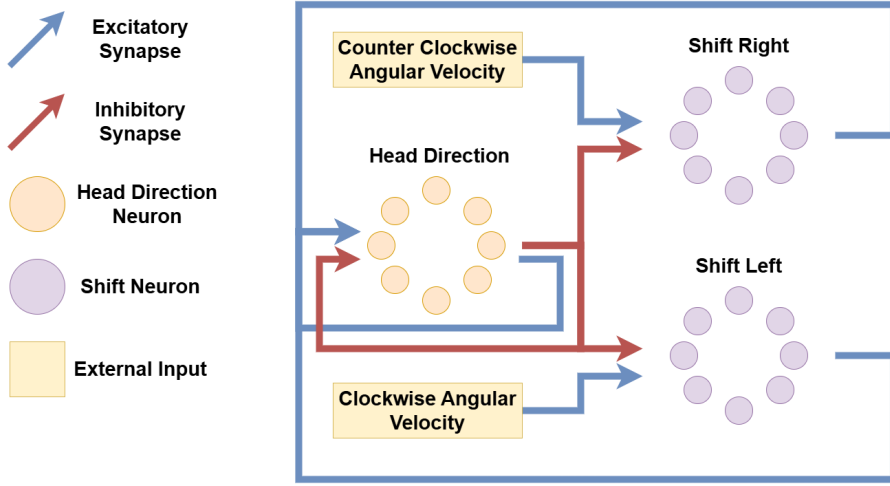


Figure 3.1: An overview of the layers in the Head Direction network, and how they are connected. For a detailed, neuron level connectivity mapping see Figure 3.5

the HOA clockwise and counterclockwise, respectively.

The angular velocity signal was passed directly to the shift neurons as an input current that was linearly proportional to the angular velocity. This allowed the network to be assessed for angular velocity integration while avoiding the need for an additional preprocessing or feature extraction stage. As a result, the effect of the network connectivity and shift-layer dynamics could be assessed directly. The Head Direction layer uses rate coding, where firing activity denotes the proximity of each neuron to the centre of the HOA, and therefore to the represented head direction. The shift layers combine input from the Head Direction layer with the angular velocity input using a time-to-first-spike encoding, where shorter spike times correspond to faster shifts of the HOA.

Initially, it was attempted to calculate the weights using the methods described in [195]. Given that this work features rate-based models the firing rate of a LIF neuron is calculated from the inter-spike interval and used as a surrogate. This is described by the following equations. For details of each parameter, and the final selected values, see Table 3.2.

Membrane time constant:

$$\tau_m = C_m R_m \quad (3.1)$$

Equilibrium potential:

$$V_\infty(t) = E_L + R_m(I_{syn}(t) + I_e(t)) \quad (3.2)$$

Input currents:

$$I = I_{syn} + I_e \quad (3.3)$$

Membrane Potential:

$$\frac{dV_m}{dt} = -\frac{V_m - E_L}{\tau_m} + \frac{I}{C_m} \quad (3.4)$$

For a constant input current,

$$V_m(t) = V_\infty + (V_0 - V_\infty) \exp\left(\frac{-t}{\tau_m}\right) \quad (3.5)$$

where $V_0 = V_m(t = 0)$

To find the inter-spike interval without the refractory period ($t_{isi} - t_{ref}$) we assert that $V_m = V_{th}$ and $V_0 = V_{reset}$

$$V_{th} = V_\infty + (V_{reset} - V_\infty) \exp\left(\frac{-t_{isi} - t_{ref}}{\tau_m}\right) \quad (3.6)$$

From this we find the inter-spike interval

$$t_{isi} - t_{ref} = \tau_m \ln\left(\frac{V_\infty - V_{reset}}{V_\infty - V_{th}}\right) \quad (3.7)$$

This can also be expressed in the form $\ln(1 + z)$ as

$$t_{isi} = \tau_m \ln\left(1 + \frac{V_{th} - V_{reset}}{V_\infty - V_{th}}\right) + t_{ref} \quad (3.8)$$

and can therefore be approximated using $\ln(1 + z) \approx z$, giving

$$t_{isi} \approx \tau_m \left(\frac{V_{th} - V_{reset}}{V_\infty - V_{th}}\right) + t_{ref} \quad (3.9)$$

given $f = \frac{1}{t_{isi}}$

$$f = \frac{1}{\tau_m \ln\left(1 + \frac{V_{th} - V_{reset}}{V_\infty - V_{th}}\right) + t_{ref}} \approx \frac{1}{\tau_m \left(\frac{V_{th} - V_{reset}}{V_\infty - V_{th}}\right) + t_{ref}} \quad (3.10)$$

$$f = \frac{1}{\tau_m \ln\left(1 + \frac{V_{th} - V_{reset}}{E_L + R_m(I) - V_{th}}\right) + t_{ref}} \approx \frac{1}{\tau_m \left(\frac{V_{th} - V_{reset}}{E_L + R_m(I) - V_{th}}\right) + t_{ref}} \quad (3.11)$$

$$\lim_{I \rightarrow \infty} f(I) = \frac{1}{t_{ref}} \quad (3.12)$$

Now that we have the f-I equation of the neuron, we create a tuning curve for the firing rate $f(\theta)$ when offset from the true direction (θ_0) by θ according to

$$f(\theta) = A + Be^{M \cos \theta - \theta_0} \quad (3.13)$$

where A , B , and M are constant parameters assigned to the values $A = 8.95$, $B = 0.344$, and $M = 5.29$, taken from [195].

Having established the rate equation and the tuning curve, f , w , and u are used to represent the firing rate from 3.13, synaptic weight, and synaptic input, respectively. The profile for the synaptic weights is found using

$$\hat{w}_n = \frac{\hat{u}_n \hat{f}_n}{\lambda + \hat{f}_n^2} \quad (3.14)$$

where $\hat{f}_n$, $\hat{w}_n$, and $\hat{u}_n$ correspond to the n th Fourier coefficient of f , w , and u respectively, and the value of $\lambda = 25824$ is taken from the source material [195].

The synaptic input is found using the inverse of the transfer function described in 3.11. The synaptic weight $w'(\Delta\theta)$ can be defined as $w_{ij} =: w'(\Delta\theta)$ for discrete head directions, where θ_{ij} is the degrees of separation between the preferred directions of the cells at index i and j . Therefore the discrete equivalent to 3.14 can be written as

$$\hat{w}'_n = \frac{\hat{u}_n \hat{f}_n}{\lambda + |\hat{f}_n|^2} \quad (3.15)$$

Then the Fast Fourier Transform was applied to the sequences of the tuning curve and the synaptic input, giving $\hat{F}$, and $\hat{U}$. $\hat{W}$ could then be found according to

$$\hat{W}[i] = \frac{\hat{U}[i] \hat{F}[i]}{\lambda + |\hat{F}[i]|^2} \quad (3.16)$$

and the weights W can be found by taking the inverse Fast Fourier Transform $W := \mathcal{F}^{-1}(\hat{W})$

However, when this was applied to the network, the spiking behaviour was volatile and it was not possible to find a compatible shifting mechanism. Shifting of the represented head direction would cause activity throughout the whole network, so it is thought that due to the nature of the LIF neuron operating on individual spikes rather than rate equations this approach wasn't compatible with the network. It also wasn't possible to adopt a rate-based model for the whole network due to the requirement for specific time-based encodings elsewhere in the system. Therefore, the Synapse weightings described below were found experimentally as discussed in Section 3.3.

In the absence of any inputs, the head direction layer must sustain its activity for the representation of direction to persist. This is achieved by connecting neurons that represent directions within 3° of one another via strong excitatory connections with a weight of 50. To maintain the HOA and prevent activity from spreading throughout the head direction layer, neurons representing directions separated by more than 5° are connected via weak inhibitory connections with a weight of -5. These connections can be seen in Figure 3.2.

Excitation in the shift layers is driven entirely by the angular velocity inputs from the IMU. The head direction layer doesn't make excitatory connections to the shift layers but instead makes global inhibitory connections with a weight

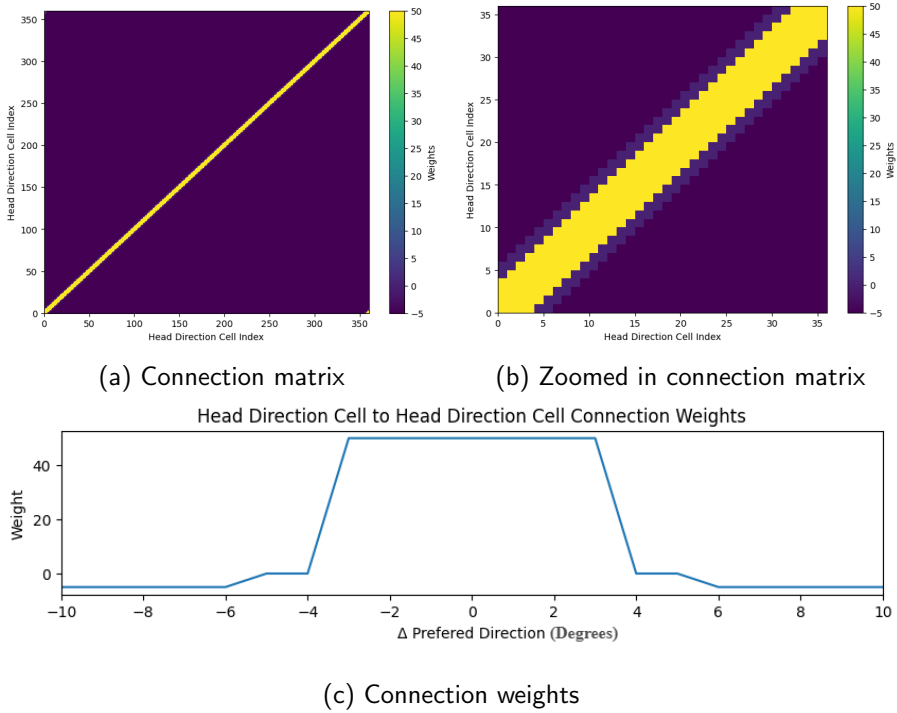


Figure 3.2: The synaptic weights between neurons in the head direction layer.

of -10 to shift neurons representing directions separated by greater than 10° . This approach was taken because combining inhibition with excitation is more resistant to erroneous shifting than using summed excitatory connections to trigger a shift [196]. These connections are shown in Figure 3.3.

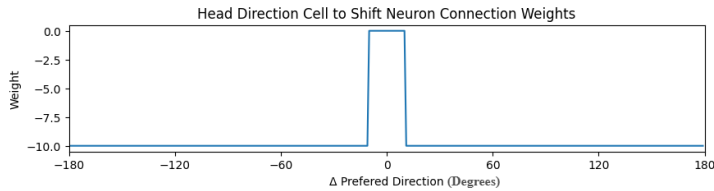


Figure 3.3: The synaptic weights between the head direction and shift layers.

These weights are considered reasonable for this implementation because they produce the intended balance of local excitation, global inhibition, and directional shifting. Specifically for this implementation, this means that the HOA is self-sustaining and stable without angular velocity input, but can be shifted when the corresponding shift layer is activated. These connections were selected empirically as simulation parameters rather than having a biological derivation.

Each neuron in the shift clockwise (counter clockwise) layer is connected to a cluster of neurons in the head direction layer that represent directions within 5° of their own but offset 1 degree to the clockwise (counter clockwise) to shift the HOA to the clockwise (counter clockwise). This uses a strong excitatory connection with a weight of 72 to overcome existing activity that may be occurring in the head direction layer; see Figure 3.4.

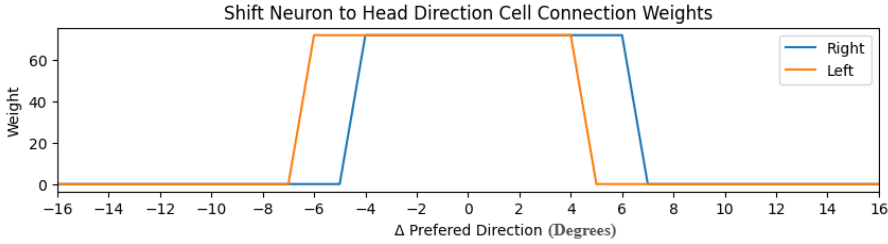


Figure 3.4: The synaptic weights between each of the shift layers and the head direction layer.

For a summary of all the network parameters described previously, see Tables 3.1 and 3.2, where all the neuron parameter values, and synaptic connection ranges and weights are detailed. For a visualisation of the network connectivity with details reduced to the connection polarity, see Figure 3.5

Table 3.1: The connectivity and synaptic weights found within the Head Direction network.

Presynaptic Neuron	Postsynaptic Neuron	Weight	Range (Degrees)
Head direction	Head direction	-5	$abs(i) > 5$
Head direction	Head direction	50	$abs(i) < 5$
Head direction	Shift	-10	$abs(i) \geq 10$
Shift right	Head direction	72	$abs(i - 1) \leq 5$
Shift left	Head direction	72	$abs(i + 1) \leq 5$

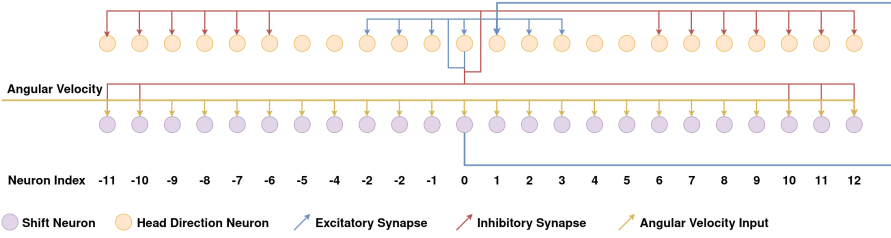


Figure 3.5: The connectivity within the Head Direction network at the neuron level with connections focused around the neuron at index 0. Neuron 0 of the Head Direction layer is connected via excitatory connections to its nearest neighbours and inhibits distant head direction neurons. It also inhibits distant shift neurons. The angular velocity input is fed as an excitatory signal directly into the shift layer, but is counteracted in distant neurons by the inhibition. Activity stimulated in Neuron 0 of the Shift layer connects via excitatory connections to Neuron 1 of the Head Direction layer, meaning activity shifts in the Head Direction layer.

Table 3.2: Neuron model parameters used in the Head Direction network.

Parameter	Head Direction Layer	Shift Layers
Neuron model	<code>iaf_psc_alpha</code>	<code>iaf_psc_alpha</code>
Resting potential, E_L	-54 mV	-54.89 mV
Membrane capacitance, C_m	125 pF	66.67 pF
Membrane time constant, τ_m	25 ms	20 ms
Membrane resistance, R_m	0.2 G Ω	0.3 G Ω
Refractory period, t_{ref}	1.67 ms	1 ms
Threshold potential, V_{th}	-55 mV	-55 mV
Reset potential, V_{reset}	-65 mV	-70 mV
Excitatory synaptic time constant, $\tau_{\text{syn,ex}}$	2 ms	2 ms
Inhibitory synaptic time constant, $\tau_{\text{syn,in}}$	2 ms	2 ms
Constant input current, I_e	0 pA	0 pA

The parameters listed in Tables 3.1 and 3.2 were selected through empirical refinement, as described in Section 3.3.2. Initial values were informed by NEST [106] defaults to produce stable LIF neuron activity and to maintain a self-sustaining HOA in the head direction layer. The connectivity weights were then adjusted to balance maintaining the HOA without external input, preventing saturation of the network, and allowing the shift layers to move the HOA in response to angular velocity input. Parameter values were adjusted iteratively until the network produced the desired response and were then tested on the ECMD dataset.

3.3 Testing

To validate the operation of the network, the following testing procedure was designed and implemented. The process begins by confirming stable behaviour, ensuring that the network can consistently preserve its internal representation of direction. Next, the network is subjected to constant angular velocity signals to ensure that the shifting mechanism is functional. The evaluation then progresses to measuring the rate of shifting and examining how the rate responds to different magnitudes of input value; here parameters are adjusted to achieve an accurate mapping between angular velocity and shift rate. The parameter sets which most accurately mapped angular velocity to shift rate were then tested on the ECMD dataset to assess real world performance.

3.3.1 Angular Velocity Analysis

Throughout this chapter, angular velocity inputs are reported in rad/s, as these are the units used in the dataset recordings. The output of the Head Direction network is reported in degrees because the head direction layer contains 360 neurons, with each neuron corresponding to a 1° increment. Therefore, angular velocity inputs and network direction outputs use different units; (rad/s) is used for the input signal, while degrees are used for the network.

An analysis of the angular velocity values found in the dataset was performed to identify the range of input values that required a linear f-I response. This is done using the absolute values of the angular velocity data for the dataset. By separating the right and left shift mechanisms the network only needs to respond to positive shifting stimuli. The angular velocities were first plotted using a histogram to show the distribution of values (see Figure 3.6) which revealed that the recordings were focused around 0 rad/s. To get an understanding of the specific values to focus accuracy of the network on the range, mean, and standard deviation were then calculated; see Table 3.3. This revealed that the network needed to be most accurate from 0 rad/s to 0.1 rad/s but that the network could be exposed to angular velocities approaching 1 rad/s. Having further established the range of values to prioritise in testing, a boxplot of the angular velocities in each scenario of the ECMD dataset was created; see Figure 3.7. Given the differences in road layouts and conditions it was anticipated that the different scenarios would present different distributions of angular velocities and therefore may result in the need to adjust the range of controlled angular velocities that were tested. However, upon reviewing the distribution of each scenario they were determined to be fairly consistent and the initial range of values was deemed sufficient for assessing the appropriateness of the network.

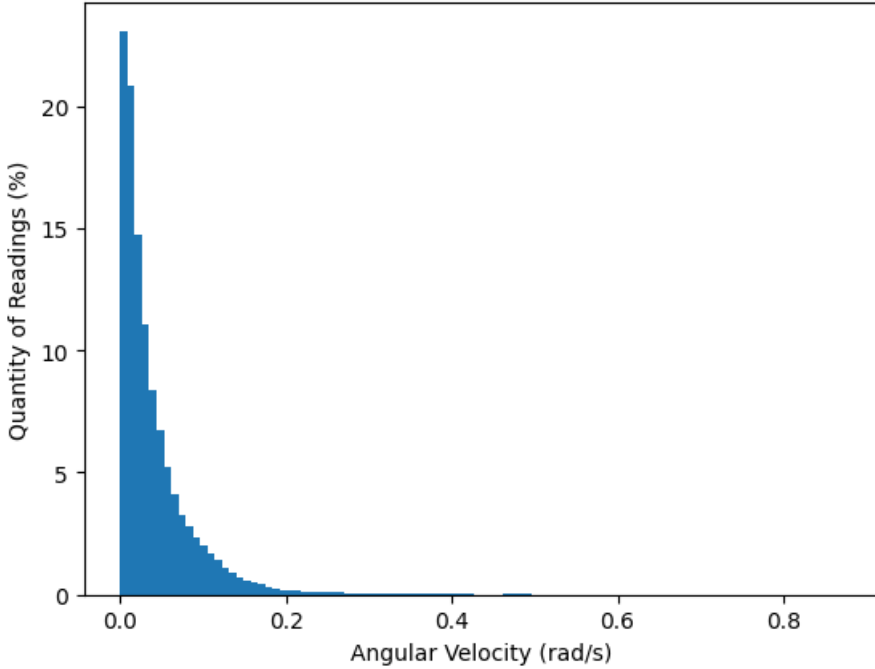


Figure 3.6: The distribution of angular velocity recordings from the ECMD dataset.

Table 3.3: A summary of descriptive statistics for the angular velocities recorded in the ECMD dataset.

Measurement	Angular Velocity (rad/s)
Minimum	0.000
Maximum	0.872
Mean	0.042
Standard Deviation	0.052

3.3.2 Network Tuning

Having analysed the distribution of the angular velocities recorded in the ECMD dataset, the shift rate of the network was measured against angular velocities ranging from 0 rad/s to 1 rad/s, and smaller intervals from 0 rad/s to 0.1 rad/s. Given the focus of angular velocity recordings about 0 rad/s the initial neuron parameters featured in the network were adjusted to make the network more prone to shifting by removing the initial region where a small input current doesn't cause an activation within the neuron. This was done by increasing the resting potential to be approximately equal to the threshold potential, effectively shifting the f-I curve upwards to start at the origin to

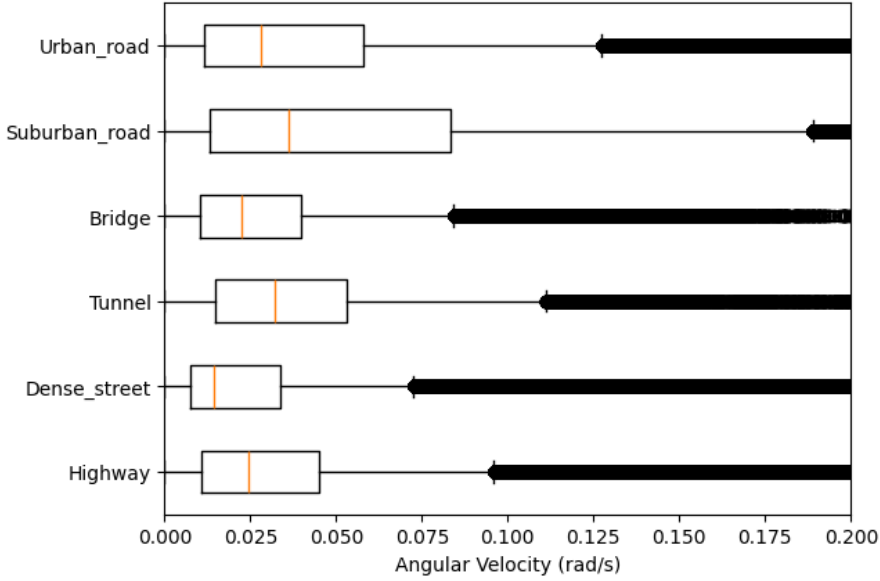


Figure 3.7: The angular velocity distributions for the ECMD dataset broken down by scene.

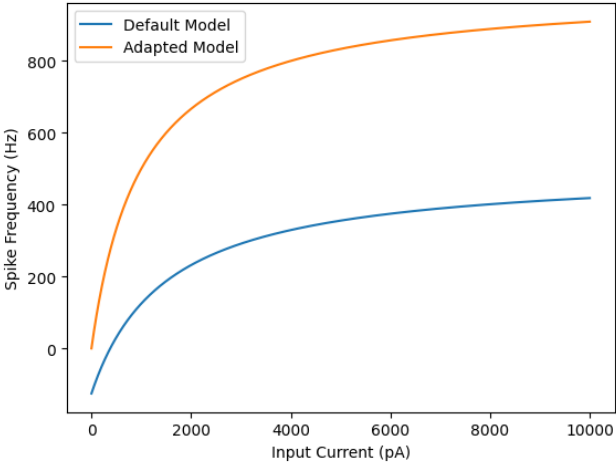
more closely match a linear response; see Figure 3.8. From here, a range neuron and network parameters were tested to find the most accurate mapping from angular velocity to shift rate.

The accuracy of the network's shift rate was assessed by applying a constant input current, representing an angular velocity input, for 50 seconds and recording the amount of rotation experienced by the network. To prevent the cyclical connectivity of the network from interfering with measurements, the network was extended to a width where the HOA would not reach the edge and circle back round. Therefore, the rotation in degrees could be recorded as the difference in index between the neuron at the centre of the HOA at the end of the simulation and the start of the simulation; and the shift rate could be found according to,

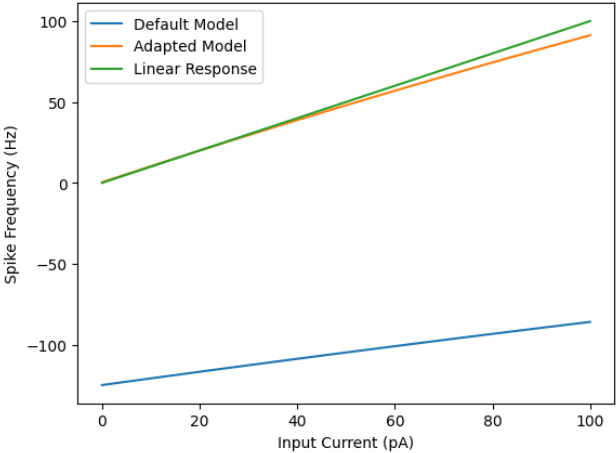
$$r = \Delta i \times t \quad (3.17)$$

where r , Δi , and t are the shift rate, change in index, and simulation time, respectively.

Using this testing methodology, a range of parameters were tested and the shift rates were plotted against the ground truth angular velocity to find the accuracy of the network's shift rate; see Figure 3.9. The responses were assessed and a selection of the most accurate parameter sets were selected to be tested on the ECMD dataset; because the shift responses showed different profiles and therefore deviated from the ground truth in different regions, it was unknown which would provide the best direction tracking when applied



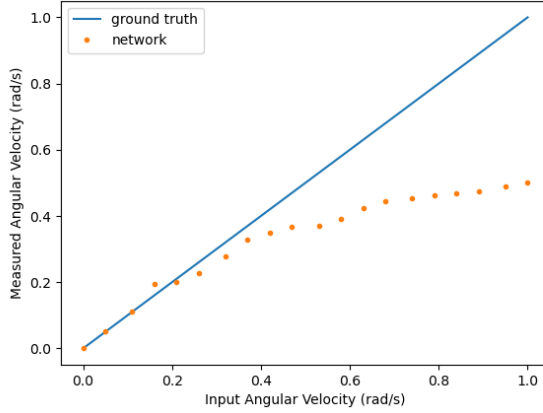
(a)



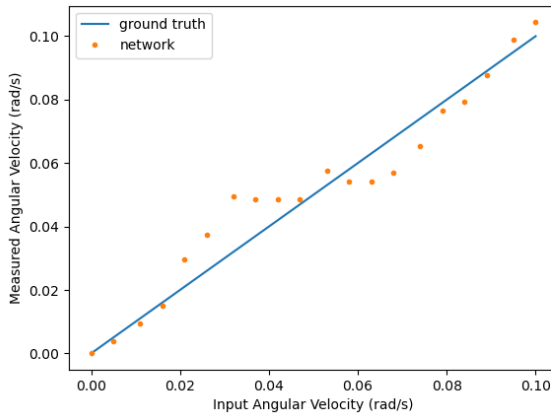
(b)

Figure 3.8: The effect of increasing the resting potential of a LIF neuron on it's f-I response.

to real-world recordings. Hence, multiple parameter sets were tested against a subset of the dataset, and the most accurate was chosen; the shift rate in response to constant, controlled angular velocities is shown in Figure 3.9.



(a)



(b)

Figure 3.9: The shift rate of the Head Direction network when exposed to constant angular velocity inputs. (a) Inputs from 0 to 1 rad/s at 0.05 rad/s intervals to cover the entire range of angular velocities recorded in the ECMD dataset. (b) Inputs from 0 to 0.1 rad/s to cover the majority of the angular velocities recorded in the ECMD dataset at a 0.005 rad/s increments.

Based on the response observed in Figure 3.9 and the connectivity described in Figure 3.5, for smaller angular velocities, which are most commonly observed in the dataset, we are seeing the expected behaviour. Before the angular

velocity begins, the network maintains a stationary HOA that inhibits activity in the shift layers for indexes that are removed from the index of the HOA. When the small angular velocity is injected into the corresponding shift layer, only the shift neurons aligned with the head direction layer HOA become active as the rest of the layer is inhibited. This activity then connects from the shift layer back to the head direction layer, offset by one index in the direction of the shift layer, causing the head direction layer HOA to be skewed in the direction of the shift. With head direction layer connectivity creating a uniform, self-sustaining HOA, when the angular velocity signal stops, the head direction layer HOA settles and is now shifted in the direction of the angular velocity.

For larger angular velocities the rate of shifting does not respond linearly to the angular velocity. Where angular velocities move beyond the range where the shift response closely matches the ground truth, the network is expected to rapidly accumulate error. For example, if the HOA is initially centred at 0° , a large angular velocity may shift the network activity to 10° whereas the ground truth has rotated to 20° . Given the nature of the Head Direction, this offset would then be maintained throughout subsequent activity unless corrected by an external reset signal.

3.4 Results

To test the accuracy of the network in a range of real-world environments, it was run on the ECMD dataset [181]. The network activity and ground truth directions are shown for all scenarios in the Appendix A.2 and the Mean Absolute Error is shown in Table 3.4.

Table 3.4: The Mean Absolute Error (MAE) recorded between the Head Direction network and the ground truth readings when applied to the ECMD dataset.

Scenario	Daytime	Difficulty	Sequence	MAE (Degrees)
Bridge	day	medium	a	5.4
		medium	b	4.8
	night	difficult	a	1.9
		difficult	b	3.1
Dense street	day	easy	e	3.7
		medium	a	3.2
		medium	b	3.9
		medium	c	2.5
		medium	circle b	7.5
	night	easy	a	7.3
Highway	day	difficult	a	14.3
		difficult	b	29.6
		easy	a	4.6
		easy	b	7.8

		medium	a	7.2
		medium	b	8.7
		medium	c	11.0
	night	difficult	c	2.3
		difficult	d	7.4
		medium	b	20.0
		medium	c	16.4
		medium	d	2.8
Suburban road	day	easy	a	9.2
		easy	b	13.8
		easy	c	2.2
		easy	d	9.9
		easy	e	13.1
		easy	f	8.6
		medium	a	69.3
		medium	b	7.7
		medium	c	3.6
		medium	d	2.8
	night	difficult	a	4.2
		medium	a	5.8
		medium	b	27.5
Tunnel	day	difficult	a	5.2
		difficult	b	6.3
		easy	a	8.9
		easy	b	6.2
		easy	c	6.8
		easy	d	2.4
		medium	a	4.9
Urban road	day	easy	a	5.8
		easy	b	10.3
		easy	c	12.6
		easy	d	5.4
		medium	a	11.6
	night	difficult	a	9.7
		difficult	b	2.4
		difficult	c	8.6
		difficult	rainy a	1.4
		difficult	rainy b	10.2
		easy	a	9.0
		easy	b	3.3
		easy	rainy a	11.9
		medium	a	11.0
		medium	b	3.1
		medium	c	5.5

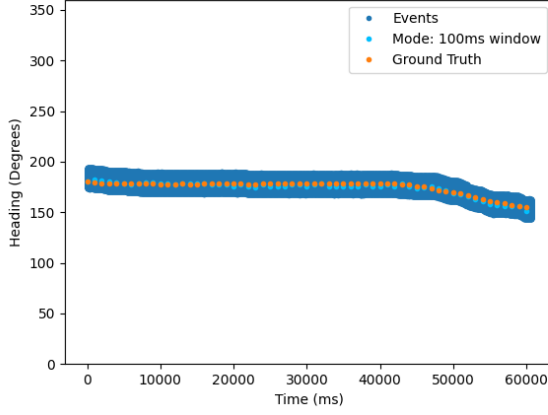
		medium	d	13.5
		medium	rainy a	2.5
		medium	rainy b	10.8

The effectiveness of this network is monitored using the Mean Absolute Error between the ground truth direction and the centre of the HOA in the head direction layer of the network. Given the normal distribution of the head direction layer's firing it is approximated that the network's position is found to be the neuron with the highest firing rate for the 100ms preceding the ground truth reading. Hence, the network position is taken to be the modal neuron index for firing activity in a 100ms window.

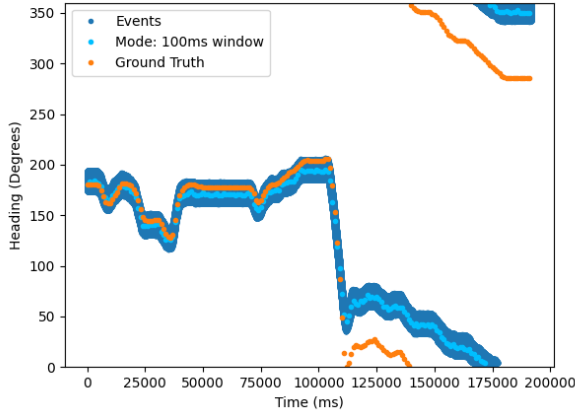
The network was able to track most sequences with a single digit Mean Absolute Error, and achieved a modal error of 3° when rounding to integer values; this was done because at a precision of 1 decimal place most scenarios had distinct values meaning the modal value was not informative. The most accurate sequence, Urban Road Night Difficult Rainy A (see Figure 3.10a), achieved a Mean Absolute Error of 1.4° , while the least accurate sequence, Suburban Road Day Medium A (see Figure 3.10b), achieved a Mean Absolute Error of 69.3° . However, this result is considered to be an outlier with all of the other sequences experiencing a Mean Absolute Error of less than 30° . In addition to all of the results being included in the Appendix A.2, a selection can also be seen in Figure 3.10.

When viewing the network activity against the ground truth direction, it is apparent that there are two types of error that occur in the network. The first is drift, where the network direction gradually deviates from the ground truth direction because of there being a small difference between the vehicle's angular velocity and the angular velocity seen in the network. This results in smaller errors, which grow in size as the simulation continues and can be seen occurring several times in Figure 3.11.

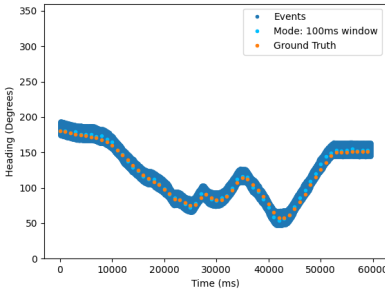
The more notable source of errors, which causes larger and more persistent errors, is when an offset occurs between the network direction and the ground truth direction. Here, a large error accumulates rapidly, which is then maintained as the network continues to shift accurately; see Figure 3.12. This behaviour is particularly important when interpreting the two largest errors in Table 3.4: Highway Day Difficult b and Suburban Road Day Medium b. In both cases, the high MAE is a result of an offset error that occurs midway through the sequence and following a large angular velocity. The large angular velocity causes the network shift response to diverge from the ground truth rotation rapidly and drastically. Once this offset is introduced, subsequent shifts can still follow the shape of the ground truth direction but remain offset from it, causing the large error to persist throughout the remainder of the sequence.



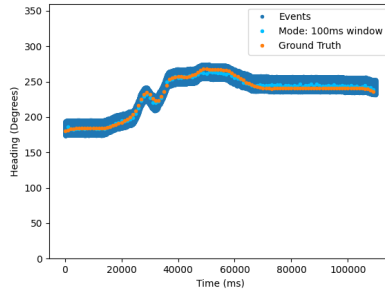
(a)



(b)

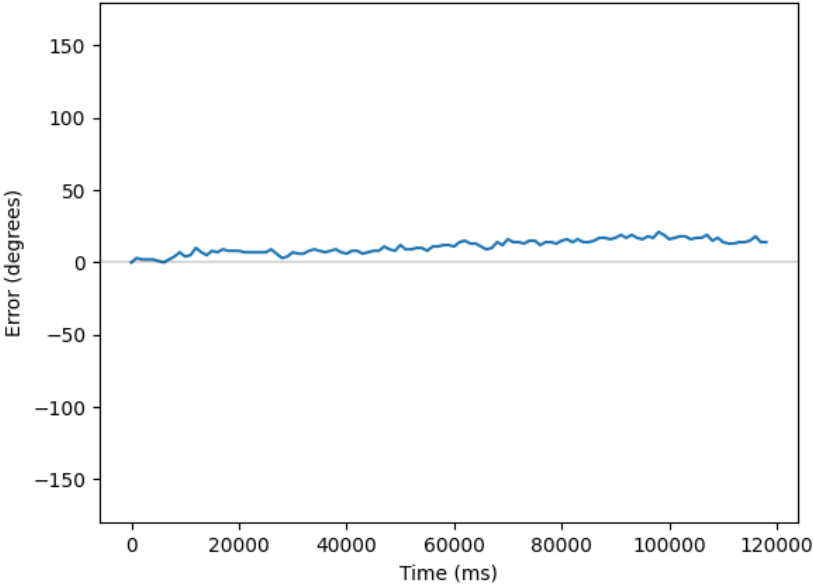


(c)

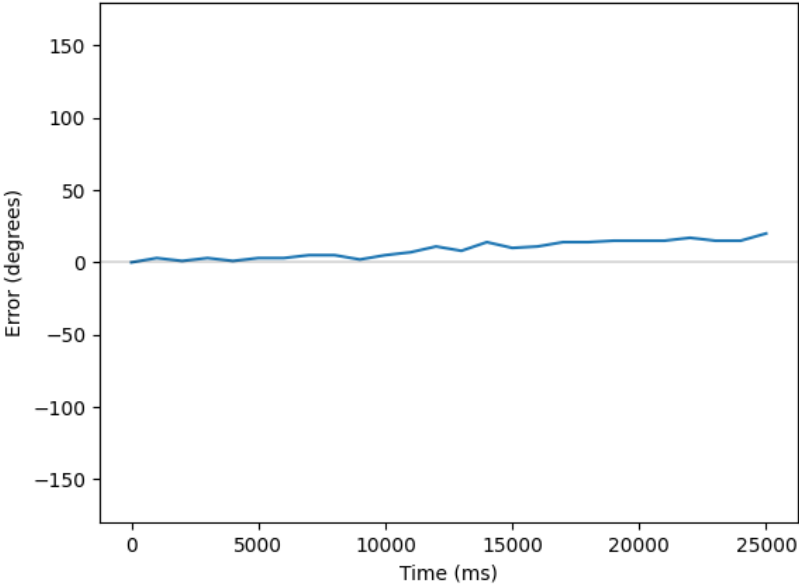


(d)

Figure 3.10: A subset of Head Direction network activity and ground truth data for the ECMD dataset. (a) The sequence resulting in the lowest MAE. (b) The sequence resulting in the highest MAE. (c)-(d) Additional sequences showing accurate direction tracking behaviour.

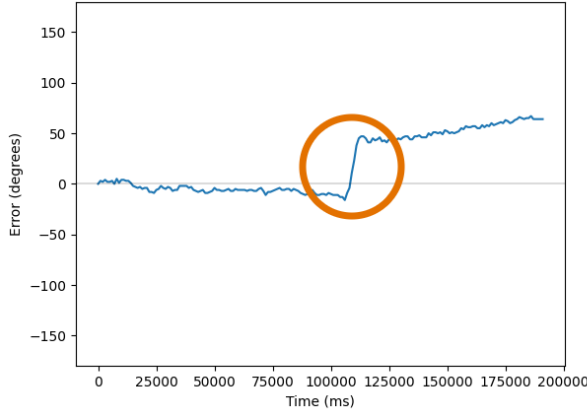


(a)

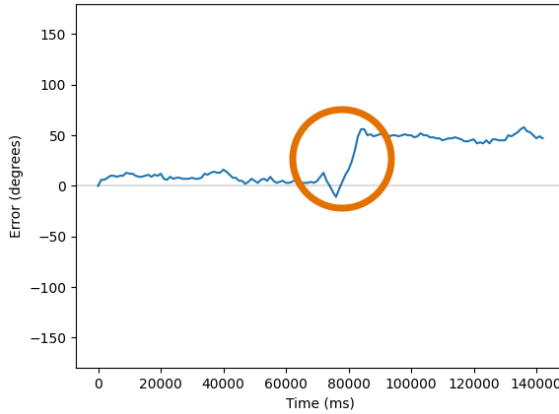


(b)

Figure 3.11: The error in the Head Direction network where drift over time is observed.



(a)



(b)

Figure 3.12: The error in the Head Direction network where a large error appears rapidly and persists, resulting in a continued offset between the ground truth direction and network direction. The beginning of the offset is circled in orange, after which the persistence of the error is visible in the blue error signal.

Given the presence of drift, the error of a sequence was plotted against its runtime; see Figure 3.13. The presence of a positive trend was expected to be seen in this data however, the plot actually reveal a clustering, implying that drift is not a primary cause of error in the system. Therefore, further analysis was performed on the results.

When grouping the sequences by Daytime, Difficulty, and Scenario (see Table 3.5), we see that the greatest mismatch in errors occurs when the

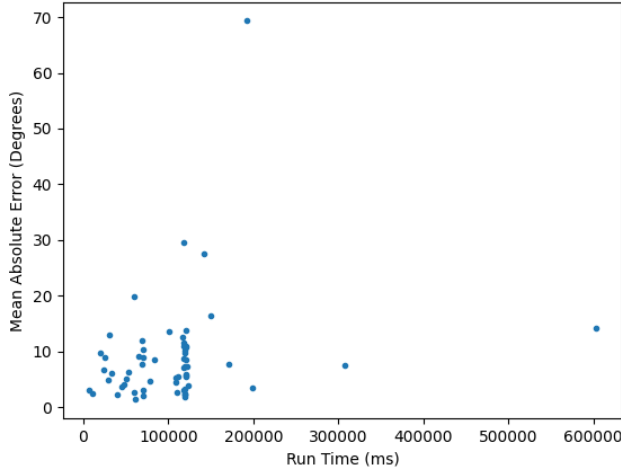


Figure 3.13: The mean absolute error plotted against the duration for each sequence of the ECMD dataset.

sequences are grouped by scenario. This implies that environmental factors could be influencing the error observed in the network; however, it could also be because this grouping has the most categories and, therefore, the smallest group size, these groups are more affected by single results with a large error. Anecdotally, the offset errors seen in Figure 3.12 occur within the Highway and Suburban Road scenarios, where the largest errors are also seen in Table 3.5 and Figure 3.14. It is hypothesised that an environmental factor might trigger the occurrence of an offset although a unique characteristic of the activity about these occurrences hasn't been identified in the activity plots.

Table 3.5: The average MAE when grouping the sequences of the ECMD dataset by time of day, difficulty, and scenario.

Group	Value	Average MAE (Degrees)
Daytime	Day	9.5
	Night	8.1
Difficulty	Easy	7.8
	Medium	10.5
	Difficult	7.6
Scenario	Bridge	3.8
	Dense Street	4.7
	Highway	11.0
	Suburban Road	13.7
	Tunnel	5.8
	Urban Road	7.8

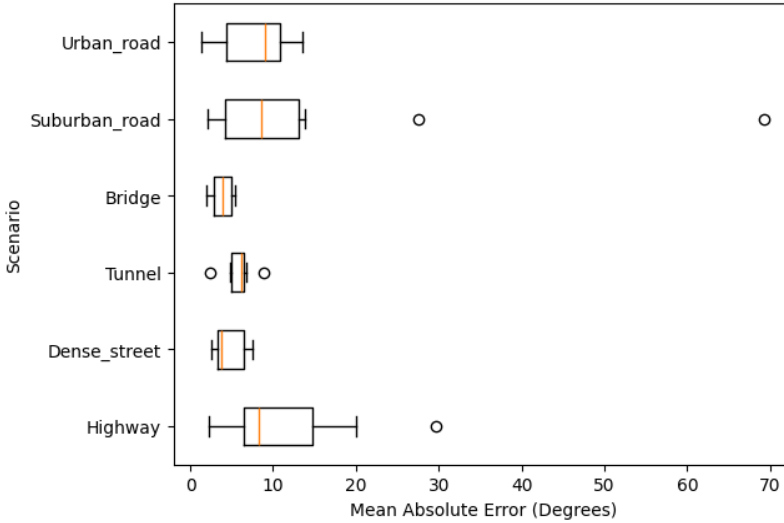


Figure 3.14: The distribution of the MAE when grouped by scenario.

3.5 Discussion

The results obtained from the empirically derived Head Direction network demonstrate that a spiking neural network can be implemented using a bio-inspired ring attractor network to track direction across a range of driving scenarios. In the majority of the sequences from the ECMD dataset, the network achieved single-digit accuracy, indicating that the bio-inspired approach provides sufficient accuracy for real-world navigation tasks. The 3° modal error also confirms that the network reliably maintains an accurate representation of direction in most scenarios.

Despite most results being positive, two main sources of error were observed, drift and offset. Drift occurred gradually when small discrepancies between the angular velocity input and the network's shift rate accumulated over time. Although the magnitude of these errors was relatively small, their presence highlights the sensitivity of direction tracking to mismatch between parameters and noise that can occur in both the network and the sensors.

In contrast, offset errors represented more significant problems that arise when the network loses touch with the ground truth, resulting in a large persistent deviation which was reflected in the recorded errors despite the shift rate continuing to function accurately afterwards. Such errors were particularly detrimental, as they are uncommon and therefore unlikely to be corrected locally; for example, through drift in the opposite direction.

These results show that the Head Direction network alone is insufficient for continued accuracy as there is a need to correct the persistent offsets and gradual build up of drift as these cannot be corrected locally by the network's dynamics. As a result, the system should incorporate external cues to relocalise to known directions when they are detected. Equally, improving the accuracy

of the network’s shift response is important to reduce the effect of drift and will probably also reduce the occurrences of offset errors.

From a hardware perspective, the Head Direction network is well suited to implementation on neuromorphic hardware because it uses LIF neurons, sparse spiking activity, and a fixed structure with a low synaptic density. The network contains 1,080 neurons across the head direction and shift layers, which is far smaller than the capacity of existing neuromorphic platforms discussed in Section 1.2.1. Further optimisation could be achieved by reducing the angular resolution where 1° precision is not required.

Taken together, these results and the consequent findings demonstrate both the promise and the limitations of spiking neural networks for head direction tracking in real-world applications. The ability to repeatedly maintain accurate direction estimates in diverse real-world driving conditions validates the approach as a biologically inspired solution. However, the observed drift and offset errors highlight the need for integrating complementary signals and refining the existing network mechanisms. Following the same principles that proved successful in the generation of the Head Direction network, the following chapter discusses how this is extended over 2 dimensions using a Grid network to integrate velocity.

Chapter 4

Velocity Integration

4.1 Introduction

Accurate localisation is a requirement for effective autonomous navigation, particularly in environments lacking external cues. In biological systems, spatial awareness is achieved through a complex network of specialised neuron populations interacting with one another, most notably grid cells and place cells. Velocity integration can be used to localise within self contained coordinate systems through by replicating the behaviours observed in grid cells, enabling neuromorphic navigation.

Biological grid cells exhibit a hexagonal firing pattern that performs path integration, encoding a relative spatial position. In artificial systems, spiking neurons, replicating grid cells, are used to continuously integrate motion cues. In biological systems such a cue may originate from speed cells, but in neuromorphic application this signal can be derived from acceleration recordings from an IMU, or directly through recording velocity. This grid network constructs an internal coordinate system that can be maintain its representation of position as well as updated in real time, enabling the continuous tracking of the agents position within the self contained coordinate system.

This chapter presents the theoretical framework for a velocity integration system based on grid cell models. Regarding implementation and validation, we present a network architecture and connectivity that creates a stable representation of position within the grid coordinate system and shows shifting behaviour, but are unable to create the predictable shift response to speed signals needed for velocity integration.

Unlike the Head Direction, Place Cell, and Vision networks, here, a final set of parameters is not presented because the implementation did not produce a predictable shift response. Instead, the chapter reports the network architecture and connection polarity, as well as observed limitations. This reflects the intention to investigate extending the shift mechanism developed for the Head Direction network into two-dimensional velocity integration. As no parameter set produced reliable shifting, reporting a single final parameter table would imply a functional implementation that was not achieved.

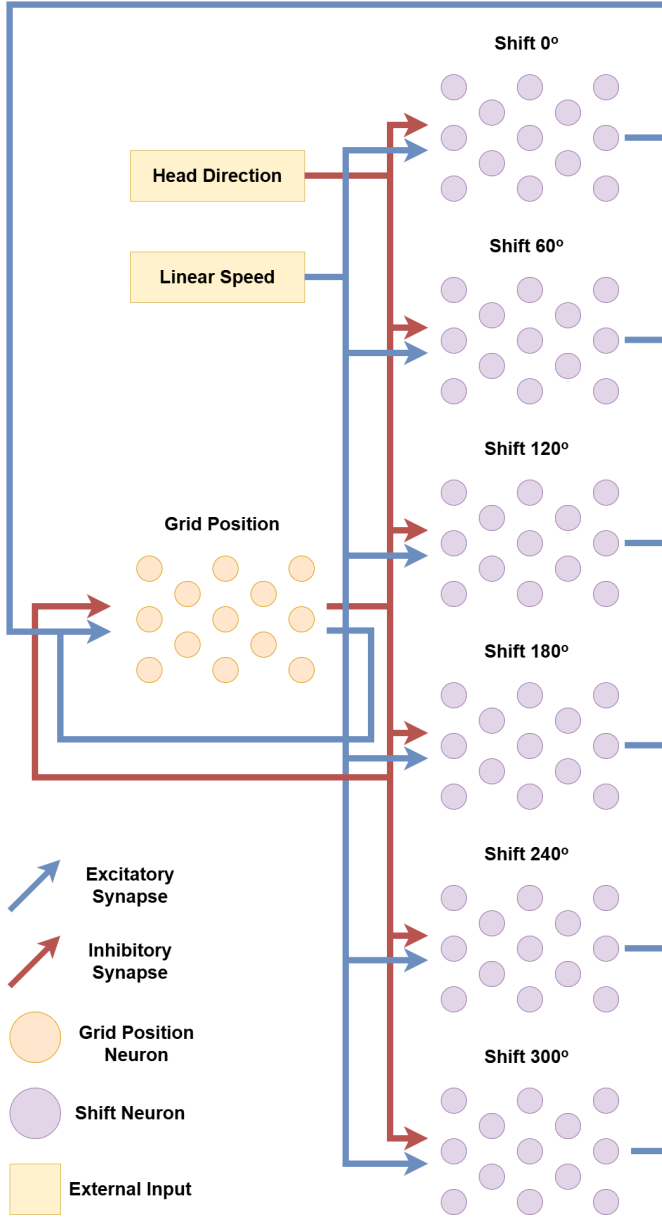


Figure 4.1: An overview of the layers in the Grid network, and how they are connected. For detailed, neuron level connectivity mapping see Figures 4.2, 4.3, and 4.4. If the Grid Position neuron at (5,5) is active then it excites its neighbours and inhibits distant Grid Position neurons. It also inhibits distant neurons in the six Shift layers. The Head Direction signal inhibits the Shift layers that don't match the current Head Direction, meaning, only the Shift neurons at similar (x, y) coordinates, in the Shift layer representing the current Head Direction aren't inhibited. Linear Speed attempts to excite all of the Shift neurons, but only those that aren't inhibited spike. These Shift neurons connect back to the Grid Position layer one step in their designated direction, meaning activity in the Grid Position layer shifts.

4.2 Network Connectivity

The Grid network applies the same concepts as those seen in the head direction network (see Chapter 3 and Figure 4.1), but does so with an added dimension to integrate 2-dimensional movement rather than 1-dimensional changes in direction. As such, the Grid network is formed using a torus network structure, facilitating cyclical connectivity for continuously tracking all directions of planar movement. The network consists of seven layers, each layer being made up of an offset 10 by 10 grid of neurons. Within each layer of neurons, every other row of the 10 by 10 grid is offset to create a regular, continuous triangular lattice where each neuron represents a vertex; this spatial distribution of the grid neurons can be seen in the connectivity plots such as Figure 4.2. These layers are the grid position layer and six shift layers. Similarly to the head direction network, the grid position layer uses a self-sustaining HOA [153] to represent the agent's current position within the grid. The six shift layers are used to stimulate a movement of the HOA to be centred about their respective neighbouring grid position neuron.

Conceptually the Grid network continues to mimic the head direction network with the need for the representation of position to persist in the absence of any input. As such, the neurons within the grid position layer feature excitatory connections to their neighbouring neurons. To avoid the aforementioned excitatory connections resulting in activations spreading across the whole network, each neuron also makes inhibitory connections to those neurons that are more than two 'steps' removed from their position. These connections can be seen in Figure 4.2.

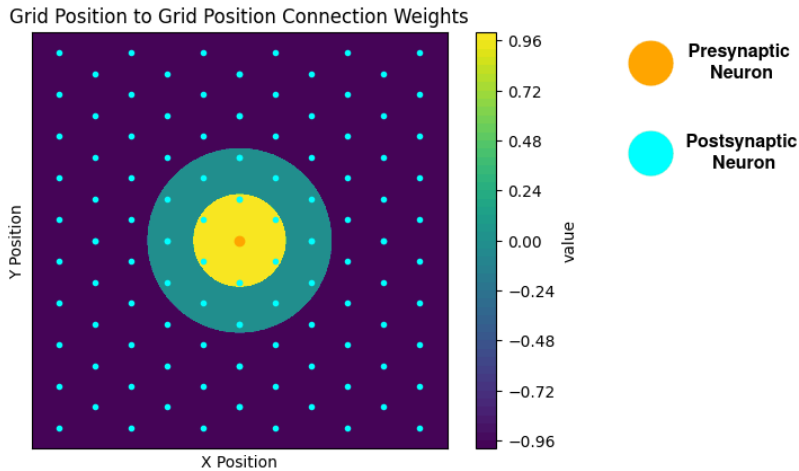


Figure 4.2: The connection polarity between neurons in the grid position layer.

The shift layers in the Grid network are driven by the speed signal discussed in Section 4.3.1. The speed signal is connected to every neuron in all six shift layers via equally weighted excitatory Synapses to drive equivalent excitation

throughout the network. To modulate shifting activity, there are inhibitory signals that originate from two sources, the grid position layer and the head direction network. The grid position layer prevents shift neurons that are not located in the same position as the current HOA from becoming active, thus preventing erroneous shifts in position. This is done through inhibitory connections to all shift neurons greater than two 'steps' from the position of the grid position neuron; see Figure 4.3. The head direction network is used to inhibit all of the shift layers which represent a that is different to the one represented by the HOA within the head direction network. No specific connectivity or weights were established for these inhibitory connections, as the grid network, for reasons which are discussed later in Section 4.3.

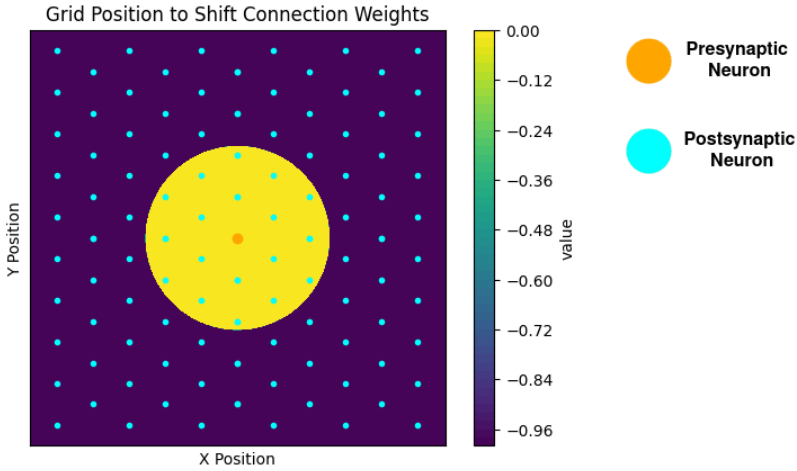


Figure 4.3: The regions where grid position neurons make inhibitory connections to neurons in the shift layers.

The six shift layers allow for the shifting of activity in the grid position layer by one 'step' at increments of 60° . This is done by each layer forming strong excitatory connections to neurons within one 'step' of the neuron that is offset from their current position by one 'step' in the respective directions; see Figure 4.4.

A summary of the aforementioned connectivity is found in Table 4.1

Table 4.1: The connectivity used in the Grid network

Presynaptic Neuron	Postsynaptic Neuron	Weight	Range
Grid position	Grid position	-ve	$\sqrt{(x - x_1)^2 + (y - y_1)^2} > 2$
Grid position	Grid position	+ve	$\sqrt{(x - x_1)^2 + (y - y_1)^2} \leq 1$
Grid position	Shift	-ve	$\sqrt{(x - x_1)^2 + (y - y_1)^2} > 2$
Shift	Grid position	+ve	$\sqrt{(x - x_1 - \Delta x)^2 + (y - y_1 - \Delta y)^2} \leq 1$

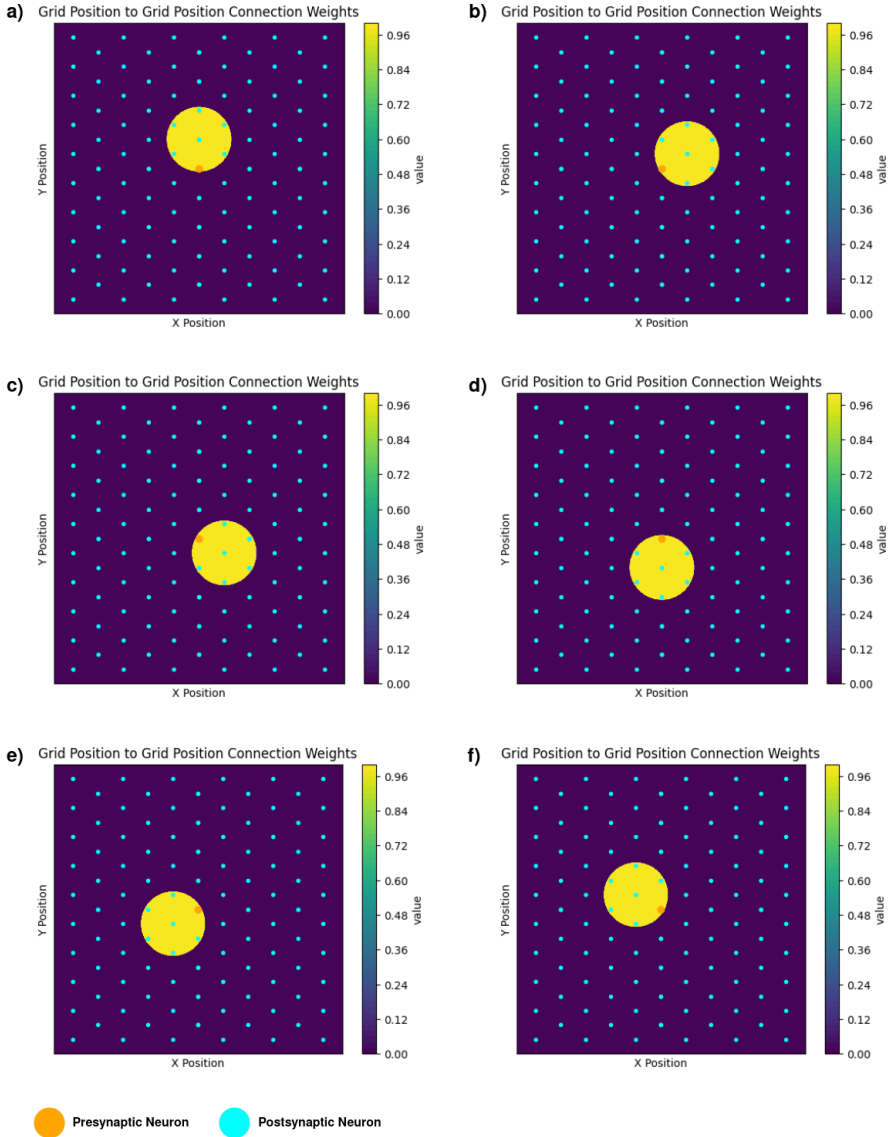


Figure 4.4: The regions where the neurons in the shift layers make excitatory connections to neurons in the grid position layer.

The speed signal was passed directly to neurons in the six shift layers as an excitatory input current, allowing the magnitude of the input to modulate the movement of the HOA via the Shift layers. Directional information was not encoded as a separate feature; instead, the Head Direction network was intended to select the relevant shift layer by inhibiting the shift layers that did not correspond to the represented direction of travel. This encoding was dictated by the spiking activity used in the Head Direction network, where

motion is represented through movement of an existing HOA. As a result, the Grid network could be assessed on whether its connectivity and shift-layer dynamics were capable of producing stable velocity integration directly from speed input.

The speed signal was passed directly to the Shift neurons as an input current that was linearly proportional to the speed of the vehicle. This allowed the network to be assessed for velocity integration without the need for preprocessing, meaning the effect of the network connectivity and shift dynamics could be assessed directly. Within the Grid Position layer, rate coding is used, where the spike frequency denotes the proximity of each neuron to the centre of the HOA, and therefore to the represented position. The Shift layers combine inputs from the Grid Position layer with the speed input, with the active shift layer determined by the direction represented by the Head Direction network. The Shift layers then use time-to-first-spike to determine the rate at which activity moves within the Grid Position layer.

4.3 Testing

To verify the functionality of the network, the following testing schedule was conceived. Initially, stable behaviour of the network is verified, with the network maintaining its representation of position. The network is then exposed to a speed signal along a single shift direction to ensure a functioning shifting mechanism. This progresses to recording the rate of movement and investigating how the rate responds to different input values, and changing parameters to accurately map speed to the rate of shifting. The ability to accurately scale shifting according to speed is where the network failed. When testing network parameters, a set of parameters wasn't found that allowed the shifting rate of the network to scale with the speed signal so the experiments following this could not be run but are mentioned here to document the intended testing process. From here directionality is introduced through an artificial head direction signal, which is derived from the ground truth direction to control for errors. The connectivity is designed to inhibit movement that doesn't approximately match the head direction; and weights are tested to scale the rate of shifting off the shift axes. With shifting effectively managed for all directions, different magnitudes of step size can now be introduced by changing network parameters to create a range of different grid networks, operating over different scales, which when combined encode absolute position in their activity. At this point, the grid networks are developed and all of the aspects have been independently tested, so it can be integrated into the wider system by replacing the ground truth direction with the head direction network, and by adding the place cell network, discussed in the next chapter.

4.3.1 Input Signals

The Grid network requires a speed signal to induce a shift of the HOA. The IMU provides acceleration data so originally it was intended to drive the Grid

network via a speed cell which tracked the speed of the vehicle through the acceleration data recorded from the IMU. Investigating the accuracy of a speed signal derived from IMU acceleration data revealed that it didn't provide an accurate estimate of speed, and calculating speed from ground truth positions was used instead. To investigate how the IMU acceleration data correlated to speed, an estimate of the speed was calculated from it according to,

$$v = u + at \quad (4.1)$$

where v is the current speed; u and a are the speed and acceleration at the time of the previous IMU reading; and t is the difference in seconds between the current time and the time of the previous IMU reading. This was compared against the speed estimated using the ground truth positions, assuming a direct line of travel between points.

$$d = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (4.2)$$

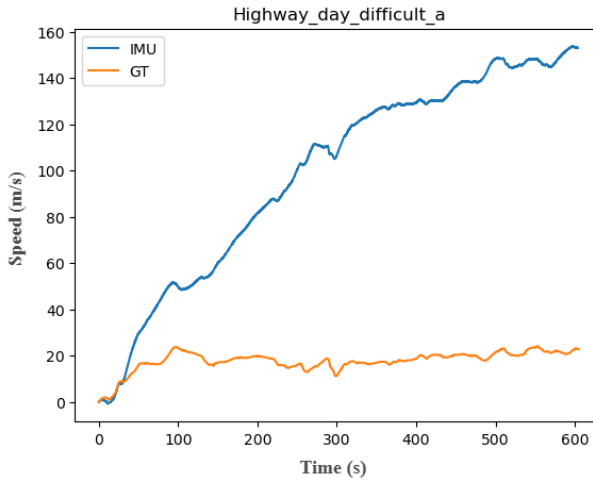
$$v = \frac{d}{t} \quad (4.3)$$

Where d is the distance between the ground truth readings, calculated using (x, y) and (x_0, y_0) the current and previous coordinates; and v is the current speed, estimated using the distance and t , the time between the ground truth readings.

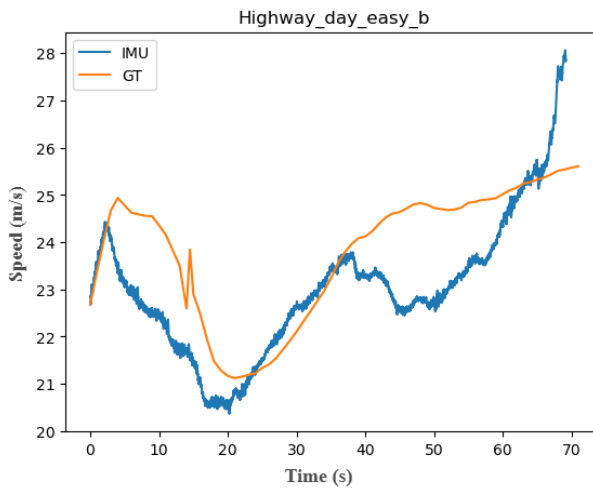
A subset of the results of this analysis is shown in Figure 4.5. Given that the speed calculated from the acceleration measurements was so significantly different from those estimated through the ground truth, and that there was no discernable pattern in the errors, it was decided to proceed with the speed estimate from ground truth as the speed signal.

4.3.2 Speed Analysis

Having determined the source of the speed signal, and following from the previous analysis of the angular velocities the same approach was employed for analysing the speed signals (Section 3.3.1). The distribution of the speed signals was plotted initially using a histogram (see Figure 4.6, revealing a much larger range of values, as well as different distribution of values. It was observed that, again, a significant number reading were focused around the origin (0 m/s) and there was also a peak in the range of 15-20m/s. Mapping the larger range of values found in the speed data, to the range found in the angular velocity data can be handled by weighting the input Synapses, however it was anticipated that mapping the distribution to the networks shifting profile would be more challenging than for angular velocities. This is because having peaks at the edges of the distribution requires accuracy at both extremes of the mapping, and it is therefore challenging to map to the logarithmic profile of the networks shift response to the linear response of the input throughout the majority observed values.



(a)



(b)

Figure 4.5: Estimations of the vehicle's speed based on acceleration recordings from the IMU, and using the position recorded in the ground truth data from the ECMD dataset.

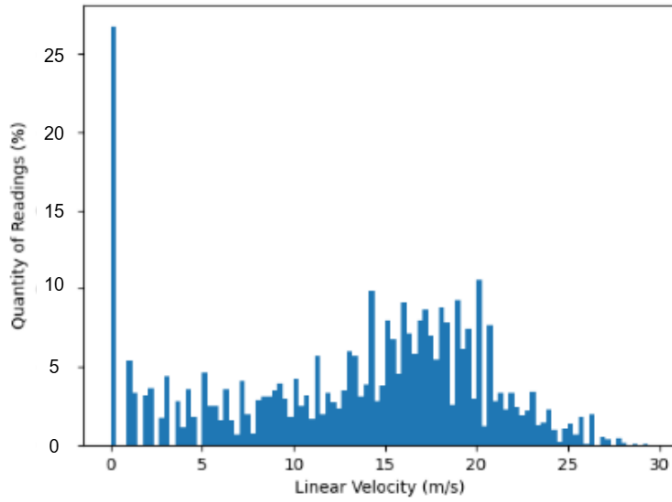


Figure 4.6: The distribution of speeds estimated from the ground truth data from the ECMD dataset.

Calculating the range, mean, and standard deviation further reinforced the previous interpretation, showing that the speed data featured more of a normal distribution; see Table 4.2. The mean and standard deviation highlighted the need for accuracy through the middle of the range, 6-20m/s.

Table 4.2: A summary of descriptive metrics for the speeds estimated from the ground truth data from the ECMD dataset.

Measurement	Speed (m/s)
Minimum	0.0
Maximum	29.4
Mean	13.2
Standard Deviation	7.1

Finally, the speed data was grouped by scenario and plotted using boxplots (see Figure 4.7 to show if the signals were consistent between environments. Here, it was noted that different scenarios varied significantly in their distribution, e.g. Dense Street vs Bridge, shown in Figure 4.7. This highlights the need for accuracy across the whole mapping from velocity to shift rate for the Grid network, and if this couldn't be achieved was expected to show in results being strongly correlated to their environment.

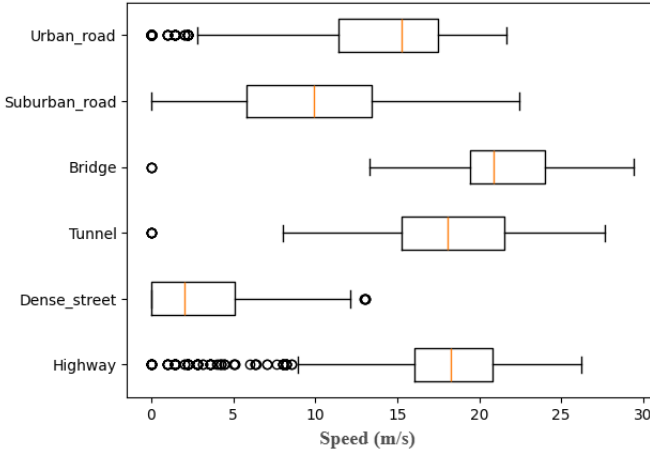


Figure 4.7: The distributions of speeds for the ECMD dataset, grouped by scene.

4.3.3 Network Tuning

By implementing the connection specification described in Section 4.2 the Grid network produces a stable HOA, when the network isn't receiving any inputs (see Figure 4.8b and 4.8a). It is also observed that the HOA shows a uniform spike rate according to distance from the receptive field. To verify a correct spatial representation at the fringes of the network a position signal was also induced at the corner of the network, showing how activity wraps around, making cyclical connections in all directions; see Figure 4.8d and 4.8c.

Moving the HOA using the shifting mechanism is then tested by inhibiting all of the shift layers except one. This is done to prevent the network from attempting to shift in all directions at once, thus resulting in no movement except for that which is induced through noise. A speed signal is introduced, which is passed to all shift neurons. The shift neurons that are local to the position of the HOA in the uninhibited shift layer become excited and create a shift signal in their given direction, resulting in movement of the HOA. While the network is stimulated by a speed signal, activity continues to shift along the axis of the shift layer, resulting in uniform activity along this axis; see Figure 4.9.

With the network successfully shifting activity, and the circular connectivity working correctly, the same strategy previously used to record the shift rate of the head direction network was used to record the shift rate of the Grid network. Speed inputs, in the form of constant input currents, were injected into the network. The network was extended in the direction of shifting (vertically) to prevent the HOA from circling around the network, and the difference in start and end position was recorded to determine the shift rate.

Following this test protocol, a range of parameter sets were tested, and the shift rates were plotted against a linear shift response. In contrast to the

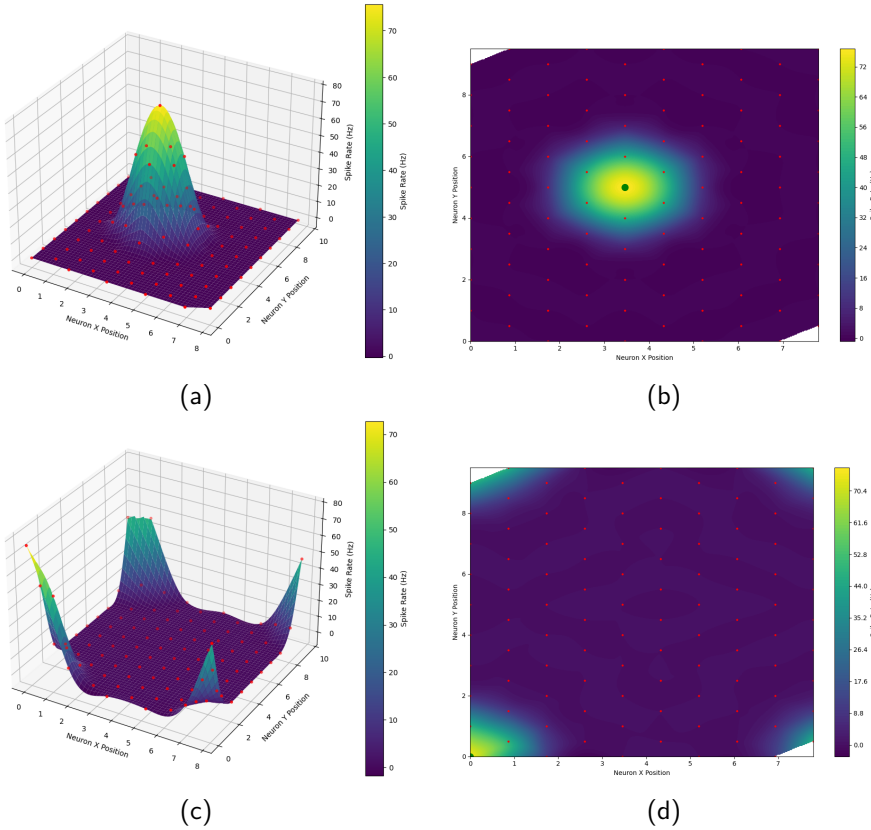


Figure 4.8: The HOA in the grid position layer of the Grid network, tested at the centre and fringe of the network.

angular velocity test, given that grid cells operate at a range of scales, meaning the step size of the network is arbitrary; the focus of this test was not to directly map the shift rate to the input speed. Instead, a linear shift response was the goal and a way to modulate the gradient of this shift response would facilitate the instantiation of multiple Grid networks, operating at different scales and providing distinct coordinate systems.

The input weight or the membrane time constant was expected to modulate the shift response of the network by scaling the input or scaling the volatility of the network, respectively. However, the results of this testing revealed significant instability within the network. Most tests caused the network to saturate with activity spreading across all grid cells or remain static with the input inducing no movement in the HOA. It was not possible to invoke a shift response over a meaningful range of inputs and therefore the network failed to perform velocity integration.

The input weight and the membrane time constant were selected as the primary parameters for this search because they were expected to modulate

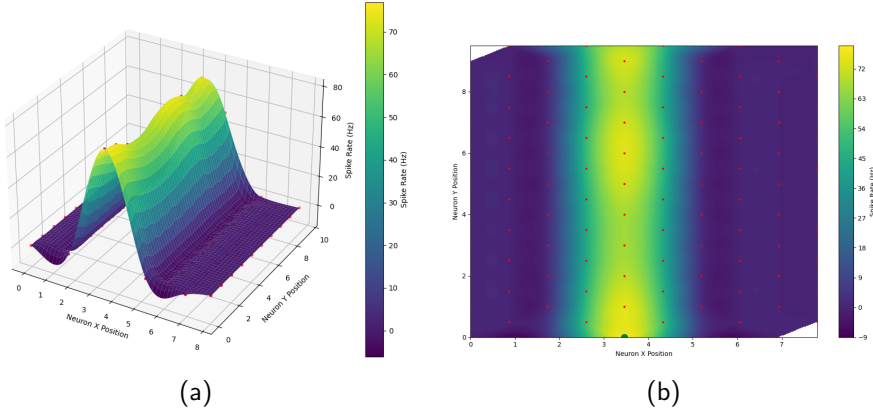


Figure 4.9: The Grid network activity shifting as expected along the axis of one of the shift layers to verify the functionality of the shift mechanism.

the shift response of the network by scaling the input strength and altering the responsiveness of the neurons, respectively. The search was performed by fixing one parameter before iterating through the second parameter. Once these responses had been evaluated, the first parameter was then incremented before iterating through possible values of the second parameter again, until the parameter space was exhausted. The search was stopped when parameter changes no longer produced any movement of the HOA or resulted in instability and saturation of the network. Therefore, while this does not rule out the possibility that an alternative network design could produce velocity integration, the parameter space tested here did not produce a stable and controllable shift response.

The Grid network parameters were explored using the Head Direction network as a starting point, because both networks rely on shifting a stable HOA through recurrent attractor dynamics. However, extending the shift mechanism from a one-dimensional ring to a two-dimensional toroid increased the complexity of the connectivity and made the Head Direction parameters ineffective. The parameter values were adjusted iteratively to test whether the network could produce movement of the HOA, whether this movement was controllable, and whether the network avoided saturation. Since no set of tested parameters produced reliable velocity integration, a different network structure is needed.

In summary, the network showed the desired stable behaviour, maintaining a HOA. The shift mechanism was also shown to be effective, with the HOA moving through the network with the introduction of a speed signal. However, when the shift rate of the network was measured, it wasn't possible to scale the shift rate with the input meaning that the network can't accurately track movement at different input speeds. Therefore, the interaction between the Head Direction network and the Grid network couldn't be tested, and the grid network couldn't be applied to real-world data through the ECMD dataset. To

account for this in the rest of the project, the expected Grid network activity was calculated from the ground truth positions (referred to as the artificial grid signal) and used as a surrogate.

4.4 Discussion

Having previously been unable to achieve the desired static and shifting behaviour, through a theoretical approach, an empirical approach was taken from the outset to developing the Grid network. This was done because such an approach had proven successful with the Head Direction network where there are many parallels. Translating this implementation from 1 dimension to 2 dimensions resulted in the desired static behaviour and evidenced a functioning shift mechanism.

Controlling the shift behaviour through modulation of the network and neuron parameters failed to produce the desired shift response to input speed signal, meaning that a functional Grid network couldn't be implemented.

While the grid network demonstrated the ability to sustain stable activity and to shift in response to an excitatory speed input, the inability to scale the shift rate in proportion to the velocity input revealed a fundamental limitation in the current design. The expected modulation of neuronal and network did not produce the required shift response for velocity integration. Instead, the network frequently lost stability, either saturating with universal excitation or failed to respond, with the HOA remaining static. This shows that the network, in its current state, is unable to respond effectively to the inputs found in the ECMD dataset.

It is thought that these shortcoming are caused by two distinct sources, the input distribution, and the spatial disparity between the Head Direction network and the Grid network. The angular velocity recordings featured a narrow range focused around 0 rad/s, the speed data exhibited multiple peaks and a greater spread of values from approximately 0-30m/s. This made it particularly difficult to achieve an accurate mapping between the input speed and the shift rate that was valid across the full range of values. The speed data also presented variations in its distribution between environments, a feature that wasn't present in the angular velocity data. However, this largely was not experienced by the network due to its failure to pass testing before being exposed to the real-world data.

The most probable cause for the premature failure of the Grid network was the disparity between the Grid network and the Head Direction network, where many of its features were borrowed from. The 2 dimensional nature of the Grid network meant that increasing connectivity by 1 'step' increases the number of neurons that the Presynaptic neuron connects to by a factor of 6, preventing granular tuning of network connectivity. It is thought that the network is inherently less stable as a result of this and therefore a theoretical approach is needed, as this facilitates more precise tuning to elicit the desired behaviour.

Despite being ineffective in this instance, conceptually, the Grid network is

well designed for implementation on neuromorphic hardware; using LIF neurons, sparse activity, and low synapse density. With each grid scale consisting of 700 neurons, the neuron count remains low even when multiple scales are considered. However, the results in this chapter show that the current connectivity and parameters do not yet provide a stable velocity integration. This means that the network cannot properly be appraised for implementation on neuromorphic hardware, as the final, working implementation has not been established. It may be that once working and transitioning to hardware the network is significantly larger and features different connectivity.

Despite being ineffective in this instance, the Grid network remains conceptually well designed for implementation on neuromorphic hardware because it uses LIF neurons and sparse activity. With each grid scale using 700 neurons, the neuron count remains low even when multiple scales are considered. However, the results in this chapter show that the current connectivity and parameters do not yet provide stable velocity integration. Therefore, the network cannot be fully appraised for hardware implementation until a functional version has been established. It may be that once the network is working and able to be mapped to hardware, a larger network or a different connectivity pattern is required.

It is assumed that were the Grid network to function as intended, the limitations seen in the Head Direction network would also be present here. Given the assumed, shared limitations, it is also fair to assume that incorporating external cues to trigger re-localisation within the local coordinate system would account for persistent and gradual errors.

Although this implementation proved ineffective, stable state behaviour and a shifting mechanism were shown to work. The concept also has biological evidence and can still theoretically be translated into an artificial implementation through a different approach. Therefore, it is believed that Grid networks are still a valid component of a bio-inspired neuromorphic navigation system and going forward grid cell activity is emulated from ground truth recordings to facilitate the completion of the project, without drifting into the realm of infeasibility.

Chapter 5

Position Tracking

5.1 Introduction

Localisation is the process by which a system determines its position within an environment. In biological systems, this capability is performed by specialised place cells which are neurons that activate when an animal is in a specific location within an environment. Each place cell corresponds to a particular region in space, and when combined with other place cells, they form a map of the environment through their combined activity. Neuromorphic navigation systems can localise an agent within an environment using the same principles, using place cells to represent specific known locations.

While grid cell activity repeats across an environment, each place cell represents a distinct Place Field. Such a particular representation requires larger amounts of preprocessing of the inputs than seen in the previous networks, as well as more complex connectivity which in this case is achieved through training. Here, this preprocessing is achieved by using grid signals to induce place cell activity. The place cells are trained to be triggered by the grid cells exclusively at their assigned location so that after training whenever the same combination of grid activity is encountered the place cell again becomes active.

Chapter 4 attempted to create an internal coordinate system that performs velocity integration. Using the integrated signals that are expected from that work, this Chapter implements a method for training place cells to perform localisation within a known environment. Using artificial grid signals to mimic velocity integration over the ECMD dataset, allowing the place cells to continuously localise the vehicle throughout the simulation.

5.2 Network and Training

Unlike the Head Direction and Grid networks the place cell network doesn't consist of multiple layers, or feature any connections within the layer. Instead, the place cell network acts as an interpreter, condensing the activity of several Grid networks, each operating at different scales, into a single position signal

which is more easily digested by humans, or used downstream by other networks.

The Place Cell network receives grid-cell activity directly as its input, allowing each place cell to operate independently as a coincidence detector for a given Place Field. Unlike the Head Direction and Grid networks, the timing of spikes is not used to encode a shift rate or motion update in this network. Instead, activity in a place cell indicates that the combination of active grid cells corresponds to the Place Field associated with that place cell. This is closer to a temporal code than a rate-based encoding, as the relevant information is whether a place cell becomes active at a specific moment in time.

The Place Cell network parameters listed in Table 5.1 were selected to test whether grid-cell activity could be learned to represent a Place Field. The default `iaf_psc_alpha` neuron parameters were used, avoiding neuronal parameter tuning, with the expectation that training the Synapses would be sufficient to form the required associations. The λ and α parameters of the `stdp_synapses` were manually adjusted during training to produce stable Place Cell activation without causing saturation. Initially, λ was set higher and α lower to encourage synaptic weights to increase, with these values reducing as training progressed, to avoid network saturation. Training signals were selected to assign a distinct Place Cell for each Place Field.

Table 5.1: Key neuron and synapse parameters used in the Place Cell network.

Parameter	Value
Neuron model	<code>iaf_psc_alpha</code>
Resting potential, E_L	-70 mV
Membrane capacitance, C_m	250 pF
Membrane time constant, τ_m	10 ms
Refractory period, t_{ref}	2 ms
Threshold potential, V_{th}	-55 mV
Reset potential, V_{reset}	-70 mV
Excitatory synaptic time constant, $\tau_{syn,ex}$	2 ms
Inhibitory synaptic time constant, $\tau_{syn,in}$	2 ms
Constant input current, I_e	0 pA
Plastic synapse model	<code>stdp_synapse</code>
acshortstdp step size, λ	Manually adjusted throughout training
acshortstdp asymmetry parameter, α	Manually adjusted throughout training

5.2.1 Distance Analysis

Before the place cell network can be trained, a Place Field for each place cell must be established so that an accurate ground truth signal can be created. The Place Fields must be sufficiently precise as to create an informative signal without being so fine grained that computational efficiency is jeopardised. Therefore given the specific application of the place cells to driving scenarios,

the distances covered by the ECMD dataset, and general road geometry were analysed to determine an appropriate scale for the Place Fields.

In contrast to previous analyses, where the mean and standard deviation were the focus, the minimum and maximum distances covered along the x and y axes were deemed the most important. The reason for this is that this is where the limits of the required neuron count and the required spatial granularity are tested. These values can be seen in Table 5.2.

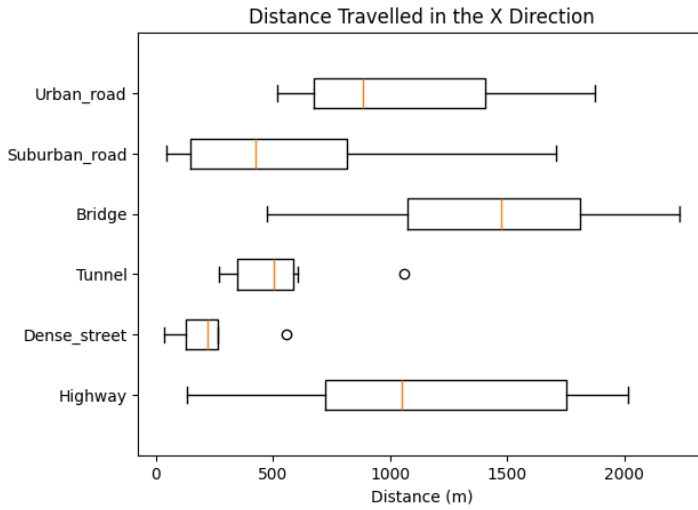
Table 5.2: The minimum and maximum distances covered by a single sequence in the ECMD dataset along each axis.

Axis	Minimum (m)	Maximum (m)
X	35	2235
Y	3	9431

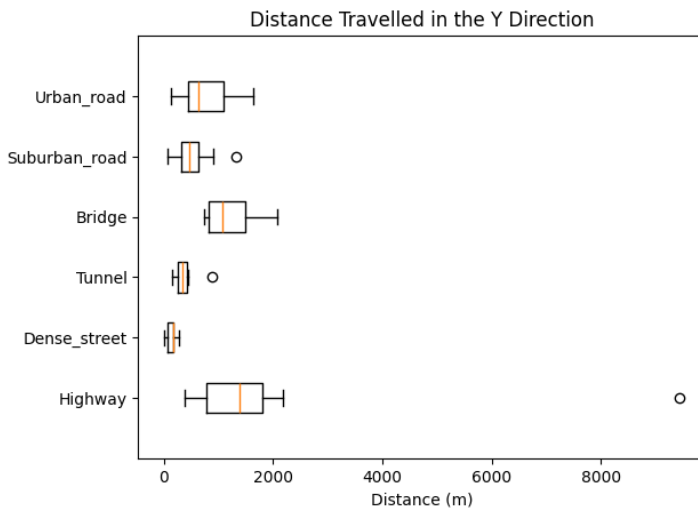
These values were then contextualised by plotting the distance distribution for each scenario (see Figure 5.1), revealing whether the maximum values were typical or should be handled individually by extending the size of the place cell network. The maximum y distance of 9431m is found to be significantly larger than any other distance along both the x and y axis, with most sequences covering approximately 2000m.

The ground truth positions for all routes were plotted (see Appendix A.5) and the existence of distinct paths that warranted distinguishing was analysed (e.g., roads intersecting a map). For a subset of the aforementioned position plots, see Figure 5.2.

Having established the upper limit for the scale of distances that required unique positions to be tracked, a lower limit, which dictates the Place Field scale, is investigated by analysing road geometry, more specifically lane widths. In the United Kingdom, a single lane has a width of up to 3.65m, with a two lane road of up to 7.3m in width [197], [198]. Given these measurements, the width of a Place Field is dictated as 15m; meaning that this is the step size between the central point of adjacent Place Fields. At this scale, it is unlikely that adjacent roads share the same Place Field, while still only requiring a single place cell to map across the width of larger roads, such as motorways and dual carriageways. By dictating this level of granularity, and with prior knowledge of the area covered by the ECMD routes we know that 200 place cells are sufficient to cover all of the routes, excluding the longest sequence where 750 place cells are needed. Having determined the scale at which the place cells operate; the training and grid signals can be calculated.



(a)



(b)

Figure 5.1: The distributions of the distances covered in the ECMD dataset, grouped by scene.

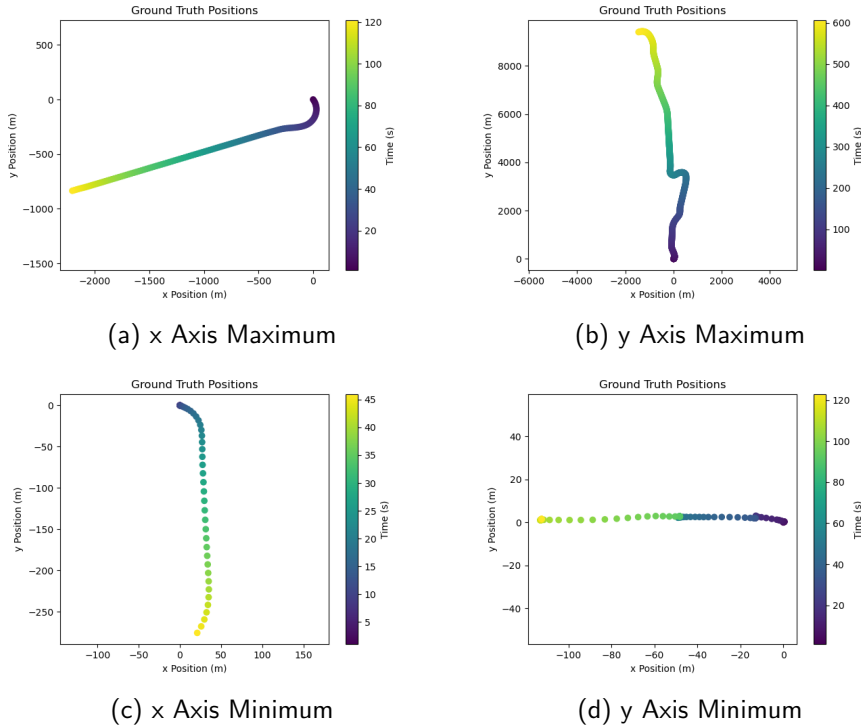


Figure 5.2: The ground truth positions for the ECMD sequences that covered the largest, and smallest amount of distance along each axis.

5.2.2 Training Specification

The place cell network operates over two phases, which dictate its constitution; training, and deployment. During the training phase, the place cells are connected to the Grid networks through plastic Synapses that modulate their weight according to STDP as defined by [199], [200]. After the training phase has been completed, the final synaptic weights are recorded from the plastic Synapses and applied to the network using static Synapses to prevent online learning when the network is deployed; see Figure 5.3. Following deployment, activity is triggered solely through grid activity.

Training the place cell network requires three input sources, an inhibitory signal, a training signal, and grid activity. The inhibitory signal supplies a constant current of 2nA (a synaptic weight of -1 makes this act as -2nA) that is connected globally in a one-to-all fashion, i.e., the inhibitory signal is connected to all the place cells throughout the duration of the training phase. Thus, preventing unwanted spiking that would result in unwanted learning. With the inhibitory signal preventing unwanted spikes, the network requires a complementary excitatory signal to induce the desired spiking behaviour from each place cell; this is the role of the training signal.

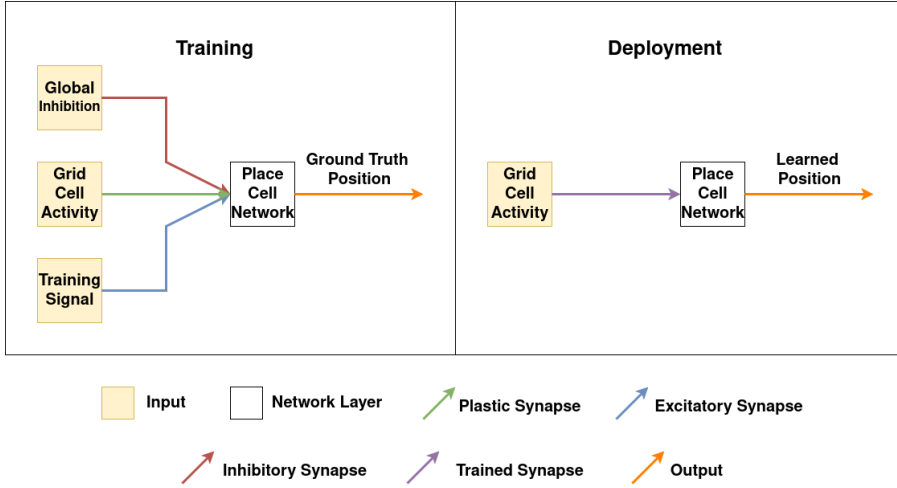


Figure 5.3: An overview of the layers in the Place Cell network and how they are connected in the training and deployment phases. The Grid Cells are connected to the place cell network via all to all STDP Synapses, totalling 140,000 Synapses at the start of training. This is for a network consisting of 200 Place Cells, 100 Grid Position neurons per scale, and 7 scales of Grid. Following training all of these connections are maintained, however many will be 0 or close to 0, and could be pruned for computational efficiency.

5.2.3 Input Signals

To enable the operation of the Place Cell network, artificially induced grid signals are developed. The advantages of this approach are two fold. Firstly, this provides contingency in the case that the Grid network isn't successfully implemented; for this reason it was needed in this project. It also means that the place cells learn the 'correct' grid signals for a Place Field instead of erroneous signals where drift has occurred over the duration of the simulation. This was done by creating a spiking input, with the same spatial structure described in Chapter 4, that is connected to a grid layer, which also uses the hexagonal lattice. Internally the grid layer is completely unconnected and therefore relies entirely on the spiking input layer for its activity. Each spiking input within the spiking input layer connects to the grid cell representing the same position with a weight of 1000 and the neighbouring grid cells with a weight of 600 to create a HOA. These weights were selected to ensure that the neuron corresponding to the Place Field receives the strongest excitation and, therefore, fires maximally, while neighbouring cells receive sufficient excitation to broaden the activity into a HOA through a lower spike frequency, avoiding the grid signal appearing to have multiple centres. When the activity moves within the input layer, it also moves in the grid layer without any residual signals at the previous position; see Figure 5.4.

With the ability to move the grid signals, find where the grid position

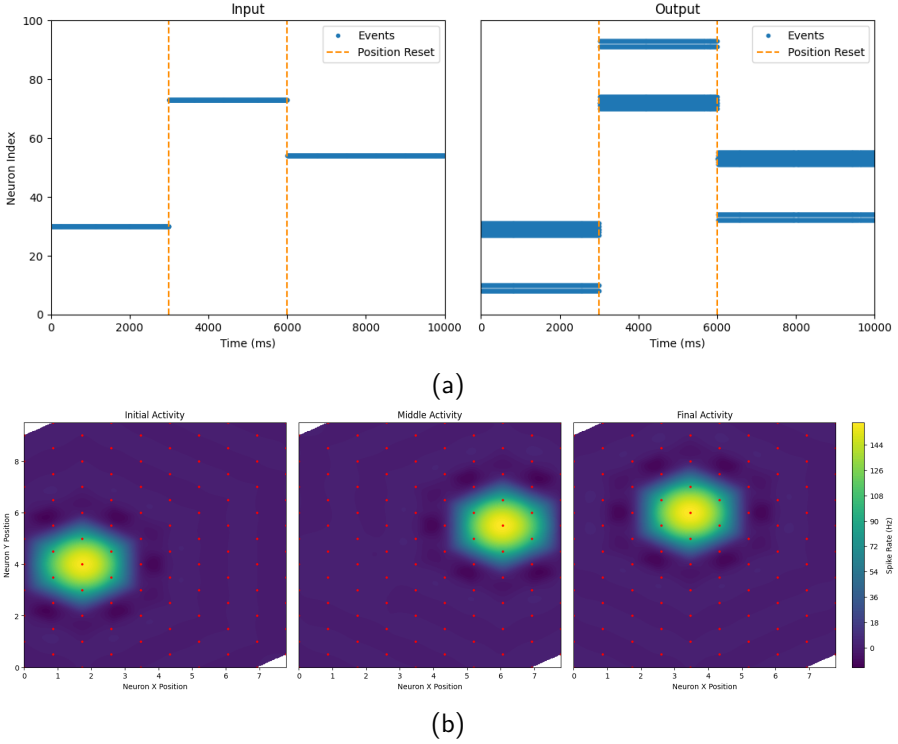


Figure 5.4: The movement of the artificial grid signal as the ground truth position is updated. (a) The ground truth signal is shown on the left and the right shows the artificial grid layer activity. The two areas of activity over each period on the right are a result of how the neurons are indexed and represent a cluster when plotted spatially (see (b)). (b) The network activity plotted spatially from the start to the first ground truth update; between the two ground truth updates; and from the second ground truth update until the end of the simulation, from left to right.

should be active at any given time must be calculated. The following describes the generic transformation from the ground truth coordinate system to the grid coordinate system, determining which grid cell should fire at a given real-world position.

The positions (x_i, y_j) of each grid cell within the Grid network are described by

$$x_i = i\sqrt{\frac{3}{4}} \quad (5.1)$$

$$y_j = \begin{cases} j, & \text{when } j \text{ is odd,} \\ j + 0.5, & \text{when } j \text{ is even.} \end{cases} \quad (5.2)$$

where i and j are the indices of the neuron along the x and y axes, respectively.

A ground truth coordinate (x_{gt}, y_{gt}) is reduced to the grid coordinate system according to

$$x_{gc} = \frac{x_{gt}}{d_s} \bmod \left(I \times \sqrt{\frac{3}{4}} \right) \quad (5.3)$$

$$y_{gc} = \frac{y_{gt}}{d_s} \bmod (J) \quad (5.4)$$

where I and J are the largest values of i and j , respectively, and d_s is the distance represented by one step in the grid coordinate system. The grid cell with the coordinates (x_i, y_j) , which are closest to (x_{gc}, y_{gc}) is where the activity is centred. Using this method, each time a new ground truth position is recorded, the artificial grid signals can also be updated giving a new set of activity.

The analysis performed in Section 5.2.1, and a Place Field size of 15m, dictates the step size between grid positions. Seven grid signals with different step sizes are used, scaling by a factor of approximately 1.5 to 1.9 between sizes. These scaling factors were chosen to balance interpretability with optimal performance based on [201]. The resulting step sizes are,

- 10m,
- 15m,
- 25m,
- 40m,
- 75m,
- 125m,
- 200m.

By having the smallest two grid step sizes less than or equal to the Place Field step size, adjacent Place Fields are distinguished through the change of grid signals. A single network using the same scale as the Place Fields could perform this function but by using multiple step sizes, the onus is not on a single signal to make this distinction; therefore, it is less likely to be 'overpowered' by competing signals making the implementation more robust. Having large scale networks, up to 200m in step size, reduces the chances of repeated signals occurring, especially over larger sequences, seen in Figure 5.1.

Having established the artificial grid signals, it is used with ground truth data from the ECMD dataset, moving the HOA at 1 second intervals, when the ground truth data is updated. This results in activity that moves through the network according to where the vehicle path is integrated to, within the Grid network; see Figure 5.5.

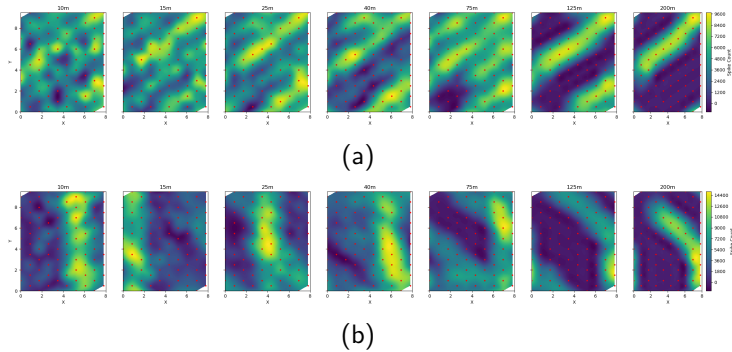


Figure 5.5: The artificial grid activity showing different behaviour based on the vehicle path.

With the input and training signals established and using the granularity of a Place Field determined in Section 5.2.1, specific Place Fields are calculated that cover the coordinates recorded in the ground truth data for ECMD sequence, and each Place Field is assigned a place cell. Each place cell has a corresponding training signal, formed by a step signal, which steps up to 3nA when the vehicle is within the cell's corresponding Place Field and steps down when it is not. Training signals are connected to place cells through excitatory connections using a one-to-one connectivity scheme. The magnitude of the inhibitory and training signals is arbitrary, provided the inhibitory signal prevents unwanted excitation, and the training signal temporarily negates the inhibition and induces activity. For details of the effects of the inhibitory and training signals on a place cell see Figure 5.6.

The time to apply the excitatory training signal is calculated by finding the minimum and maximum values along the x and y axes for each sequence. At 15m intervals along each axis, a Place Field is assigned. For each ground truth position in a sequence the index of the nearest Place Field is found so that all of the visited Place Fields have 1 second intervals where they were visited, and those that weren't visited are dropped. The remaining Place Fields are then assigned a place cell. Each place cell receives the excitatory training signal for the duration of the vehicle occupying it's assigned Place Field.

Although synaptic weight updates are performed using STDP, an unsupervised training mechanism, the use of an external training signal makes this a supervised training process. This is because the training signal acts as a supervisory signal, indicating which place cell should become active for each Place Field, while STDP determines the synaptic weights that associate the active grid pattern with that place cell.

During training, the weights between grid cells and place cells develop according to how selectively each grid cell contributes to a given Place Field. Grid cells that are active for a specific Place Field will repeatedly have concurrent activity with the corresponding place cell, causing these connections to strengthen. In contrast, grid cells that are active across multiple Place Fields will be associated with several place cells during training, preventing a single

large synaptic weight from forming. As a result, trained weights are expected to be strongest when grid activity is linked to a single Place Field and weaker when shared between multiple Place Fields. This is intended as it allows each Place Cell to learn a combination of grid signals, rather than relying on a single grid cell response, although highly selective connections may also risk becoming overfit and ineffective were the number of Place Fields to increase.

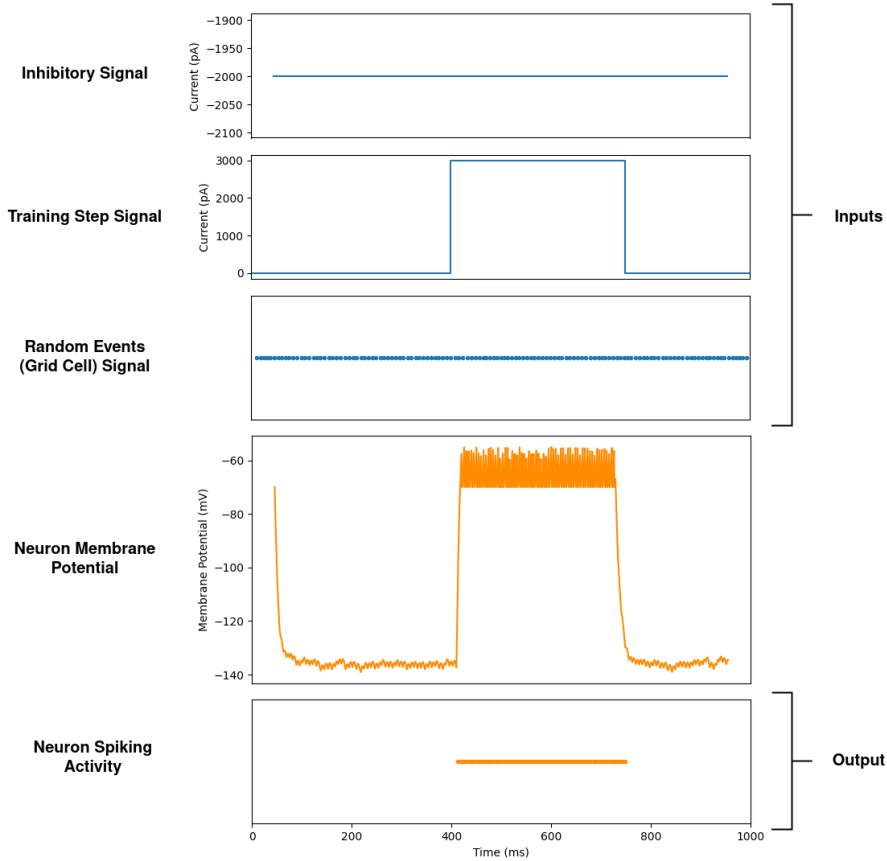


Figure 5.6: The effects of the inhibitory, training, and grid signals, that are used during the training phase, on a place cell.

All grid cells are connected to all place cells through plastic Synapses; it is worth noting that following training, some connections may possess a weight of 0 and therefore could be pruned. Using the rules of weight modulation according to STDP, when the vehicle is in a given Place Field, the grid cells that are simultaneously active have their synaptic weights increased to the corresponding place cell. Each place cell learns the unique combination of grid activity that occurs at it's given Place Field, resulting in this activity triggering the place cell independently of the training signal. At this stage, the training phase ends and the network is ready to be deployed, to track position

in real-world scenarios. An overview of the connectivity previously discussed is shown in Figure 5.7.

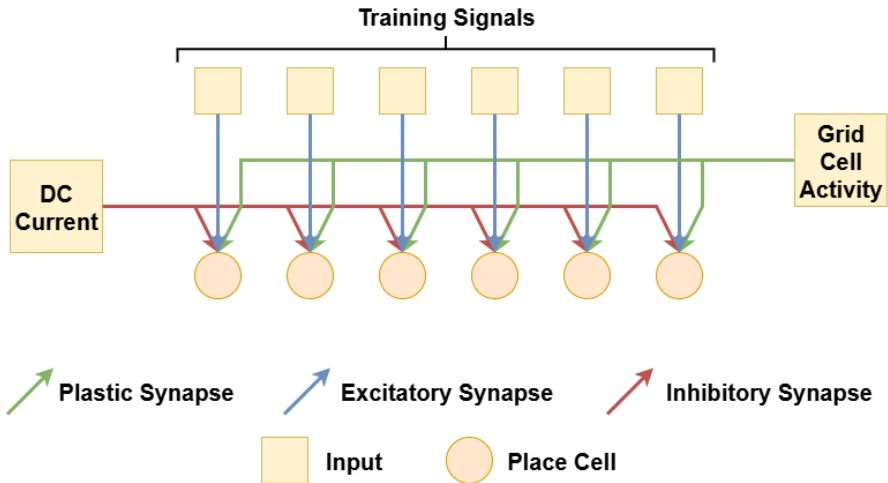


Figure 5.7: Place Cell network connectivity at the neuronal level during training.

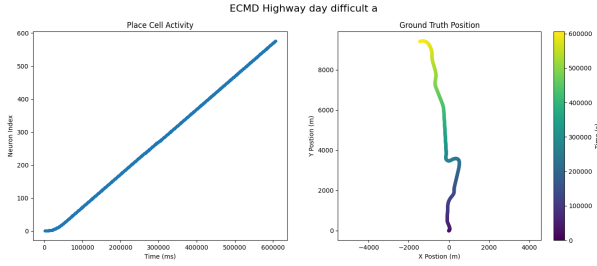
5.3 Results

Training was solely reliant upon the effects of STDP to modulate synaptic weight. Training was carried out running the network, with the training signals, on a sequence, with STDP making small adjustments to the synaptic weights during each test run. The network was then run on the sequence again, without training signals, and with the synaptic weights made static, to test if the place cells exhibited the desired behaviour. If the place cells firing was correct, training ended, otherwise this process was repeated. After the place cells had been trained on each sequences the following results were achieved.

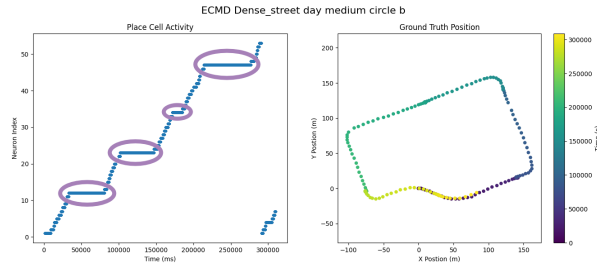
Using the connectivity and signals described in Section 5.2, the Place Cell network was trained and run on the ECMD dataset. Given the limitations of the Grid network that was developed, this was only done using artificial grid signals developed from the ground truth data. This means that the positions corresponding to place cell firing are accurate, but there is still scope to analyse the effectiveness of the training.

The place cells succeeded in learning the patterns exhibited by the grid signals, producing the activity shown in 5.8; for all of the activity plots see Appendix A.5. Due to the way in which the Place Fields were assigned to the place cells, most of the results exhibit a similar behaviour to that shown in Figure 5.8a, where activity progresses continuously through the network. This results from the vehicle travelling steadily along a road without making repeated visits to a Place Field. On shorter journeys, with less consistent speeds, the vehicle can be seen to stop in a specific Place Field before continuing on; this is most common in dense street scenarios and is probably due to traffic lights

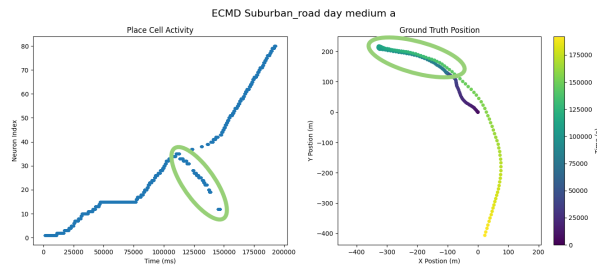
or other similar interruptions to the journey. This is shown in Figure 5.8b. The behaviour in which the Place Fields were revisited is successfully shown by Figures 5.8b, 5.8c, and ; showing the ability to use repeated grid signals to localise to the same position repeatedly. Although the Place Cell network sometimes showed activity returning to a place cell when a Place Field was revisited, this didn't happen on all of the occasions where the vehicle repeatedly travelled along the same section of road. This is seen in Figure 5.8d where it appears that the vehicle travelled along a road in one direction, made a U-turn and returned in the opposite direction. The size of the Place Field was specified to identify these journeys in opposing directions along the same road as being in the same place; however, in this sequence, most activity is assigned to new place cells. Therefore it is thought that the Place Fields aren't adequately sized to capture both sides of a road and need to be made larger to show the intended behaviour.



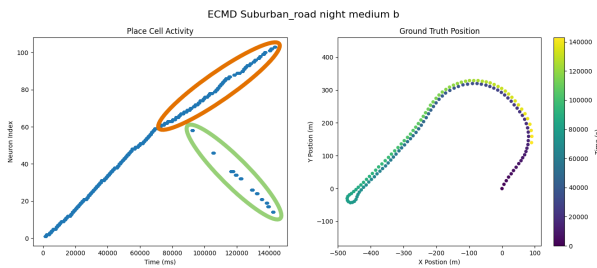
(a)



(b)



(c)



(d)

Figure 5.8: Place cell firing activity and the corresponding ground truth trajectories. (a) The place cell activity when the vehicle exclusively enters new Place Fields. (b) When the vehicle regularly stops along its journey this is also reflected in the place cells. This is circled in purple. (c) When a Place Field is revisited by the vehicle returning along the road in the opposite direction the place cell becomes active again as circled in green, but this behaviour wasn't always shown, see the orange circle in (d).

5.4 Discussion

The place cell network successfully learnt to map from multi-scale grid activity to individual Place Fields to provide a localisation signal. Using artificial grid signals, sourced from ground truth data, allowed STDP-based learning to converge without issues around drift or other sources of error. As a result, the trained network produced accurate representations of the position across the ECMD sequences. Long, non-repeating trajectories, as seen on highways and through tunnels, produced by continuous movement of activity through the Place Cell network, without activity erroneously returning to previous place cells; see Figure 5.8a. Dense urban environments produce choppy but equally accurate activity where the vehicle's progression was hindered; see Figure 5.8b. Finally, when locations were revisited, this was often, but not always detected; see Figure 5.8d. This shows that the scale and quantity of grid signals was sufficient for the place cells to discern between positions, and that the use of STDP was sufficient to learn these signals.

However, there are shortfalls to this implementation. The primary limitation is the lack of authentic grid activity. Given that the place cells are reliant upon the grid activity for localisation, means that the accuracy of this cannot be assessed because the errors originate from the grid signals. Additionally, encoding position using place cells, with a corresponding Place Field is good when used within a greater neuromorphic navigation system, isn't easily interpretable for an absolute position. This is because neither the signal profile nor the neuron index directly corresponds to the position of the vehicle. Instead, each place cell has a Place Field, which must be assigned, and then stored; to realise the position from the Place Cell network, the active cell must be identified and its index must be noted, and using this index the corresponding Place Field can be looked up to finally understand the position of the vehicle. The final limitation is that at the current Place Field step size, when returning along the opposite side of the same road, the network failed to localise to the same Place Field and instead activity moved to new place cells. Larger Place Fields should amend this limitation and should be investigated.

For mapping to neuromorphic hardware, the implementation of the Place Cell network is suitable in terms of neuron count given most ECMD sequences require approximately 200 place cells. However, the all-to-all plastic connectivity between the grid cells and place cells creates a large number of Synapses that connect relatively few neurons. This is particularly important for a hardware implementation compared with simulation due to memory and routing requirements, which are strongly affected by the number of active Synapses. Therefore, while the network is feasible to implement on hardware, it would be advisable to prune unused or weak connections after training, leaving a sparse set of grid-to-place Synapses. This would preserve the effectiveness of the training while minimising the potential issues associated with the Synapse density when mapping to hardware.

In summary, the Place Cell network demonstrates that a bio-inspired localisation mechanism can learn in order to contribute to a larger neuromorphic navigation system; successfully mapping from signals mimicking grid cells

to discrete Place Fields. The use of artificial grid signals prevents analysis of the robustness and error rates of this implementation, but the proof of concept is valid. The limitations identified reveal the need to refine the implementation and highlight the need to tailor the network when adapting it to different environments and applications. The focus of future work should be on integrating authentic Grid networks and scaling the Place Fields for the environment. Addressing these challenges will allow for accurate assessment of the network and allow it to contribute as part of a larger bio-inspired navigation system.

Chapter 6

Visual Place Recognition and Reset Mechanism

6.1 Introduction

In navigation tasks, the accumulation of errors is inevitable in even the most robust systems. Without a mechanism for correction, drift can degrade localisation accuracy and lead to significant deviations from ground truth over time. To address this, navigation systems can periodically reset their estimate of position and direction by recognising previously known locations through visual place recognition.

Visual place recognition can act to close loops within navigation systems. When a system recognises a known location, it can correct the accumulated error by aligning the systems estimated position and direction with stored equivalents. The process of loop closure is essential for ensuring the accuracy of localisation signals over extended periods.

In neuromorphic navigation systems, this reset mechanism can be implemented sending a signal triggered by visual place recognition networks, which run from event-based vision sensors, to Head Direction and Grid Cell networks. A spiking neural network processes the incoming visual event data and attempts to match it to previously learnt visual representations of known places. Upon successful recognition, the system can trigger a reset signal that resets both the Head Direction and Grid Cell networks to the corresponding ground truth states. This integration of visual place recognition with bio-inspired localisation models allows the system to operate entirely using spiking neural networks and offer more robust localisation signals due to the integrated error correction.

Stereo and monocular camera arrangements can be employed in place recognition systems, each offering distinct advantages. Monocular systems are compact and energy-efficient, making them well suited for applications in robotics; however, they inherently lack depth information, which can limit their robustness. Alternately, stereo systems can be used to provide depth cues through disparity, allowing for more precise operation in spatial related

tasks and improved performance in visually similar scenes. As such, the choice between monocular and stereo configurations involves a balance between computational efficiency and spatial complexity, meaning the optimal solution is contextual.

This chapter presents the implementation of a spiking neural network for performing visual place recognition on stereo event camera inputs and its role in the reset mechanism used to correct the internal spatial signals in navigation related networks. The structure of the vision network is introduced followed by how it is trained to interface with the spatial networks. Finally, we discuss the results that are achieved by adding a reset signal to the navigation networks.

6.2 Network Structure

For computational efficiency, the event camera outputs are reduced from 640x480 to 16x12 by combining the outputs of each 40x40 array of pixels to a single event stream, which is input directly into the network. This was done by altering the data at the source, but could equally be done in real time by connecting each pixel array to a neuron within a 16x12 layer of spiking neurons. The network is made up of a 16x16x12 cuboid of neurons with the left event camera feeding into the front and the right event camera feeding into the side. The neurons within the network don't have any internal connectivity. Instead, each pixel array is connected back, through the network, to every neuron along its corresponding row, with the pixel array from the other event camera doing the same, but perpendicular, since it is connecting from the side. This means that every neuron represents a distinct pair of pixel arrays. The pixel arrays are connected to the neurons with a synaptic weight of 300, meaning that the neuron will only fire when both corresponding pixel arrays are active. Therefore, the network pairs activity between the two event cameras along the rows, converting the two 2D firing patterns into 3D firing. This is shown in Figure 6.1.

The structure of the network preserves the spatial position of events for each camera frame. Since the ECMD dataset provides stereo event-camera data, the same physical object may generate events at different horizontal positions in the left and right event streams. This horizontal difference is the disparity between the cameras and provides a depth cue. In this implementation, disparity is not explicitly calculated; instead, it is captured through the relative activity patterns produced by the network from the left and right event-camera inputs. As a result, the visual reset network can learn from the spatial pattern of event activity, where the (x,y) location of active neurons encodes both scene structure and stereo differences between the two cameras.

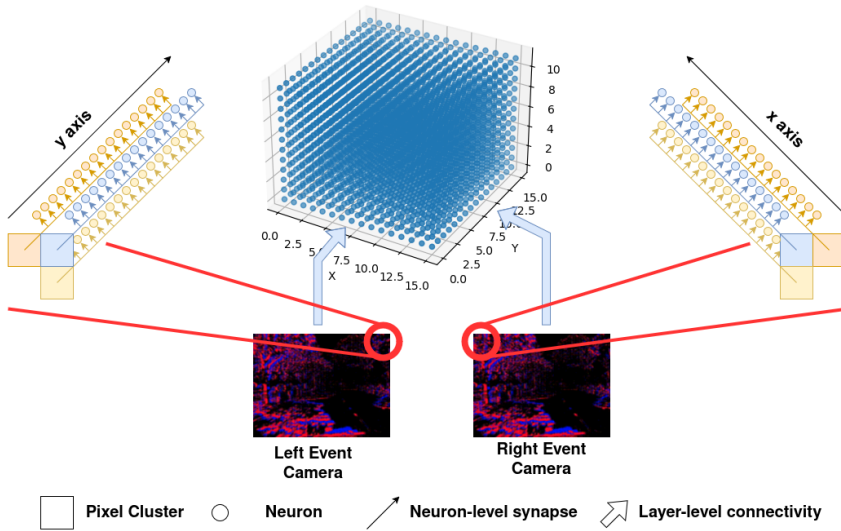


Figure 6.1: The structure and connectivity of the visual processing part of the stereo vision network. Showing the cuboid shape of the network and how the two sets of event camera data are passed into the network perpendicularly. Also shown is how the event cameras connect along an entire row.

The stereo vision network uses neurons as coincidence detectors, where activity indicates that a particular pattern of event-camera input has occurred. Neurons respond to coincident activity between the left and right event cameras, allowing disparity to be captured through differences in the spatial location of activity between the two inputs. This provides an implicit depth cue without explicitly calculating disparity as a separate feature. The visual input is therefore encoded directly from event-camera activity rather than through a separate feature extraction stage. As with the Place Cell network, the relevant output is the activation of neurons in response to coincident input activity, rather than a precise temporal code or a manually extracted visual feature.

Feature extraction was not included because the aim of this chapter was to assess whether a visual reset signal could be produced within the wider event-based localisation system, rather than to introduce additional complexity. Such a network would require separate design, training, optimisation, and evaluation, adding further complexity on top of the Head Direction, Grid, Place Cell, and reset mechanisms already developed in this thesis. Furthermore, given the size of the ECMD dataset, manually identifying reliable landmarks and optimal reset times across all sequences was not feasible. Using a simple network and encoding allowed the reset mechanism to be evaluated while keeping the implementation focused on the wider neuromorphic navigation system.

The visual reset parameters listed in Table 6.1 were selected to test whether event-camera activity could produce a reset signal for the localisation system.

The default `iaf_psc_alpha` neuron parameters were used to avoid neuron-level tuning, with the expectation that training the Synapses would be sufficient. The parameters of the `stdp_synapses` were also left at their default values, except for λ and α , which were manually adjusted during training to produce a reset response without causing immediate saturation. Initially, the λ was large and the α was small to cause synaptic weights to increase, but these were reduced as the network was trained to avoid saturation. The reset times were selected to provide known training points within each sequence, but were only minimally optimised. The focus was to explore the feasibility of the reset mechanism as part of the larger system.

Table 6.1: Key neuron and synapse parameters used in the Visual Reset network.

Parameter	Value
Neuron model	<code>iaf_psc_alpha</code>
Resting potential, E_L	-70 mV
Membrane capacitance, C_m	250 pF
Membrane time constant, τ_m	10 ms
Refractory period, t_{ref}	2 ms
Threshold potential, V_{th}	-55 mV
Reset potential, V_{reset}	-70 mV
Excitatory synaptic time constant, $\tau_{\text{syn,ex}}$	2 ms
Inhibitory synaptic time constant, $\tau_{\text{syn,in}}$	2 ms
Constant input current, I_e	0 pA
Plastic synapse model	<code>stdp_synapse</code>
acrshortstdp step size, λ	Manually adjusted throughout training
acrshortstdp asymmetry parameter, α	Manually adjusted throughout training

6.3 Network Operation

The vision networks connect to the output layers using an all-to-all connectivity, so that every vision neuron is connected to every possible output neuron, through STDP Synapses during training, and static, trained Synapses during deployment. The STDP Synapses are trained using the same signals and methodology described in 5.2.2, and then converted to static Synapses once the desired behaviour has been achieved.

To provide an output reset signal to the localisation networks, the vision network must be able to interface with them, mapping the reset signal to the correct spatial position. This could be done by connecting the vision network directly to the localisation networks, but this would require the learned activity to become dominant to overpower existing activity, without becoming too strong and saturating through space or time. Instead, the vision network creates an output layer for each localisation network that uses the same spatial

structure, where each neuron connects to the corresponding neuron in the spatial network and its neighbour, via strong excitatory connections. This means that the vision network only has to excite a single neuron, which will amplify the reset signal, and pass it to a group of neurons so the new signal is self sustaining. The general connectivity for the training and deployment phases for both vision networks is shown in Figure 6.2, as well as the output layers to interface with the spatial networks that have previously been described in the project. The vision network doesn't connect directly to the Place Cell network because the place cell activity is derived from the grid cell activity. So, by resetting the activity of the grid cells, the Place Cell network also resets to the known position.

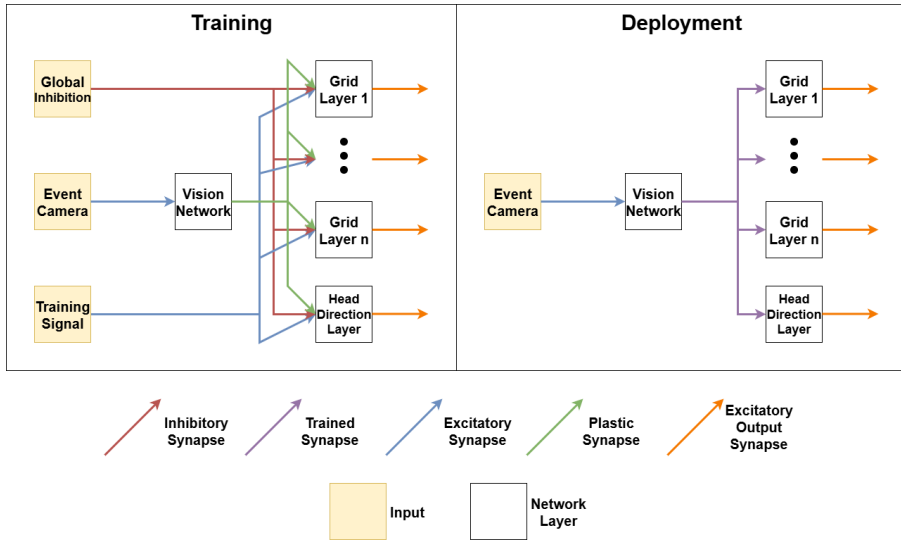
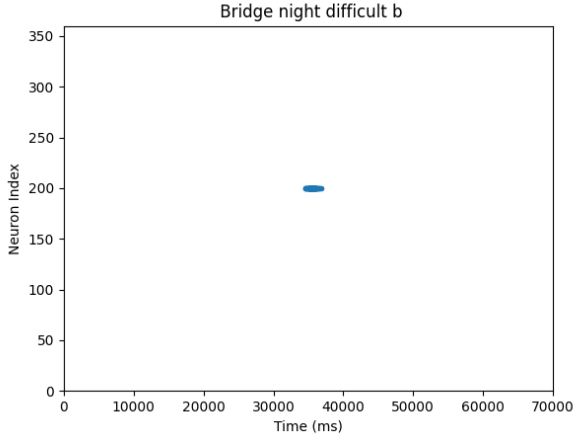


Figure 6.2: An overview of the areas of the vision network and how they are connected in the training and deployment phases.

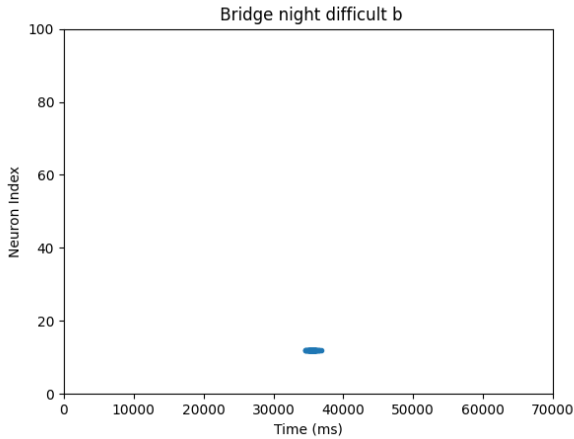
The trained weights in the visual reset network are considered reasonable when they produce a reset response at the intended time in the sequence, without causing persistent activity or saturation. During training, visual neurons that are active at the selected reset time strengthen their connections to the corresponding spatial reset neuron, causing the network to associate that specific event-camera activity with that known direction and position. However, because the reset location is trained using a single traversal of a sequence, the resulting weights may become overfit to the exact visual activity observed at that point. The current ECMD dataset does not provide repeated traversals of the same location under different conditions, so the network cannot learn which visual features remain stable across changes in lighting, traffic, weather, or viewing position. As a result, trained reset weights demonstrate that a visual reset signal can be formed, but cannot be considered robust landmark recognition.

The vision network is trained to provide a reset signal at the midpoint of

each sequence in the ECMD dataset. This halves the maximum continuous period to which the network is exposed to drift and should therefore have the largest impact on accuracy, without the need to individually analyse each sequence to find an optimal reset time. The results of this generic reset signal are discussed in the next section.



(a)



(b)

Figure 6.3: Reset signals from the vision network. (a) The head direction reset signal. (b) The grid cell reset signal with a step size of 10m.

6.4 Results

The network was successfully trained to reset the head direction and grid signals, and theoretically can also reset the place cell signals through resetting

grid signals. Both monocular and stereo systems successfully learnt to associate the event camera data at the midpoint of each sequence using the ground truth position and direction, triggering a reset signal in each of the output layers; see Figure 6.3. Given that the place cell network is run from artificial grid signals, it isn't possible to induce a reset through the Grid Cell network. However, given that movement in the grid signals has been shown to move place cell activity, and that a grid reset signal has been established, it is reasonably assumed that the reset can induce a correction in the place cell network.

Further inspection of the activity in the output layers reveals that the head direction and grid layers provide the same output signal. This is because the network was only trained to recognise a single position, at a single time, so the output layers could have been replaced by a single neuron and provided the same functionality, but wouldn't offer the ability to scale to multiple reset positions or directions. When looking into the plausibility of triggering multiple resets in the network the behaviour for when training was continued beyond the desired reset being learnt was analysed. This revealed that the neuron that is reset is quickly saturated with activity, throughout the simulation time; see Figure 6.4, implying that the combination of the training scheme and vision network doesn't provide a robust solution, and that the attempting to train the same direction or grid position to be associated with different scenes wouldn't be possible and would lead to saturation.

The vision network produced a reset in the head direction network for all of the ECMD sequences (see Appendix A.7), and the new MAE is recorded and compared in Table 6.2. Introducing a reset into the simulation reduced the network error in the majority of sequences, and visual analysis of the sequences for which the error increased reveals that adjusting the reset time for these sequences would have produced improved accuracy; see Figures 6.5c and 6.5d.

The effective use of a reset is seen in Figure 6.5b, where drift has accumulated during the first half of the sequence, which is corrected through the reset mechanism. The direction is then tracked accurately for the remainder of the sequence, reducing the observed MAE by over 13°. In contrast to this Figure 6.5d shows the sequence that already exhibited the largest error without a reset. This is exaggerated through the introduction of a reset, as the reset was triggered slightly before the major offset error occurred, and moved the activity in the direction of the offset error, therefore increasing it. This highlights the importance of the timing of reset signals as ideally, it would be triggered immediately following the offset error. However, this presents the issue of needing to know when such errors may occur which isn't necessarily possible in real-world deployment.

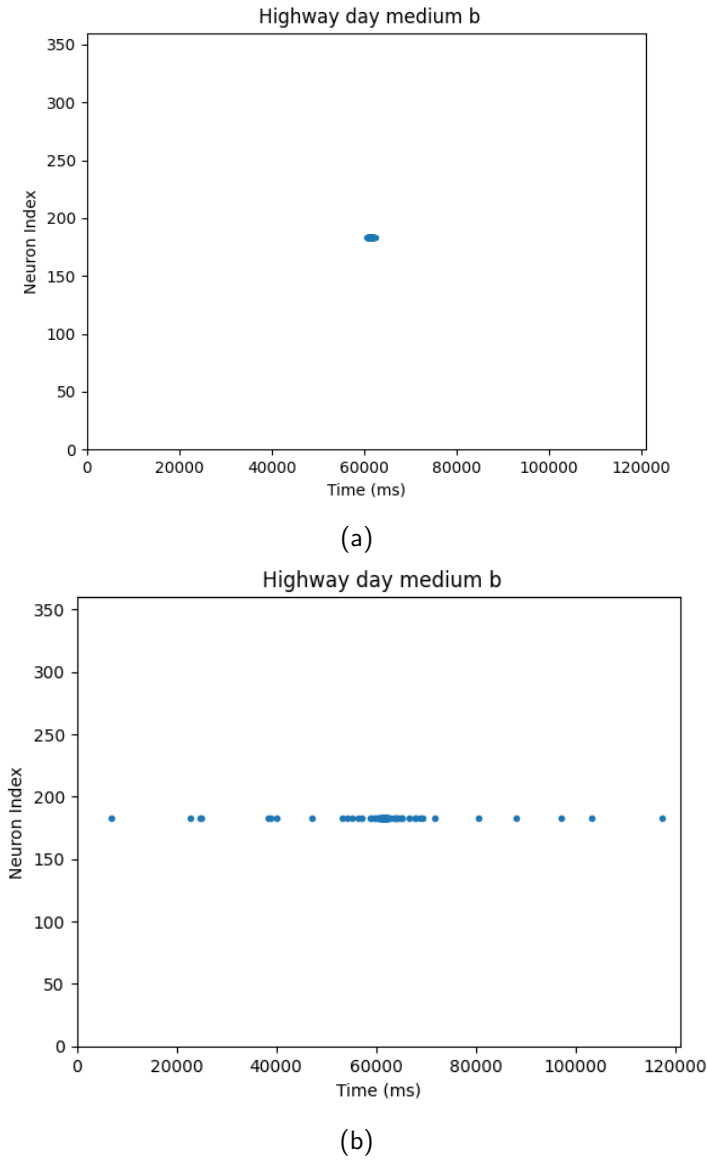


Figure 6.4: It was seen that when training was run beyond the desired reset signal being achieved that the neuron would quickly saturate, no longer providing a useful reset signal.

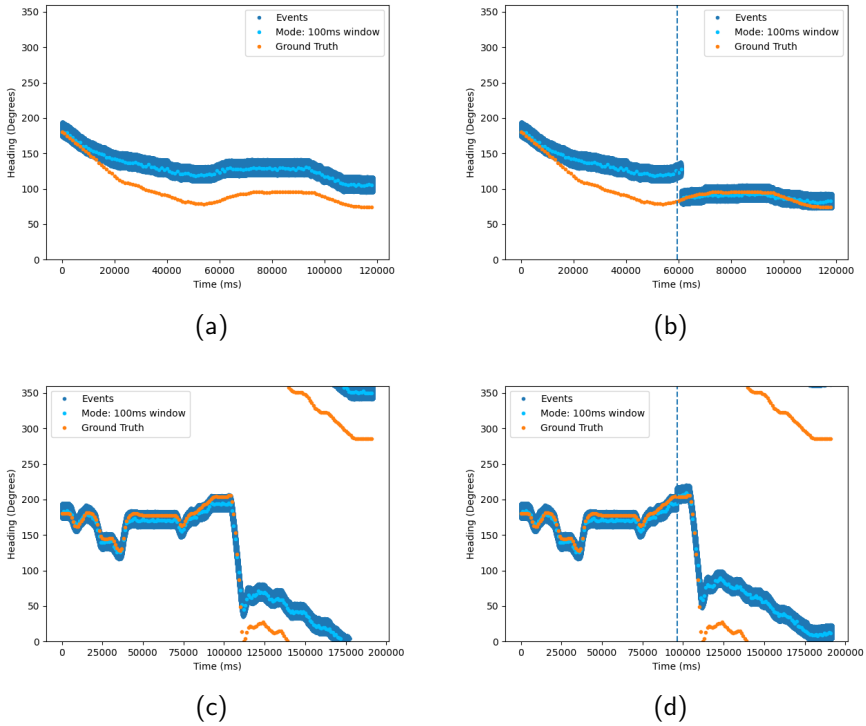


Figure 6.5: The activity in the Head Direction network with and without a reset being induced at the midpoint of the simulation. (a) and (b) show the effective use of a reset to reduce the recorded error. (c) and (d) show where a reset increased the recorded error.

Table 6.2: Errors observed in the Head direction network for the ECMD dataset with and without the reset mechanism. Recorded using the Mean Absolute Error (MAE).

Scenario	Daytime	Difficulty	Sequence	MAE without Reset (Degrees)	MAE with Reset (Degrees)
Bridge	day	medium	a	5.4	2.1
		medium	b	4.8	3.5
	night	difficult	a	1.9	1.7
		difficult	b	3.1	2.5
Dense street	day	easy	e	3.7	2.1
		medium	a	3.2	5.9

		medium	b	3.9	4.9
		medium	c	2.5	2.9
		medium	circle b	7.5	9.0
	night	easy	a	7.3	5.4
Highway	day	difficult	a	14.3	12.0
		difficult	b	29.6	16.0
		easy	a	4.6	6.5
		easy	b	7.8	6.5
		medium	a	7.2	5.1
		medium	b	8.7	6.6
		medium	c	11.0	6.1
	night	difficult	c	2.3	2.9
		difficult	d	7.4	6.7
		medium	b	20.0	8.7
		medium	c	16.4	7.0
		medium	d	2.8	5.3
Suburban road	day	easy	a	9.2	5.0
		easy	b	13.8	5.7
		easy	c	2.2	2.1
		easy	d	9.9	16.4
		easy	e	13.1	8.8
		easy	f	8.6	4.3
		medium	a	69.3	89.3
		medium	b	7.7	6.3
		medium	c	3.6	5.5
		medium	d	2.8	2.0
	night	difficult	a	4.2	4.2
		medium	a	5.8	11.3
		medium	b	27.5	27.6
Tunnel	day	difficult	a	5.2	3.3
		difficult	b	6.3	5.7
		easy	a	8.9	7.0
		easy	b	6.2	6.0
		easy	c	6.8	4.1
		easy	d	2.4	4.4
		medium	a	4.9	4.4
Urban road	day	easy	a	5.8	11.6
		easy	b	10.3	3.8
		easy	c	12.6	9.7
		easy	d	5.4	4.7
		medium	a	11.6	7.8
	night	difficult	a	9.7	5.6
		difficult	b	2.4	2.3

		difficult	c	8.6	8.1
		difficult	rainy a	1.4	2.2
		difficult	rainy b	10.2	4.6
		easy	a	9.0	5.8
		easy	b	3.3	3.4
		easy	rainy a	11.9	6.6
		medium	a	11.0	6.7
		medium	b	3.1	3.1
		medium	c	5.5	5.3
		medium	d	13.5	7.3
		medium	rainy a	2.5	2.5
		medium	rainy b	10.8	4.4

Given that the distribution of head direction errors have previously been identified to be associated with the scenario, this was again investigated; see Figure 6.6. The distribution of errors typically remained consistent with or without a reset, but the median error and interquartile range were both notably reduced across scenarios, showing that not only was the performance better but that the signal could more be relied upon more consistently.

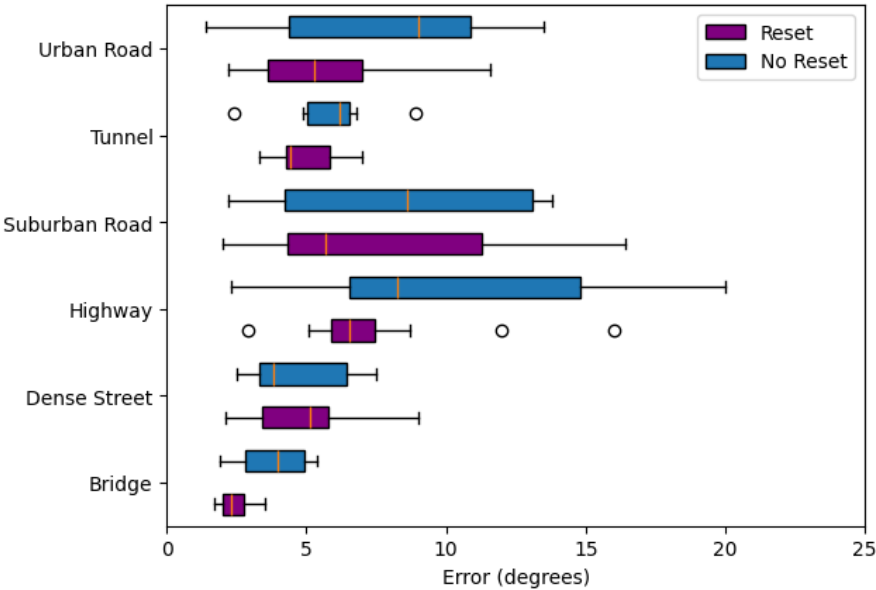


Figure 6.6: The error distributions by scenario with and without the visual reset signal.

6.5 Discussion

The integration of a visual place recognition network with the spatial navigation system provided a bio-inspired solution for loop closure in the head direction and grid cell networks. The use of event-based stereo vision allowed the network to detect scenes through temporal resolution and depth cues, implied through disparity, providing a more robust recognition of known scenes. The reset mechanism demonstrated a reduction of the MAE in many of the ECMD sequences, and the tightened the error distributions across most driving scenarios. These results evidence the feasibility of a neuromorphic navigation system by integrating loop closure, and achieving it in a fully event-based architecture. It also highlights the effectiveness of visual place recognition as a trigger for loop closure, effectively resetting to the known position.

Despite the results showing promise for the concept, several limitations of the current implementation were shown. The reset mechanism was trained to recognise a single scene per sequence, the timing of which was not optimised. Some resets were triggered at suboptimal times and increased the observed errors; however, this could be overcome by curating the reset time for each sequence. Ideally, the reset time would be identified dynamically from the visual and velocity inputs. A more robust approach would be to trigger resets only when the visual network detects a reliable landmark, or when the motion input indicates a period likely to induce accumulated error. Such approaches would reduce the dependence on manually selected reset times and would allow the reset mechanism to operate more autonomously, either by improving landmark-based loop closure or by correcting errors after periods of unreliable path integration. However, this would introduce additional complexity through a feature detection or landmark selection stage. More concerning is that the vision network is sensitive to visual inputs, quickly saturating after the first reset signal is learnt, implying that learning multiple reset times would result in saturation of the network. It is believed that increasing the network size and not grouping the event camera pixels could resolve this, but at great computational cost.

The absence of a functioning grid cell network meant that the grid reset signal couldn't be applied to evidence the ability of the vision network to reset position. Despite this, it is still fairly assumed that the implementation of a grid reset signal is successful because the desired reset signal was achieved, and resetting the grid signal works to move place cell activity. So the only missing link is connecting the reset signal to a functioning grid network, and the use of appropriate synaptic weights and connectivity would ensure that the reset signal can override existing activity in a grid network.

For mapping to neuromorphic hardware, the visual reset network must also consider compatibility with the event-camera input, although this should not be problematic as both share the same operating principles using sparse and asynchronous events. The current implementation of the network itself uses dense connectivity between the visual network and the reset output layers, which may place a significant synaptic load on the neuromorphic hardware. This could be made more efficient by replacing dense, untrained mappings

with sparse, trained Synapses where training can be performed offline, and in simulation. This would reduce the memory and routing requirements imposed by Synapses while preserving the low-latency benefit of producing an event-based reset signal for the localisation networks.

In summary, the visual place recognition and reset mechanism presented here demonstrates the feasibility of a fully event-based approach to loop closure for neuromorphic navigation, through the integration of visual cues. Although effective in correcting accumulated errors under controlled conditions, its current limitations highlight the challenge faced when developing a system that operates across the complex, variable environments encountered in real-world driving, thereby showing the need for greater complexity going forward.

Chapter 7

Conclusion

This thesis has explored a bio-inspired, fully event-based system for localisation in driving scenarios, demonstrating the feasibility of combining SNNs with event-cameras to create an end-to-end neuromorphic system. Taking inspiration from mammalian navigation systems, the work has shown how head direction, grid, and place cell networks can be adapted for use with inertial cues, recorded using an IMU, and visual cues in real-world driving scenarios. The inclusion of a stereo visual place recognition network provided a loop closure mechanism that enabled the correction of accumulated errors, showing the capabilities of entirely neuromorphic solutions in localisation tasks, and how these can leverage the low latency and low power benefits of this computational approach.

The results show that the head direction network, implemented as an event-based ring attractor network, was able to track direction with single digit accuracy in a range of driving scenarios, highlighting the potential of SNNs for real-world real-time navigation. However, it also revealed challenges related to the implementation, such as drift caused by small mismatches between the angular velocity and the shift rate of the network. In addition, the network suffered from occasional large offset errors that persisted in the absence of external correction.

The Grid network, demonstrated basic stability and a shift mechanism, but did not achieve a reliable mapping from the speed input to shift rate. This emphasises the need for a theory based network design when increasing the complexity of the network, in this case through added dimensionality. Although this implementation of the Grid network wasn't successful it is still believed that a network operating using the same principles can be constructed to achieve this task.

The Place Cell network was trained from artificial grid cell signals, operating at multiple scales, using STDP, and successfully mapped the artificial grid cell signals to distinct Place Fields, thus providing a single spiking representation of position. The primary limitation of Place Cell network was that when the vehicle travelled repeatedly along the same road, activity in the network didn't always return to the same place cell instead moving to a new one. This shows that the Place Fields should be scaled to wholly represent the breadth of the

road to avoid assigning multiple Place Fields, and therefore place cells, to a single road section.

The visual place recognition network demonstrated the capacity to trigger effective reset signals that reduced the MAE of the Head Direction network in many sequences, validating the concept of event-based visual place recognition for loop closure. However, the sensitivity of the vision network to saturation after learning a single reset location highlights the challenge of scaling to multiple reset without scaling the network, incurring significant computational cost.

Overall, this thesis provides a proof of concept for an end-to-end neuromorphic localisation system, from the sensor to the network's output. It shows that bio-inspired models can be adapted for engineering applications and highlights the potential of Neuromorphic Computing for low latency, low power solutions to autonomous navigation, with the specific contributions highlighted in the following section. In addition to the capabilities that are demonstrated, it also serves to show many avenues for future work, as detailed in the final Section 7.2.

7.1 Contributions

This work makes several novel contributions to the field of neuromorphic navigation by demonstrating how bio-inspired neural network architectures can be adapted and integrated into an event-driven localisation system for real-world driving scenarios. While previous studies have explored individual bio-inspired components such as head direction or grid cells in simplified environments, this thesis advances the field by attempting to create a combined system that applies SNNs based on head direction, grid, and place cells to localisation using real driving data.

The first key contribution lies in extending ring attractor networks for head direction tracking to operate using spike timing, and applying this to real-world driving scenarios. This is furthered by the addition of loop closure through a visual place recognition network, with the whole system proven to operate on real-world driving data. Secondly, while the method of tracking position through grid and place cells remains incomplete, the training of place cells to learn grid signals for continuously tracking position in driving scenarios is also a novel contribution.

Across the contribution chapters, the proposed networks feature many characteristics that are suitable for implementation on neuromorphic hardware. The use of LIF neurons, sparse spiking activity, and event-based inputs allows the system to utilise neuromorphic principles at all stages, thus translating well to hardware. The Head Direction and Grid networks use predefined attractor dynamics to represent direction and position, while the Place Cell and visual reset networks demonstrate how learning can be used to provide localisation and loop closure within an event-based network. It is in the latter, trained networks that suitability for neuromorphic hardware cannot be judged from neuron count alone, as dense Synapse connectivity can place significant

demands on the hardware platform. Therefore, mapping to hardware would be best performed after offline training to allow pruning to reduce the number of Synapses. In addition to being mapped onto hardware, there are many directions in which this work can be taken further to continue to contribute to the field of Neuromorphic Computing; these are discussed in the next section.

7.2 Future Work

This thesis begins to demonstrate the feasibility of combining event-based processing through bio-inspired representations of space for applied localisation tasks. However, there is significant scope to go beyond what has been detailed in this thesis, with many possible directions. These are completing a planar navigation system, progressing the system onto neuromorphic hardware, extending networks to operate across three dimensions, and enabling autonomy through landmark detection and learning, and dynamic resetting.

The obvious scope for future work is the development of a fully functioning Grid network capable of scalable velocity integration thus creating a complete planar localisation network. The main challenge lies in achieving a shift response that responds proportionally to the speed input and can be adjusted to facilitate multiple scales of grid step size. Since the Grid network is central to the proposed navigation system, this would allow velocity integration to provide a continuous internal position signal. This position could then be interpreted by the Place Cell network, with loop closure triggered through the visual reset mechanism, resulting in a full planar VIO system. Further work should also consider how Place Field size is selected for different environments. The results in this thesis show that improperly sized Place Fields can cause repeated traversal of the same road to activate different place cells. A future system should adapt the scale of Place Fields to the environment, balancing the precision of localisation against the need to recognise the same physical location when travelling along different lanes or in different directions.

The second direction is mapping the complete system onto neuromorphic hardware, assessing efficiency and performance in an applied real world system. The networks implemented in this thesis use simple LIF neurons and well established training mechanisms, which means that mapping from simulation to hardware should be straightforward. The neuron counts for each subnetwork and the total network are also well within the capacities of current neuromorphic platforms: the Head Direction network uses 1,080 neurons, the Grid network uses 700 neurons per scale, the stereo vision network uses 3,072 neurons, and the Place Cell network uses approximately 200 place cells for most ECMD sequences. These values are small relative to the platforms discussed in Section 1.2.1, which have been developed to support a large number of spiking neurons and Synapses [20], [22], [94], [98], [99]. The more significant constraint when transitioning to hardware is therefore likely to be the density of synaptic connectivity, for example, the Place Cell network using all-to-all connectivity to the many scales of Grid networks. To combat this, pruning strategies should be implemented before transitioning to hardware; removing trained Synapses with

negligible or zero weight, therefore replacing fully connected mappings with sparse connectivity between the subnetworks. This would improve hardware friendliness while retaining the event-based and low-latency advantages of the system.

A hardware implementation would also allow the performance of the system to be evaluated in terms of latency, power consumption, and the sparsity of events. In simulation, all neuronal and Synapse states are calculated at each time step regardless of activity. In neuromorphic hardware, power consumption is more closely related to the membrane potential and volume of synaptic events. This is beneficial for the proposed system given the sparse encoding used in the Head Direction, Grid, Place Cell, and Vision networks, where only small regions of each network denote activity at a given time. Therefore, future work should assess the actual and predicted power consumption along with the localisation capabilities of the network.

Another direction for future work lies in extending the dimensionality of the system, using the same principles the localisation signals can be extended from tracking planar movements into three-dimensional space. This would be critical for applications where travel does not occur exclusively along a surface, such as for UAVs and submersibles. In these instances, vertical movement is possible and rotation is not constrained around the vertical axis. Extending the system to accommodate an additional dimension would require the Head Direction network to represent orientation beyond a single angle, moving from a ring attractor to a spherical attractor. Similarly, the Grid network would need to integrate velocity across all three dimensions rather than along a plane, thus requiring a 4D equivalent of its current toroidal attractor network. This would increase both the dimensionality and the connectivity of the spatial networks, exacerbating the synaptic density problem discussed previously, if such a network were mapped onto hardware. However, such an extension would significantly increase the applicability of the system, enabling neuromorphic localisation for aerial, underwater, and other unconstrained environments.

The introduction of autonomy through landmark detection and dynamic resetting would enhance the applicability of the system to the real world. In this thesis, the visual place recognition network is trained to produce reset signals at predetermined points in each sequence. An ideal system would autonomously detect reliable landmarks, associate them with the current Head Direction and Place Cell representations, and store them as future reset locations. When the same landmark is later detected, the system could dynamically reset to the stored spatial representations, removing the need to predefine reset locations and allowing the system to build a map of known landmarks during exploration.

Developing such a reset mechanism would require datasets that are specifically designed with repetitions of the same routes and landmark encounters. Existing event-camera datasets provide recordings across different environments and sensor configurations [181], [182], [185], [186]; however, a particularly useful dataset for this system would contain many recordings of the same route under different conditions. These should include different times of day, seasons, weather conditions, traffic densities, and directions of travel. Such a dataset would allow future work to implement visual networks that can learn stable

landmarks that remain recognisable despite variation in the environment. This would directly support the development of autonomous landmark learning for robust loop closure and dynamic reset mechanisms.

Together, these directions would move the system from a proof of concept to a robust exploratory framework for navigation based on neuromorphic hardware and design principles. Completing planar velocity integration would provide a functional baseline system that is capable in many day-to-day scenarios. Mapping networks to hardware establishes the performance, efficiency, and practicality of this approach when applied in the real world. Extending the system to the third dimension expands the range of scenarios in which the system can be deployed. Finally, enabling autonomy through landmark learning allows the system to operate in unknown environments, performing an exploratory function.

Bibliography

- [1] M. S. Gazzaniga, R. B. Ivry and G. R. Mangun, *Cognitive Neuroscience: The Biology of the Mind*. 2019, vol. 5 (cit. on pp. 2, 3).
- [2] B. Cheng and D. M. Titterington, “Neural networks: A review from a statistical perspective,” *Statistical Science*, vol. 9, no. 1, pp. 2–30, 1994, ISSN: 08834237. Accessed: 10th Dec. 2024. [Online]. Available: <http://www.jstor.org/stable/2246275> (cit. on p. 2).
- [3] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire and T. McGinnity, “A review of learning in biologically plausible spiking neural networks,” *Neural Networks*, vol. 122, pp. 253–272, Feb. 2020, ISSN: 08936080. DOI: 10.1016/j.neunet.2019.09.036. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608019303181> (cit. on p. 2).
- [4] T. S. Mosqueiro and R. Huerta, “Computational models to understand decision making and pattern recognition in the insect brain,” *Current Opinion in Insect Science*, vol. 6, pp. 80–85, Dec. 2014, ISSN: 22145745. DOI: 10.1016/j.cois.2014.10.005. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2214574514001072> (cit. on p. 2).
- [5] L. Deng et al., “Rethinking the performance comparison between snns and anns,” *Neural Networks*, vol. 121, pp. 294–307, Jan. 2020, ISSN: 08936080. DOI: 10.1016/j.neunet.2019.09.005. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608019302667> (cit. on p. 2).
- [6] W. Maass, “Networks of spiking neurons: The third generation of neural network models,” *Neural Networks*, vol. 10, pp. 1659–1671, 9 Dec. 1997, ISSN: 08936080. DOI: 10.1016/S0893-6080(97)00011-7. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608097000117> (cit. on p. 2).
- [7] D. Auge, J. Hille, E. Mueller and A. Knoll, “A survey of encoding techniques for signal processing in spiking neural networks,” *Neural Processing Letters*, vol. 53, pp. 4693–4710, 6 Dec. 2021, ISSN: 1370-4621. DOI: 10.1007/s11063-021-10562-2. [Online]. Available: <https://link.springer.com/10.1007/s11063-021-10562-2> (cit. on pp. 2, 3, 19).

- [8] P. U. Diehl, G. Zarrella, A. Cassidy, B. U. Pedroni and E. Neftci, "Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware," in *2016 IEEE International Conference on Rebooting Computing (ICRC)*, 2016, pp. 1–8. DOI: 10.1109/ICRC.2016.7738691 (cit. on p. 2).
- [9] G. Datta and P. A. Beerel, "Can deep neural networks be converted to ultra low-latency spiking neural networks?" In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022, pp. 718–723. DOI: 10.23919/DATE54114.2022.9774704 (cit. on p. 2).
- [10] F Ponulak and A Kasinski, "Introduction to spiking neural networks: Information processing, learning and applications," *Acta Neurobiol Exp*, vol. 71, pp. 409–433, 2011 (cit. on p. 2).
- [11] A. Delorme and S. Thorpe, "Face identification using one spike per neuron: Resistance to image degradations," *Neural Networks*, vol. 14, pp. 795–803, 6–7 Jul. 2001, ISSN: 08936080. DOI: 10.1016/S0893-6080(01)00049-1. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608001000491> (cit. on p. 2).
- [12] M. Bouvier et al., "Spiking neural networks hardware implementations and challenges," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 15, pp. 1–35, 2 Apr. 2019, ISSN: 1550-4832. DOI: 10.1145/3304103. [Online]. Available: <https://dl.acm.org/doi/10.1145/3304103> (cit. on p. 2).
- [13] L. Paninski, J. Pillow and J. Lewi, "Statistical models for neural encoding, decoding, and optimal stimulus design," in *Computational Neuroscience: Theoretical Insights into Brain Function*, ser. Progress in Brain Research, P. Cisek, T. Drew and J. F. Kalaska, Eds., vol. 165, Elsevier, 2007, pp. 493–507. DOI: [https://doi.org/10.1016/S0079-6123\(06\)65031-0](https://doi.org/10.1016/S0079-6123(06)65031-0). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0079612306650310> (cit. on p. 2).
- [14] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier and Y. Tian, "Deep residual learning in spiking neural networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang and J. W. Vaughan, Eds., vol. 34, Curran Associates, Inc., 2021, pp. 21 056–21 069. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/file/afe434653a898da20044041262b3ac74-Paper.pdf (cit. on p. 2).
- [15] J. K. Eshraghian et al., "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, vol. 111, no. 9, pp. 1016–1054, 2023. DOI: 10.1109/JPROC.2023.3308088 (cit. on p. 2).
- [16] C. B. Delahunt, J. A. Riffell and J. N. Kutz, "Biological mechanisms for learning: A computational model of olfactory learning in the manduca sexta moth, with applications to neural nets," *Frontiers in Computational Neuroscience*, vol. 12, Dec. 2018, ISSN: 1662-5188. DOI: 10.3389/fncom.2018.00102. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fncom.2018.00102/full> (cit. on p. 2).

- [17] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, pp. 1629–1636, 10 1990, ISSN: 00189219. DOI: 10.1109/5.58356. [Online]. Available: <http://ieeexplore.ieee.org/document/58356/> (cit. on pp. 2, 24).
- [18] S. Furber, "Large-scale neuromorphic computing systems," *Journal of Neural Engineering*, vol. 13, p. 051001, 5 Oct. 2016, ISSN: 1741-2560. DOI: 10.1088/1741-2560/13/5/051001. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1741-2560/13/5/051001> (cit. on p. 2).
- [19] B. V. Benjamin et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proceedings of the IEEE*, vol. 102, pp. 699–716, 5 May 2014, ISSN: 0018-9219. DOI: 10.1109/JPROC.2014.2313565. [Online]. Available: <http://ieeexplore.ieee.org/document/6805187/> (cit. on p. 2).
- [20] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, pp. 82–99, 1 Jan. 2018, ISSN: 0272-1732. DOI: 10.1109/MM.2018.112130359. [Online]. Available: <https://ieeexplore.ieee.org/document/8259423/> (cit. on pp. 2, 24, 25, 117).
- [21] F. Akopyan et al., "Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, pp. 1537–1557, 10 Oct. 2015, ISSN: 02780070. DOI: 10.1109/TCAD.2015.2474396 (cit. on pp. 2, 25).
- [22] S. B. Furber, F. Galluppi, S. Temple and L. A. Plana, "The spinnaker project," *Proceedings of the IEEE*, vol. 102, pp. 652–665, 5 May 2014, ISSN: 0018-9219. DOI: 10.1109/JPROC.2014.2304638. [Online]. Available: <https://ieeexplore.ieee.org/document/6750072/> (cit. on pp. 2, 24, 25, 117).
- [23] F. Ponulak, *Resume-new supervised learning method for spiking neural networks*, 2005. [Online]. Available: <http://d1.cie.put.poznan.pl/fp/> (cit. on pp. 2, 20).
- [24] Y. Hao, X. Huang, M. Dong and B. Xu, "A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule," *Neural Networks*, vol. 121, pp. 387–395, Jan. 2020, ISSN: 08936080. DOI: 10.1016/j.neunet.2019.09.007. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608019302680> (cit. on p. 2).
- [25] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers in Neuroscience*, vol. Volume 11 - 2017, 2017, ISSN: 1662-453X. DOI: 10.3389/fnins.2017.00682. [Online]. Available: <https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2017.00682> (cit. on p. 2).

- [26] R. Huerta and T. Nowotny, “Fast and robust learning by reinforcement signals: Explorations in the insect brain,” *Neural Computation*, vol. 21, pp. 2123–2151, 8 Aug. 2009, ISSN: 0899-7667. DOI: 10.1162/neco.2009.03-08-733. [Online]. Available: <https://direct.mit.edu/neco/article/21/8/2123-2151/7461> (cit. on p. 2).
- [27] F. Faghihi, A. A. Moustafa, R. Heinrich and F. Wörgötter, “A computational model of conditioning inspired by drosophila olfactory system,” *Neural Networks*, vol. 87, pp. 96–108, Mar. 2017, ISSN: 08936080. DOI: 10.1016/j.neunet.2016.11.002. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608016301666> (cit. on p. 2).
- [28] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt and H. van Hasselt, “Multi-task deep reinforcement learning with popart,” 2019, pp. 3796–3803. [Online]. Available: www.github.com/deepmind/scalable (cit. on p. 2).
- [29] K. Yamazaki, V.-K. Vo-Ho, D. Bulsara and N. Le, “Spiking neural networks and their applications: A review,” *Brain Sciences*, vol. 12, no. 7, p. 863, 2022. DOI: 10.3390/brainsci12070863 (cit. on p. 2).
- [30] E. R. Kandel, J. Koester, S. Mack and S. Siegelbaum, *Principles of neural science*, vol. 4, ISBN: 9781259642241 (cit. on p. 3).
- [31] T. Trappenberg, *Fundamentals of computational neuroscience* (cit. on p. 3).
- [32] E. Pereda, R. Q. Quiroga and J. Bhattacharya, “Nonlinear multivariate analysis of neurophysiological signals,” *Progress in Neurobiology*, vol. 77, pp. 1–37, 1-2 Sep. 2005, ISSN: 03010082. DOI: 10.1016/j.pneurobio.2005.10.003. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S030100820500119X> (cit. on p. 3).
- [33] A. I. Gulyás, M. Megías, Z. Emri and T. F. Freund, “Total number and ratio of excitatory and inhibitory synapses converging onto single interneurons of different types in the cal area of the rat hippocampus,” *Journal of Neuroscience*, vol. 19, pp. 10 082–10 097, 22 Nov. 1999, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.19-22-10082.1999. [Online]. Available: <https://www.jneurosci.org/content/19/22/10082>[https://www.jneurosci.org/content/19/22/10082abstract](https://www.jneurosci.org/content/19/22/10082.abstract) (cit. on p. 5).
- [34] M. Megías, Z. Emri, T. F. Freund and A. I. Gulyás, “Total number and distribution of inhibitory and excitatory synapses on hippocampal cal pyramidal cells,” *Neuroscience*, vol. 102, pp. 527–540, 3 Feb. 2001, ISSN: 0306-4522. DOI: 10.1016/S0306-4522(00)00496-6 (cit. on p. 5).
- [35] A. L. Hodgkin and A. F. Huxley, “A quantitative description of membrane current and its application to conduction and excitation in nerve,” *The Journal of Physiology*, vol. 117, pp. 500–544, 4 Aug. 1952, ISSN: 0022-3751. DOI: 10.1113/jphysiol.1952.sp004764. [Online]. Available: <https://physoc.onlinelibrary.wiley.com/doi/10.1113/jphysiol.1952.sp004764> (cit. on pp. 7, 9).

- [36] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, vol. 5, pp. 115–133, 4 Dec. 1943, ISSN: 0007-4985. DOI: 10.1007/BF02478259. [Online]. Available: <http://link.springer.com/10.1007/BF02478259> (cit. on p. 7).
- [37] L. Lapique, "Recherches quantitatives sur l'excitation electrique des nerfs taitee comme une polarization," *Journal de physiologie et de pathologie générale*, 1907 (cit. on pp. 7, 12).
- [38] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological Review*, vol. 65, pp. 386–408, 6 1958, ISSN: 1939-1471. DOI: 10.1037/h0042519. [Online]. Available: <https://doi.apa.org/doi/10.1037/h0042519> (cit. on p. 7).
- [39] M. A. Nielsen, *Neural networks and deep learning*, misc, 2018. [Online]. Available: <http://neuralnetworksanddeeplearning.com/> (cit. on p. 8).
- [40] W. Gerstner and W. M. Kistler, *Spiking Neuron Models Single Neurons, Populations, Plasticity*. 2008, vol. 4 (cit. on pp. 8, 12).
- [41] E. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, pp. 1063–1070, 5 Sep. 2004, ISSN: 1045-9227. DOI: 10.1109/TNN.2004.832719. [Online]. Available: <http://ieeexplore.ieee.org/document/1333071/> (cit. on p. 8).
- [42] N. Brunel and M. C. W. van Rossum, "Quantitative investigations of electrical nerve excitation treated as polarization," *Biological Cybernetics*, vol. 97, pp. 341–349, 5-6 Dec. 2007, ISSN: 0340-1200. DOI: 10.1007/s00422-007-0189-6. [Online]. Available: <http://link.springer.com/10.1007/s00422-007-0189-6> (cit. on p. 12).
- [43] N. Brunel and M. C. W. van Rossum, "Lapicque's 1907 paper: From frogs to integrate-and-fire," *Biological Cybernetics*, vol. 97, pp. 337–339, 5-6 Dec. 2007, ISSN: 0340-1200. DOI: 10.1007/s00422-007-0190-0. [Online]. Available: <http://link.springer.com/10.1007/s00422-007-0190-0> (cit. on p. 12).
- [44] A. Gilra and W. Gerstner, "Predicting non-linear dynamics by stable local learning in a recurrent spiking neural network," *eLife*, vol. 6, Nov. 2017, ISSN: 2050-084X. DOI: 10.7554/eLife.28295. [Online]. Available: <https://elifesciences.org/articles/28295> (cit. on p. 15).
- [45] M. Kaiser, "A tutorial in connectome analysis: Topological and spatial features of brain networks," *NeuroImage*, vol. 57, pp. 892–907, 3 Aug. 2011, ISSN: 10538119. DOI: 10.1016/j.neuroimage.2011.05.025. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S1053811911005131> (cit. on p. 15).
- [46] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *Neural and Evolutionary Computing*, Nov. 2015. [Online]. Available: <http://arxiv.org/abs/1511.08458> (cit. on p. 16).

- [47] A. Khan, A. Sohail, U. Zahoor and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review*, vol. 53, pp. 5455–5516, 8 Dec. 2020, ISSN: 0269-2821. DOI: 10.1007/s10462-020-09825-6. [Online]. Available: <https://link.springer.com/10.1007/s10462-020-09825-6> (cit. on p. 16).
- [48] S. Schaefer, D. Gehrig and D. Scaramuzza, "Aegnn: Asynchronous event-based graph neural networks," *IEEE*, Jun. 2022, pp. 12 361–12 371, ISBN: 978-1-6654-6946-3. DOI: 10.1109/CVPR52688.2022.01205. [Online]. Available: <https://ieeexplore.ieee.org/document/9878467/> (cit. on p. 16).
- [49] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe and T. Masquelier, "StdP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, Mar. 2018, ISSN: 08936080. DOI: 10.1016/j.neunet.2017.12.005. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608017302903> (cit. on p. 16).
- [50] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe and T. Masquelier, "Bio-inspired digit recognition using reward-modulated spike-timing-dependent plasticity in deep convolutional networks," *Pattern Recognition*, vol. 94, pp. 87–95, Oct. 2019, ISSN: 00313203. DOI: 10.1016/j.patcog.2019.05.015. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0031320319301906> (cit. on p. 16).
- [51] G. Bellec, D. Salaj, A. Subramoney, R. Legenstein and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking neurons," *32nd Conference on Neural Information Processing Systems*, Mar. 2018. [Online]. Available: <http://arxiv.org/abs/1803.09574> (cit. on p. 16).
- [52] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, pp. 1550–1560, 10 1990, ISSN: 15582256. DOI: 10.1109/5.58337 (cit. on pp. 16, 20).
- [53] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, pp. 602–610, 5-6 Jul. 2005, ISSN: 0893-6080. DOI: 10.1016/J.NEUNET.2005.06.042 (cit. on p. 16).
- [54] A. Wendt, L. Paulun and N. Kasabov, "A retinotopic spiking neural network system for accurate recognition of moving objects using neucube and dynamic vision sensors," *Frontiers in Computational Neuroscience*, vol. 12, 2018. DOI: 10.3389/fncom.2018.00042. [Online]. Available: www.frontiersin.org (cit. on pp. 16, 39, 41).
- [55] R. M. Grieves and K. J. Jeffery, "The representation of space in the brain," *Behavioural Processes*, vol. 135, pp. 113–131, Feb. 2017, ISSN: 03766357. DOI: 10.1016/j.beproc.2016.12.012. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0376635716302480> (cit. on pp. 16, 34, 36, 37, 40).

- [56] N. K. Kasabov, "Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data," *Neural Networks*, vol. 52, pp. 62–76, Apr. 2014, ISSN: 08936080. DOI: 10.1016/j.neunet.2014.01.006. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608014000070> (cit. on p. 17).
- [57] G. Tanaka et al., "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, Jul. 2019, ISSN: 08936080. DOI: 10.1016/j.neunet.2019.03.005. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608019300784> (cit. on p. 17).
- [58] A. Ghani, T. M. McGinnity, L. P. Maguire and J. Harkin, "Neuro-inspired speech recognition with recurrent spiking neurons," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5163 LNCS, pp. 513–522, PART 1 2008, ISSN: 1611-3349. DOI: 10.1007/978-3-540-87536-9_53. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-87536-9_53 (cit. on p. 17).
- [59] R. Gütiğ, "To spike, or when to spike?" *Current Opinion in Neurobiology*, vol. 25, pp. 134–139, Apr. 2014, ISSN: 09594388. DOI: 10.1016/j.conb.2014.01.004. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0959438814000129> (cit. on p. 17).
- [60] W. Guo, M. E. Fouda, A. M. Eltawil and K. N. Salama, "Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems," *Frontiers in Neuroscience*, vol. 15, Mar. 2021, ISSN: 1662-453X. DOI: 10.3389/fnins.2021.638474. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fnins.2021.638474/full> (cit. on p. 18).
- [61] T. Gollisch and M. Meister, "Rapid neural coding in the retina with relative spike latencies," *Science*, vol. 319, pp. 1108–1111, 5866 Feb. 2008, ISSN: 0036-8075. DOI: 10.1126/science.1149639. [Online]. Available: <https://www.science.org/doi/10.1126/science.1149639> (cit. on p. 18).
- [62] S. Park, S. Kim, H. Choe and S. Yoon, "Fast and efficient information transmission with burst spikes in deep spiking neural networks," *ACM*, Jun. 2019, pp. 1–6, ISBN: 9781450367257. DOI: 10.1145/3316781.3317822. [Online]. Available: <https://dl.acm.org/doi/10.1145/3316781.3317822> (cit. on p. 18).
- [63] G. Portelli et al., "Rank order coding: A retinal information decoding strategy revealed by large-scale multielectrode array retinal recordings," *eneuro*, vol. 3, ENEURO.0134–15.2016, 3 May 2016, ISSN: 2373-2822. DOI: 10.1523/ENEURO.0134–15.2016. [Online]. Available: <https://www.eneuro.org/lookup/doi/10.1523/ENEURO.0134–15.2016> (cit. on p. 18).

- [64] A. Delorme, L. Perrinet and S. J. Thorpe, "Networks of integrate-and-fire neurons using rank order coding b: Spike timing dependent plasticity and emergence of orientation selectivity," *Neurocomputing*, vol. 38-40, pp. 539–545, Jun. 2001, ISSN: 09252312. DOI: 10.1016/S0925-2312(01)00403-9. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0925231201004039> (cit. on p. 18).
- [65] L. L. Colgin, "Mechanisms and functions of theta rhythms," *Annual Review of Neuroscience*, vol. 36, pp. 295–312, Volume 36, 2013 Jul. 2013, ISSN: 0147006X. DOI: 10.1146/ANNUREV-NEURO-062012-170330/CITE/REFWORKS. [Online]. Available: <https://www.annualreviews.org/content/journals/10.1146/annurev-neuro-062012-170330> (cit. on p. 18).
- [66] M. A. Montemurro, M. J. Rasch, Y. Murayama, N. K. Logothetis and S. Panzeri, "Phase-of-firing coding of natural visual stimuli in primary visual cortex," *Current Biology*, vol. 18, pp. 375–380, 5 Mar. 2008, ISSN: 09609822. DOI: 10.1016/j.cub.2008.02.023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0960982208001681> (cit. on p. 18).
- [67] T. O. West et al., "Propagation of beta/gamma rhythms in the cortico-basal ganglia circuits of the parkinsonian rat," *Journal of Neurophysiology*, vol. 119, pp. 1608–1628, 5 May 2018, ISSN: 0022-3077. DOI: 10.1152/jn.00629.2017. [Online]. Available: <https://www.physiology.org/doi/10.1152/jn.00629.2017> (cit. on p. 18).
- [68] A. Safa, I. Ocket, A. Bourdoux, H. Sahli, F. Catthoor and G. Gielen, "Continuously learning to detect people on the fly: A bio-inspired visual system for drones," Feb. 2022. [Online]. Available: <http://arxiv.org/abs/2202.08023> (cit. on p. 20).
- [69] S. M. Bohte, J. N. Kok and H. La Poutr , "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1–4, 17–37, 2002. DOI: 10.1016/s0925-2312(01)00658-0 (cit. on p. 20).
- [70] S. K. Esser, R. Appuswamy, P. A. Merolla, J. V. Arthur and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing," Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/10a5ab2db37feedfdeaab192ead4ac0e-Paper.pdf (cit. on p. 20).
- [71] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in Neuroscience*, vol. 14, Feb. 2020, ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00119. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2020.00119/full> (cit. on p. 20).

- [72] E. O. Neftci, C. Augustine, S. Paul and G. Detorakis, “Event-driven random back-propagation: Enabling neuromorphic deep learning machines,” *Frontiers in Neuroscience*, vol. 11, Jun. 2017, ISSN: 1662-453X. DOI: 10.3389/fnins.2017.00324. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fnins.2017.00324/full> (cit. on p. 20).
- [73] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman and G. Hinton, “Backpropagation and the brain,” *Nature Reviews Neuroscience* 2020 21:6, vol. 21, pp. 335–346, 6 Apr. 2020, ISSN: 1471-0048. DOI: 10.1038/s41583-020-0277-3. [Online]. Available: <https://www.nature.com/articles/s41583-020-0277-3> (cit. on p. 20).
- [74] D. O. Hebb, *The Organization of Behavior A Neurophysiological Theory*. 1949, ISBN: 978-0805843002 (cit. on pp. 20, 21).
- [75] H. Barlow, “Unsupervised learning,” *Neural Computation*, vol. 1, pp. 295–311, 3 Sep. 1989, ISSN: 0899-7667. DOI: 10.1162/NECO.1989.1.3.295. [Online]. Available: <https://dx.doi.org/10.1162/neco.1989.1.3.295> (cit. on p. 21).
- [76] Z. Ghahramani, “Unsupervised learning,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3176, pp. 72–112, 2004, ISSN: 1611-3349. DOI: 10.1007/978-3-540-28650-9_5. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-28650-9_5 (cit. on p. 21).
- [77] N. Caporale and Y. Dan, “Spike timing-dependent plasticity: A hebbian learning rule,” *Annual Review of Neuroscience*, vol. 31, no. 1, 25–46, 2008. DOI: 10.1146/annurev.neuro.31.060407.125639 (cit. on p. 21).
- [78] B. Kosko, “Bidirectional associative memories: Unsupervised hebbian learning to bidirectional backpropagation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 103–115, 2021. DOI: 10.1109/TSMC.2020.3043249 (cit. on p. 21).
- [79] P. D. Roberts and C. C. Bell, “Spike timing dependent synaptic plasticity in biological systems,” *Biological Cybernetics*, vol. 87, pp. 392–403, 5-6 Dec. 2002, ISSN: 03401200. DOI: 10.1007/s00422-002-0361-y. [Online]. Available: <http://link.springer.com/10.1007/s00422-002-0361-y> (cit. on p. 21).
- [80] N. Rathi, P. Panda and K. Roy, “StdP-based pruning of connections and weight quantization in spiking neural networks for energy-efficient recognition,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, pp. 668–677, 4 Apr. 2019, ISSN: 0278-0070. DOI: 10.1109/TCAD.2018.2819366. [Online]. Available: <https://ieeexplore.ieee.org/document/8325325/> (cit. on p. 21).
- [81] N. A. Rahman and N. Yusoff, “Modulated spike-time dependent plasticity (stdp)-based learning for spiking neural network (snn): A review,” *Neurocomputing*, vol. 618, p. 129170, 2025. DOI: 10.1016/j.neucom.2024.129170 (cit. on p. 21).

- [82] A. M. Dacks, J. A. Riffell, J. P. Martin, S. L. Gage and A. J. Nighorn, "Olfactory modulation by dopamine in the context of aversive learning," *Journal of Neurophysiology*, vol. 108, pp. 539–550, 2 Jul. 2012, ISSN: 0022-3077. DOI: 10.1152/jn.00159.2012. [Online]. Available: <https://www.physiology.org/doi/10.1152/jn.00159.2012> (cit. on p. 22).
- [83] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Computation*, vol. 19, pp. 1468–1502, 6 Jun. 2007, ISSN: 0899-7667. DOI: 10.1162/neco.2007.19.6.1468. [Online]. Available: <https://direct.mit.edu/neco/article/19/6/1468-1502/7190> (cit. on p. 22).
- [84] R. Legenstein, D. Pecevski and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Computational Biology*, vol. 4, no. 10, 2008. DOI: 10.1371/journal.pcbi.1000180 (cit. on p. 22).
- [85] Z. Huang et al., "Towards high-performance spiking transformers from ann to snn conversion," in *Proceedings of the 32nd ACM International Conference on Multimedia*, ser. MM '24, New York, NY, USA: Association for Computing Machinery, 2024, 10688–10697, ISBN: 9798400706868. DOI: 10.1145/3664647.3680620. [Online]. Available: <https://doi.org/10.1145/3664647.3680620> (cit. on p. 22).
- [86] B. Rueckauer and S.-C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," vol. 2018-May, IEEE, Apr. 2018, pp. 1–5, ISBN: 978-1-5386-4881-0. DOI: 10.1109/ISCAS.2018.8351295. [Online]. Available: <https://ieeexplore.ieee.org/document/8351295/> (cit. on p. 22).
- [87] T. Bu, W. Fang, J. Ding, P. DAI, Z. Yu and T. Huang, "Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks," in *International Conference on Learning Representations*, 2022 (cit. on p. 23).
- [88] A. Taherkhani, A. Belatreche, Y. Li and L. P. Maguire, "DL-resume: A delay learning-based remote supervised method for spiking neurons," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, pp. 3137–3149, 12 Dec. 2015, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2015.2404938. [Online]. Available: <http://ieeexplore.ieee.org/document/7063227/> (cit. on p. 24).
- [89] A. Taherkhani, A. Belatreche, Y. Li and L. P. Maguire, "A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, pp. 5394–5407, 11 Nov. 2018, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2018.2797801. [Online]. Available: <https://ieeexplore.ieee.org/document/8305661/> (cit. on p. 24).

- [90] S. Davidson and S. B. Furber, "Comparison of artificial and spiking neural networks on digital hardware," *Frontiers in Neuroscience*, vol. 15, Apr. 2021, ISSN: 1662-453X. DOI: 10.3389/fnins.2021.651141. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fnins.2021.651141/full> (cit. on p. 24).
- [91] X. Zou, S. Xu, X. Chen, L. Yan and Y. Han, "Breaking the von neu-mann bottleneck: Architecture-level processing-in-memory technology," *Science China Information Sciences*, vol. 64, pp. 1–10, 6 Jun. 2021, ISSN: 18691919. DOI: 10.1007/S11432-020-3227-1/METRICS. [Online]. Available: <https://link.springer.com/article/10.1007/s11432-020-3227-1> (cit. on p. 24).
- [92] J. Hasler and B. Marr, "Finding a roadmap to achieve large neur-omorphiic hardware systems," *Frontiers in Neuroscience*, vol. 7, 7 SEP 2013, ISSN: 1662-453X. DOI: 10.3389/fnins.2013.00118. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fnins.2013.00118/abstract> (cit. on p. 24).
- [93] C. Mead, "How we created neuromorphic engineering," *Nature Elec-tronics*, vol. 3, pp. 434–435, 7 Jul. 2020, ISSN: 2520-1131. DOI: 10.1038/s41928-020-0448-2. [Online]. Available: <https://www.nature.com/articles/s41928-020-0448-2> (cit. on p. 24).
- [94] P. A. Merolla et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, pp. 668–673, 6197 Aug. 2014, ISSN: 0036-8075. DOI: 10.1126/science.1254642. [Online]. Available: <https://www.science.org/doi/10.1126/science.1254642> (cit. on pp. 24, 25, 117).
- [95] M. K. Kim, Y. Park, I. J. Kim and J. S. Lee, "Emerging materials for neuromorphic devices and systems," *iScience*, vol. 23, p. 101 846, 12 Dec. 2020, ISSN: 2589-0042. DOI: 10.1016/J.ISCI.2020.101846 (cit. on p. 24).
- [96] A. Vanarse, A. Osseiran and A. Rassau, "A review of current neur-omorphiic approaches for vision, auditory, and olfactory sensors," *Fron-tiers in Neuroscience*, vol. 10, Mar. 2016, ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00115. [Online]. Available: <http://journal.frontiersin.org/Article/10.3389/fnins.2016.00115/abstract> (cit. on pp. 24, 25).
- [97] J. Kaiser, H. Mostafa and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (decolle)," *Frontiers in Neuroscience*, vol. 14, May 2020, ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00424. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2020.00424/full> (cit. on p. 24).
- [98] C. Pehle et al., "The brainscales-2 accelerated neuromorphic system with hybrid plasticity," *Frontiers in Neuroscience*, vol. 16, p. 795 876, Feb. 2022, ISSN: 1662453X. DOI: 10.3389/FNINS.2022.795876/BIBTEX. [Online]. Available: www.frontiersin.org (cit. on pp. 24, 117).

- [99] G. Orchard et al., “Efficient neuromorphic signal processing with loihi 2,” in *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, 2021, pp. 254–259. DOI: 10.1109/SiPS52927.2021.00053 (cit. on pp. 25, 117).
- [100] C. Mayr, S. Hoppner and S. Furber, “Spinnaker 2: A 10 million core processor system for brain simulation and machine learning,” *Concurrent Systems Engineering Series*, 2019. DOI: 10.3233/978-1-61499-949-2-277 (cit. on p. 25).
- [101] M.-H. Tayarani-Najaran and M. Schmuker, “Event-based sensing and signal processing in the visual, auditory, and olfactory domain: A review,” *Frontiers in Neural Circuits*, vol. 15, May 2021, ISSN: 1662-5110. DOI: 10.3389/fncir.2021.610446. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fncir.2021.610446/full> (cit. on p. 25).
- [102] G. Gallego et al., “Event-based vision: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 154–180, 1 Jan. 2022, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2020.3008413. [Online]. Available: <https://ieeexplore.ieee.org/document/9138762/> (cit. on pp. 26, 27).
- [103] P. Lichtsteiner, C. Posch and T. Delbruck, “A 128x128 120 db 15 us latency asynchronous temporal contrast vision sensor,” *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 566–576, 2 2008, ISSN: 0018-9200. DOI: 10.1109/JSSC.2007.914337. [Online]. Available: <http://ieeexplore.ieee.org/document/4444573/> (cit. on p. 26).
- [104] T. pandas development team, *Pandas-dev/pandas: Pandas*, version latest, Feb. 2020. DOI: 10.5281/zenodo.3509134. [Online]. Available: <https://doi.org/10.5281/zenodo.3509134> (cit. on p. 29).
- [105] Wes McKinney, “Data Structures for Statistical Computing in Python,” in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a (cit. on p. 29).
- [106] M.-O. Gewaltig and M. Diesmann, “Nest (neural simulation tool),” *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007 (cit. on pp. 29, 55).
- [107] J. Taube, R. Muller and J. Ranck, “Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis,” *The Journal of Neuroscience*, vol. 10, pp. 420–435, 2 Feb. 1990, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.10-02-00420.1990. [Online]. Available: <https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.10-02-00420.1990> (cit. on pp. 33–35).
- [108] T. Hafting, M. Fyhn, S. Molden, M. B. Moser and E. I. Moser, “Microstructure of a spatial map in the entorhinal cortex,” *Nature* 2005 436:7052, vol. 436, pp. 801–806, 7052 Jun. 2005, ISSN: 1476-4687. DOI: 10.1038/nature03721. [Online]. Available: <https://www.nature.com/articles/nature03721> (cit. on pp. 33, 35).

- [109] J. O'Keefe and J. Dostrovsky, "The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat," *Brain Research*, vol. 34, pp. 171–175, 1 Nov. 1971, ISSN: 00068993. DOI: 10.1016/0006-8993(71)90358-1. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0006899371903581> (cit. on pp. 33, 36).
- [110] P. E. Sharp, H. T. Blair and J. Cho, "The anatomical and computational basis of the rat head-direction cell signal," *Trends in Neurosciences*, vol. 24, no. 5, pp. 289–294, 2001, ISSN: 0166-2236. DOI: 10.1016/S0166-2236(00)01797-5. [Online]. Available: [https://doi.org/10.1016/S0166-2236\(00\)01797-5](https://doi.org/10.1016/S0166-2236(00)01797-5) (cit. on p. 34).
- [111] J. S. Taube, "Head direction cells recorded in the anterior thalamic nuclei of freely moving rats," *Journal of Neuroscience*, vol. 15, pp. 70–86, 1 Jan. 1995, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.15-01-00070.1995. [Online]. Available: <https://www.jneurosci.org/content/15/1/70https://www.jneurosci.org/content/15/1/70.abstract> (cit. on p. 34).
- [112] J. S. Taube, "The head direction signal: Origins and sensory-motor integration," *Annual Review of Neuroscience*, vol. 30, pp. 181–207, Volume 30, 2007 Jul. 2007, ISSN: 0147006X. DOI: 10.1146/ANNUREV.NEURO.29.051605.112854/1. [Online]. Available: <https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.29.051605.112854> (cit. on p. 34).
- [113] R. M. Yoder and J. S. Taube, "Head direction cell activity in mice: Robust directional signal depends on intact otolith organs," *Journal of Neuroscience*, vol. 29, pp. 1061–1076, 4 Jan. 2009, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.1679-08.2009. [Online]. Available: <https://www.jneurosci.org/content/29/4/1061https://www.jneurosci.org/content/29/4/1061.abstract> (cit. on pp. 34, 35).
- [114] R. M. Yoder and J. S. Taube, "The vestibular contribution to the head direction signal and navigation," *Frontiers in Integrative Neuroscience*, vol. 8, p. 71638, APR Apr. 2014, ISSN: 16625145. DOI: 10.3389/FNINT.2014.00032/BIBTEX (cit. on p. 34).
- [115] G. M. Muir et al., "Disruption of the head direction cell signal after occlusion of the semicircular canals in the freely moving chinchilla," *Journal of Neuroscience*, vol. 29, pp. 14521–14533, 46 Nov. 2009, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.3450-09.2009. [Online]. Available: <https://www.jneurosci.org/content/29/46/14521https://www.jneurosci.org/content/29/46/14521.abstract> (cit. on pp. 34–36).
- [116] R. G. Robertson, E. T. Rolls, P. Georges-François and S. Panzeri, "Head direction cells in the primate pre-subiculum," *Hippocampus*, vol. 9, pp. 206–219, 3 1999. DOI: [https://doi.org/10.1002/\(SICI\)1098-1063\(1999\)9:3<206::AID-HIP02>3.0.CO;2-H](https://doi.org/10.1002/(SICI)1098-1063(1999)9:3<206::AID-HIP02>3.0.CO;2-H). [Online]. Available: [https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%](https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%281098-1063(1999)9:3%3C206::AID-HIP02%3E3.0.CO;2-H)

- 291098-1063%281999%299%3A3%3C206%3A%3AAID-HIP02%3E3.0.CO%3B2-H (cit. on pp. 34, 35).
- [117] E. Ben-Yishay, K. Krivoruchko, S. Ron, N. Ulanovsky, D. Derdikman and Y. Gutfreund, "Directional tuning in the hippocampal formation of birds," *Current Biology*, vol. 31, 2592-2602.e4, 12 Jun. 2021, ISSN: 0960-9822. DOI: 10.1016/J.CUB.2021.04.029 (cit. on pp. 34, 35).
 - [118] L. M. Giocomo, T. Stensola, T. Bonnevie, T. V. Cauter, M. B. Moser and E. I. Moser, "Topography of head direction cells in medial entorhinal cortex," *Current Biology*, vol. 24, pp. 252-262, 3 Feb. 2014, ISSN: 0960-9822. DOI: 10.1016/J.CUB.2013.12.002 (cit. on p. 34).
 - [119] A. Rubin, M. M. Yartsev and N. Ulanovsky, "Encoding of head direction by hippocampal place cells in bats," *Journal of Neuroscience*, vol. 34, pp. 1067-1080, 3 Jan. 2014, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.5393-12.2014. [Online]. Available: <https://www.jneurosci.org/content/34/3/1067><https://www.jneurosci.org/content/34/3/1067.abstract> (cit. on p. 35).
 - [120] E. Vinepinsky, L. Cohen, S. Perchik, O. Ben-Shahar, O. Donchin and R. Segev, "Representation of edges, head direction, and swimming kinematics in the brain of freely-navigating fish," *Scientific Reports* 2020 10:1, vol. 10, pp. 1-16, 1 Sep. 2020, ISSN: 2045-2322. DOI: 10.1038/s41598-020-71217-1. [Online]. Available: <https://www.nature.com/articles/s41598-020-71217-1> (cit. on p. 35).
 - [121] J. D. Seelig and V. Jayaraman, "Neural dynamics for landmark orientation and angular path integration," *Nature*, vol. 521, pp. 186-191, 7551 May 2015, ISSN: 0028-0836. DOI: 10.1038/nature14446. [Online]. Available: <https://www.nature.com/articles/nature14446> (cit. on p. 35).
 - [122] A. Jauffret, N. Cuperlier and P. Gaussier, "From grid cells and visual place cells to multimodal place cell: A new robotic architecture," *Frontiers in Neurorobotics*, vol. 9, 2015. DOI: 10.3389/fnbot.2015.00001 (cit. on p. 35).
 - [123] M. Fyhn, S. Molden, M. P. Witter, E. I. Moser and M. B. Moser, "Spatial representation in the entorhinal cortex," *Science*, vol. 305, pp. 1258-1264, 5688 Aug. 2004, ISSN: 00368075. DOI: 10.1126/SCIENCE.1099901/SUPPL_FILE/FYNH.SOM.PDF. [Online]. Available: <https://www.science.org/doi/10.1126/science.1099901> (cit. on p. 35).
 - [124] V. H. Brun et al., "Progressive increase in grid scale from dorsal to ventral medial entorhinal cortex," *Hippocampus*, vol. 18, pp. 1200-1212, 12 Dec. 2008, ISSN: 1098-1063. DOI: 10.1002/HIP0.20504. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/hipo.20504><https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.20504><https://onlinelibrary.wiley.com/doi/10.1002/hipo.20504> (cit. on p. 35).

- [125] A. Samsonovich and B. L. McNaughton, "Path integration and cognitive mapping in a continuous attractor neural network model," *Journal of Neuroscience*, vol. 17, pp. 5900–5920, 15 Aug. 1997, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.17-15-05900.1997. [Online]. Available: <https://www.jneurosci.org/content/17/15/5900>[https://www.jneurosci.org/content/17/15/5900abstract](https://www.jneurosci.org/content/17/15/5900.abstract) (cit. on p. 36).
- [126] Y. Burak and I. R. Fiete, "Accurate path integration in continuous attractor network models of grid cells," *PLoS Computational Biology*, vol. 5, O. Sporns, Ed., e1000291, 2 Feb. 2009, ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1000291. [Online]. Available: <https://dx.plos.org/10.1371/journal.pcbi.1000291> (cit. on p. 36).
- [127] N. Burgess, C. Barry and J. O'Keefe, "An oscillatory interference model of grid cell firing," *Hippocampus*, vol. 17, pp. 801–812, 9 Sep. 2007, ISSN: 1050-9631. DOI: 10.1002/hipo.20327. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/hipo.20327> (cit. on p. 36).
- [128] N. Ulanovsky and C. F. Moss, "Hippocampal cellular and network activity in freely moving echolocating bats," *Nature Neuroscience* 2007 10:2, vol. 10, pp. 224–233, 2 Jan. 2007, ISSN: 1546-1726. DOI: 10.1038/nn1829. [Online]. Available: <https://www.nature.com/articles/nn1829> (cit. on p. 36).
- [129] M. M. Yartsev and N. Ulanovsky, "Representation of three-dimensional space in the hippocampus of flying bats," *Science*, vol. 340, pp. 367–372, 6130 Apr. 2013, ISSN: 10959203. DOI: 10.1126/SCIENCE.1235338/SUPPL_FILE/YARTSEV.SM.PDF. [Online]. Available: <https://www.science.org/doi/10.1126/science.1235338> (cit. on p. 36).
- [130] M. Geva-Sagiv, S. Romani, L. Las and N. Ulanovsky, "Hippocampal global remapping for different sensory modalities in flying bats," *Nature Neuroscience* 2016 19:7, vol. 19, pp. 952–958, 7 May 2016, ISSN: 1546-1726. DOI: 10.1038/nn.4310. [Online]. Available: <https://www.nature.com/articles/nn.4310> (cit. on p. 36).
- [131] T. Ono, K. Nakamura, M. Fukuda and R. Tamura, "Place recognition responses of neurons in monkey hippocampus," *Neuroscience Letters*, vol. 121, pp. 194–198, 1-2 Jan. 1991, ISSN: 0304-3940. DOI: 10.1016/0304-3940(91)90683-K (cit. on p. 36).
- [132] E. Hori et al., "Place-related neural responses in the monkey hippocampal formation in a virtual space," *Hippocampus*, vol. 15, pp. 991–996, 8 Jan. 2005, ISSN: 1098-1063. DOI: 10.1002/HIPO.20108. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/hipo.20108><https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.20108><https://onlinelibrary.wiley.com/doi/10.1002/hipo.20108> (cit. on p. 36).

- [133] Y. Hazama and R. Tamura, "Effects of self-locomotion on the activity of place cells in the hippocampus of a freely behaving monkey," *Neuroscience Letters*, vol. 701, pp. 32–37, May 2019, ISSN: 0304-3940. DOI: 10.1016/J.NEULET.2019.02.009 (cit. on p. 36).
- [134] H. S. Courellis et al., "Spatial encoding in primate hippocampus during free navigation," *PLOS Biology*, vol. 17, e3000546, 12 2019, ISSN: 1545-7885. DOI: 10.1371/JOURNAL.PBIO.3000546. [Online]. Available: <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000546> (cit. on p. 36).
- [135] J. O'Keefe and D. H. Conway, "Hippocampal place units in the freely moving rat: Why they fire where they fire," *Experimental Brain Research*, vol. 31, pp. 573–590, 4 Apr. 1978, ISSN: 00144819. DOI: 10.1007/BF00239813/METRICS. [Online]. Available: <https://link.springer.com/article/10.1007/BF00239813> (cit. on p. 36).
- [136] J. Bures, A. A. Fenton, Y. Kaminsky and L. Zinyuk, "Place cells and place navigation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 94, pp. 343–350, 1 Jan. 1997, ISSN: 00278424. DOI: 10.1073/PNAS.94.1.343/ASSET/2F331802-CE4F-46DD-9B66-1D0B4FCDB94C/ASSETS/GRAPHIC/PQ0173175007.JPEG. [Online]. Available: <https://www.pnas.org/doi/abs/10.1073/pnas.94.1.343> (cit. on p. 37).
- [137] E. I. Moser, E. Kropff and M. B. Moser, "Place cells, grid cells, and the brain's spatial representation system," *Annual Review of Neuroscience*, vol. 31, pp. 69–89, Volume 31, 2008 Jul. 2008, ISSN: 0147006X. DOI: 10.1146/ANNUREV.NEURO.31.061307.090723/CITE/REFWORKS. [Online]. Available: <https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.31.061307.090723> (cit. on p. 37).
- [138] L. L. Colgin, E. I. Moser and M. B. Moser, "Understanding memory through hippocampal remapping," *Trends in Neurosciences*, vol. 31, pp. 469–477, 9 Sep. 2008, ISSN: 0166-2236. DOI: 10.1016/J.TINS.2008.06.008 (cit. on p. 37).
- [139] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor and R. Benosman, "Hfirst: A temporal approach to object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2028–2040, 2015. DOI: 10.1109/TPAMI.2015.2392947 (cit. on pp. 37, 38).
- [140] G. Orchard, A. Jayawant, G. K. Cohen and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Frontiers in Neuroscience*, vol. 9, Nov. 2015, ISSN: 1662-453X. DOI: 10.3389/fnins.2015.00437. [Online]. Available: <http://journal.frontiersin.org/Article/10.3389/fnins.2015.00437/abstract> (cit. on pp. 37, 38).

- [141] R. Serrano-Gotarredona et al., “Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory–processing– learning–actuating system for high-speed visual object recognition and tracking,” *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1417–1438, 2009. DOI: 10.1109/TNN.2009.2023653 (cit. on p. 38).
- [142] G. K. Cohen, G. Orchard, S. H. Leng, J. Tapson, R. B. Benosman and A. van Schaik, “Skimming digits: Neuromorphic classification of spike-encoded images,” *Frontiers in Neuroscience*, vol. 10, p. 191 715, APR Apr. 2016, ISSN: 1662453X. DOI: 10.3389/FNINS.2016.00184/BIBTEX. [Online]. Available: www.frontiersin.org (cit. on p. 38).
- [143] A. Amir et al., “A low power, fully event-based gesture recognition system,” vol. 2017-January, IEEE, Jul. 2017, pp. 7388–7397, ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.781. [Online]. Available: <https://ieeexplore.ieee.org/document/8100264/> (cit. on p. 38).
- [144] J. H. Lee et al., “Real-time gesture interface based on event-driven processing from stereo silicon retinas,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2250–2263, 2014. DOI: 10.1109/TNNLS.2014.2308551 (cit. on p. 38).
- [145] S. Hussaini, M. Milford and T. Fischer, “Spiking neural networks for visual place recognition via weighted neuronal assignments,” *IEEE Robotics and Automation Letters*, vol. 7, pp. 4094–4101, 2 Apr. 2022, ISSN: 2377-3766. DOI: 10.1109/LRA.2022.3149030. [Online]. Available: <https://ieeexplore.ieee.org/document/9706280/> (cit. on p. 38).
- [146] R. Mur-Artal and J. D. Tardos, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, pp. 1255–1262, 5 Oct. 2017, ISSN: 1552-3098. DOI: 10.1109/TR0.2017.2705103. [Online]. Available: <http://ieeexplore.ieee.org/document/7946260/> (cit. on p. 39).
- [147] C. Cadena et al., “Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age,” *IEEE Transactions on Robotics*, vol. 32, pp. 1309–1332, 6 Dec. 2016, ISSN: 1552-3098. DOI: 10.1109/TR0.2016.2624754. [Online]. Available: <http://ieeexplore.ieee.org/document/7747236/> (cit. on p. 39).
- [148] S. Wang, R. Clark, H. Wen and N. Trigoni, “Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050. DOI: 10.1109/ICRA.2017.7989236 (cit. on p. 39).
- [149] R. Li, S. Wang and D. Gu, “Ongoing evolution of visual slam from geometry to deep learning: Challenges and opportunities,” *Cognitive Computation*, vol. 10, pp. 875–889, 6 Dec. 2018, ISSN: 1866-9956. DOI: 10.1007/s12559-018-9591-8. [Online]. Available: <http://link.springer.com/10.1007/s12559-018-9591-8> (cit. on p. 39).

- [150] A. K. Kosta and K. Roy, "Adaptive-spikenet: Event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 6021–6027. DOI: 10.1109/ICRA48891.2023.10160551 (cit. on p. 39).
- [151] M. Milford, G. Wyeth and D. Prasser, "Ratslam: A hippocampal model for simultaneous localization and mapping," IEEE, 2004, 403–408 Vol.1, ISBN: 0-7803-8232-3. DOI: 10.1109/ROBOT.2004.1307183. [Online]. Available: <http://ieeexplore.ieee.org/document/1307183/> (cit. on pp. 39, 40).
- [152] X. Xie, R. H. R. Hahnloser and H. S. Seung, "Double-ring network model of the head-direction system," *Physical Review E*, vol. 66, p. 041902, 4 Oct. 2002, ISSN: 1063-651X. DOI: 10.1103/PhysRevE.66.041902. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.66.041902> (cit. on p. 39).
- [153] P. Song and X.-J. Wang, "Angular path integration by moving "hill of activity": A spiking neuron model without recurrent excitation of the head-direction system," *The Journal of Neuroscience*, vol. 25, pp. 1002–1014, 4 Jan. 2005, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.4172-04.2005. [Online]. Available: <https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.4172-04.2005> (cit. on pp. 39, 49, 73).
- [154] G. Tang and K. P. Michmizos, "Gridbot: An autonomous robot controlled by a spiking neural network mimicking the brain's navigational system," vol. 18, ACM, Jul. 2018, pp. 1–8, ISBN: 9781450365444. DOI: 10.1145/3229884.3229888. [Online]. Available: <https://dl.acm.org/doi/10.1145/3229884.3229888> (cit. on pp. 39, 40).
- [155] R. Kreiser, M. Cartiglia, J. N. Martel, J. Conradt and Y. Sandamirskaya, "A neuromorphic approach to path integration: A head-direction spiking neural network with vision-driven reset," vol. 2018-May, IEEE, Apr. 2018, pp. 1–5, ISBN: 978-1-5386-4881-0. DOI: 10.1109/ISCAS.2018.8351509. [Online]. Available: <https://ieeexplore.ieee.org/document/8351509/> (cit. on pp. 39, 42).
- [156] J. Cheng, L. Zhang, Q. Chen, X. Hu and J. Cai, "A review of visual slam methods for autonomous driving vehicles," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 104992, Sep. 2022, ISSN: 09521976. DOI: 10.1016/j.engappai.2022.104992. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0952197622001853> (cit. on p. 39).
- [157] S. Milz, G. Arbeiter, C. Witt, B. Abdallah and S. Yogamani, "Visual slam for automated driving: Exploring the applications of deep learning," IEEE, Jun. 2018, pp. 360–36010, ISBN: 978-1-5386-6100-0. DOI: 10.1109/CVPRW.2018.00062. [Online]. Available: <https://ieeexplore.ieee.org/document/8575523/> (cit. on p. 39).

- [158] L. Steffen et al., “Creating an obstacle memory through event-based stereo vision and robotic proprioception,” vol. 2019-August, IEEE, Aug. 2019, pp. 1829–1836, ISBN: 978-1-7281-0356-3. DOI: 10.1109/COASE.2019.8843238. [Online]. Available: <https://ieeexplore.ieee.org/document/8843238/> (cit. on p. 40).
- [159] B. Rueckauer and T. Delbruck, “Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor,” *Frontiers in Neuroscience*, vol. 10, Apr. 2016, ISSN: 1662-453X. DOI: 10.3389/fnins.2016.00176. [Online]. Available: <http://journal.frontiersin.org/Article/10.3389/fnins.2016.00176/abstract> (cit. on p. 40).
- [160] E. Piñero-Fuentes, S. Canas-Moreno, A. Rios-Navarro, T. Delbruck and A. Linares-Barranco, “Autonomous driving of a rover-like robot using neuromorphic computing,” in Springer Science and Business Media Deutschland GmbH, 2021, vol. 12862 LNCS, pp. 57–68, ISBN: 9783030850982. DOI: 10.1007/978-3-030-85099-9_5. [Online]. Available: https://link.springer.com/10.1007/978-3-030-85099-9_5 (cit. on p. 40).
- [161] N. J. Sanket et al., “Evdodgenet: Deep dynamic obstacle dodging with event cameras,” IEEE, May 2020, pp. 10 651–10 657, ISBN: 978-1-7281-7395-5. DOI: 10.1109/ICRA40945.2020.9196877. [Online]. Available: <https://ieeexplore.ieee.org/document/9196877/> (cit. on p. 40).
- [162] H. Kim, “Real-time visual slam with an event camera,” 2017 (cit. on p. 40).
- [163] A. R. Vidal, H. Rebecq, T. Horstschaefter and D. Scaramuzza, “Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 994–1001, 2 Apr. 2018, ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2793357. [Online]. Available: <https://ieeexplore.ieee.org/document/8258997/> (cit. on p. 40).
- [164] Y. Zhou, G. Gallego and S. Shen, “Event-based stereo visual odometry,” *IEEE Transactions on Robotics*, vol. 37, pp. 1433–1450, 5 Oct. 2021, Stereo event based cameras used to track motion and create a semi-dense 3D map of the scene using time surfaces and algorithms;
Section 1; Introduce event cameras and list contributions;
Section 2; Reviews Event-based Depth Estimation, Event-Based Pose Estimation, Event-Based VO and SLAM;
Section 3; Gives an overview of the system e;
Explaining what a time surface is, theirs updated at 100Hz;
Stores short history of time surfaces, after intialisation continuously estimate pose with respect to local map which are stored to get pose at any time through interpolation. Mapping module takes events, time surfaces and pose estimation to create a new local map. ;
Section 4; In depth explanation of mapping (depth estimation);
Section 5; In depth explanation of tracking;
Section 6; Used handheld, MVSEC and sim-

- ulated sequences; compares results to other methods; ISSN: 1552-3098. DOI: 10.1109/TR0.2021.3062252. [Online]. Available: <https://ieeexplore.ieee.org/document/9386209/> (cit. on p. 40).
- [165] J. Hidalgo-Carrio, G. Gallego and D. Scaramuzza, “Event-aided direct sparse odometry,” *IEEE*, Jun. 2022, pp. 5771–5780, ISBN: 978-1-6654-6946-3. DOI: 10.1109/CVPR52688.2022.00569. [Online]. Available: <https://ieeexplore.ieee.org/document/9879881/> (cit. on p. 40).
- [166] K. Xiao, G. Wang, Y. Chen, Y. Xie, H. Li and S. Li, “Research on event accumulator settings for event-based slam,” *IEEE*, Feb. 2022, pp. 50–56, ISBN: 978-1-6654-8174-8. DOI: 10.1109/ICRCA55033.2022.9828933. [Online]. Available: <https://ieeexplore.ieee.org/document/9828933/> (cit. on p. 40).
- [167] G. Tang, A. Shah and K. P. Michmizos, “Spiking neural network on neuromorphic hardware for energy-efficient unidimensional slam,” *IEEE*, Nov. 2019, pp. 4176–4181, ISBN: 978-1-7281-4004-9. DOI: 10.1109/IR0S40897.2019.8967864. [Online]. Available: <https://ieeexplore.ieee.org/document/8967864/> (cit. on pp. 40, 41).
- [168] Z. Bing et al., “End to end learning of spiking neural network based on r-stdp for a lane keeping vehicle,” *IEEE*, May 2018, pp. 4725–4732, ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8460482. [Online]. Available: <https://ieeexplore.ieee.org/document/8460482/> (cit. on pp. 40, 41).
- [169] Z. Bing, I. Baumann, Z. Jiang, K. Huang, C. Cai and A. Knoll, “Supervised learning in snn via reward-modulated spike-timing-dependent plasticity for a target reaching vehicle,” *Frontiers in Neurorobotics*, vol. 13, p. 18, May 2019, ISSN: 1662-5218. DOI: 10.3389/fnbot.2019.00018. [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnbot.2019.00018/full> (cit. on p. 40).
- [170] S. Poulter, T. Hartley and C. Lever, “The neurobiology of mammalian navigation,” *Current Biology*, vol. 28, R1023–R1042, 17 Sep. 2018, ISSN: 09609822. DOI: 10.1016/j.cub.2018.05.050. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0960982218306869> (cit. on p. 40).
- [171] D. C. Rowland, Y. Roudi, M.-B. Moser and E. I. Moser, “Ten years of grid cells,” *Annual Review of Neuroscience*, vol. 39, pp. 19–40, 1 Jul. 2016, ISSN: 0147-006X. DOI: 10.1146/annurev-neuro-070815-013824. [Online]. Available: <https://www.annualreviews.org/doi/10.1146/annurev-neuro-070815-013824> (cit. on p. 40).
- [172] R. Kreiser, A. Renner, Y. Sandamirskaya and P. Pienroj, “Pose estimation and map formation with spiking neural networks: Towards neuromorphic slam,” *IEEE*, Oct. 2018, pp. 2159–2166, ISBN: 978-1-5386-8094-0. DOI: 10.1109/IR0S.2018.8594228. [Online]. Available: <https://ieeexplore.ieee.org/document/8594228/> (cit. on pp. 40, 42).

- [173] H. Tang, R. Yan and K. C. Tan, "Cognitive navigation by neuro-inspired localization, mapping, and episodic memory," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, pp. 751–761, 3 Sep. 2018, ISSN: 2379-8920. DOI: 10.1109/TCDS.2017.2776965. [Online]. Available: <https://ieeexplore.ieee.org/document/8119542/> (cit. on p. 40).
- [174] A. Vitale, A. Renner, C. Nauer, D. Scaramuzza and Y. Sandamirskaya, "Event-driven vision and control for uavs on a neuromorphic chip," IEEE, May 2021, pp. 103–109, ISBN: 978-1-7281-9077-8. DOI: 10.1109/ICRA48506.2021.9560881. [Online]. Available: <https://ieeexplore.ieee.org/document/9560881/> (cit. on p. 41).
- [175] B. Chakraborty, U. Kamal, X. She, S. Dash and S. Mukhopadhyay, "Brain-inspired spatiotemporal processing algorithms for efficient event-based perception," IEEE, Apr. 2023, pp. 1–6, ISBN: 9783981926378. DOI: 10.23919/DATE56975.2023.10136914. [Online]. Available: <https://ieeexplore.ieee.org/document/10136914/> (cit. on p. 41).
- [176] J. Kaiser et al., "Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks," IEEE, Dec. 2016, pp. 127–134, ISBN: 978-1-5090-4616-4. DOI: 10.1109/SIMPAN.2016.7862386. [Online]. Available: <https://ieeexplore.ieee.org/document/7862386/> (cit. on p. 42).
- [177] M. B. Milde et al., "Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system," *Frontiers in Neurobotics*, vol. 11, Jul. 2017, ISSN: 1662-5218. DOI: 10.3389/fnbot.2017.00028. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fnbot.2017.00028/full> (cit. on p. 42).
- [178] H. Blum, A. Dietmüller, M. Milde, J. Conradt, G. Indiveri and Y. Sandamirskaya, "A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor," Robotics: Science and Systems Foundation, Jul. 2017, ISBN: 978-0-9923747-3-0. DOI: 10.15607/RSS.2017.XIII.035. [Online]. Available: <http://www.roboticsproceedings.org/rss13/p35.pdf> (cit. on p. 42).
- [179] R. Kreiser, G. Waibel, N. Armengol, A. Renner and Y. Sandamirskaya, "Error estimation and correction in a spiking neural network for map formation in neuromorphic hardware," IEEE, May 2020, pp. 6134–6140, ISBN: 978-1-7281-7395-5. DOI: 10.1109/ICRA40945.2020.9197498. [Online]. Available: <https://ieeexplore.ieee.org/document/9197498/> (cit. on p. 42).
- [180] A. Safa et al., "Fusing event-based camera and radar for slam using spiking neural networks with continual stdp learning," IEEE, May 2023, pp. 2782–2788, ISBN: 979-8-3503-2365-8. DOI: 10.1109/ICRA48891.2023.10160681. [Online]. Available: <https://ieeexplore.ieee.org/document/10160681/> (cit. on p. 42).

- [181] P. Chen et al., “Ecmd: An event-centric multisensory driving dataset for slam,” *IEEE Transactions on Intelligent Vehicles*, vol. 9, pp. 407–416, 1 Jan. 2024, ISSN: 2379-8904. DOI: 10.1109/TIV.2023.3339144. [Online]. Available: <https://ieeexplore.ieee.org/document/10342726/> (cit. on pp. 43, 47, 49, 61, 118).
- [182] M. Gehrig, W. Aarents, D. Gehrig and D. Scaramuzza, “Dsec: A stereo event camera dataset for driving scenarios,” *IEEE Robotics and Automation Letters*, 2021. DOI: 10.1109/LRA.2021.3068942 (cit. on pp. 43, 47, 118).
- [183] J. Jiao et al., “Fusionportable: A multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 3851–3856 (cit. on p. 43).
- [184] S Klenk, J Chui, N Demmel and D Cremers, “Tum-vie: The tum stereo visual-inertial event dataset,” in *International Conference on Intelligent Robots and Systems (IROS)*, 2021. arXiv: 2108.07329 [cs.CV] (cit. on p. 44).
- [185] K. Chaney et al., “M3ed: Multi-robot, multi-sensor, multi-environment event dataset,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2023, pp. 4015–4022 (cit. on pp. 44, 47, 118).
- [186] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar and K. Daniilidis, “The multivehicle stereo event camera dataset: An event camera dataset for 3d perception,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 2032–2039, 3 Jul. 2018, ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2800793. [Online]. Available: <http://ieeexplore.ieee.org/document/8288670/> (cit. on pp. 44, 47, 118).
- [187] J. Binas, D. Neil, S.-C. Liu and T. Delbruck, “Ddd17: End-to-end davis driving dataset,” Nov. 2017. [Online]. Available: <https://arxiv.org/abs/1711.01458v1> (cit. on p. 44).
- [188] Y. Hu, J. Binas, D. Neil, S. C. Liu and T. Delbruck, “Ddd20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction,” *2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC 2020*, Sep. 2020. DOI: 10.1109/ITSC45102.2020.9294515 (cit. on p. 44).
- [189] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce and R. Benosman, “Hats: Histograms of averaged time surfaces for robust event-based object classification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1731–1740. DOI: 10.1109/CVPR.2018.00186 (cit. on p. 44).
- [190] T. Fischer and M. Milford, “Event-based visual place recognition with ensembles of temporal windows,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 6924–6931, 4 Oct. 2020, ISSN: 23773766. DOI: 10.1109/LRA.2020.3025505 (cit. on p. 44).

- [191] H. Rebecq, D. Gehrig and D. Scaramuzza, “Esim: An open event camera simulator,” 2018 (cit. on p. 44).
- [192] Z. Zhang et al., “V2ce: Video to continuous events simulator,” Sep. 2023. [Online]. Available: <http://arxiv.org/abs/2309.08891> (cit. on p. 44).
- [193] Y. Hu, S. C. Liu and T. Delbruck, “V2e: From video frames to realistic dvs events,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1312–1321, Jun. 2020, ISSN: 21607516. DOI: 10.1109/CVPRW53098.2021.00144. [Online]. Available: <https://arxiv.org/pdf/2006.07722> (cit. on p. 44).
- [194] K Zhang, “Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory,” *The Journal of Neuroscience*, vol. 16, pp. 2112–2126, 6 Mar. 1996, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.16-06-02112.1996. [Online]. Available: <https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.16-06-02112.1996> (cit. on p. 49).
- [195] Z. Bing et al., “Toward cognitive navigation: Design and implementation of a biologically inspired head direction cell network,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 2147–2158, 5 May 2022, ISSN: 2162-237X. DOI: 10.1109/TNNLS.2021.3128380. [Online]. Available: <https://ieeexplore.ieee.org/document/9634854/> (cit. on pp. 50–52).
- [196] X. J. Wang, “Synaptic basis of cortical persistent activity: The importance of nmda receptors to working memory,” *Journal of Neuroscience*, vol. 19, pp. 9587–9603, 21 Nov. 1999, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.19-21-09587.1999. [Online]. Available: <https://www.jneurosci.org/content/19/21/9587><https://www.jneurosci.org/content/19/21/9587.abstract> (cit. on p. 53).
- [197] https://assets.publishing.service.gov.uk/media/5a7f2f39e5274a2e8ab4aa29/HS2_Rural_Road_Design_Criteria.pdf (cit. on p. 87).
- [198] <https://www.camden.gov.uk/documents/20142/18572305/Appendix+D+Highway+layout+and+traffic+assessment+of+Alternative+Scheme+Proposals.pdf/111b2c8f-d2ef-0c04-75b0-9b729de42008> (cit. on p. 87).
- [199] R. Gütig, R. Aharonov, S. Rotter and H. Sompolinsky, “Learning input correlations through nonlinear temporally asymmetric hebbian plasticity,” *Journal of Neuroscience*, vol. 23, pp. 3697–3714, 9 May 2003, ISSN: 0270-6474. DOI: 10.1523/JNEUROSCI.23-09-03697.2003. [Online]. Available: <https://www.jneurosci.org/content/23/9/3697><https://www.jneurosci.org/content/23/9/3697.abstract> (cit. on p. 89).
- [200] *Synapse type for spike-timing dependent plasticity - nest simulator documentation*. [Online]. Available: https://nest-simulator.readthedocs.io/en/latest/models/stdp_synapse.html#id5 (cit. on p. 89).

- [201] B. W. Towse, C. Barry, D. Bush and N. Burgess, “Optimal configurations of spatial scale for grid cell firing under noise and uncertainty,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, p. 20130290, 1635 Feb. 2014, ISSN: 14712970. DOI: 10.1098/RSTB.2013.0290. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC3866454/> (cit. on p. 92).

Appendix A

Additional Figures

A.1 Head Direction Network Shift Response

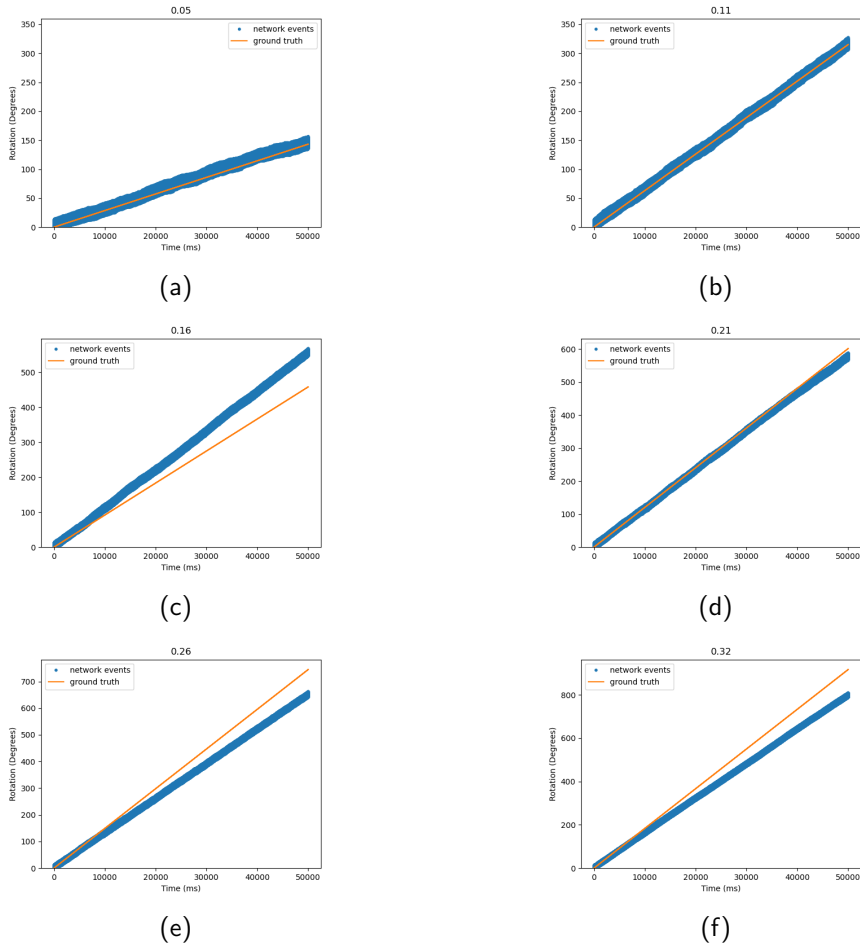


Figure A.1: Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.

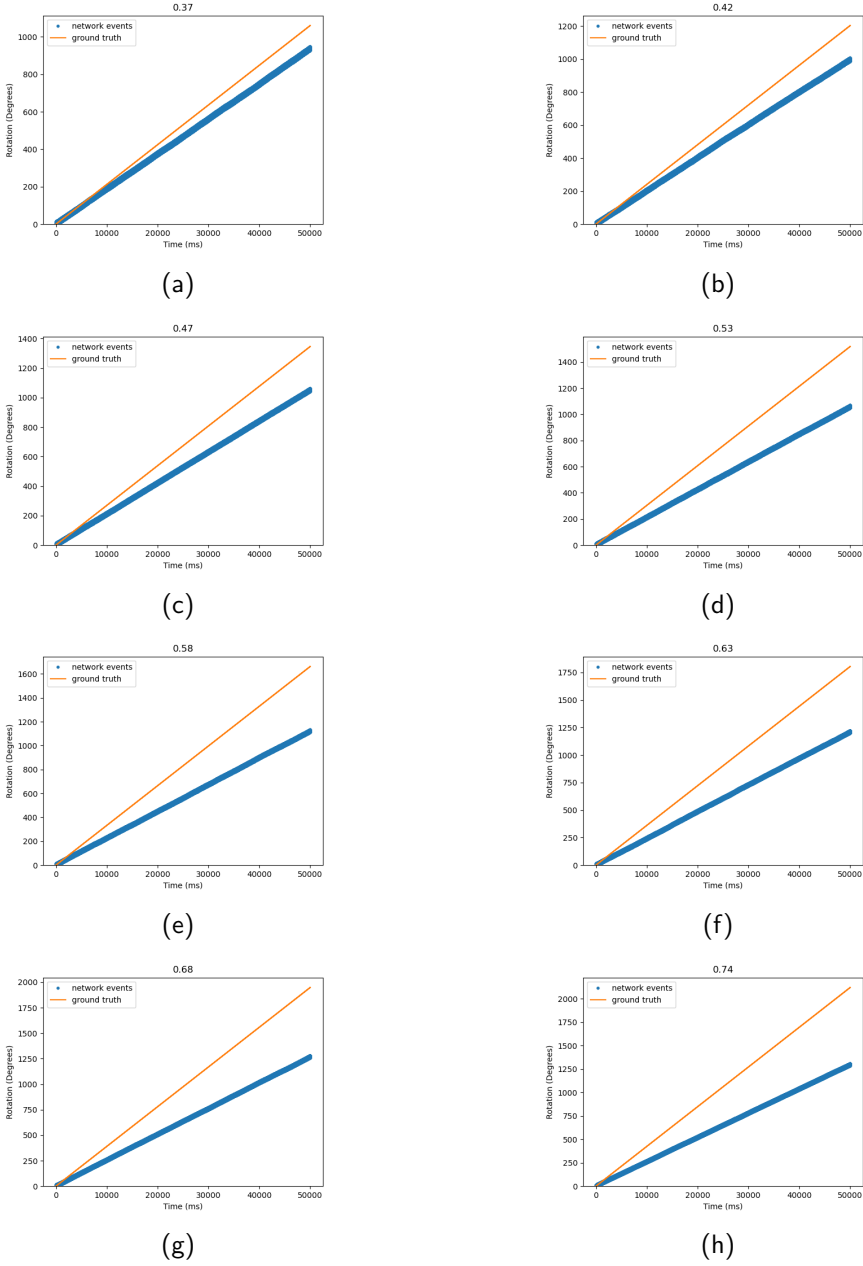


Figure A.2: Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.

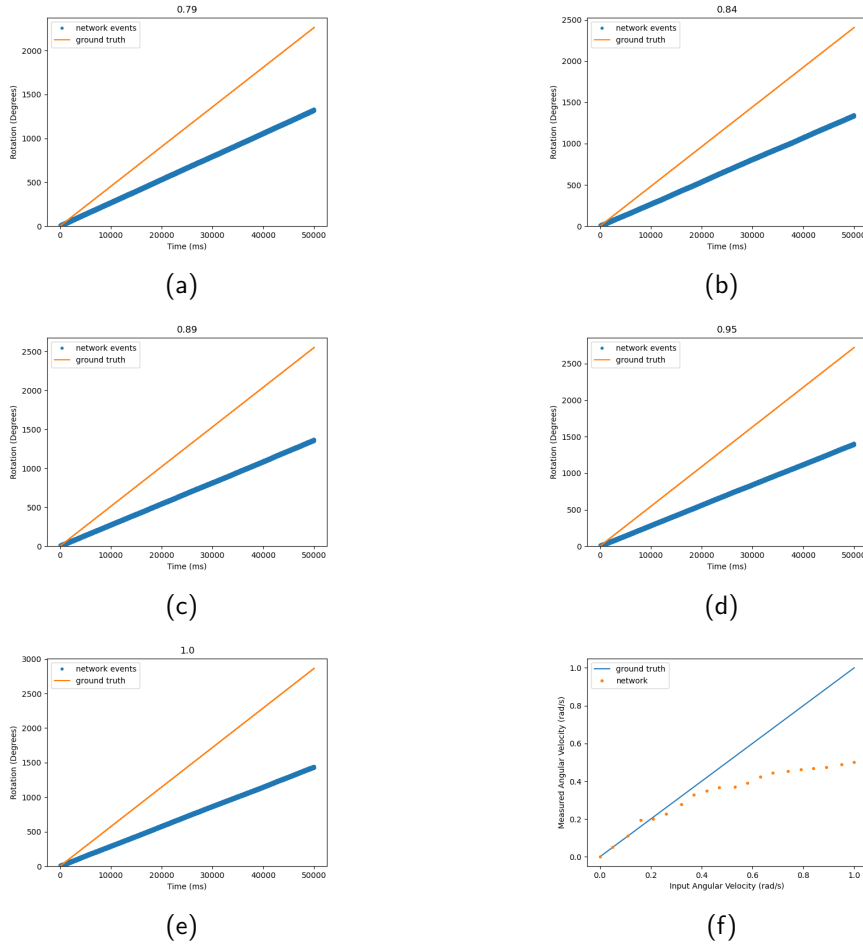


Figure A.3: Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot. (d) shows all of the measured rotations as angular velocities plotted against the input angular velocity.

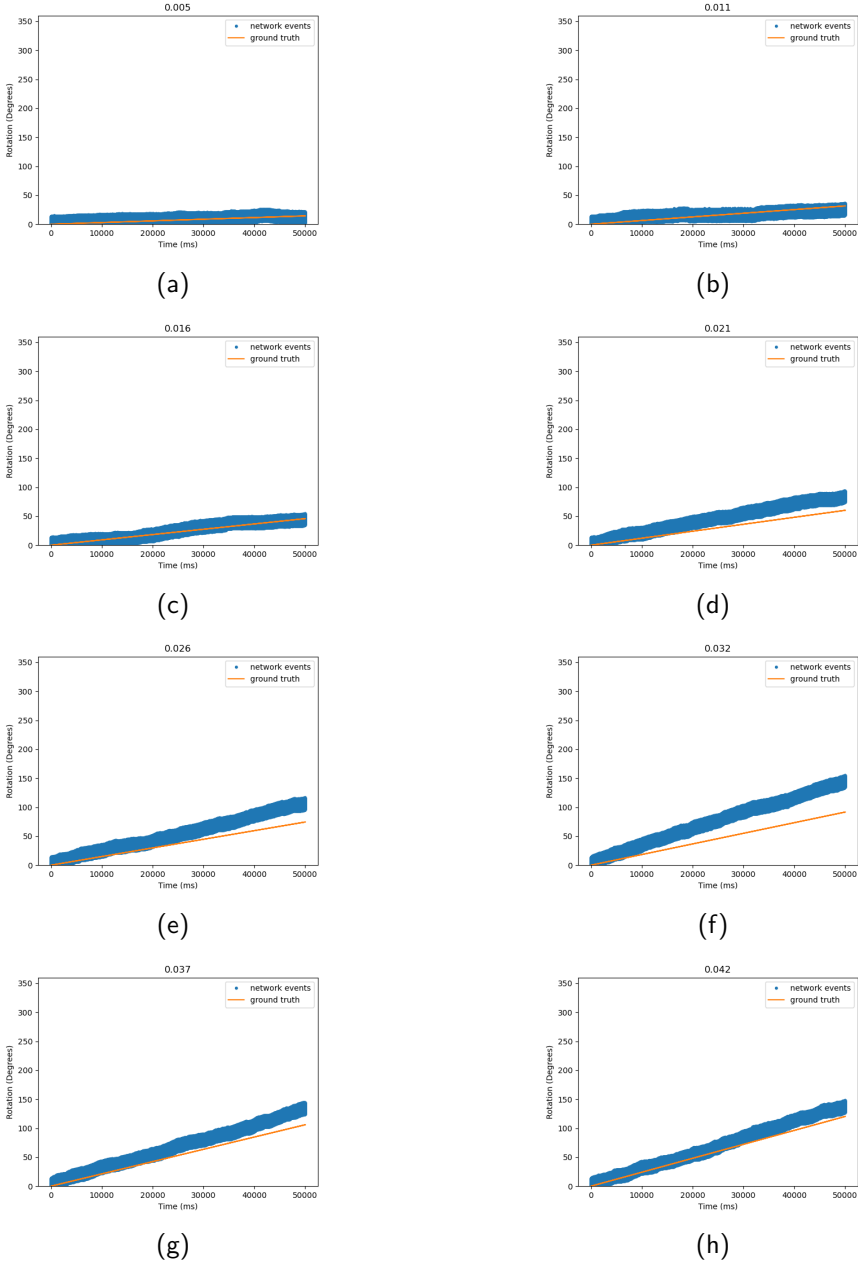
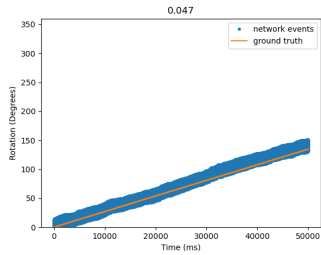
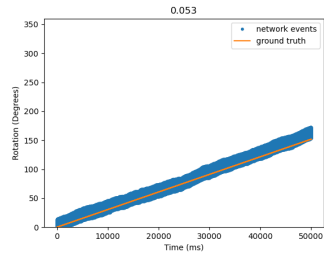


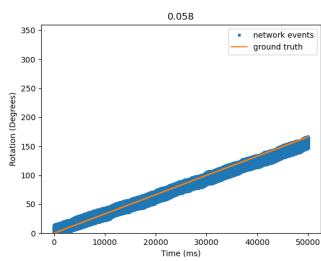
Figure A.4: Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.



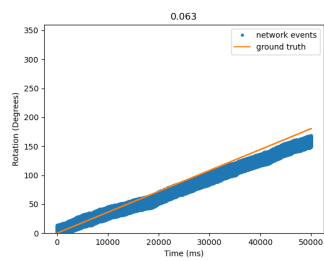
(a)



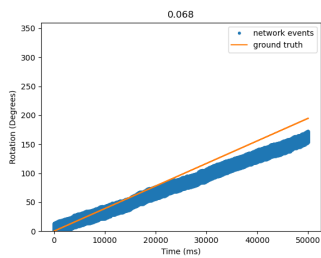
(b)



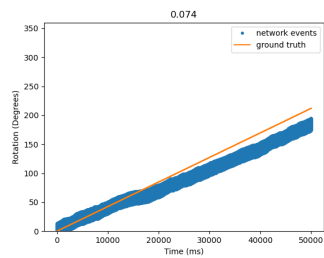
(c)



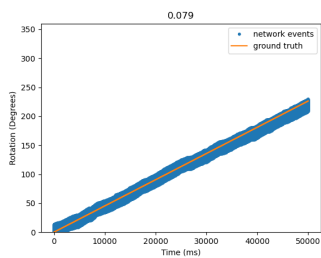
(d)



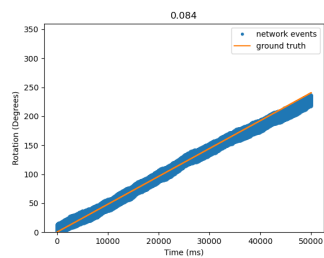
(e)



(f)



(g)



(h)

Figure A.5: Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot.

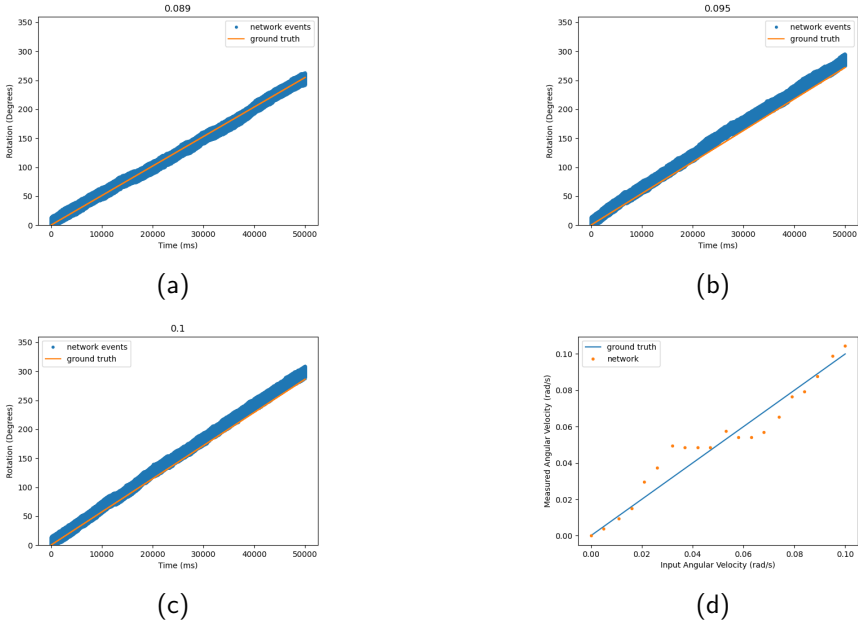
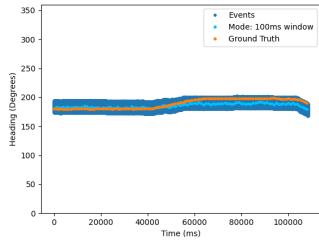
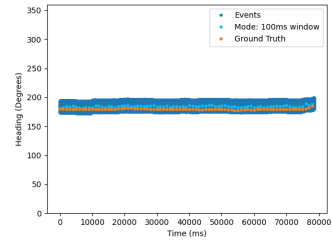


Figure A.6: Network rotation measured in degrees. The input angular velocity (recorded in rad/s) is seen in the title of each plot. (h) shows all of the measured rotations as angular velocities plotted against the input angular velocity.

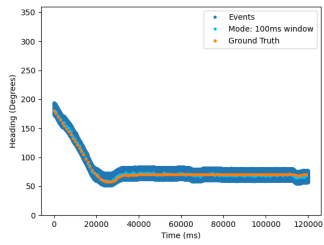
A.2 Head Direction Network Activity



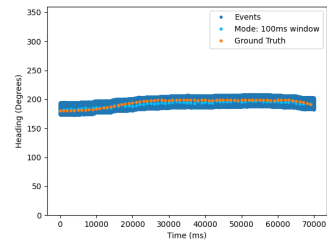
(a)



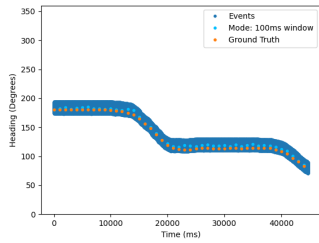
(b)



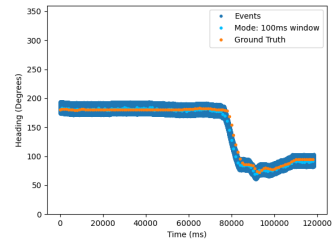
(c)



(d)



(e)



(f)

Figure A.7: Head Direction network activity for the ECMD dataset.

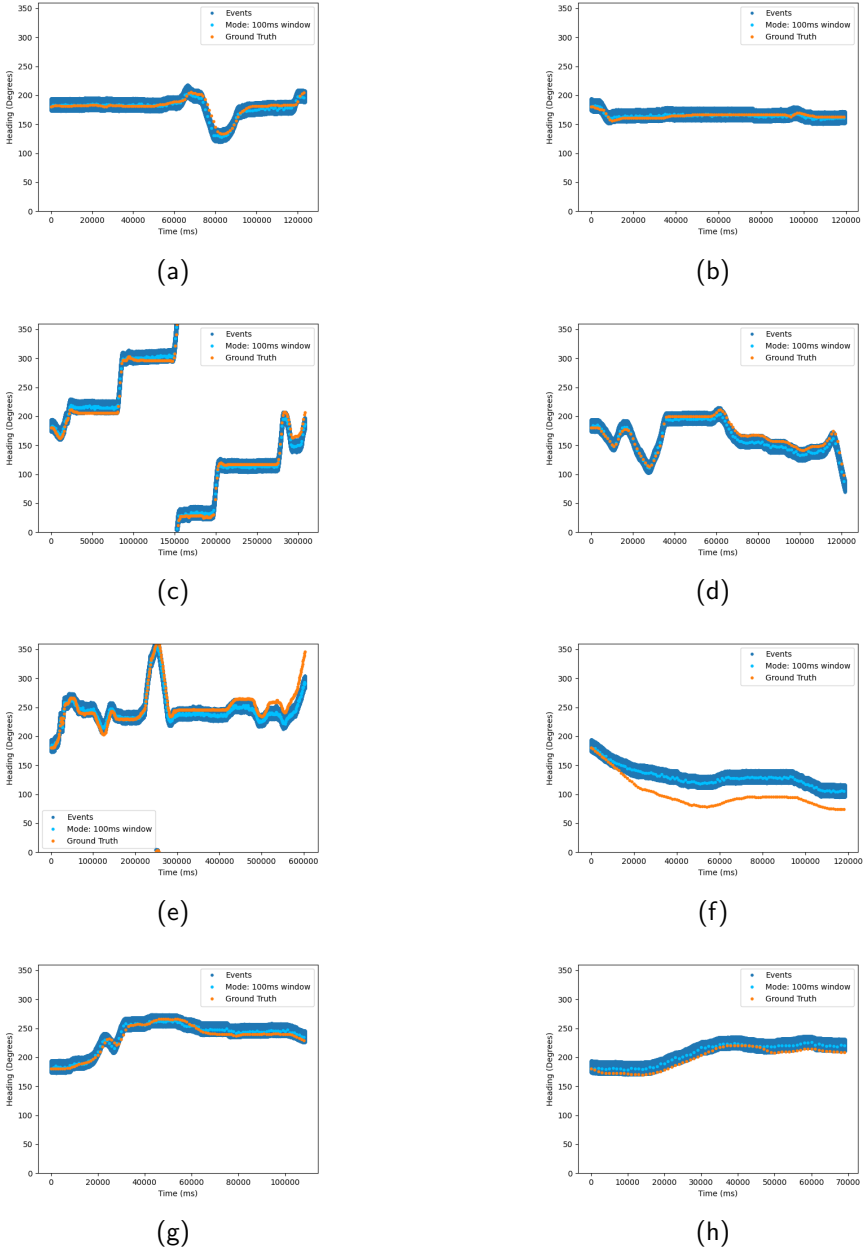
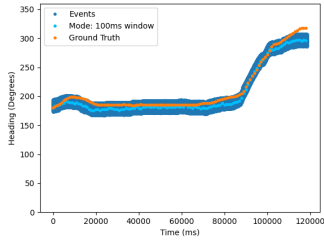
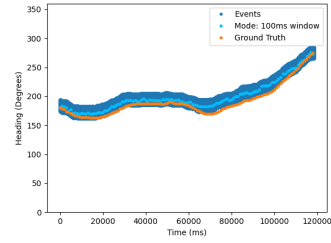


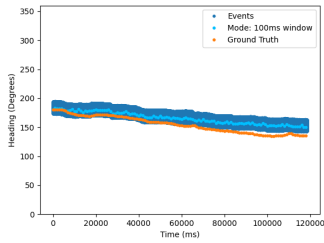
Figure A.8: Head Direction network activity for the ECMD dataset.



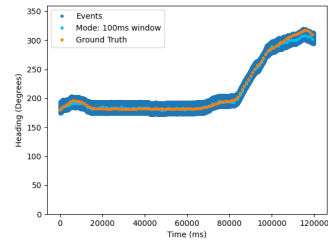
(a)



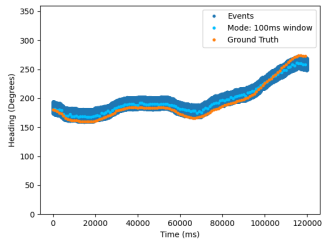
(b)



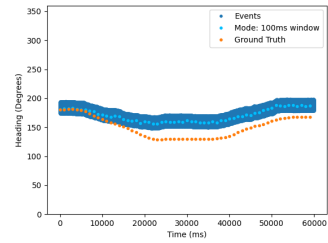
(c)



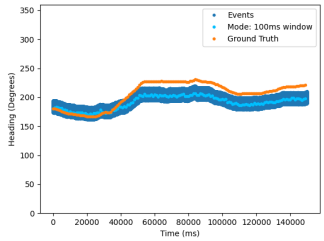
(d)



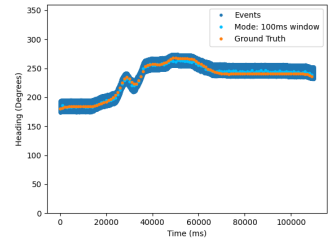
(e)



(f)

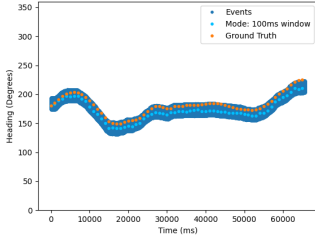


(g)

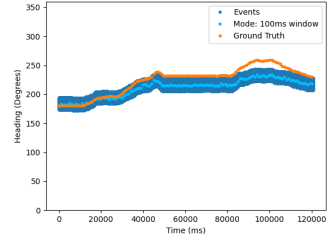


(h)

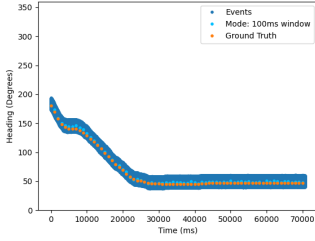
Figure A.9: Head Direction network activity for the ECMD dataset.



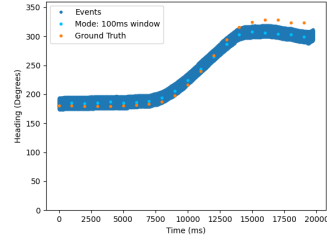
(a)



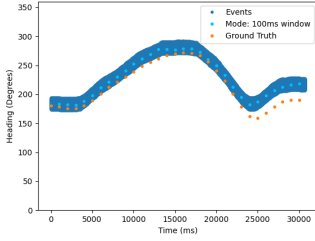
(b)



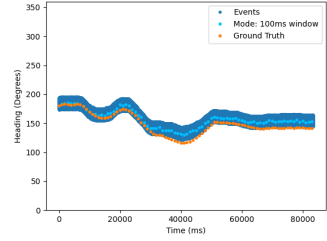
(c)



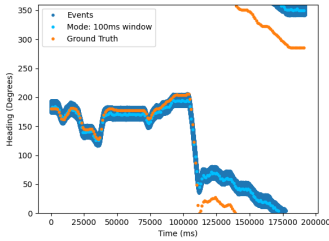
(d)



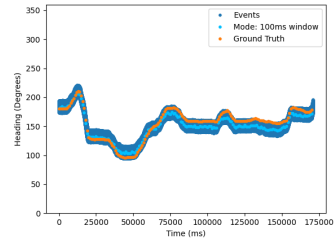
(e)



(f)

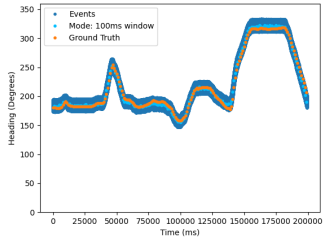


(g)

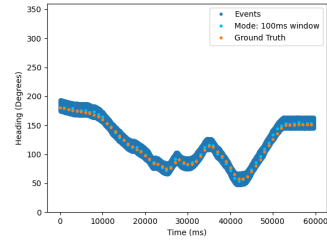


(h)

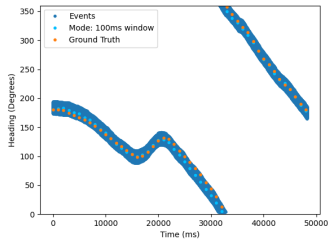
Figure A.10: Head Direction network activity for the ECMD dataset.



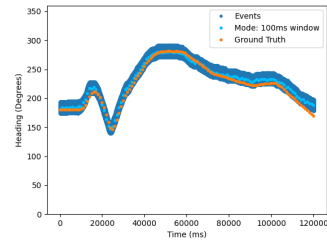
(a)



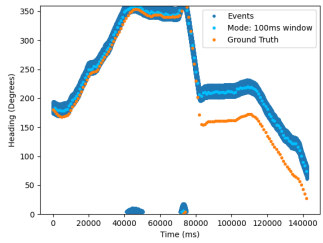
(b)



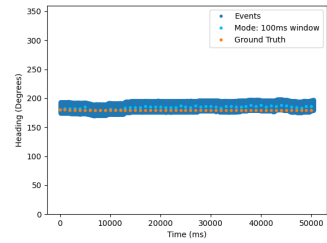
(c)



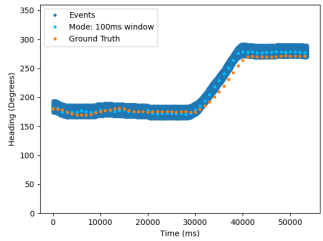
(d)



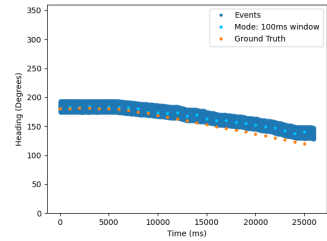
(e)



(f)

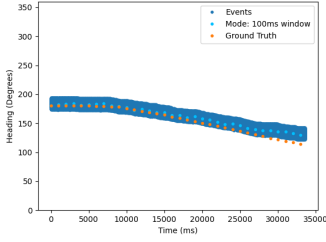


(g)

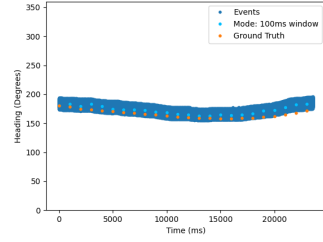


(h)

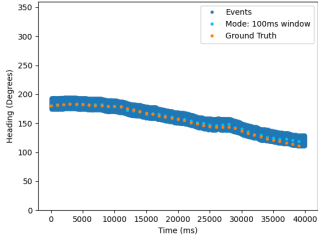
Figure A.11: Head Direction network activity for the ECMD dataset.



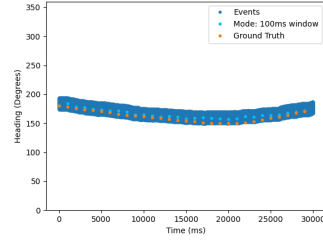
(a)



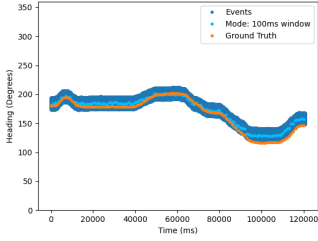
(b)



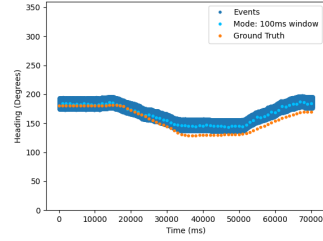
(c)



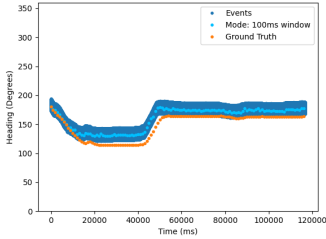
(d)



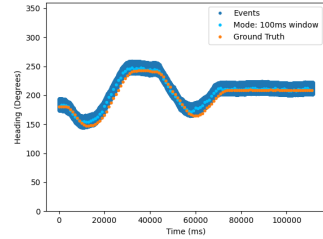
(e)



(f)

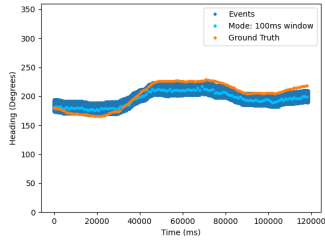


(g)

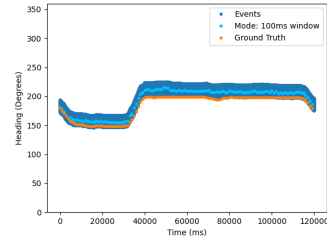


(h)

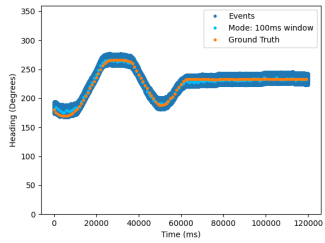
Figure A.12: Head Direction network activity for the ECMD dataset.



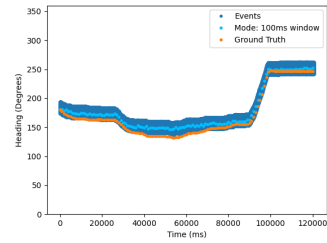
(a)



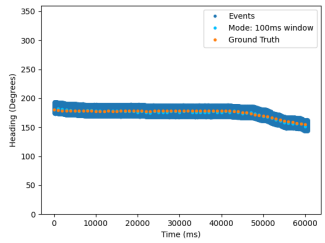
(b)



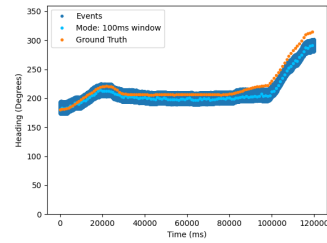
(c)



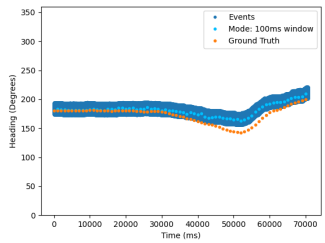
(d)



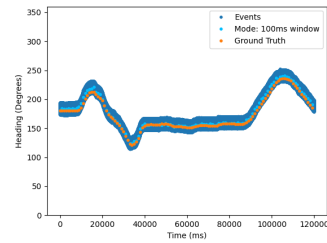
(e)



(f)

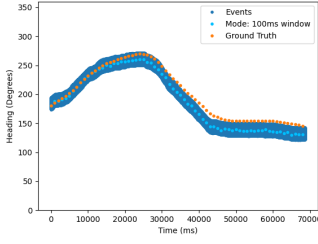


(g)

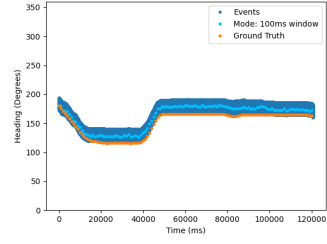


(h)

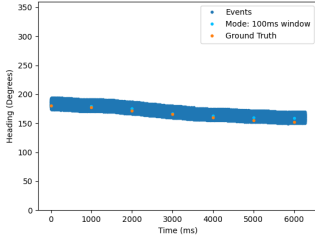
Figure A.13: Head Direction network activity for the ECMD dataset.



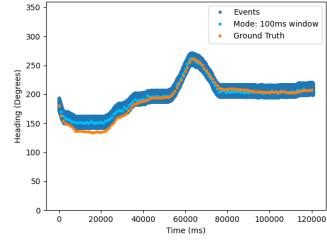
(a)



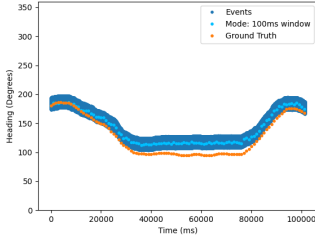
(b)



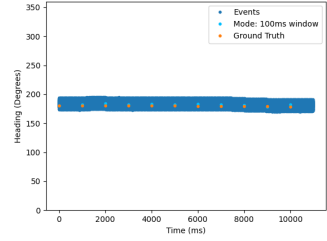
(c)



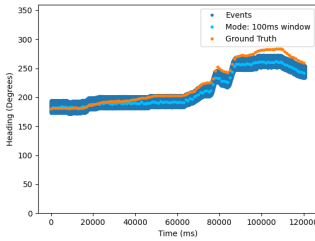
(d)



(e)



(f)



(g)

Figure A.14: Head Direction network activity for the ECMD dataset.

A.3 Head Direction Network Errors Against Time

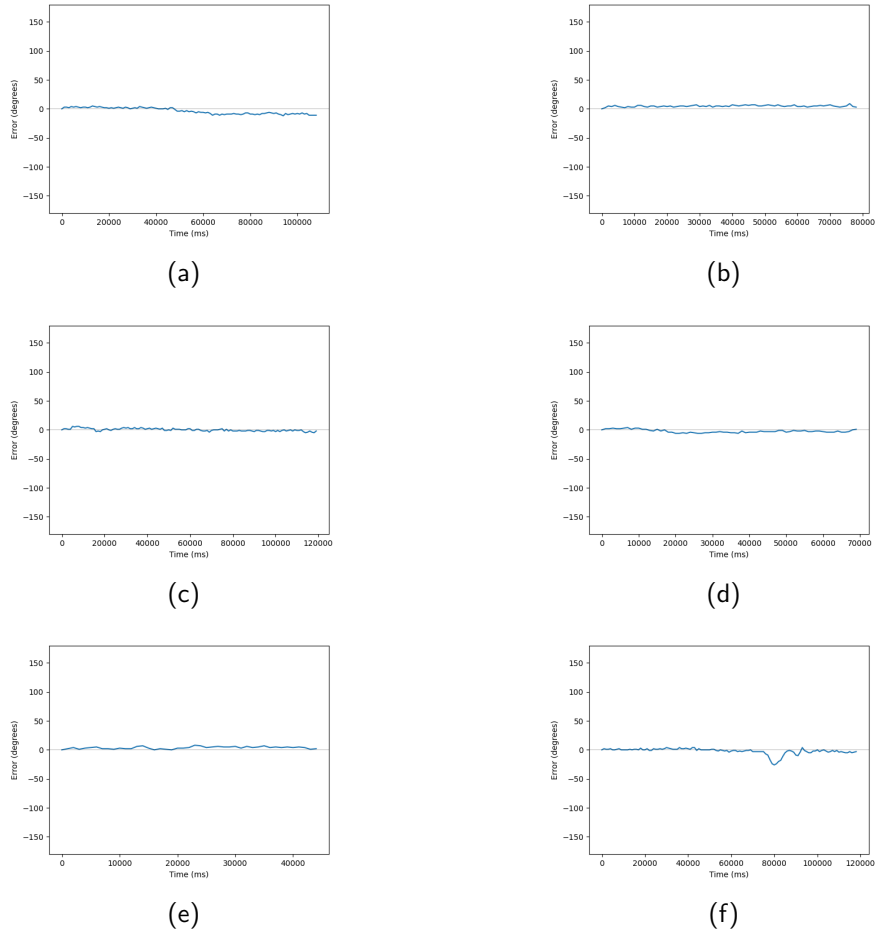


Figure A.15: Direction errors recorded in the Head Direction network when run on the ECMD dataset.

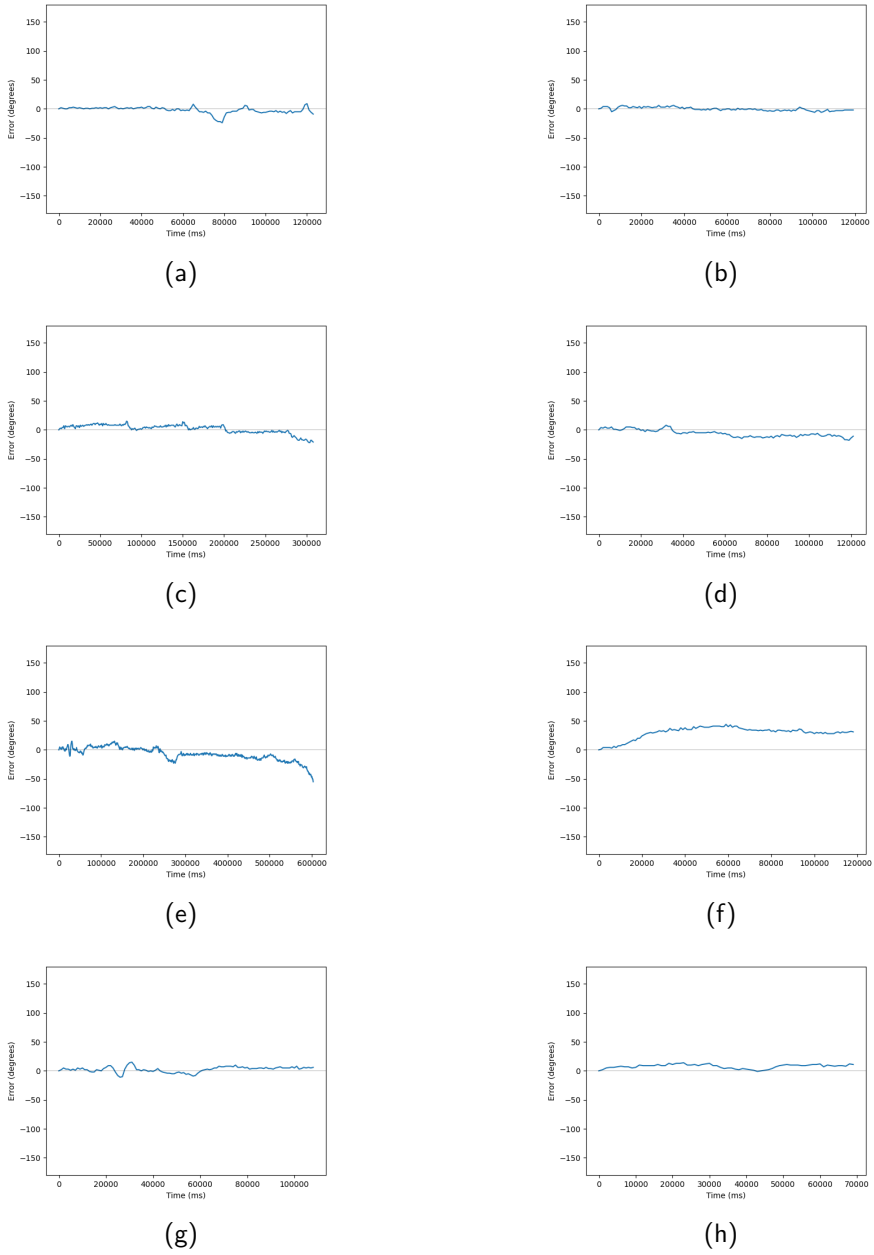
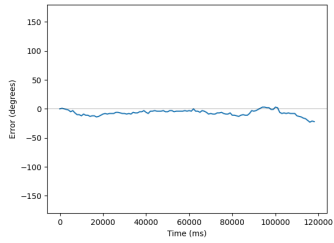
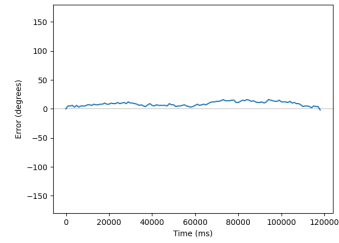


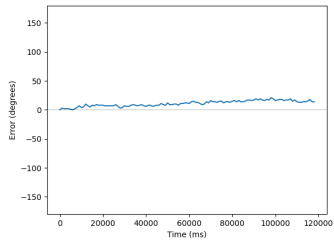
Figure A.16: Direction errors recorded in the Head Direction network when run on the ECMD dataset.



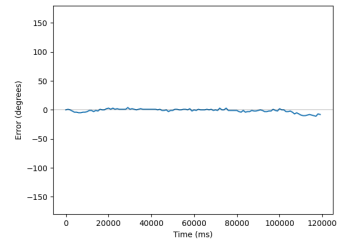
(a)



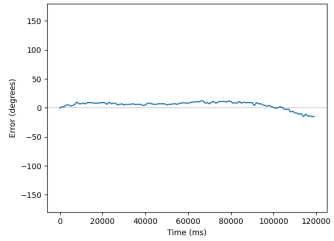
(b)



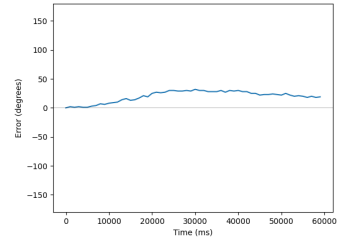
(c)



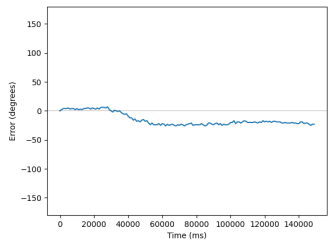
(d)



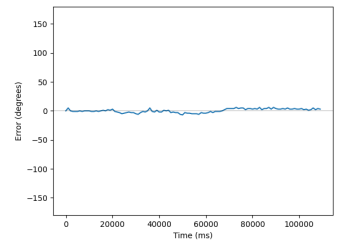
(e)



(f)

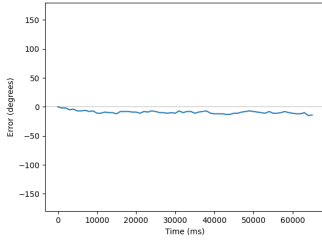


(g)

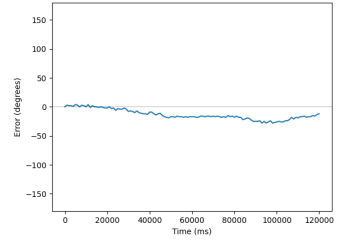


(h)

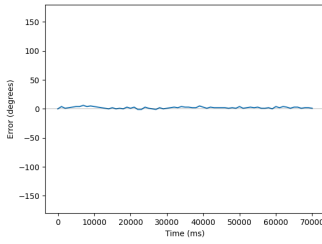
Figure A.17: Direction errors recorded in the Head Direction network when run on the ECMD dataset.



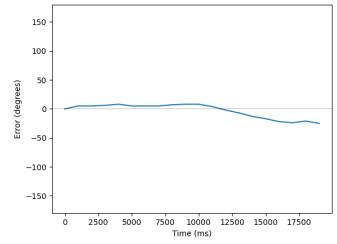
(a)



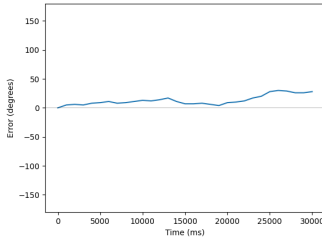
(b)



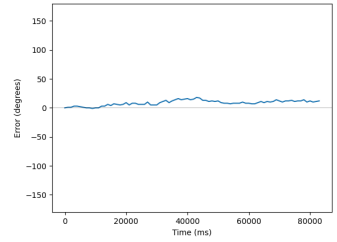
(c)



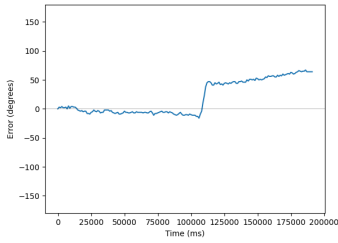
(d)



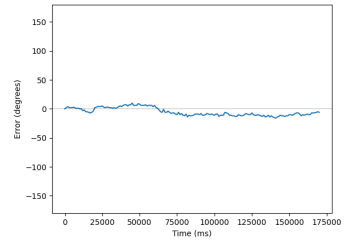
(e)



(f)

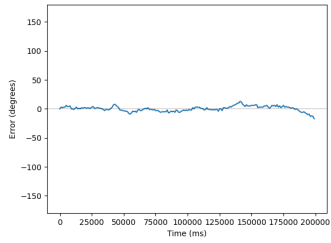


(g)

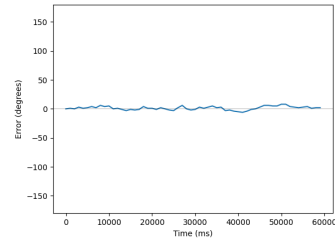


(h)

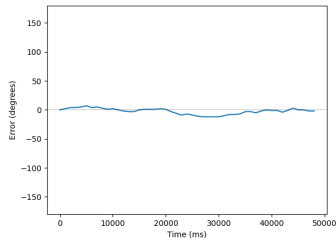
Figure A.18: Direction errors recorded in the Head Direction network when run on the ECMD dataset.



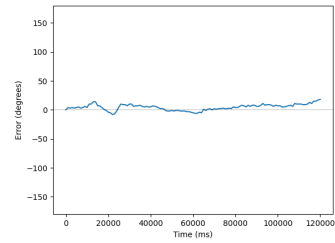
(a)



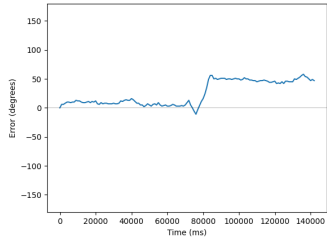
(b)



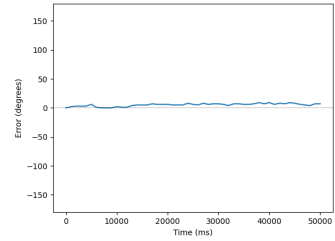
(c)



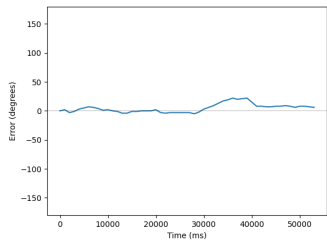
(d)



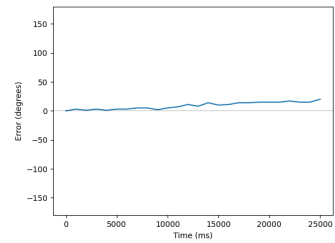
(e)



(f)

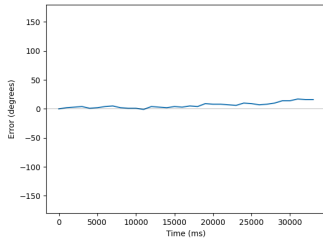


(g)

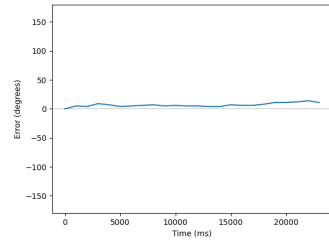


(h)

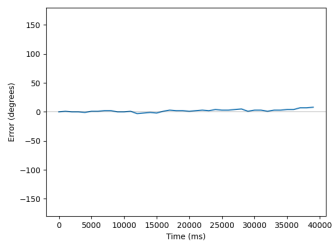
Figure A.19: Direction errors recorded in the Head Direction network when run on the ECMD dataset.



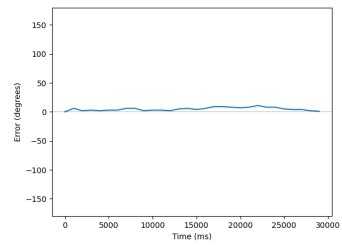
(a)



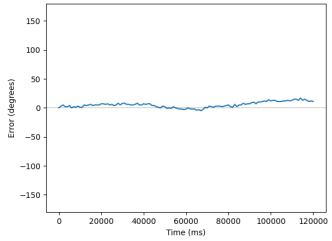
(b)



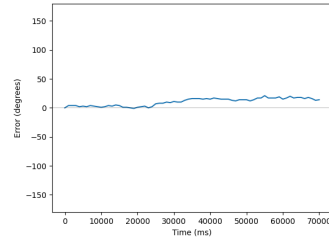
(c)



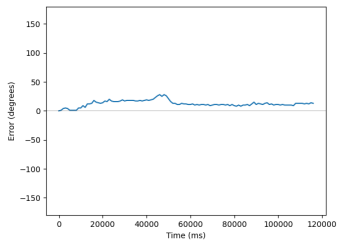
(d)



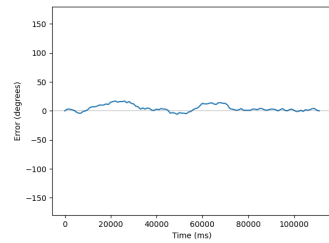
(e)



(f)

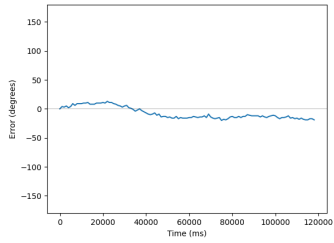


(g)

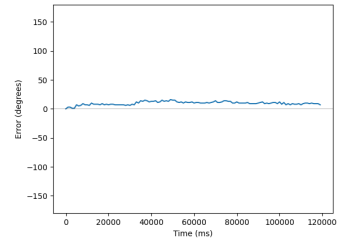


(h)

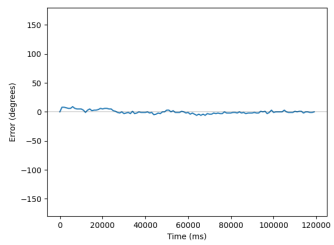
Figure A.20: Direction errors recorded in the Head Direction network when run on the ECMD dataset.



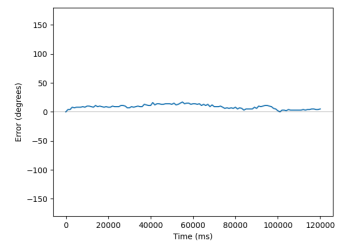
(a)



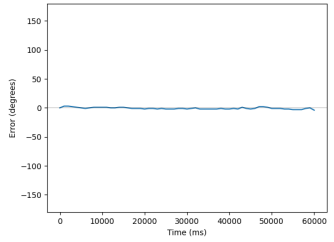
(b)



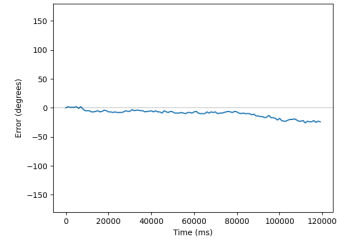
(c)



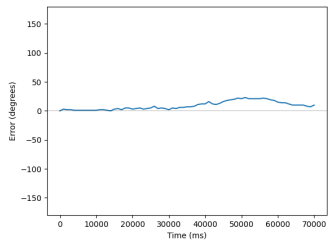
(d)



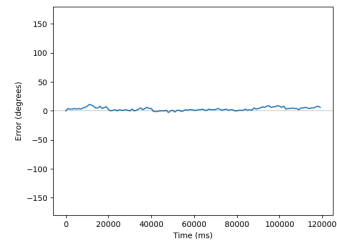
(e)



(f)

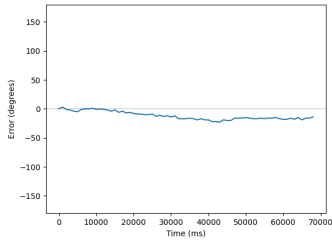


(g)

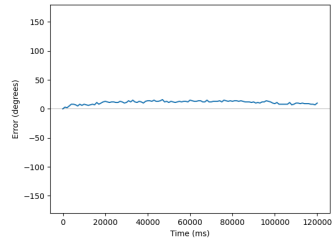


(h)

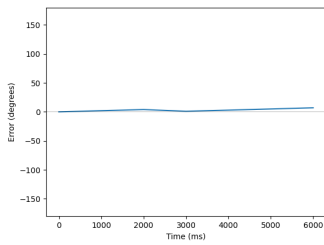
Figure A.21: Direction errors recorded in the Head Direction network when run on the ECMD dataset.



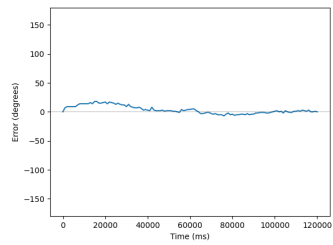
(a)



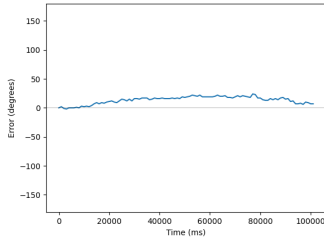
(b)



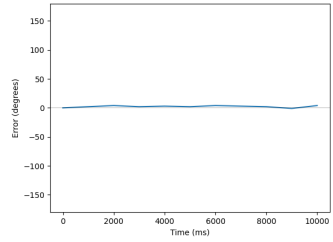
(c)



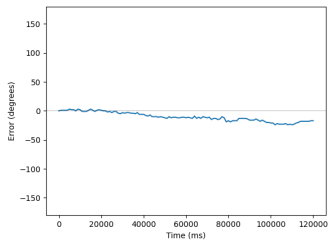
(d)



(e)



(f)



(g)

Figure A.22: Direction errors recorded in the Head Direction network when run on the ECMD dataset.

A.4 Artificial Grid Signals

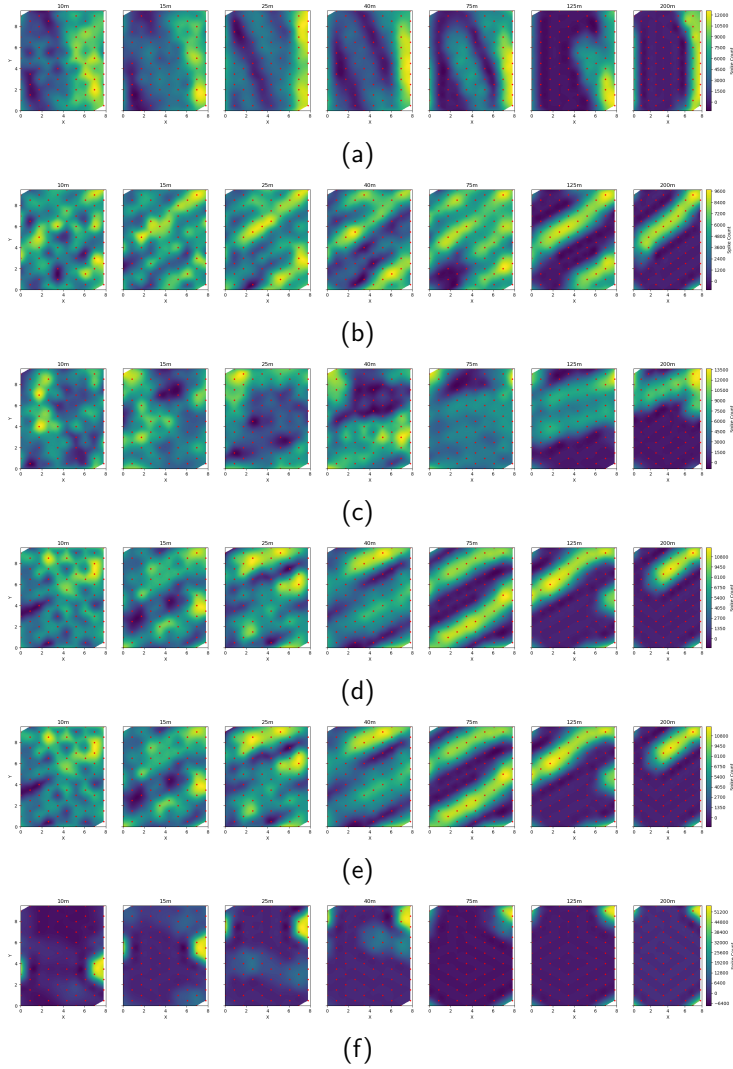


Figure A.23: The artificial grid signals developed from ground truth positions.

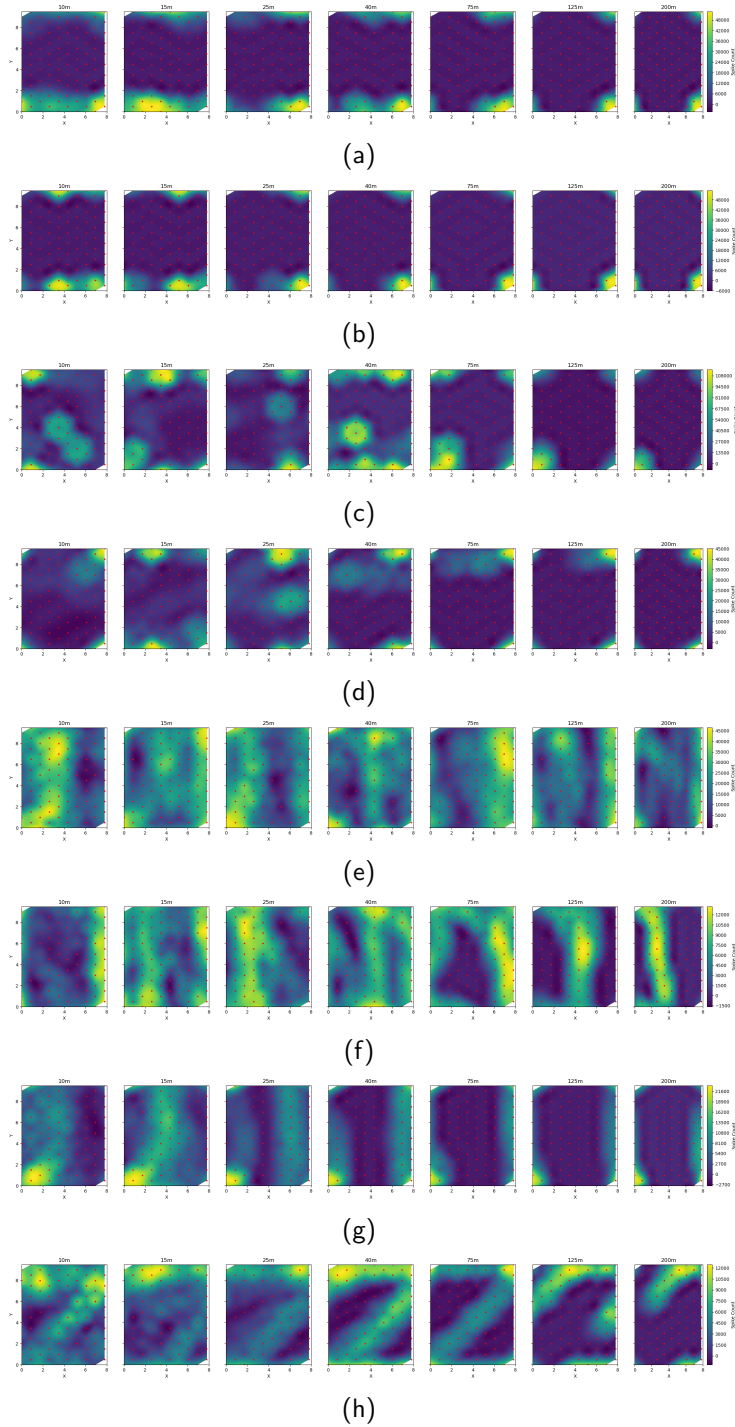


Figure A.24: The artificial grid signals developed from ground truth positions.

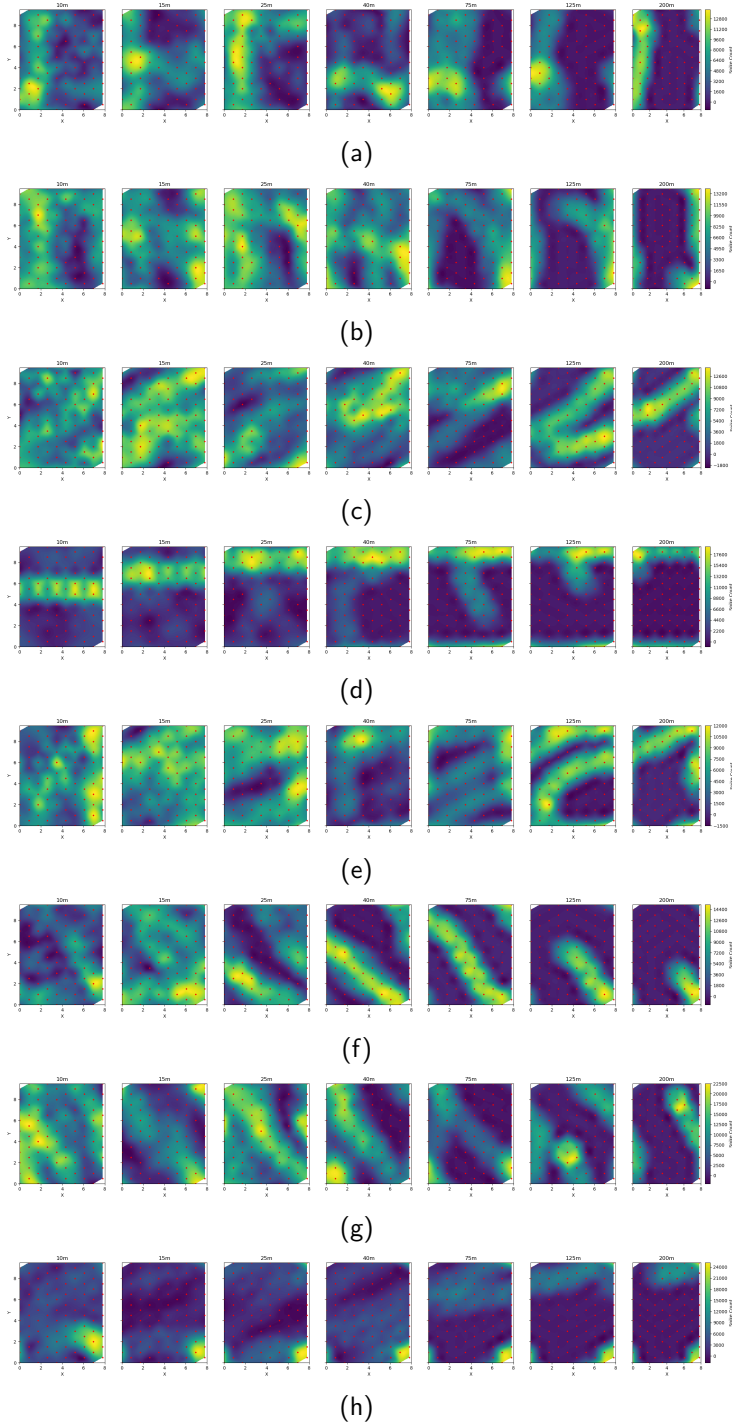


Figure A.25: The artificial grid signals developed from ground truth positions.

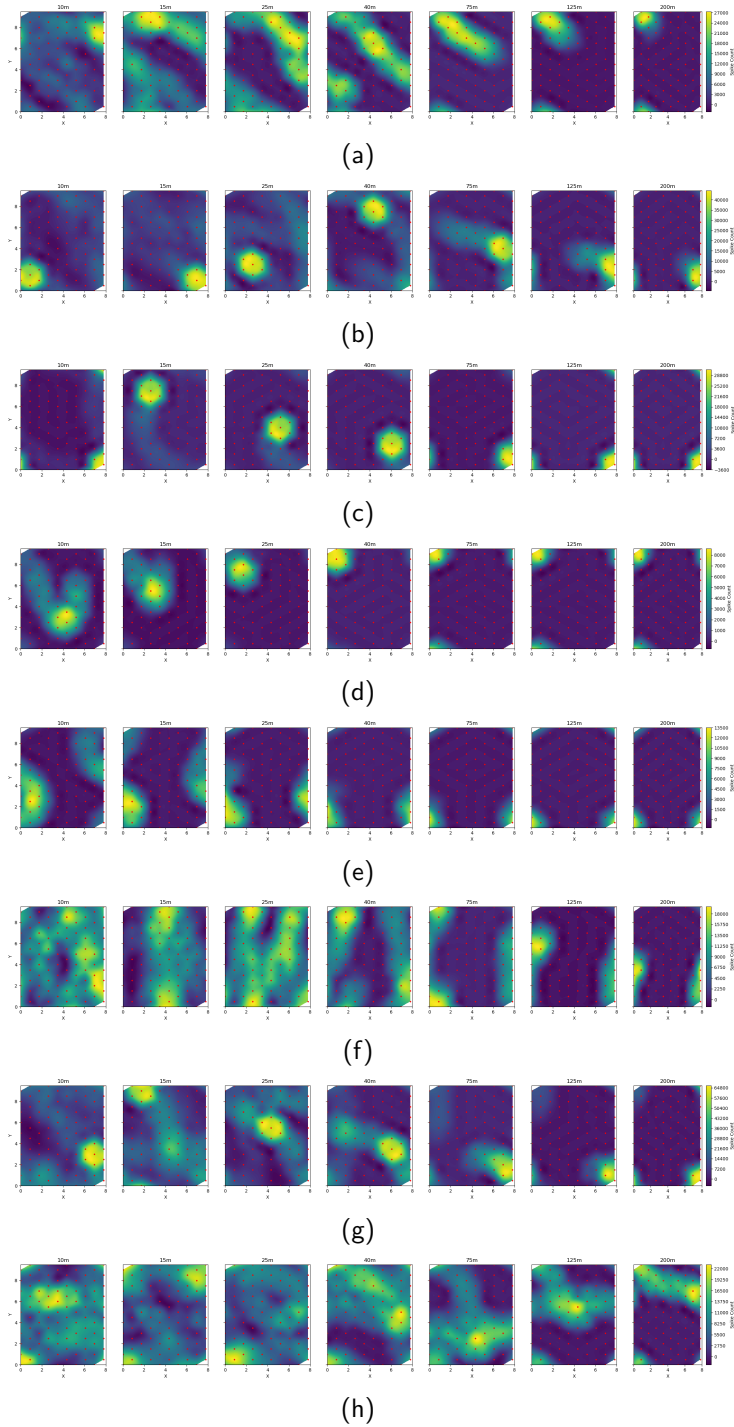


Figure A.26: The artificial grid signals developed from ground truth positions.

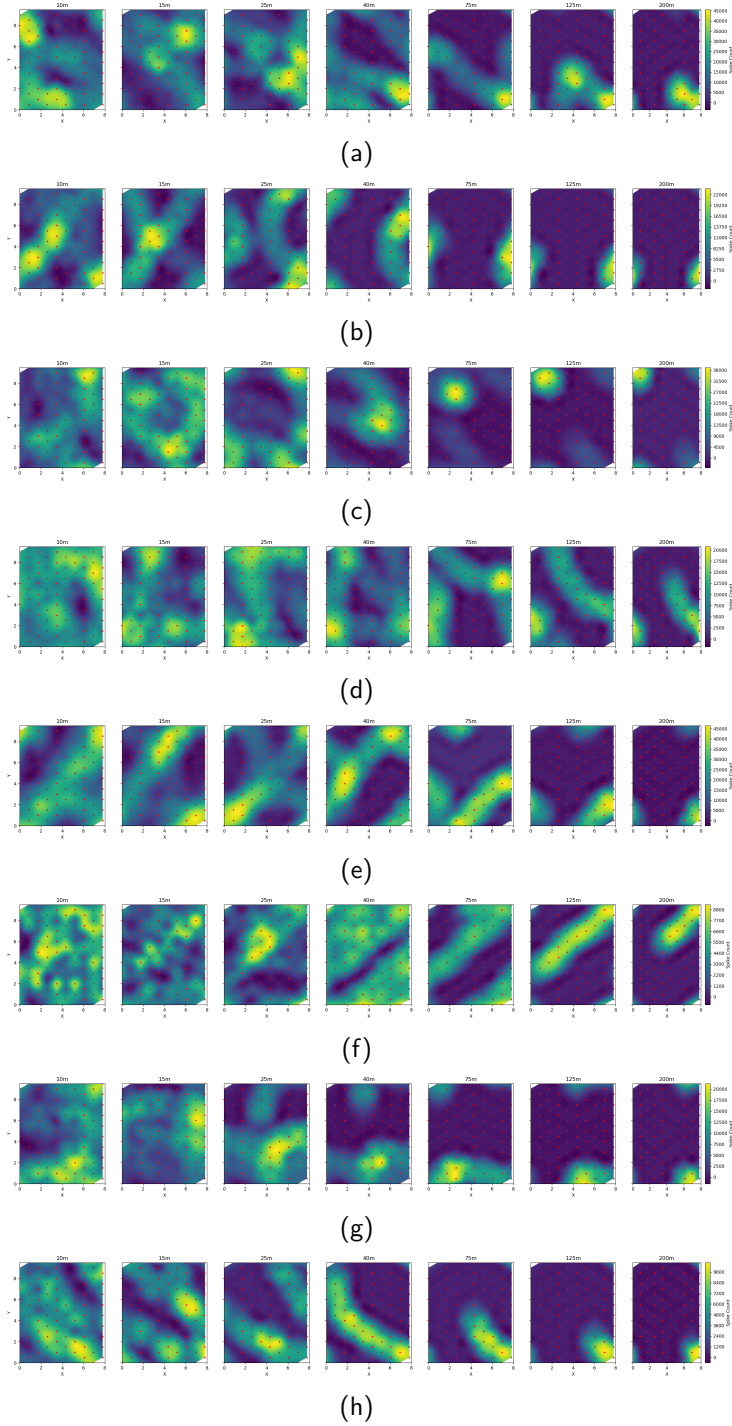


Figure A.27: The artificial grid signals developed from ground truth positions.

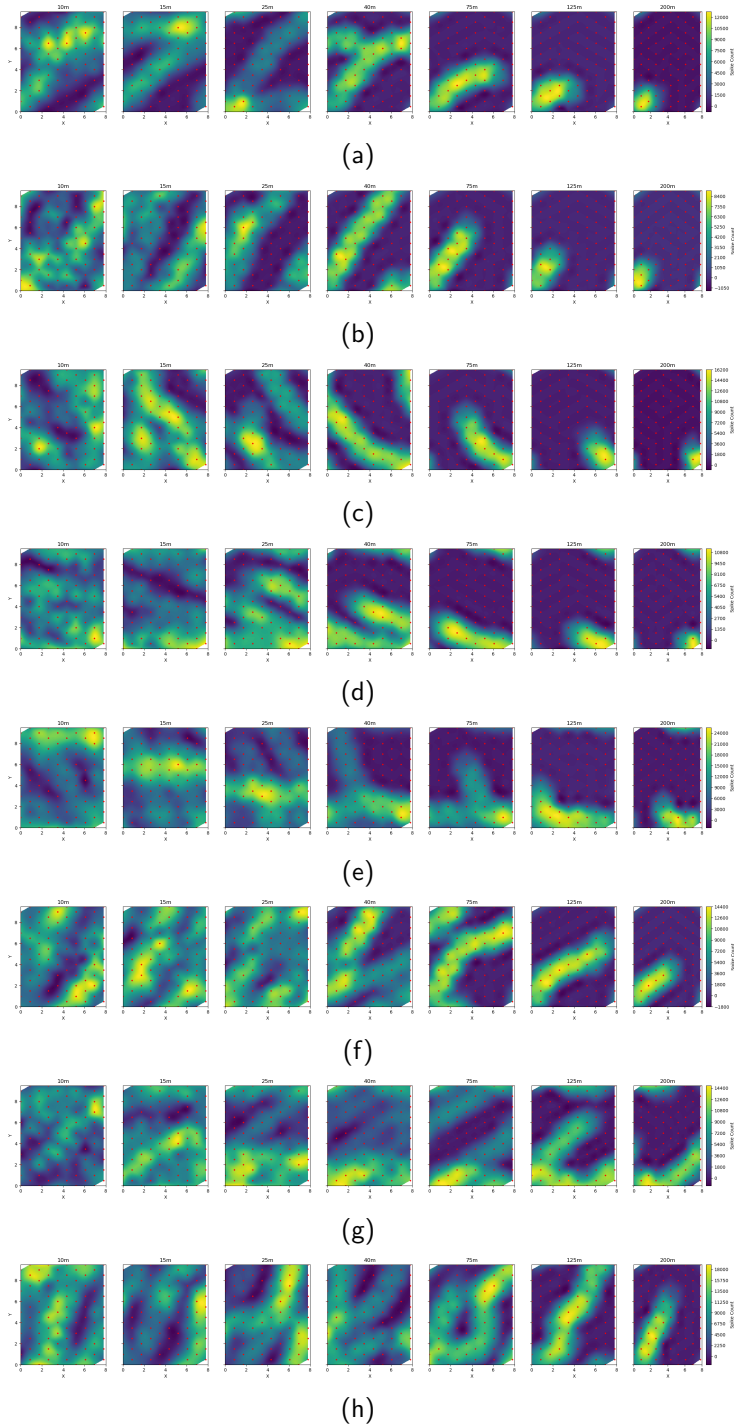


Figure A.28: The artificial grid signals developed from ground truth positions.

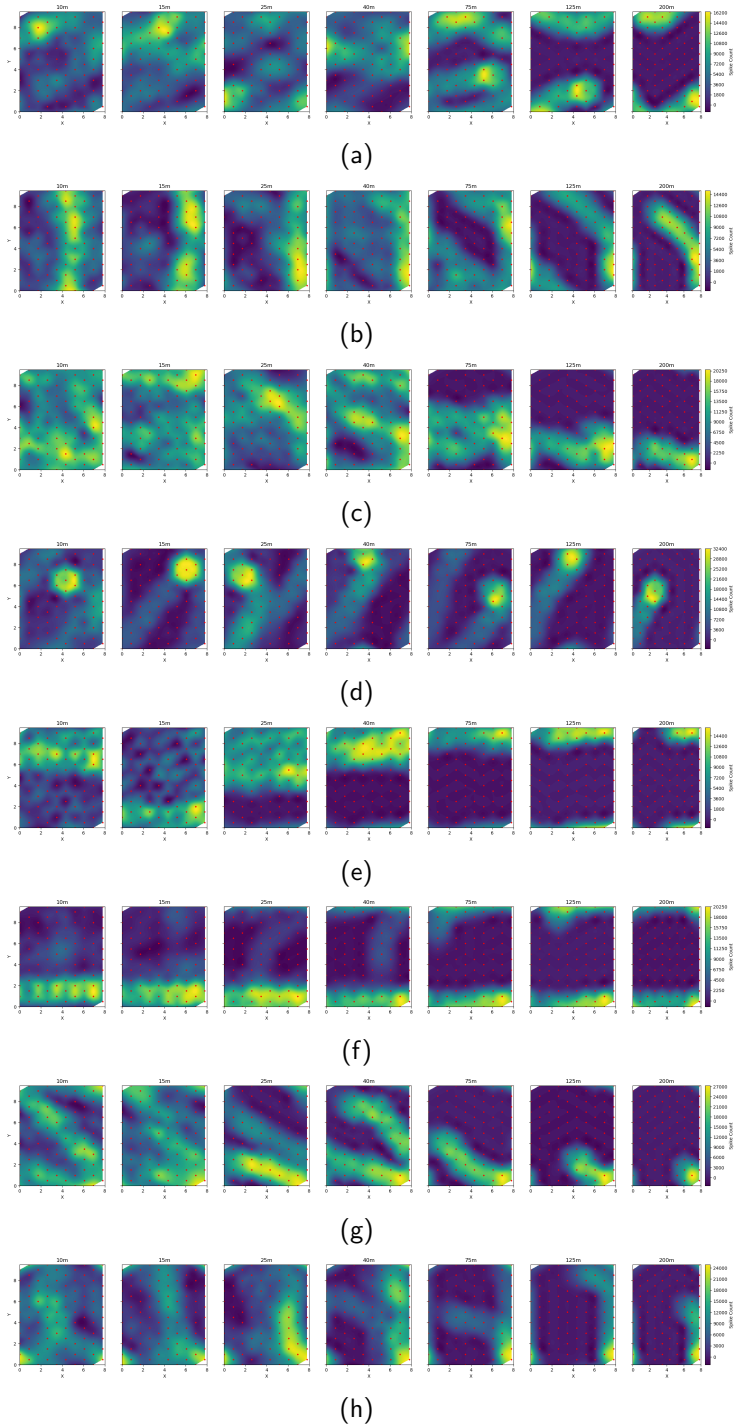


Figure A.29: The artificial grid signals developed from ground truth positions.

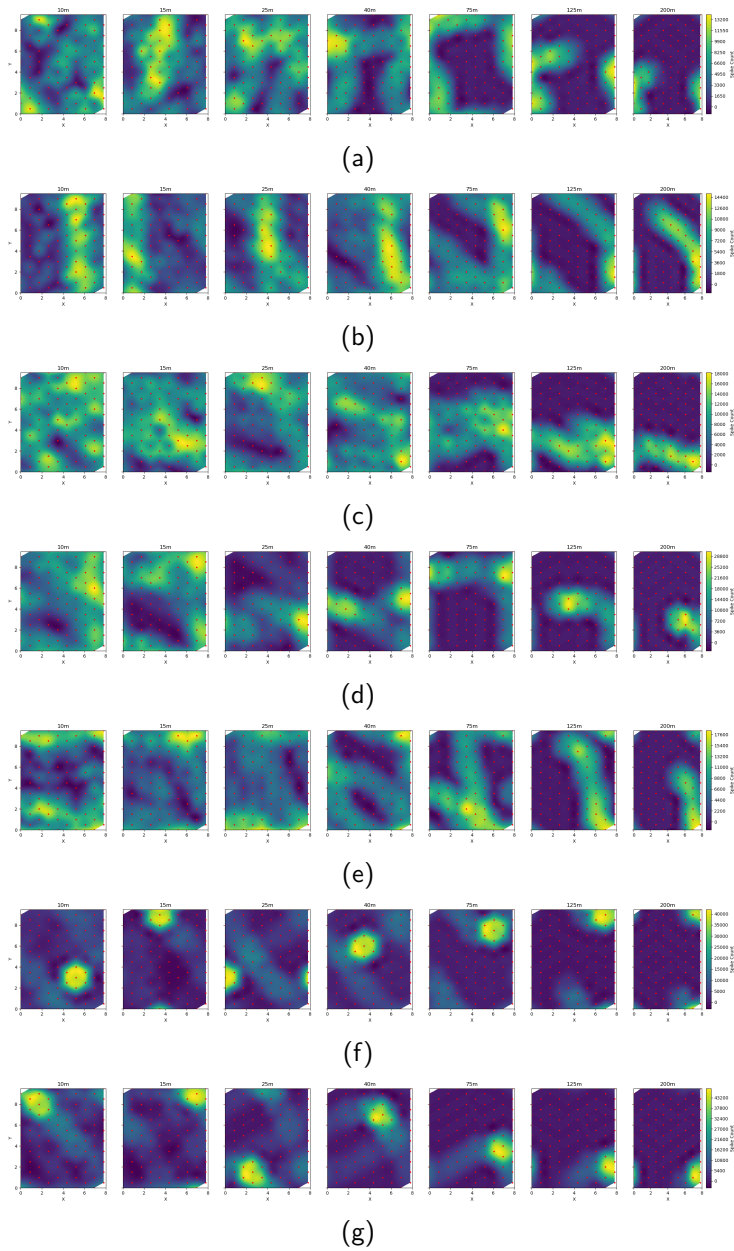


Figure A.30: The artificial grid signals developed from ground truth positions.

A.5 Place Cell Network Activity

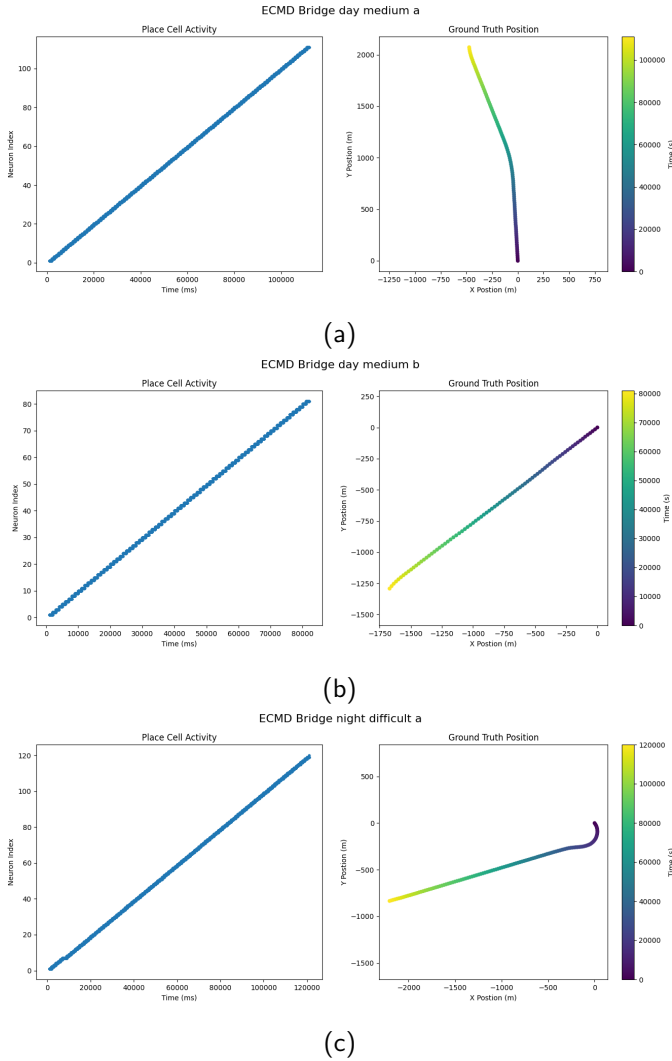
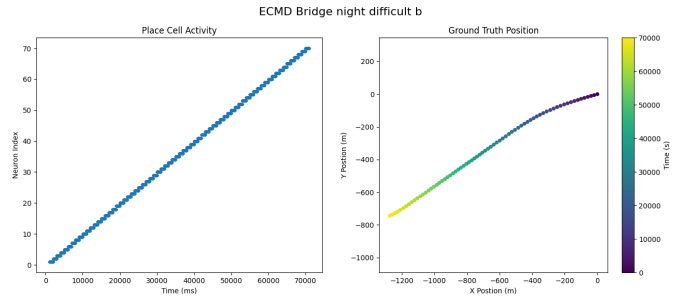
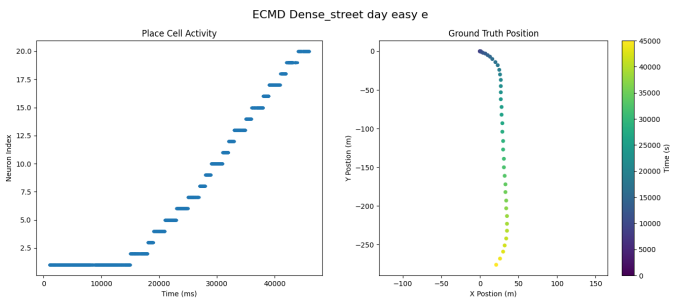


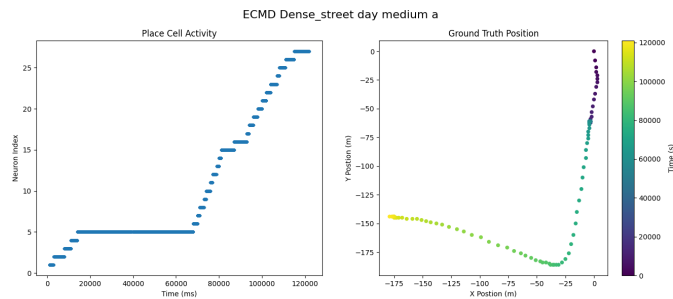
Figure A.31: Place Cell network activity accompanied by the corresponding ground truth trajectory.



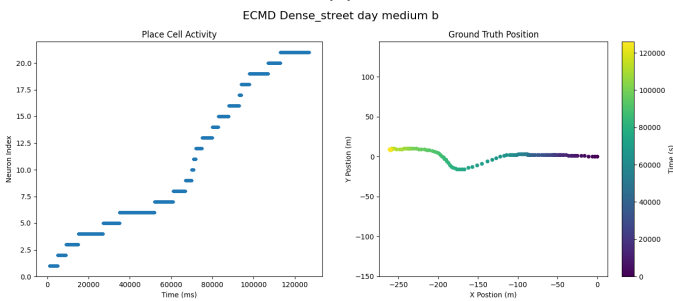
(a)



(b)

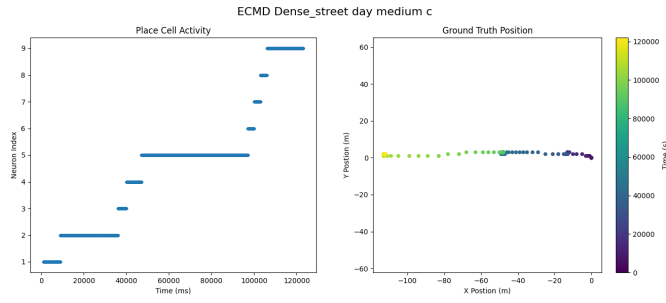


(c)

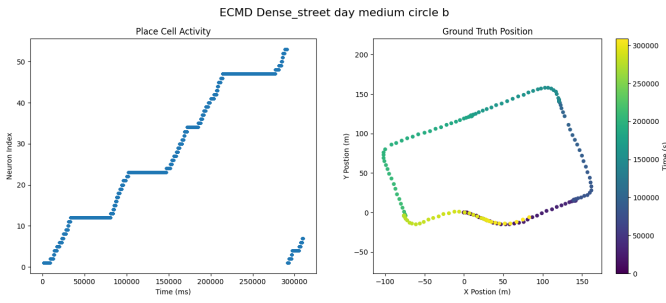


(d)

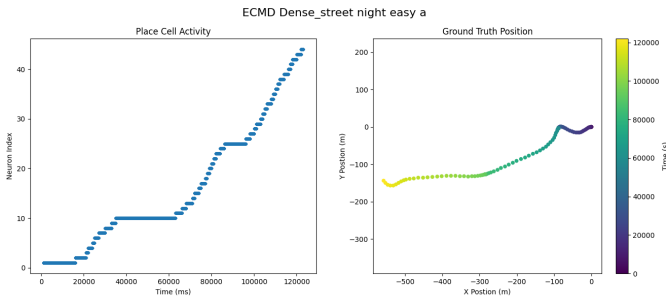
Figure A.32: Place Cell network activity accompanied by the corresponding ground truth trajectory.



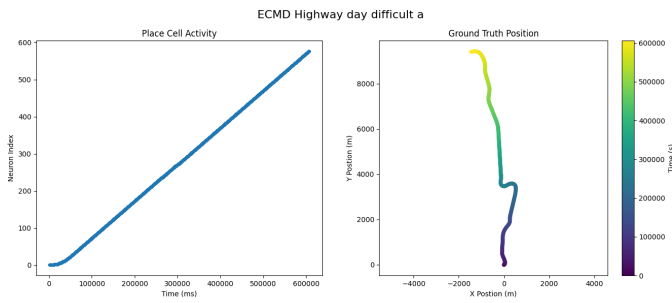
(a)



(b)

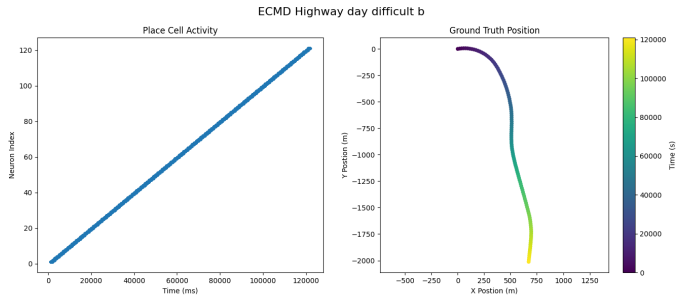


(c)

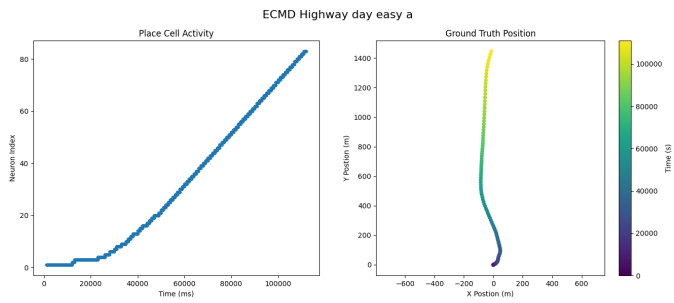


(d)

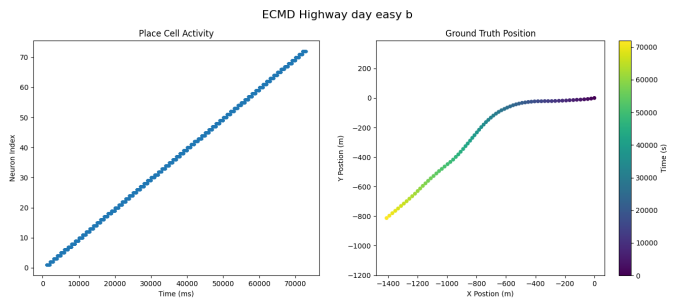
Figure A.33: Place Cell network activity accompanied by the corresponding ground truth trajectory.



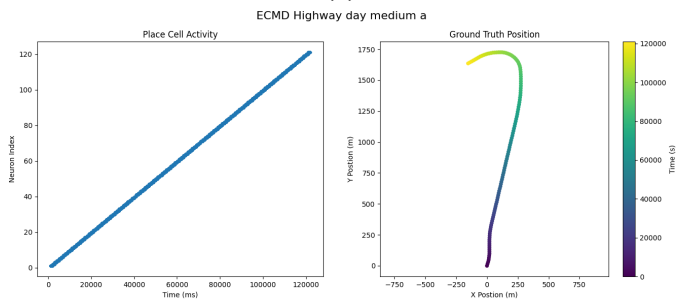
(a)



(b)

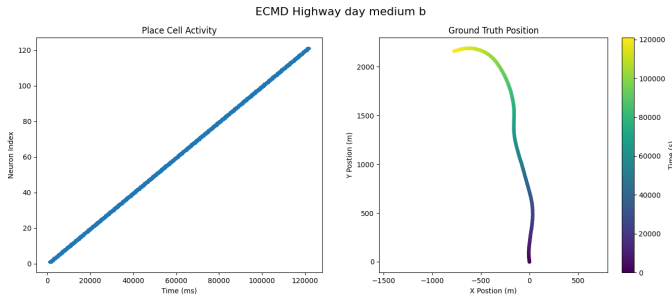


(c)

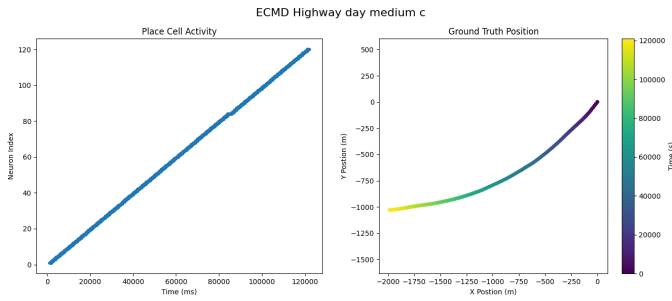


(d)

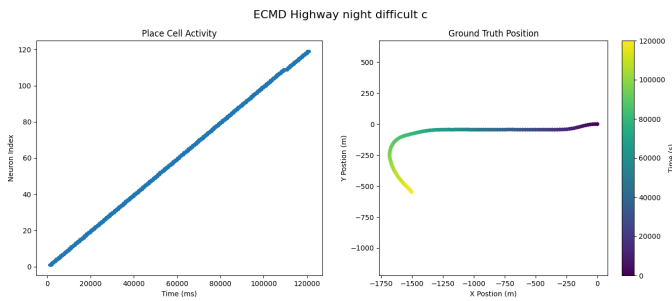
Figure A.34: Place Cell network activity accompanied by the corresponding ground truth trajectory.



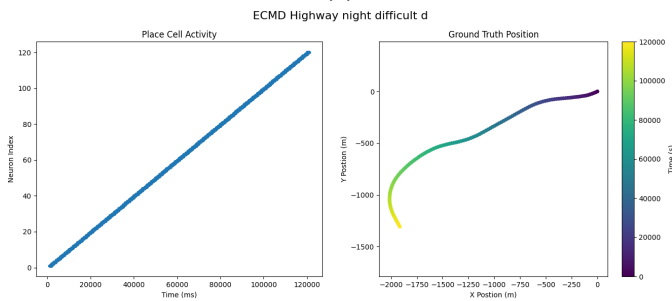
(a)



(b)



(c)



(d)

Figure A.35: Place Cell network activity accompanied by the corresponding ground truth trajectory.

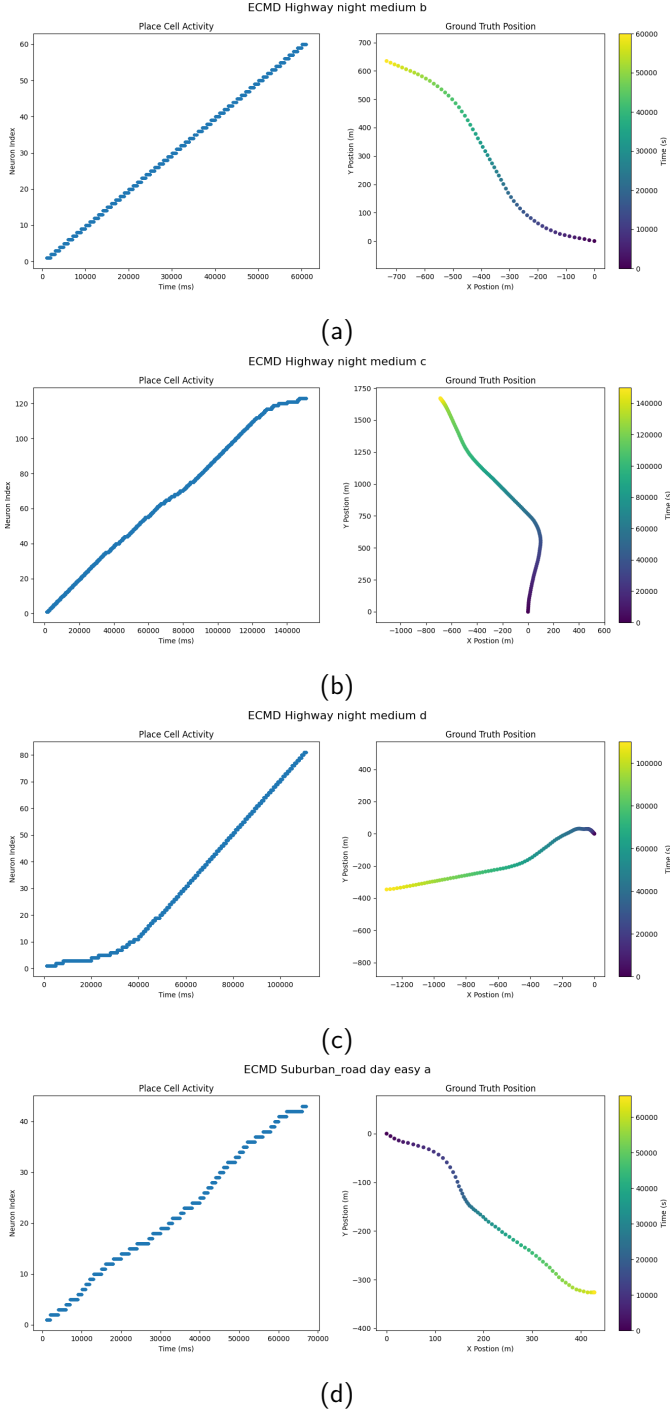


Figure A.36: Place Cell network activity accompanied by the corresponding ground truth trajectory.

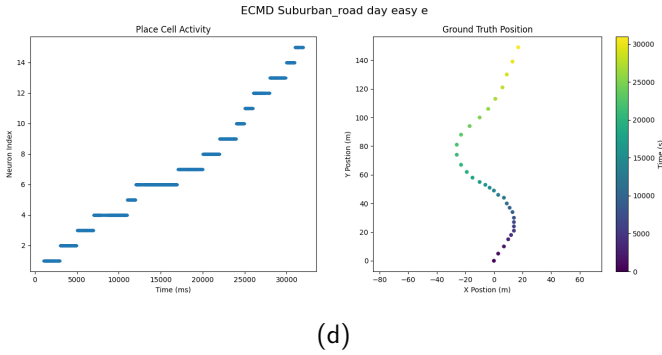
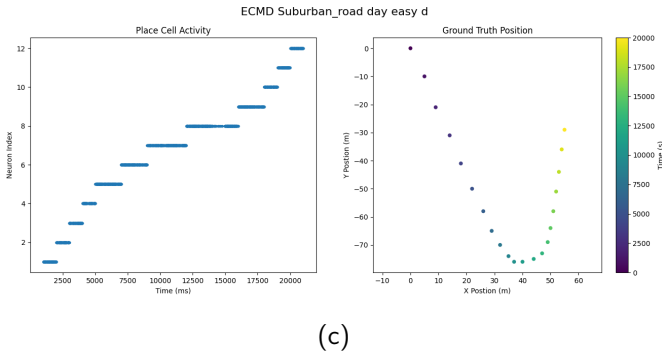
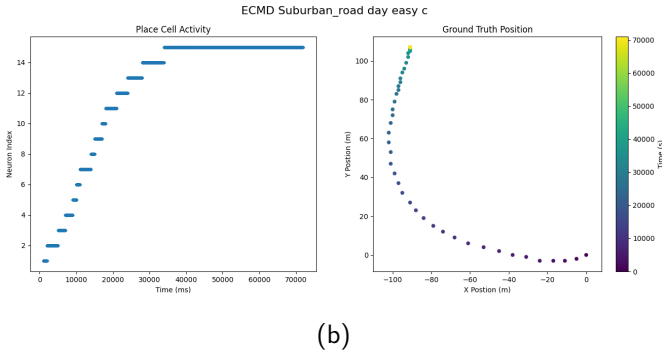
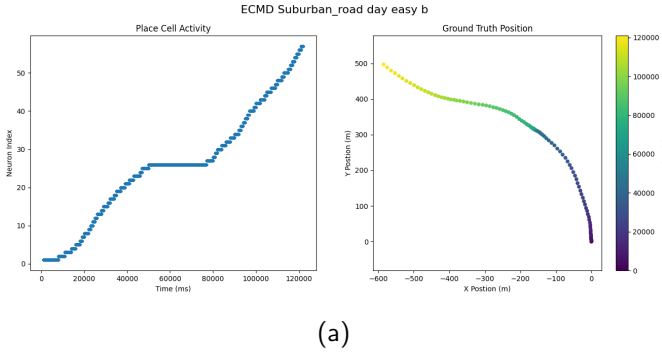
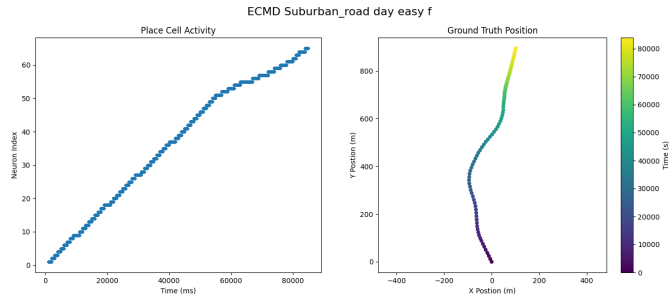
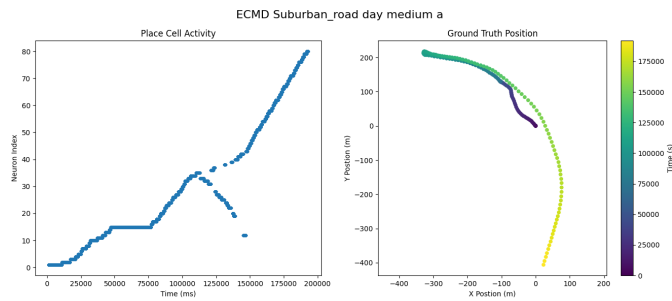


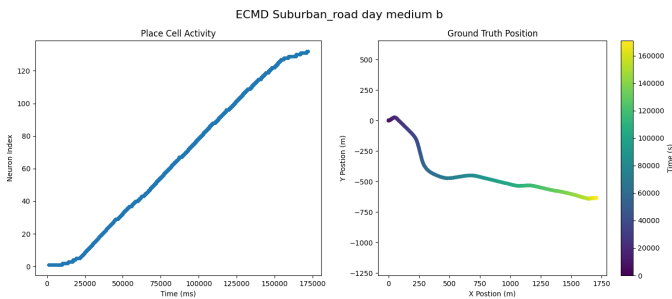
Figure A.37: Place Cell network activity accompanied by the corresponding ground truth trajectory.



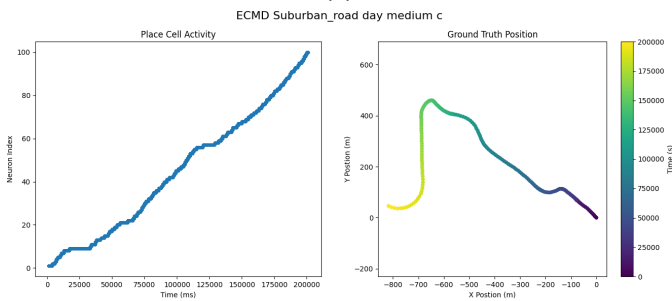
(a)



(b)

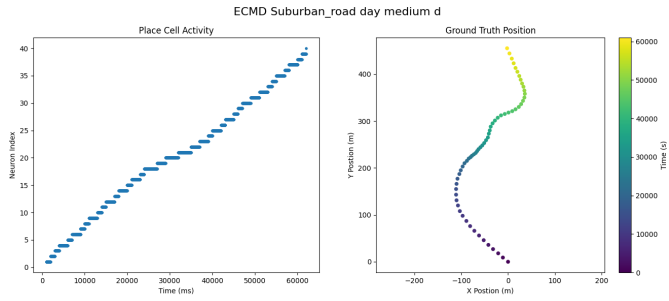


(c)

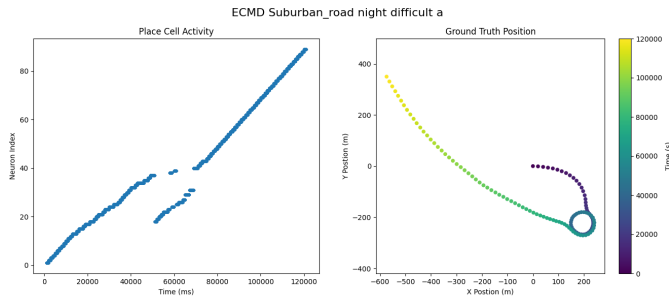


(d)

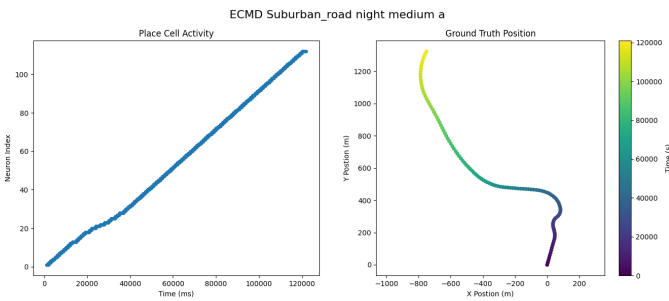
Figure A.38: Place Cell network activity accompanied by the corresponding ground truth trajectory.



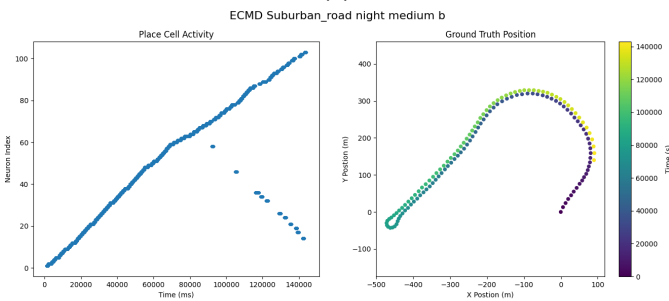
(a)



(b)



(c)



(d)

Figure A.39: Place Cell network activity accompanied by the corresponding ground truth trajectory.

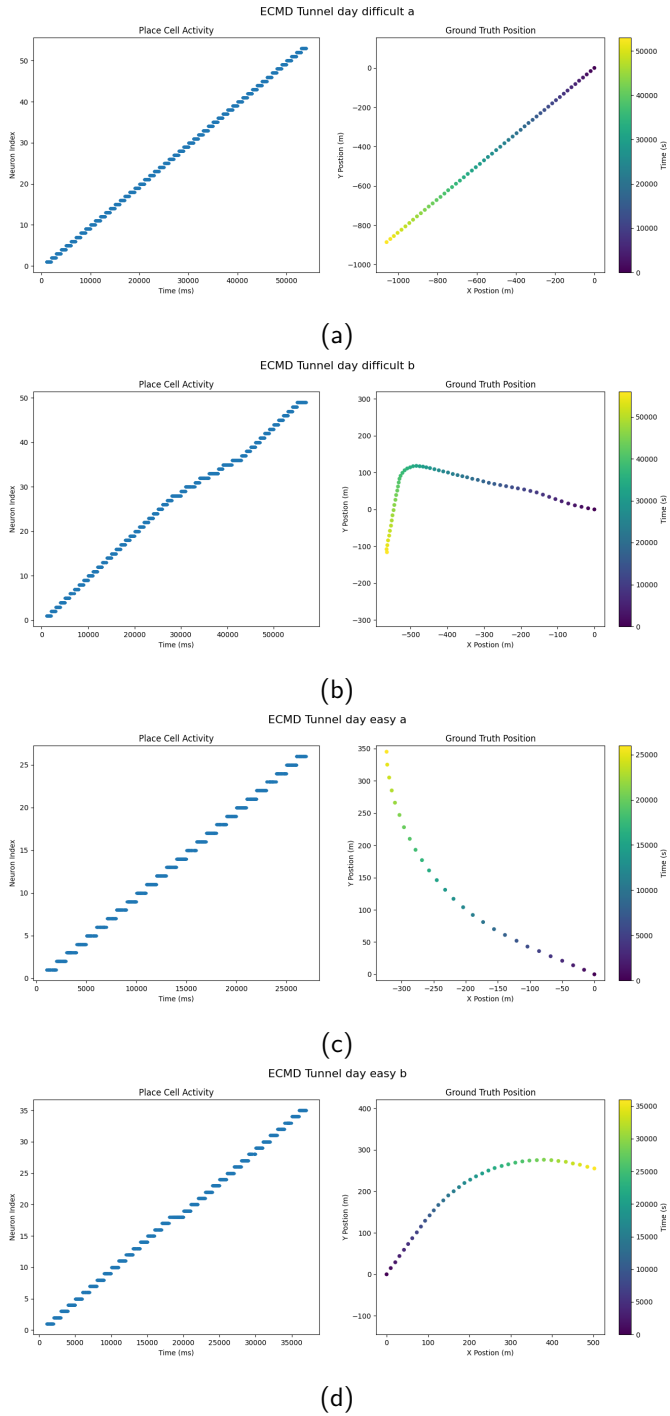
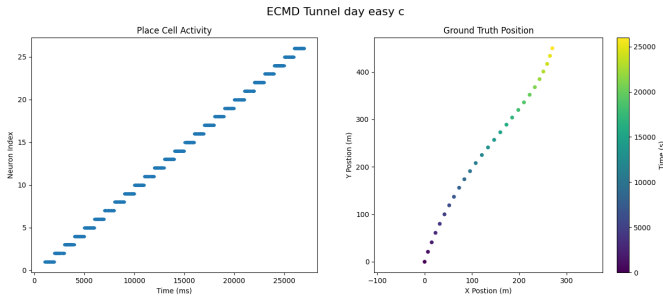
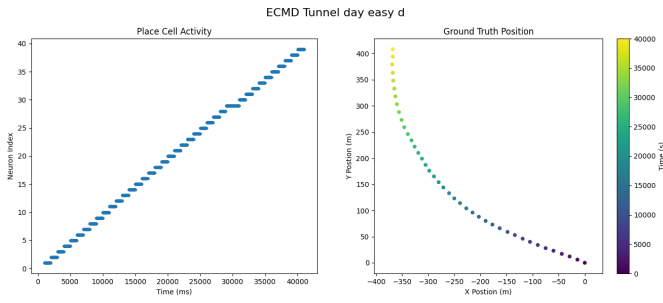


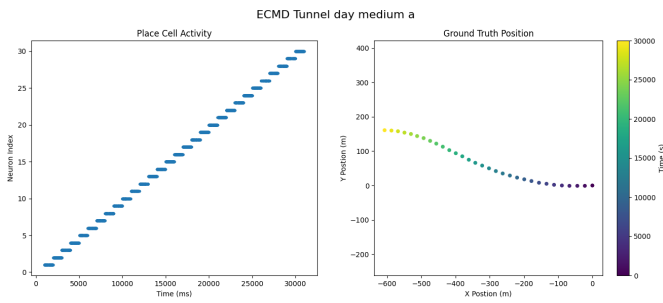
Figure A.40: Place Cell network activity accompanied by the corresponding ground truth trajectory.



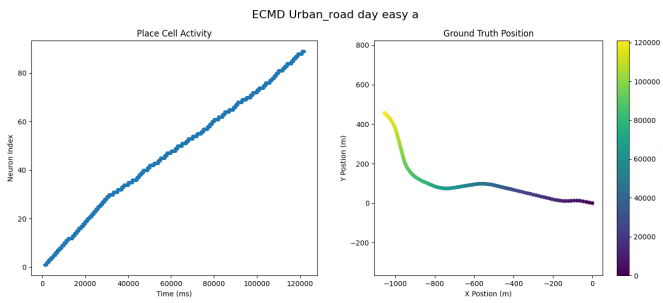
(a)



(b)

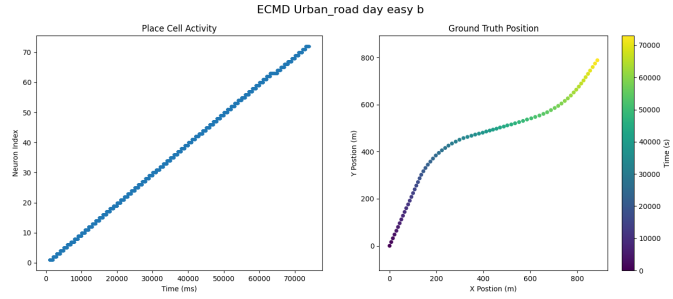


(c)

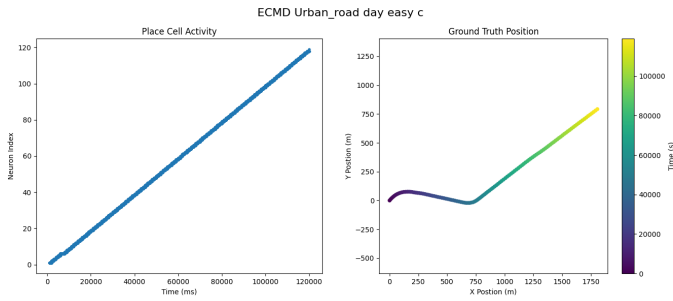


(d)

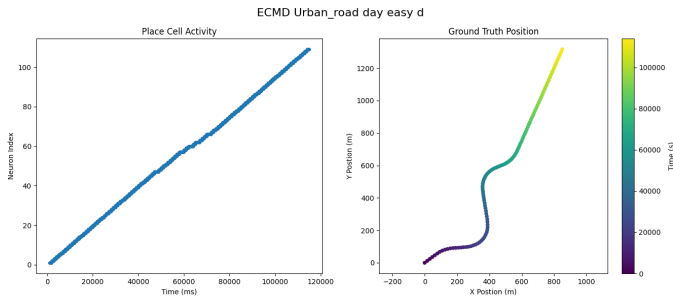
Figure A.41: Place Cell network activity accompanied by the corresponding ground truth trajectory.



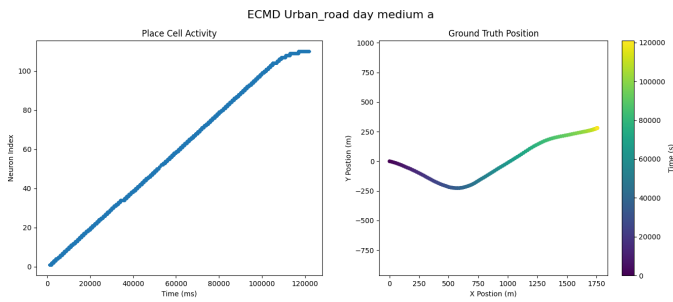
(a)



(b)

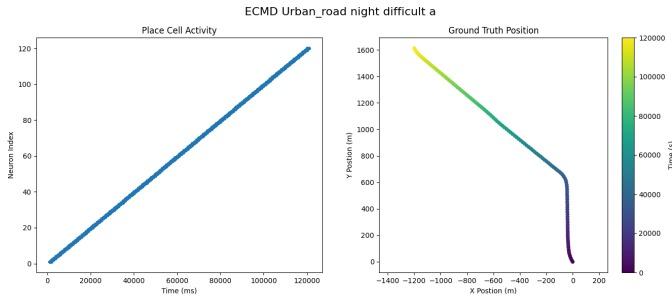


(c)

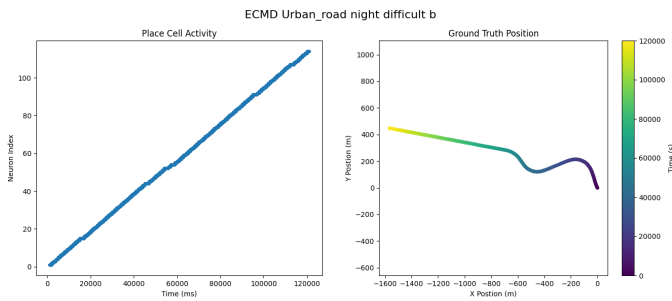


(d)

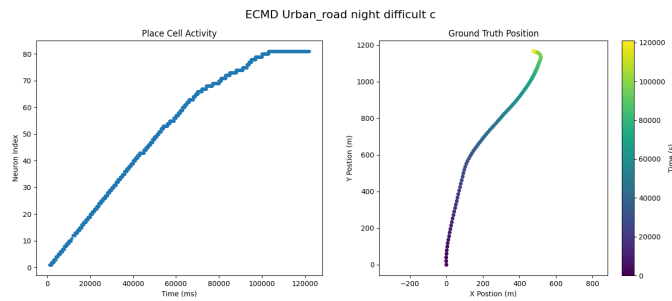
Figure A.42: Place Cell network activity accompanied by the corresponding ground truth trajectory.



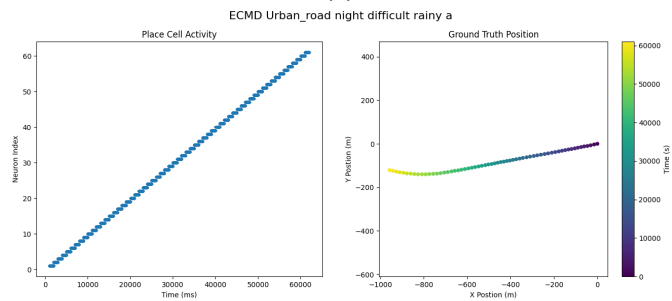
(a)



(b)

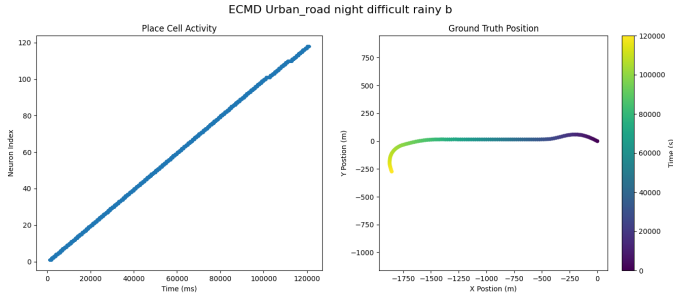


(c)

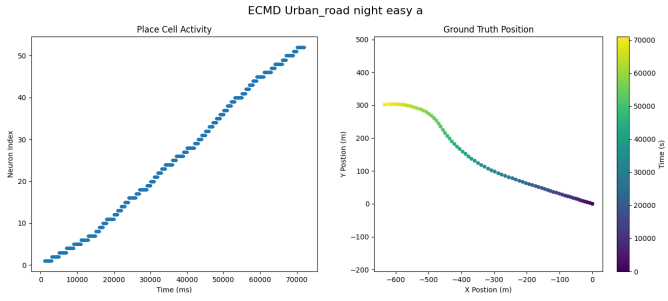


(d)

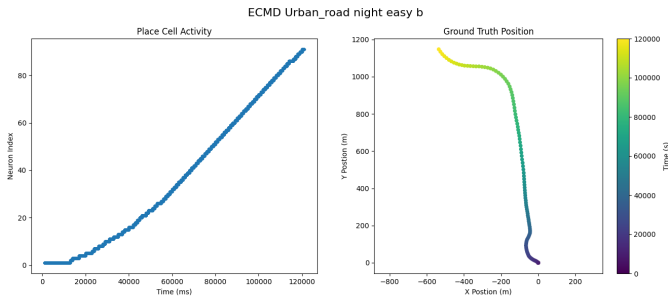
Figure A.43: Place Cell network activity accompanied by the corresponding ground truth trajectory.



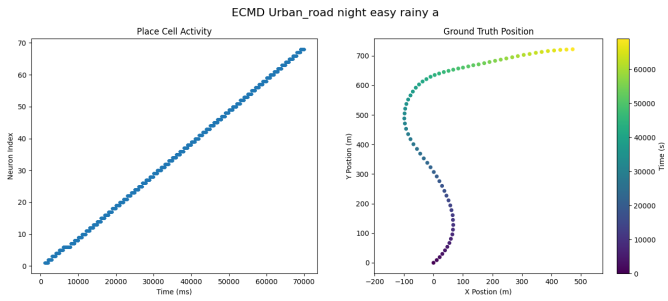
(a)



(b)

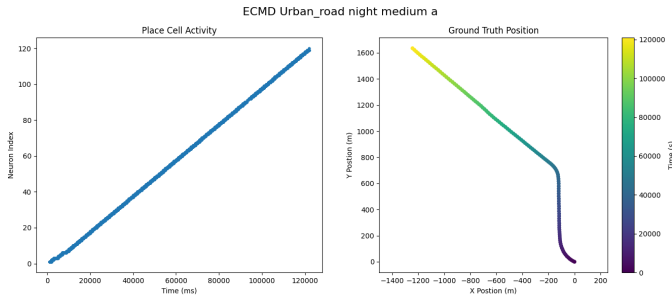


(c)

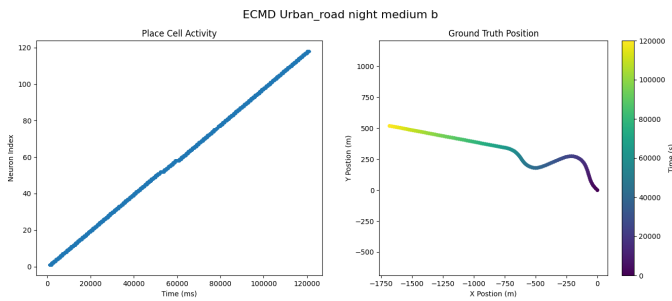


(d)

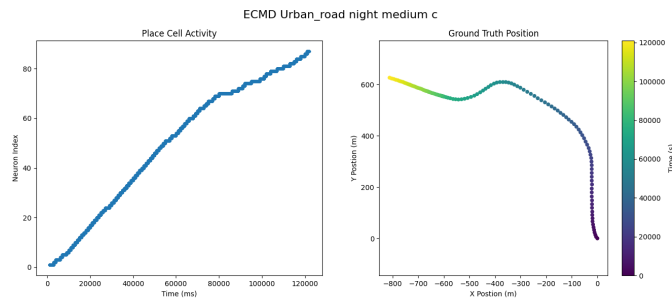
Figure A.44: Place Cell network activity accompanied by the corresponding ground truth trajectory.



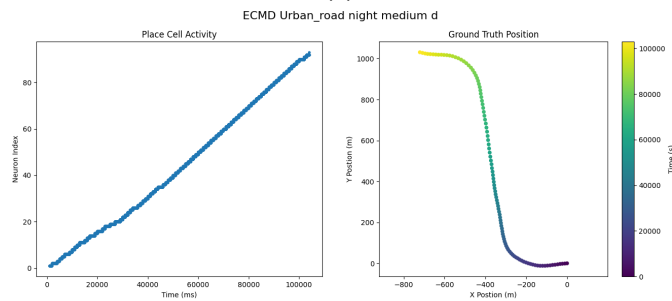
(a)



(b)



(c)



(d)

Figure A.45: Place Cell network activity accompanied by the corresponding ground truth trajectory.

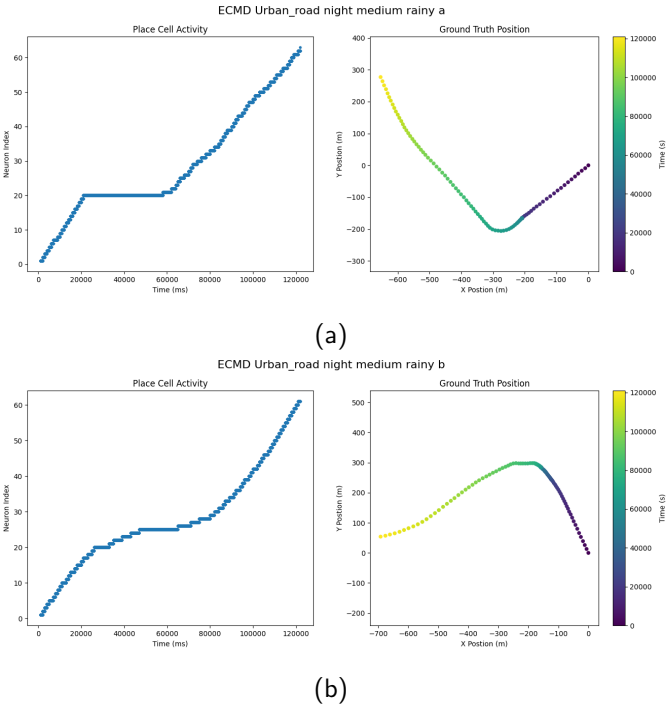


Figure A.46: Place Cell network activity accompanied by the corresponding ground truth trajectory.

A.6 Visual Place Recognition Network Reset Signals

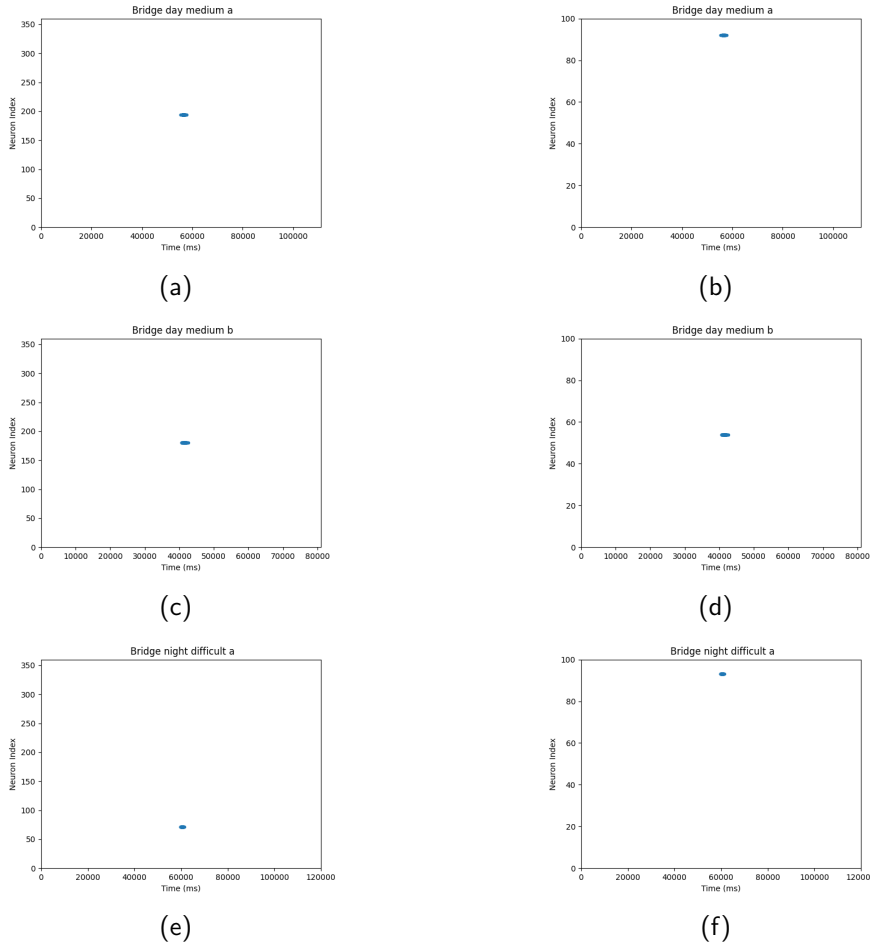


Figure A.47: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

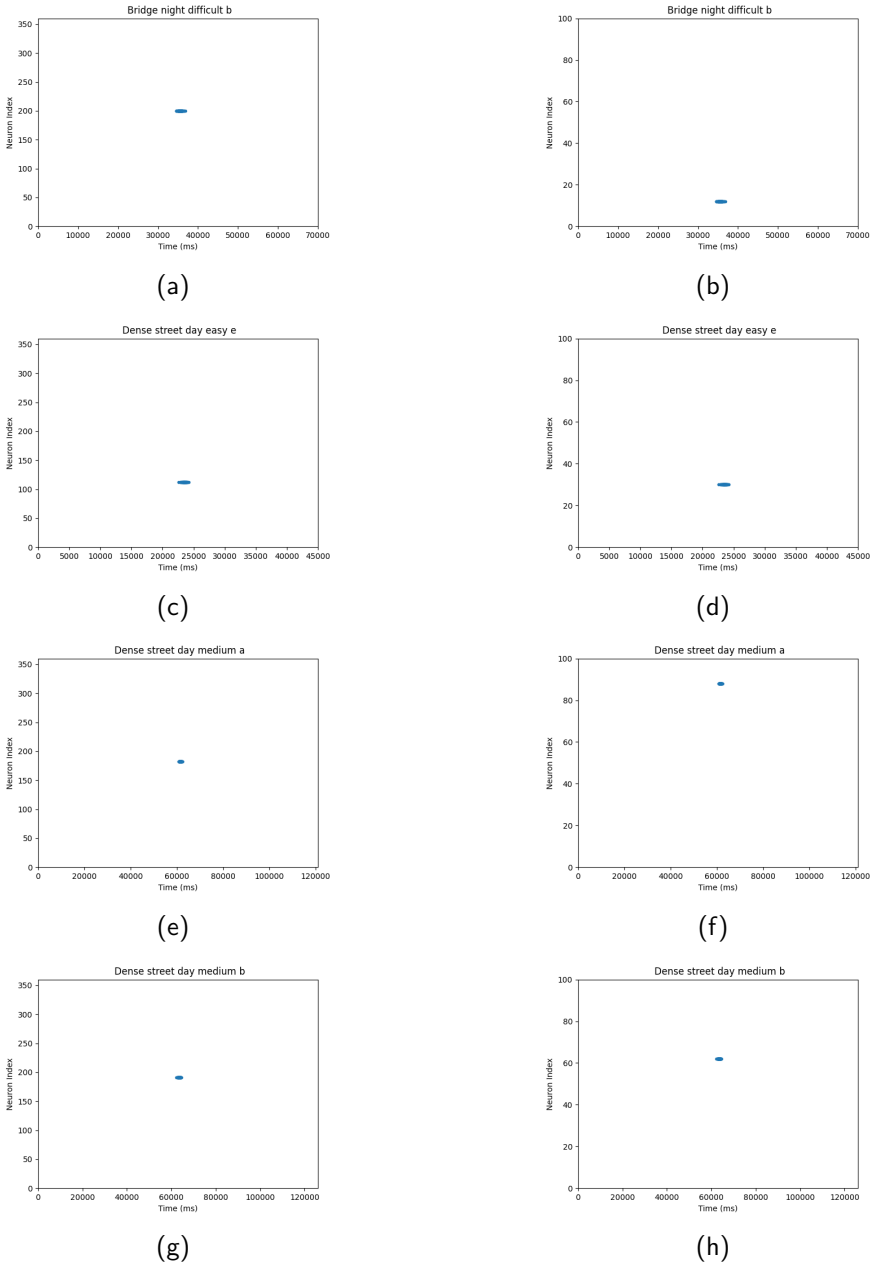


Figure A.48: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

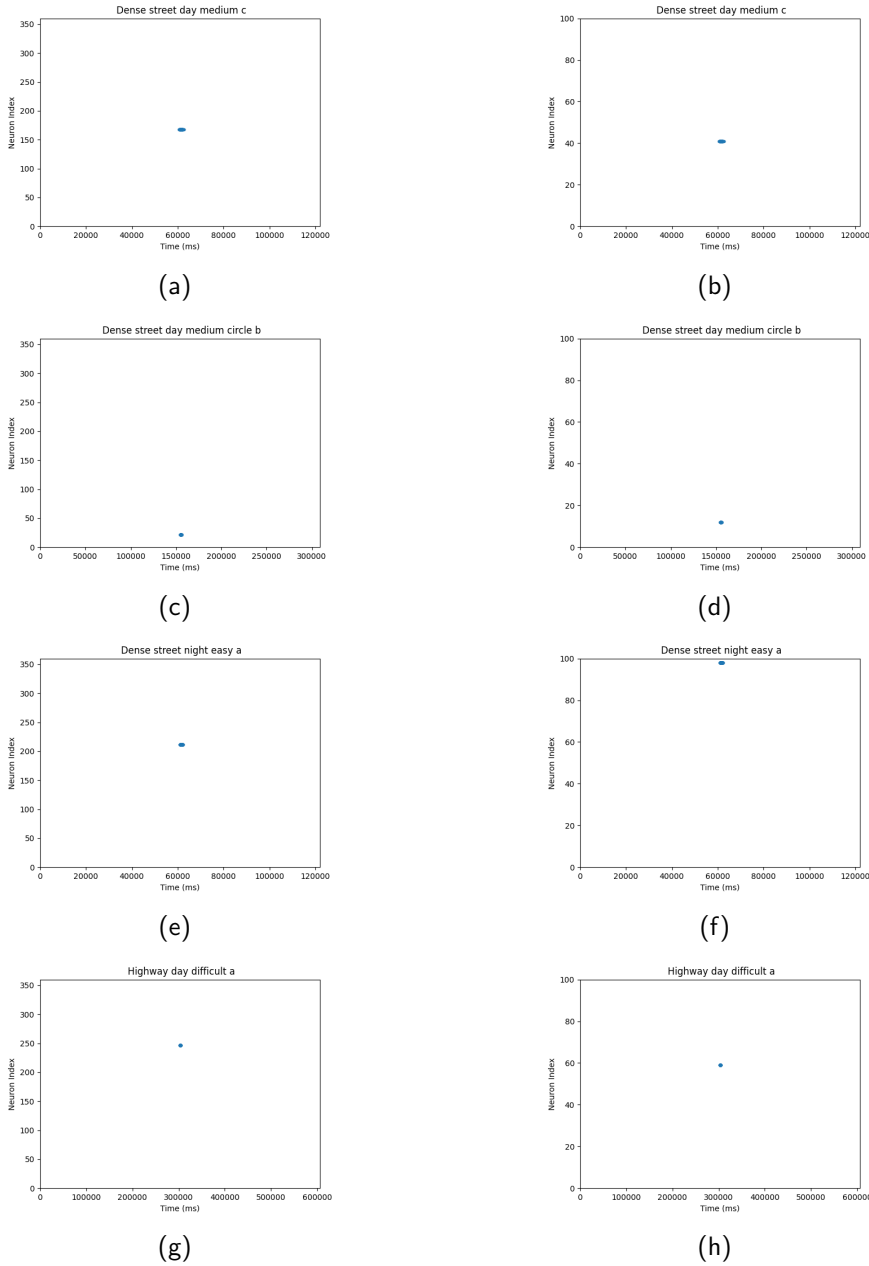


Figure A.49: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

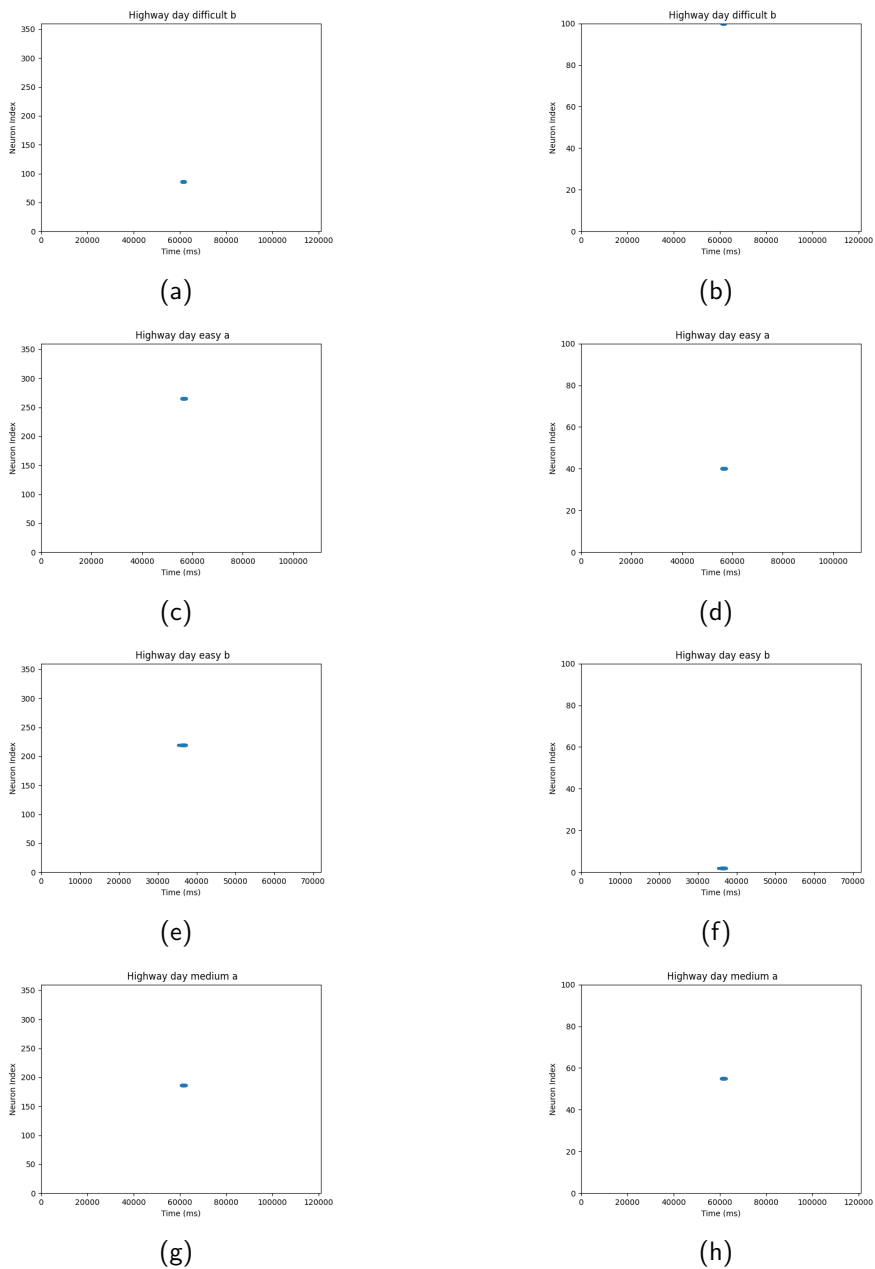


Figure A.50: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

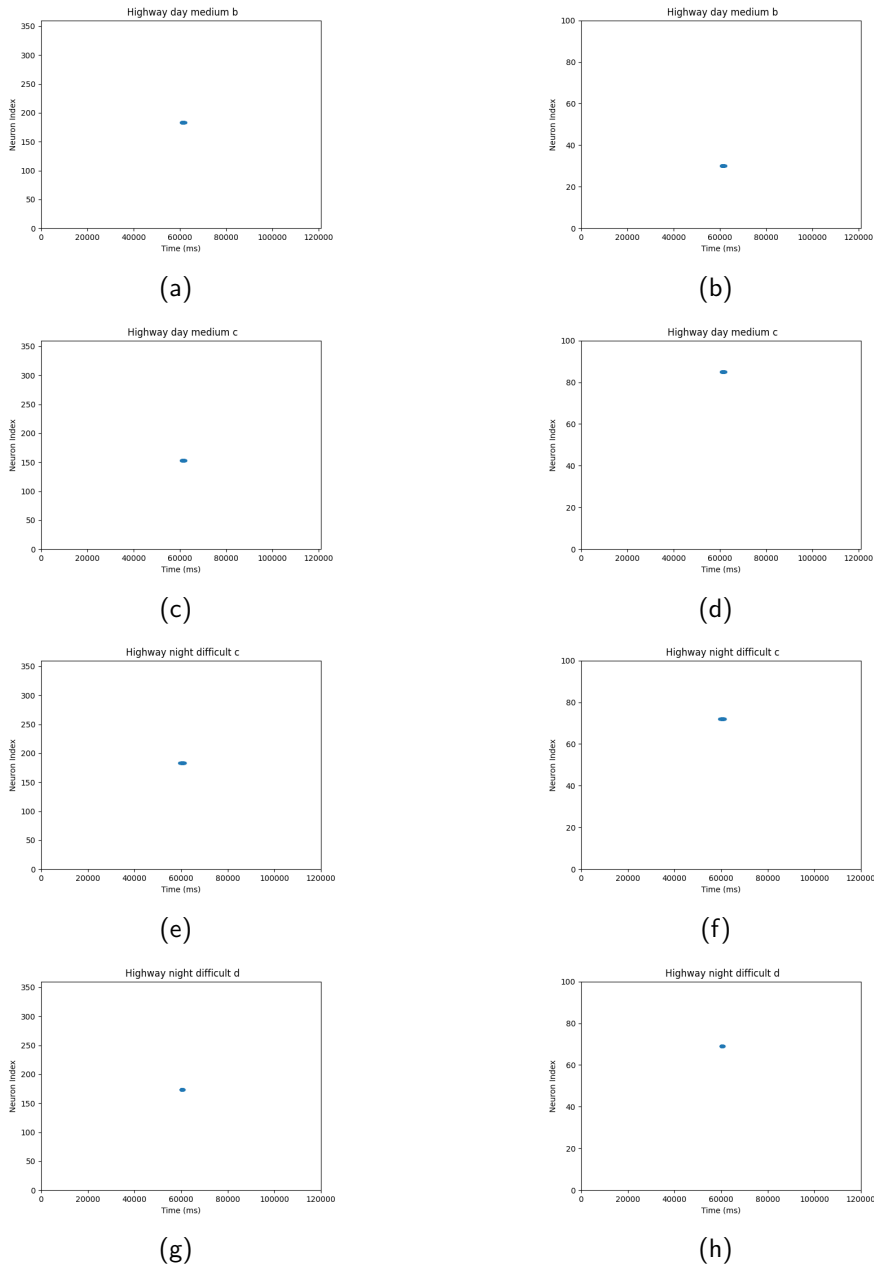


Figure A.51: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

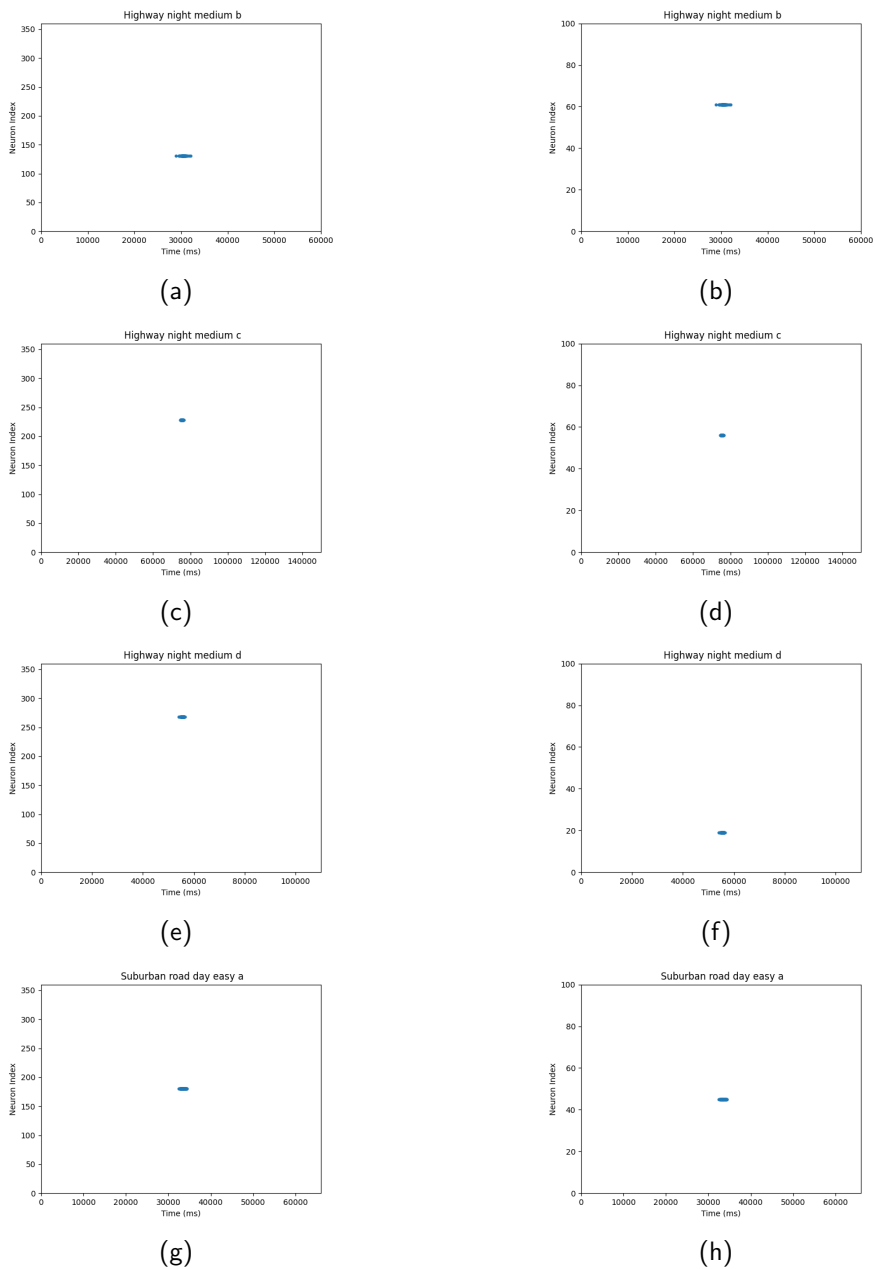


Figure A.52: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

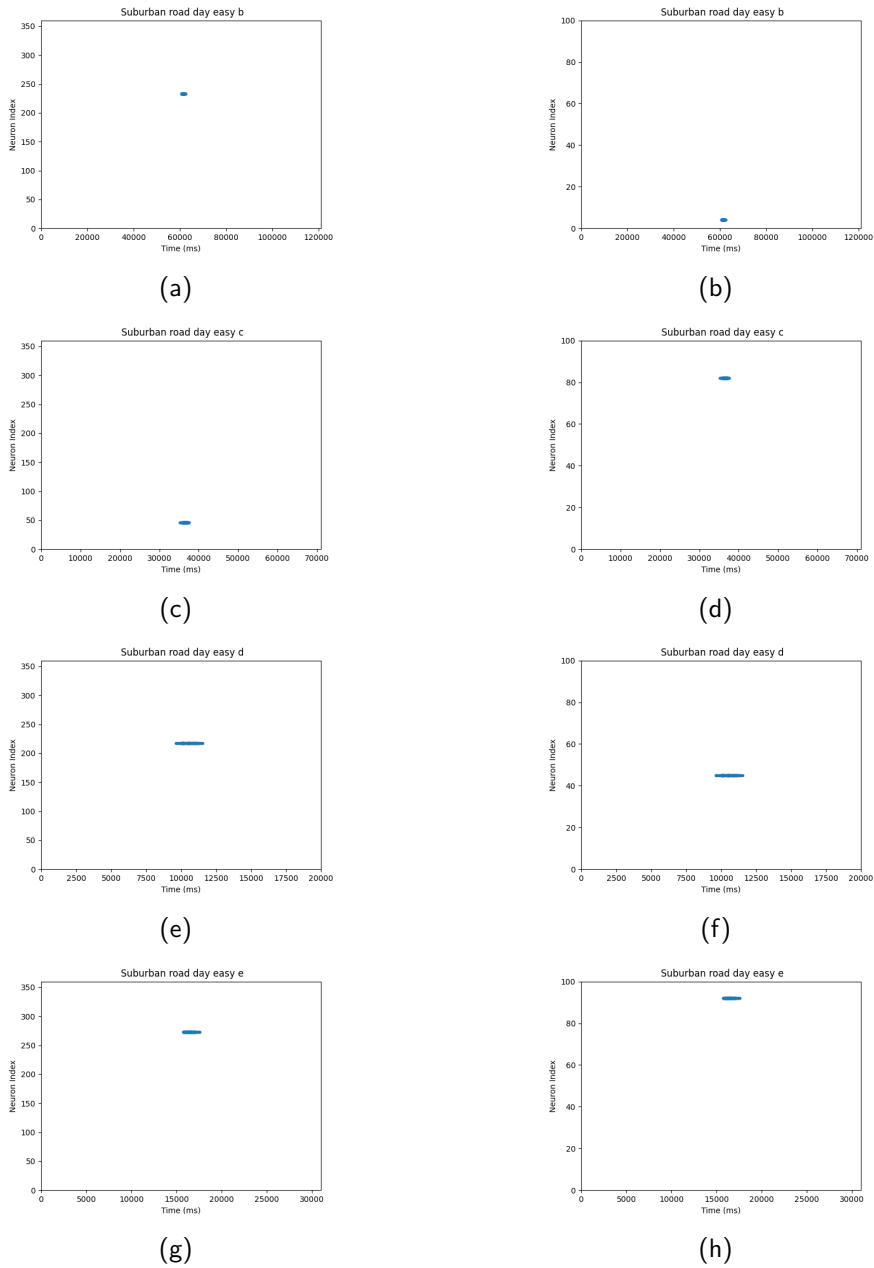


Figure A.53: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

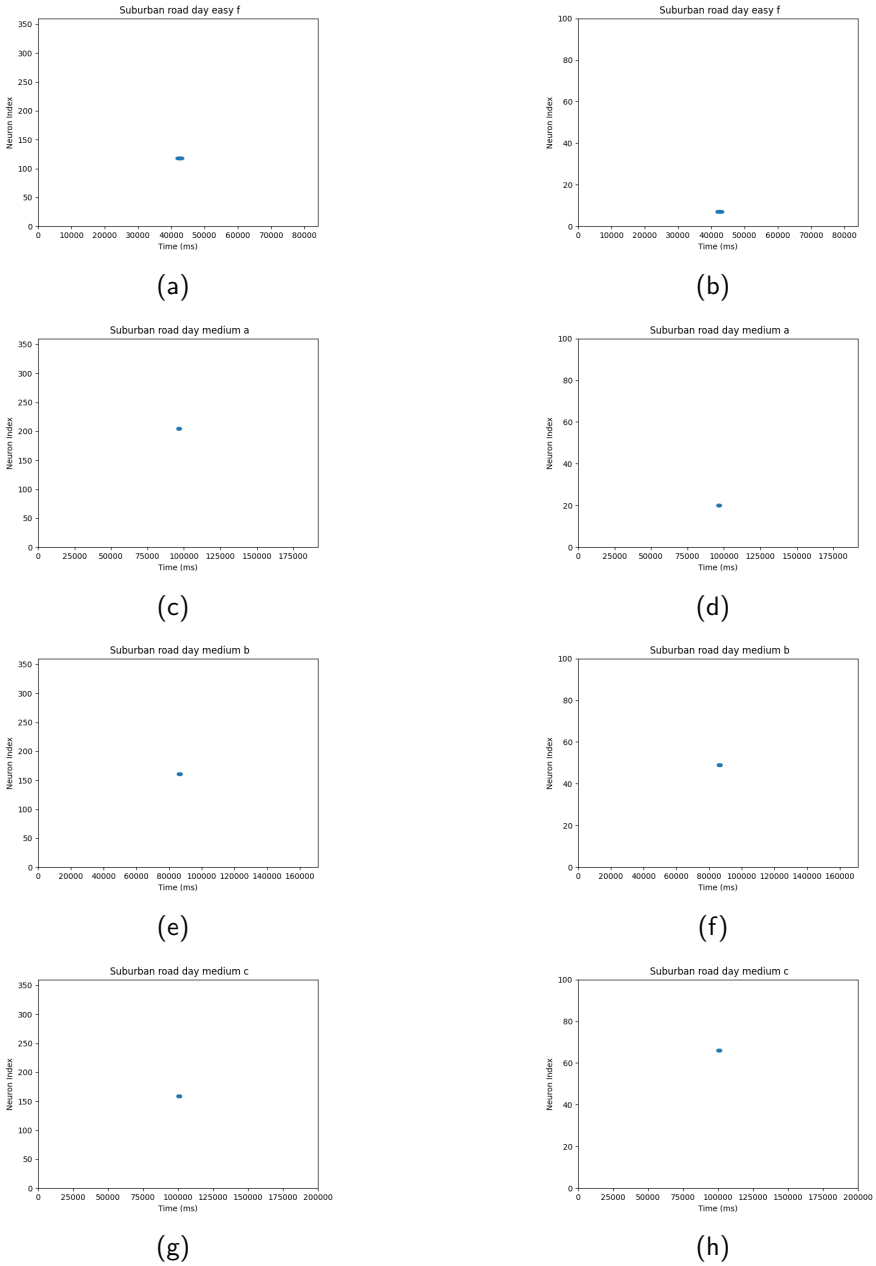


Figure A.54: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

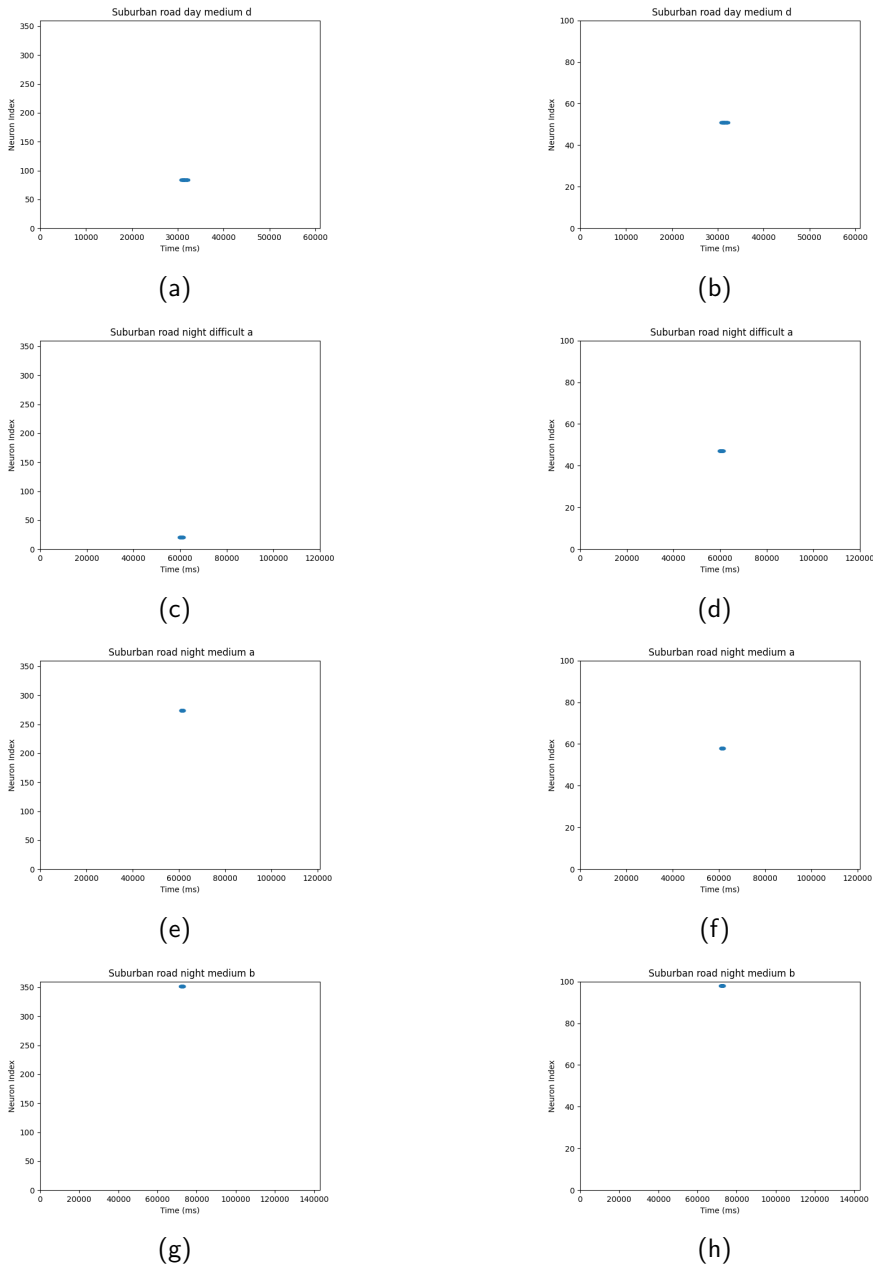


Figure A.55: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

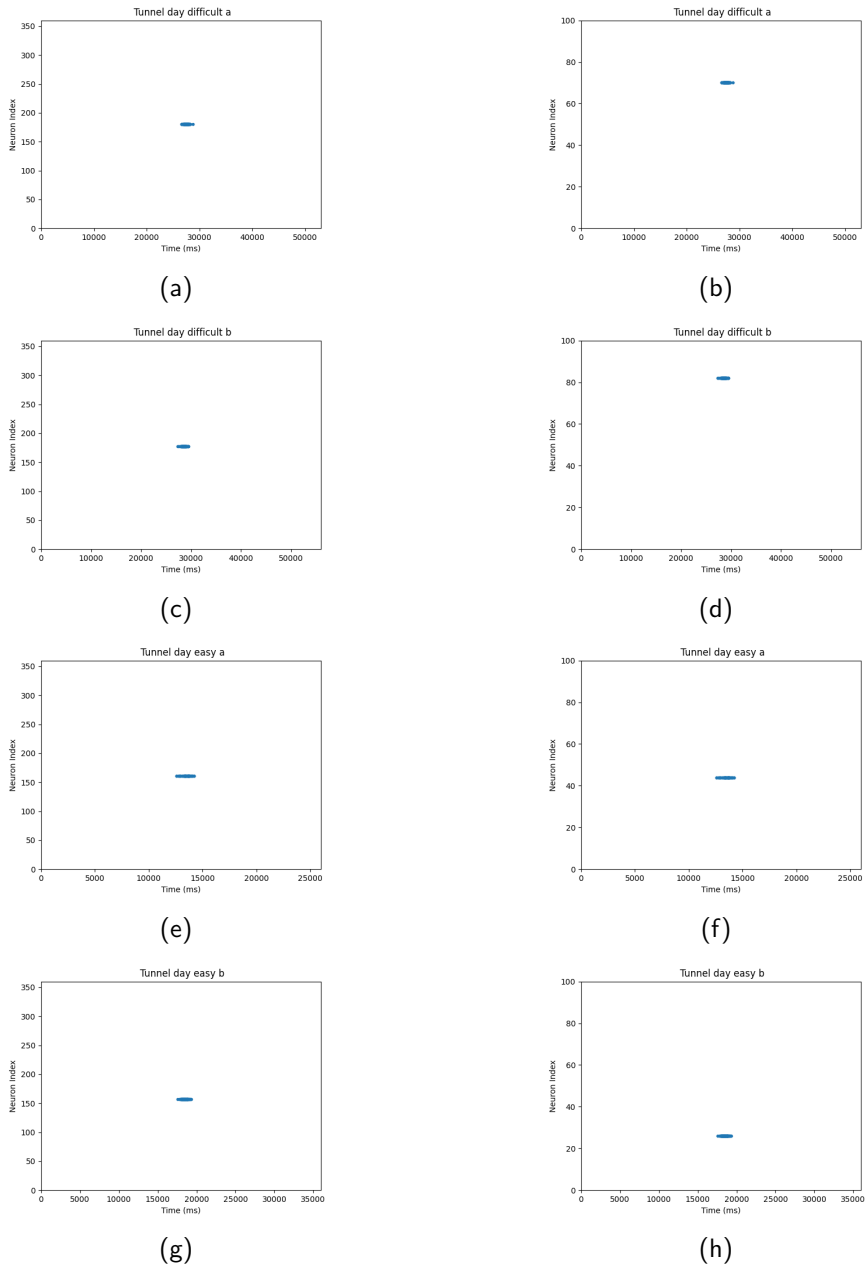


Figure A.56: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

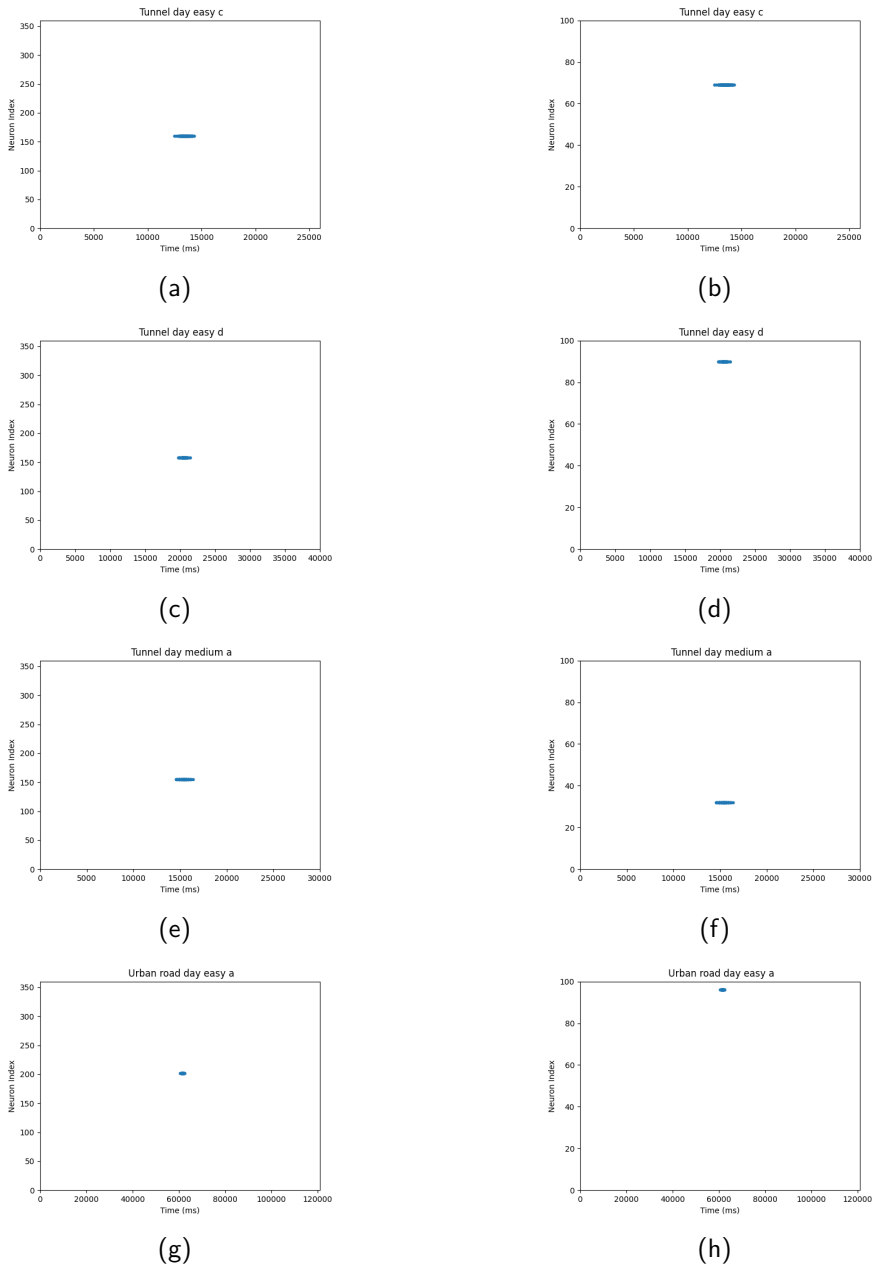


Figure A.57: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

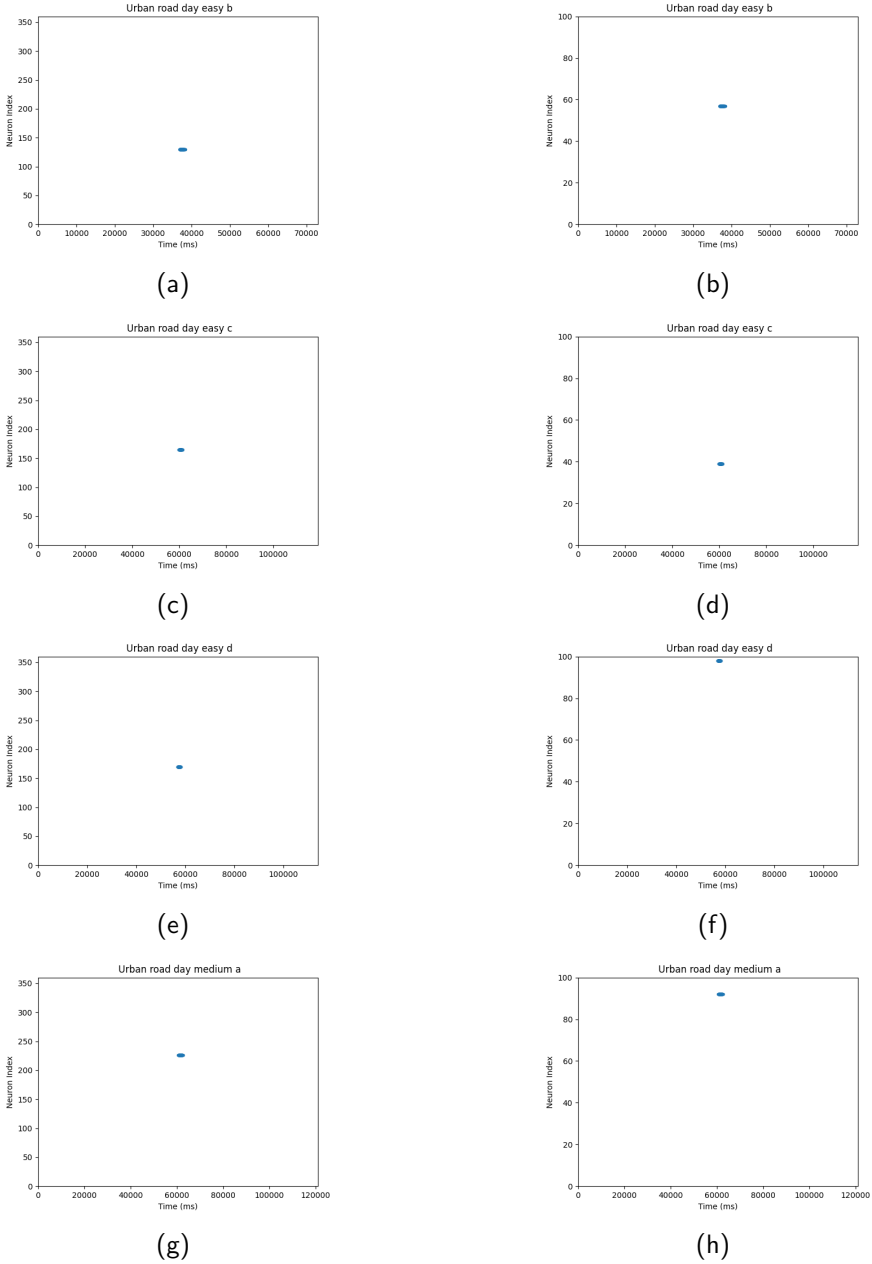


Figure A.58: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

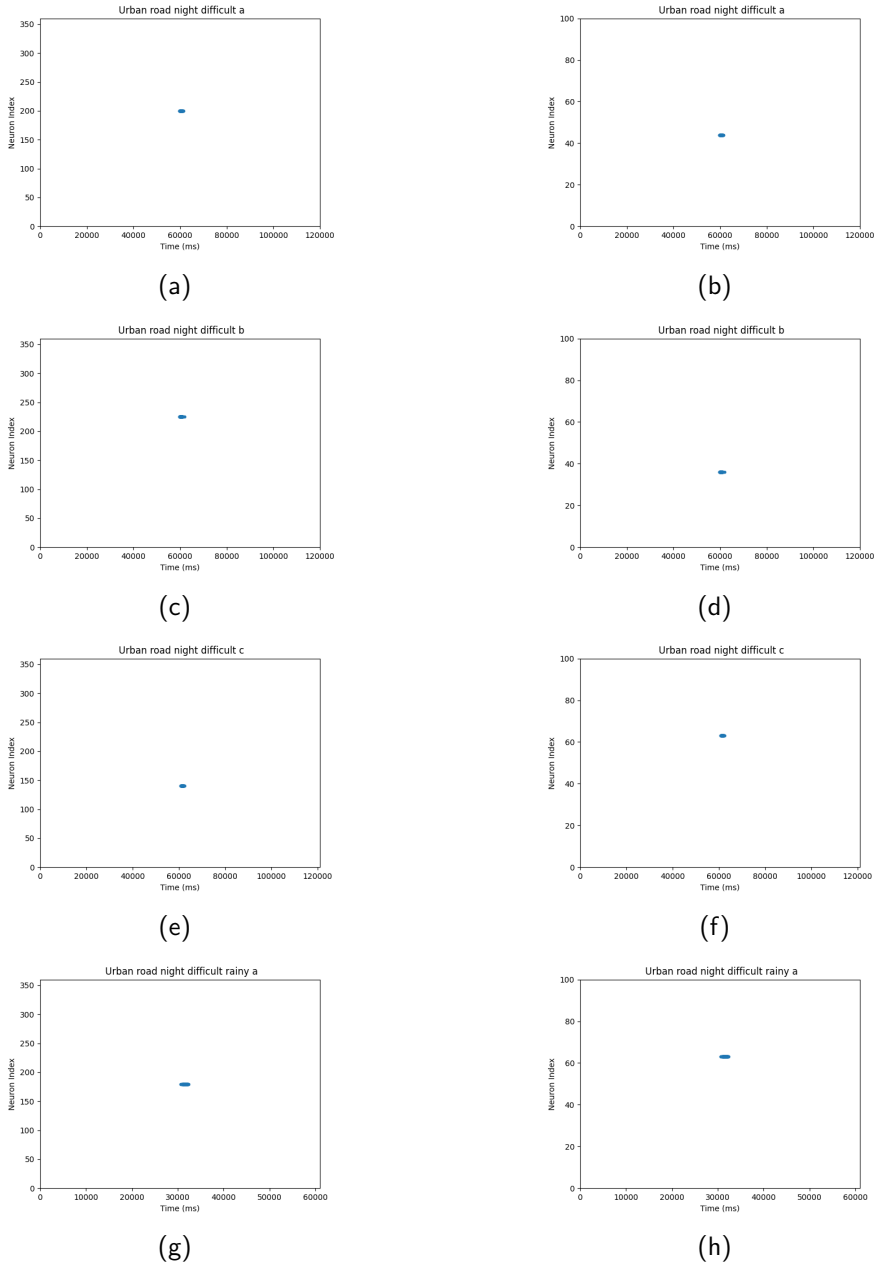


Figure A.59: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

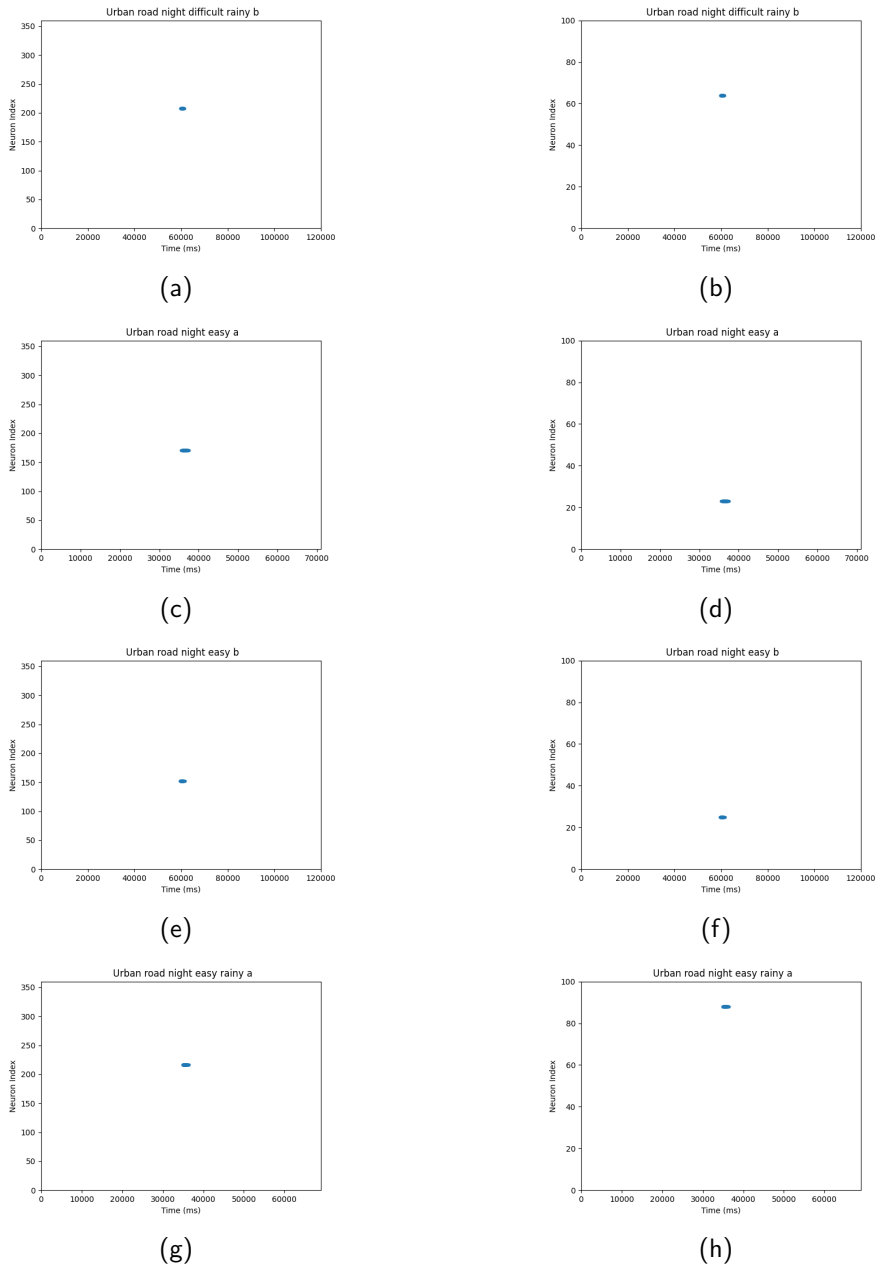


Figure A.60: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

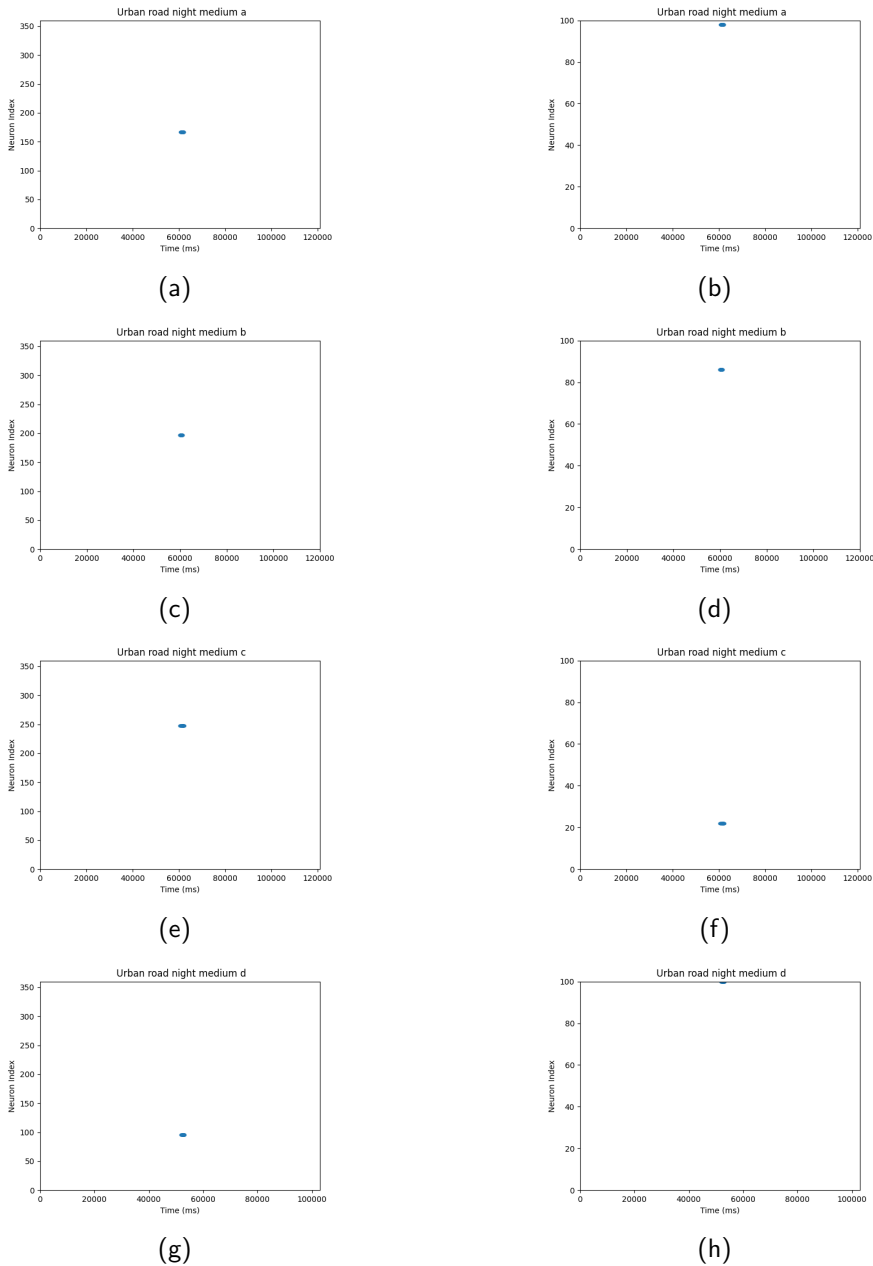


Figure A.61: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

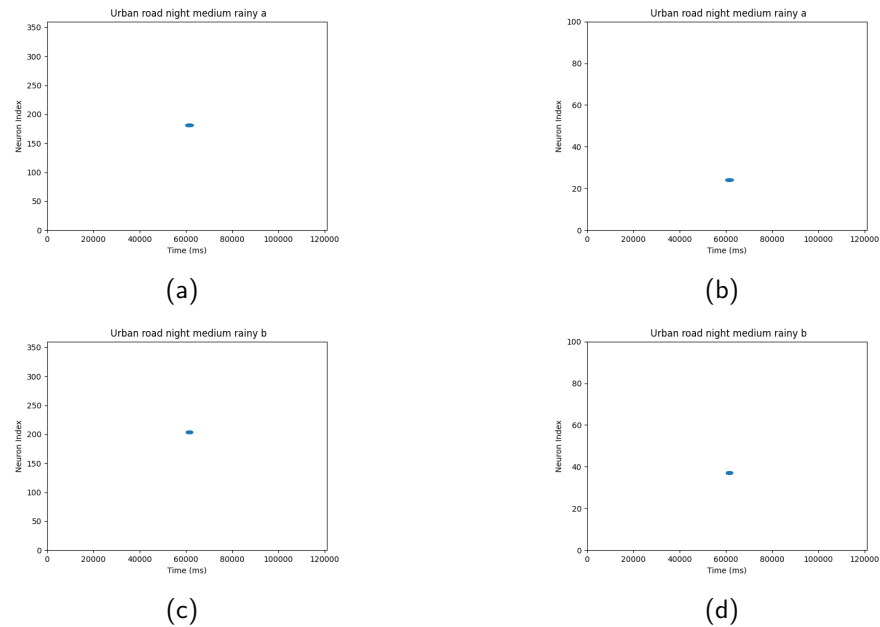


Figure A.62: The reset signals from the Vision network for each ECMD sequence on each row. The Head Direction reset shown on the left and the Grid (step size 10m) reset shown on the right.

A.7 Head Direction Network Activity with a Reset Signal

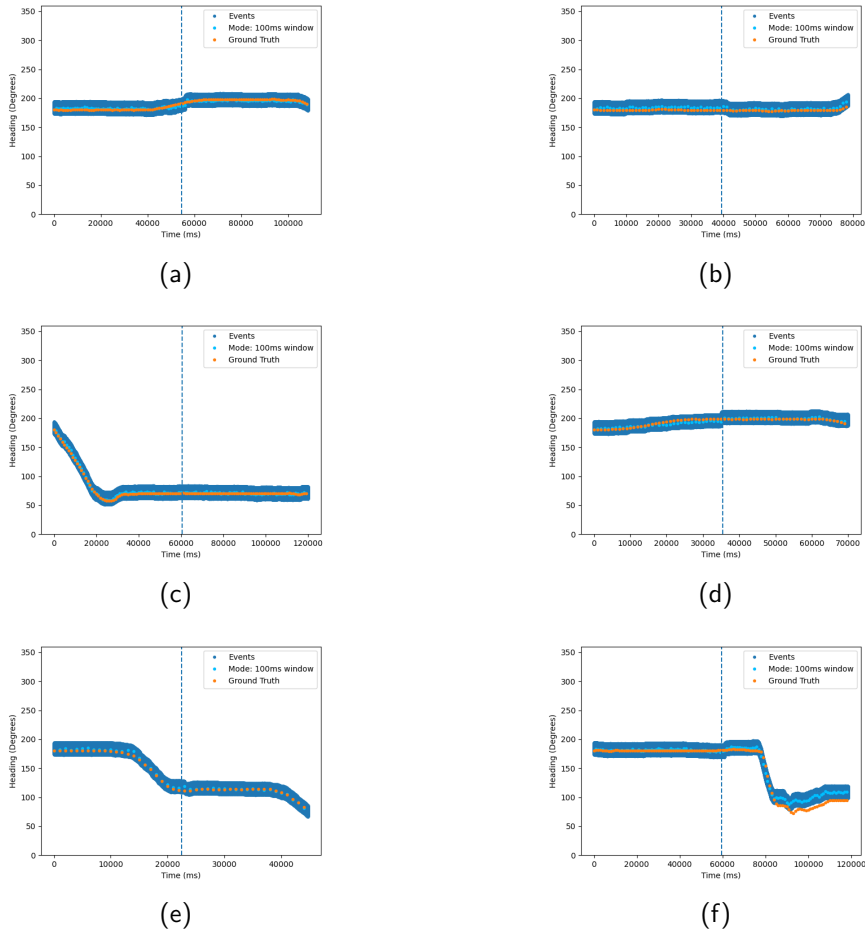


Figure A.63: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.

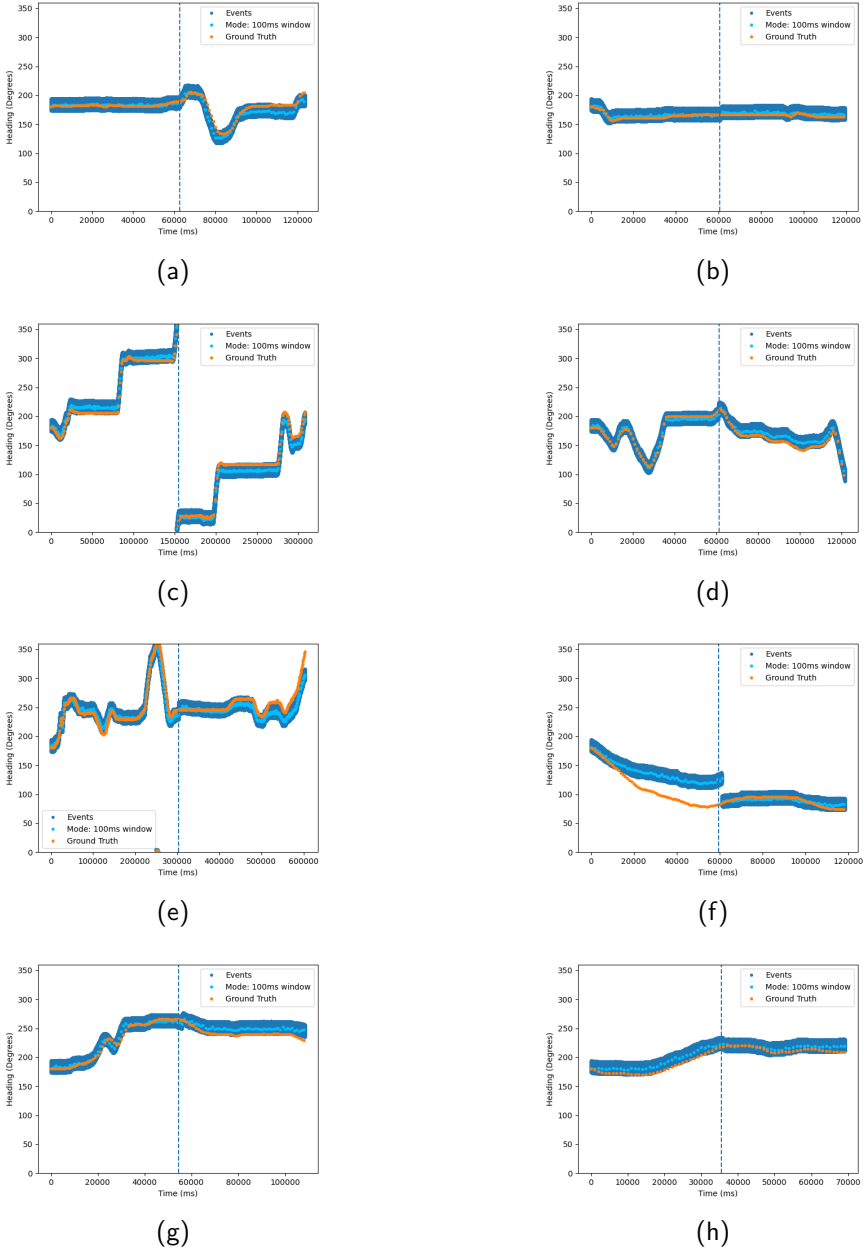
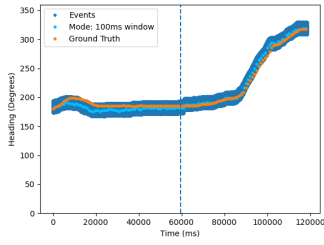
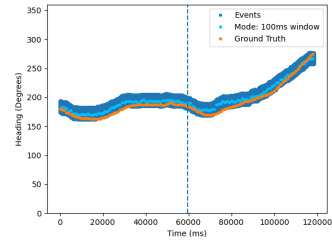


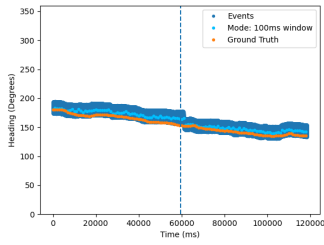
Figure A.64: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.



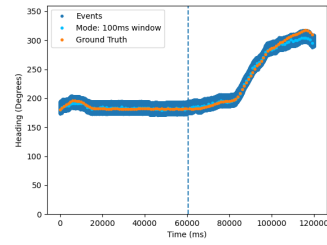
(a)



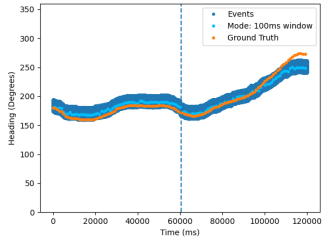
(b)



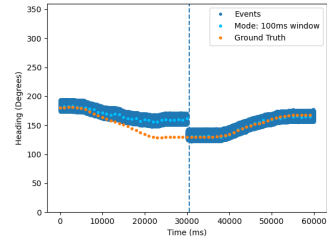
(c)



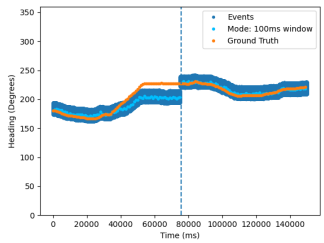
(d)



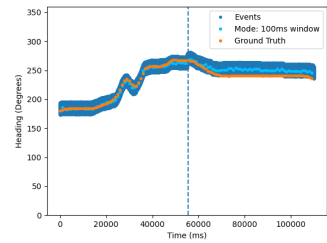
(e)



(f)



(g)



(h)

Figure A.65: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.

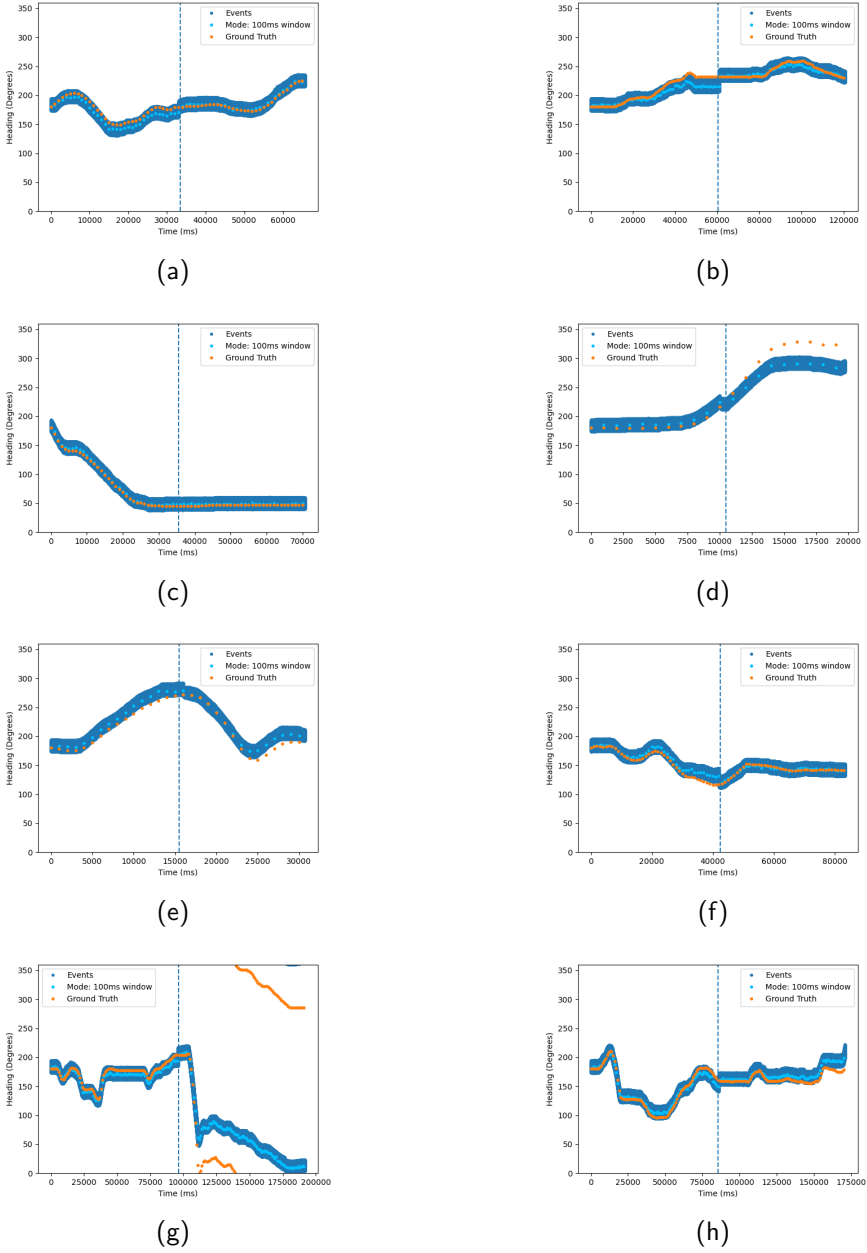
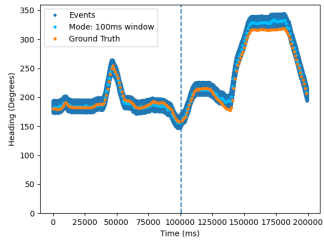
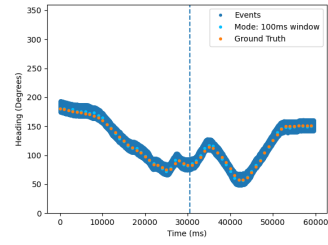


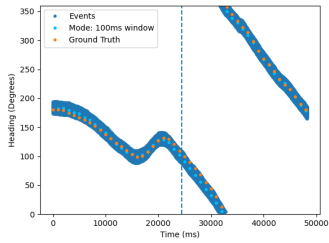
Figure A.66: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.



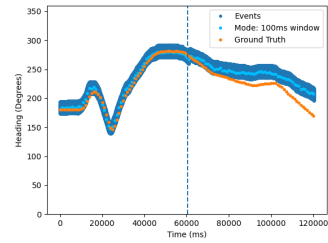
(a)



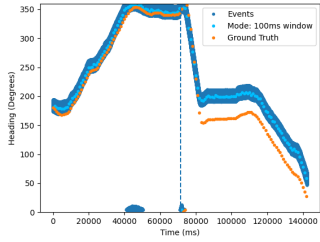
(b)



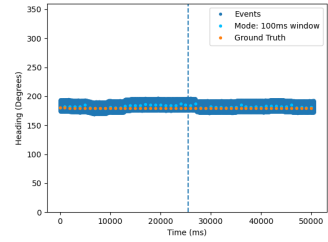
(c)



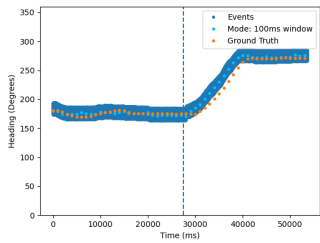
(d)



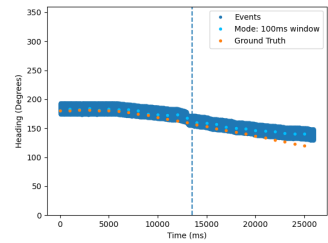
(e)



(f)

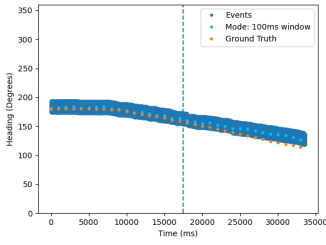


(g)

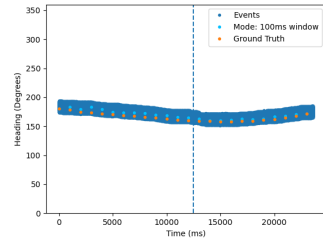


(h)

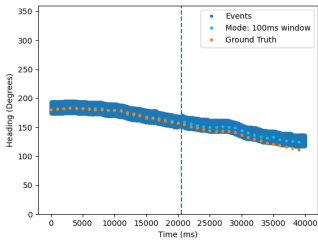
Figure A.67: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.



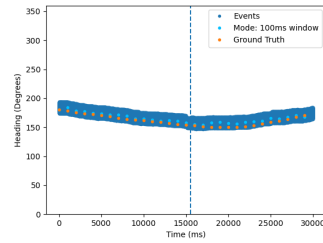
(a)



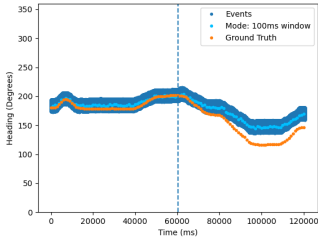
(b)



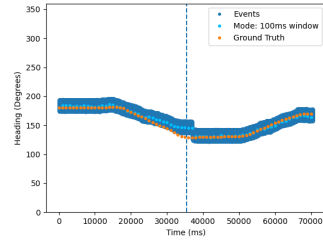
(c)



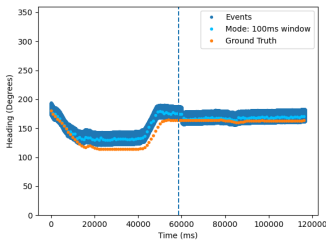
(d)



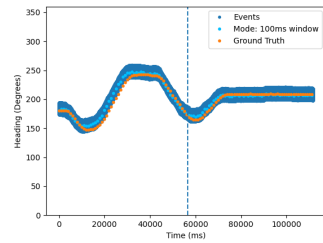
(e)



(f)



(g)



(h)

Figure A.68: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.

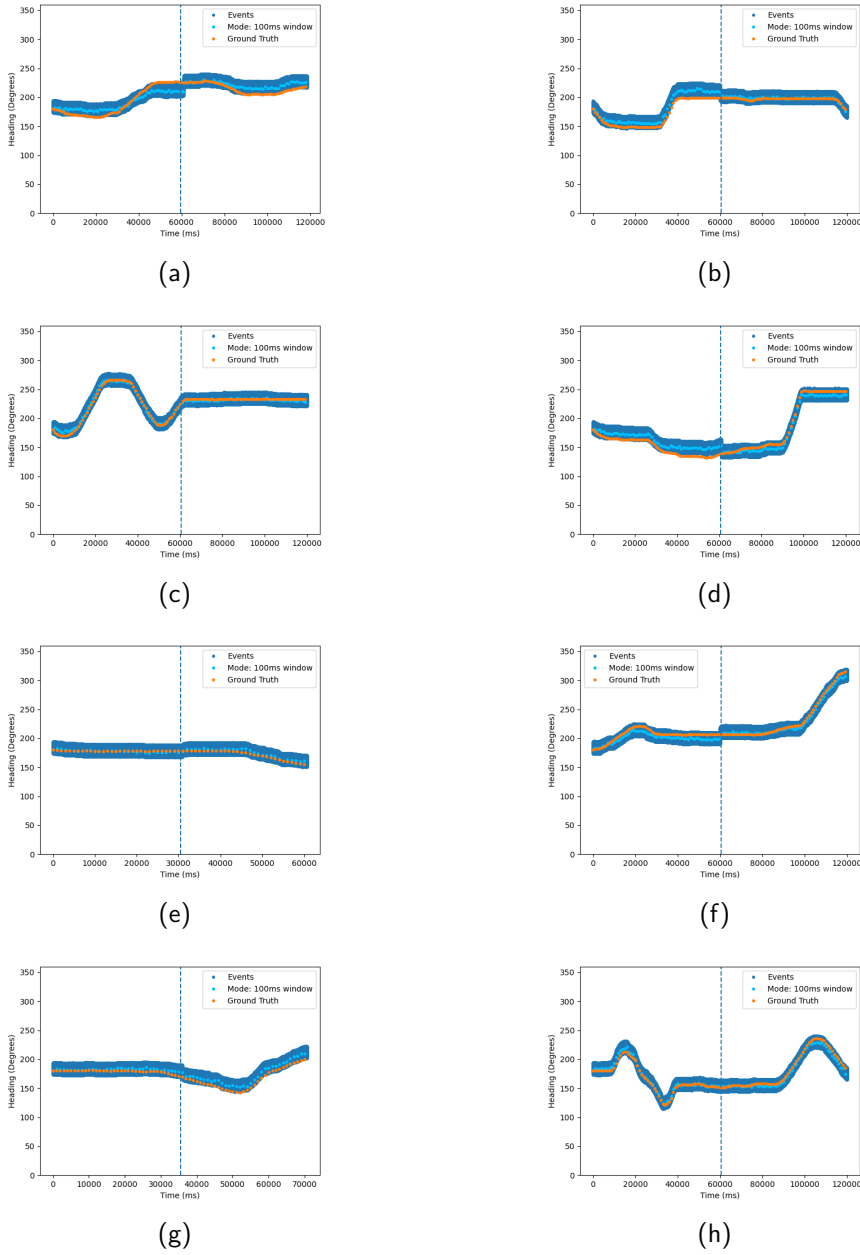


Figure A.69: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.

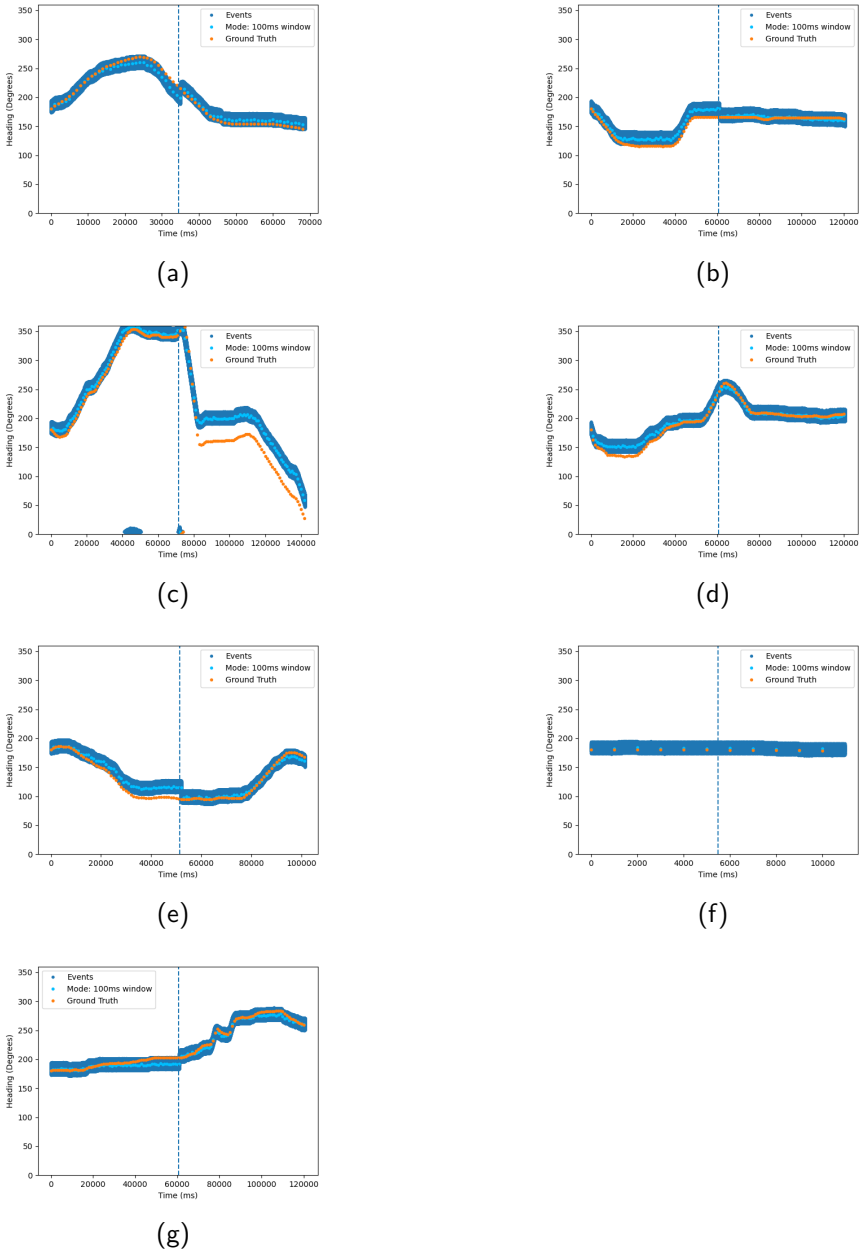


Figure A.70: Head Direction network activity with a reset at the midpoint of the simulation for the ECMD dataset.

A.8 Head Direction Network Errors Against Time with a Reset Signal

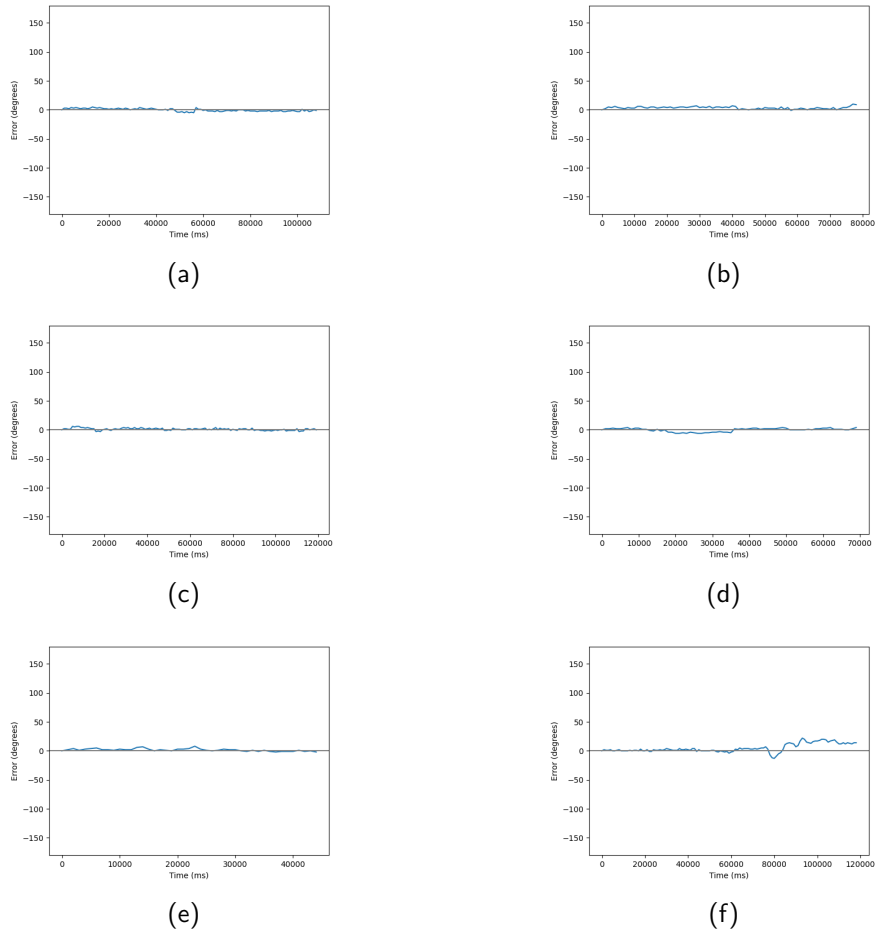


Figure A.71: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.

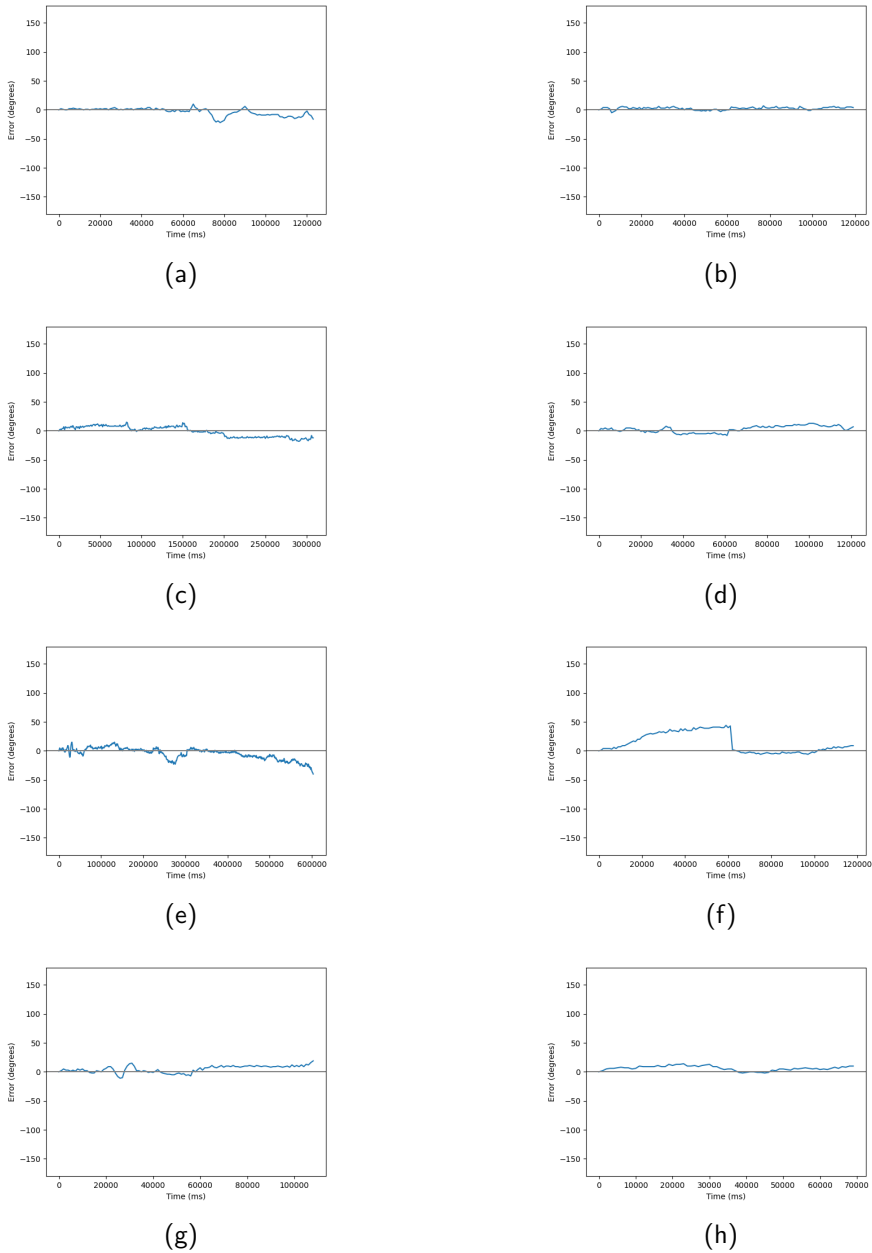
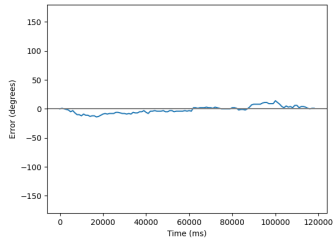
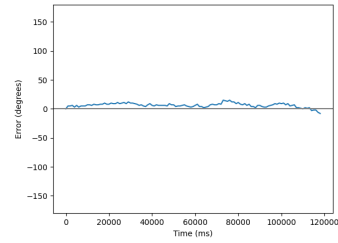


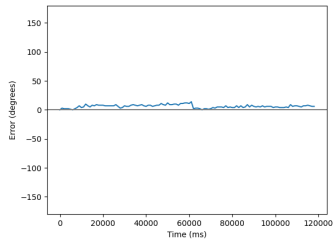
Figure A.72: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.



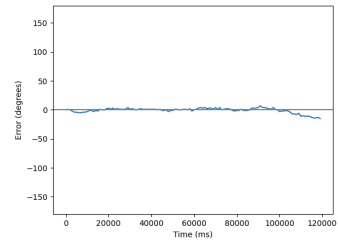
(a)



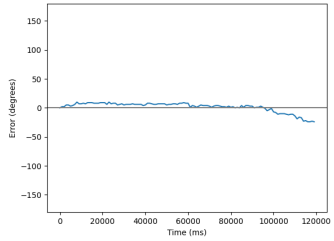
(b)



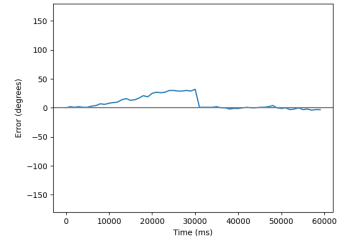
(c)



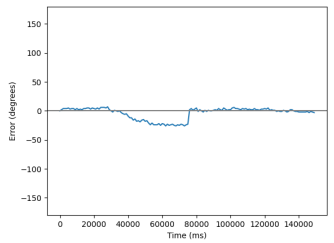
(d)



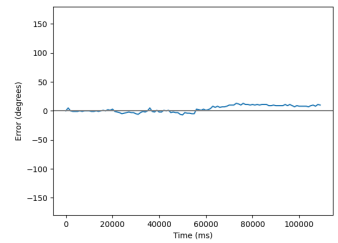
(e)



(f)



(g)



(h)

Figure A.73: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.

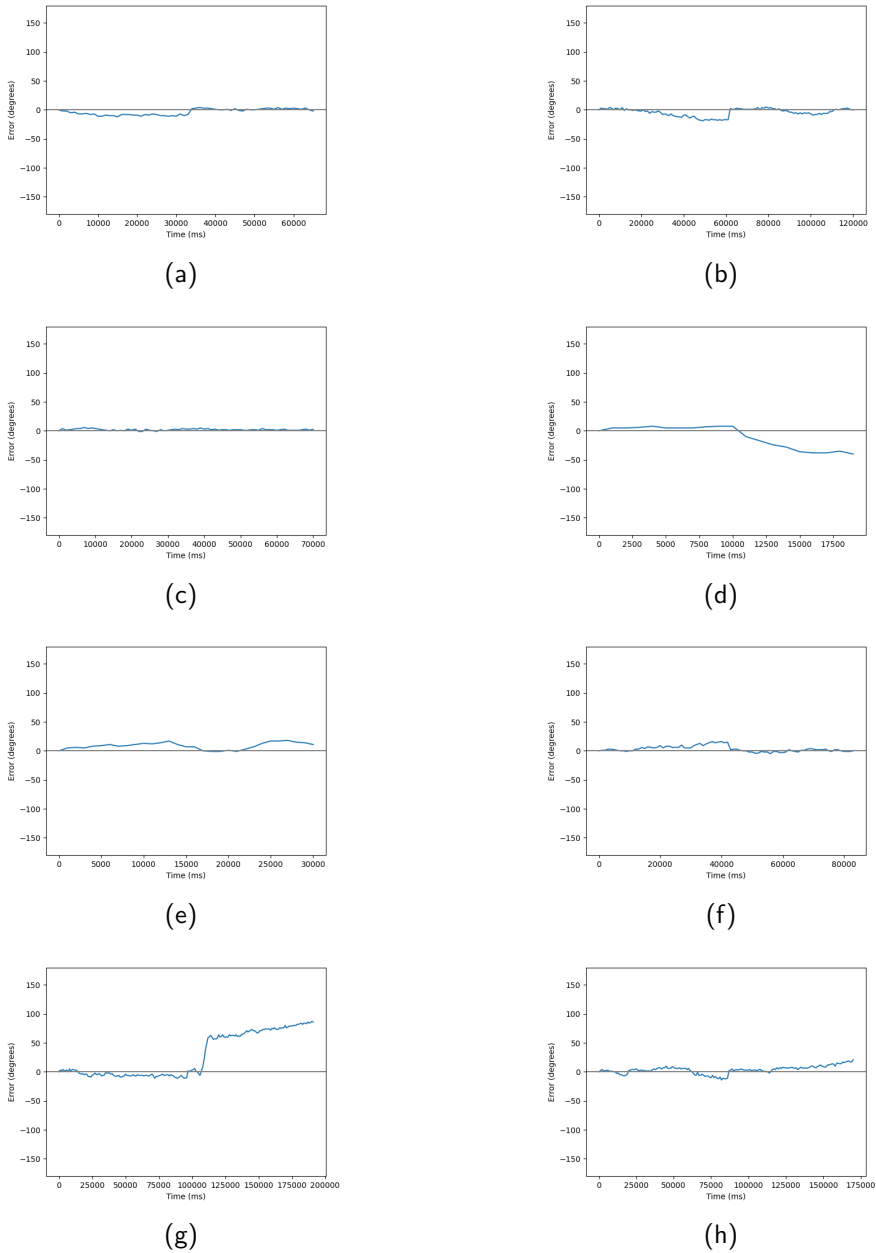
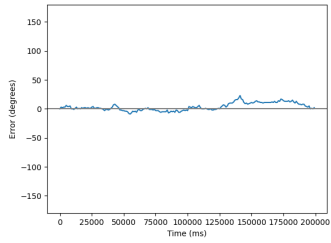
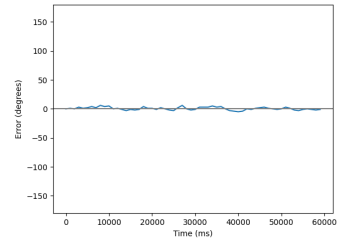


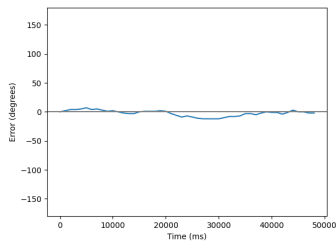
Figure A.74: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.



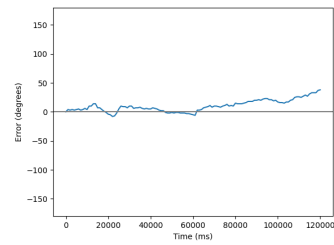
(a)



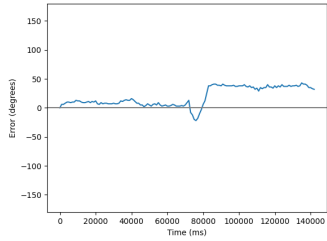
(b)



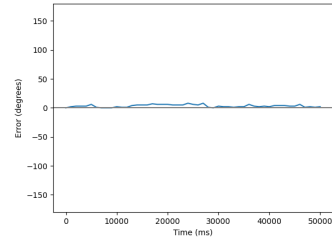
(c)



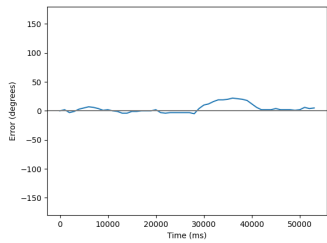
(d)



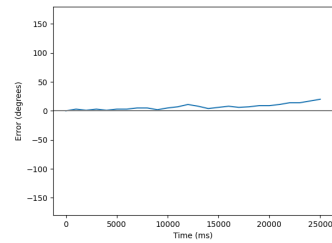
(e)



(f)



(g)



(h)

Figure A.75: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.

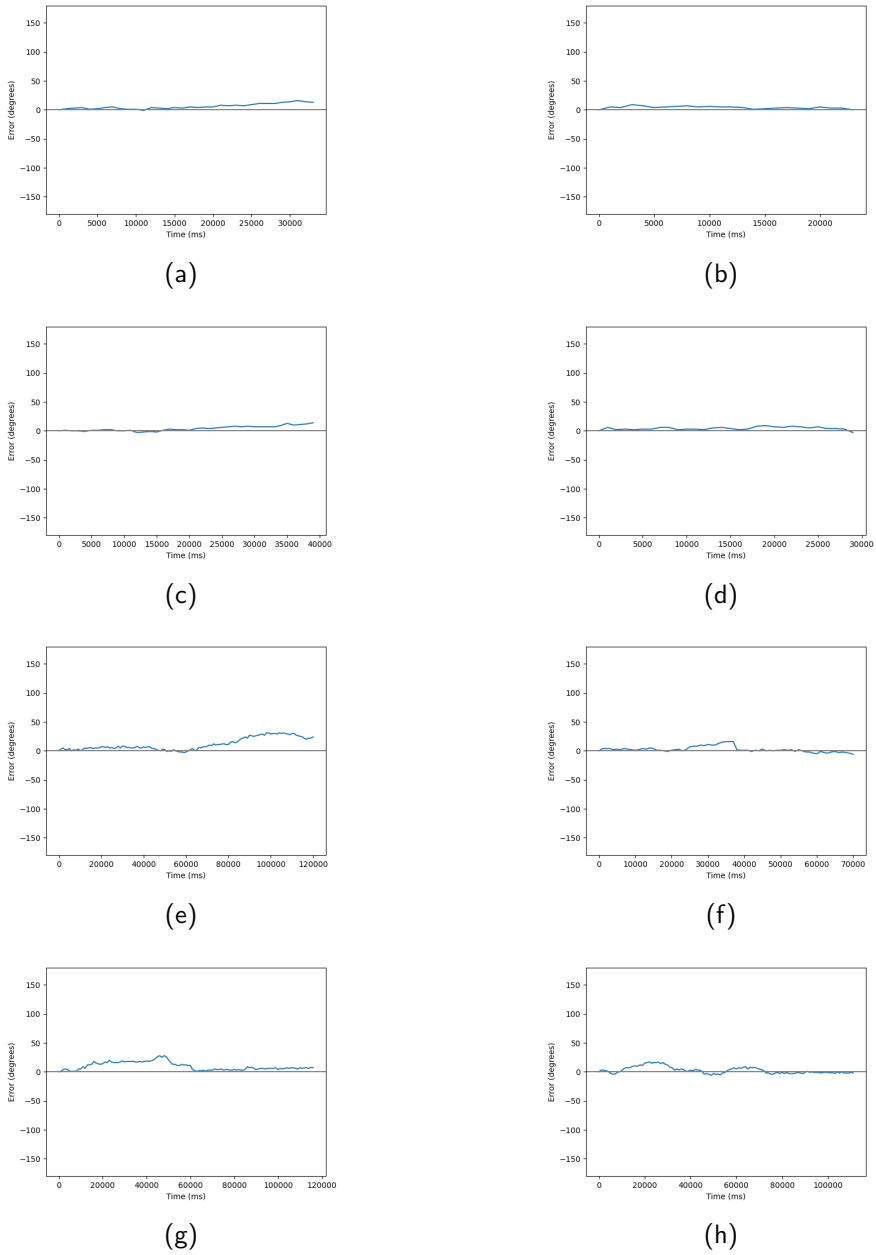
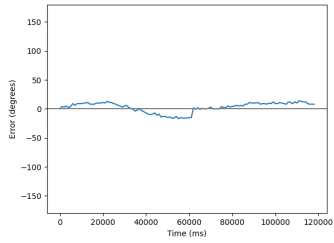
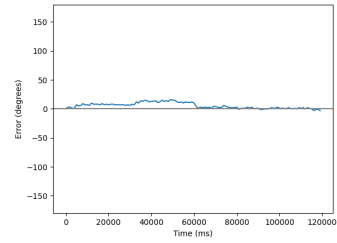


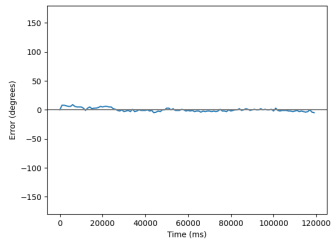
Figure A.76: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.



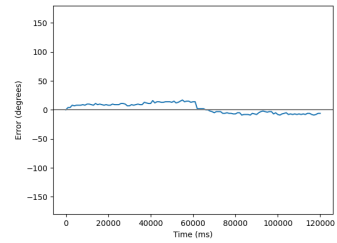
(a)



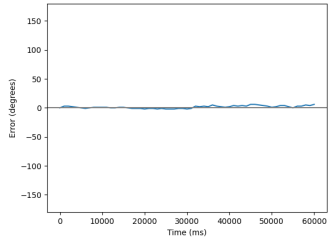
(b)



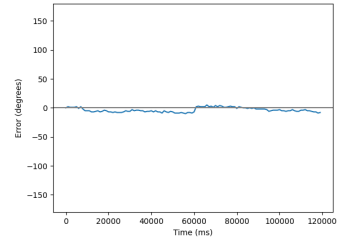
(c)



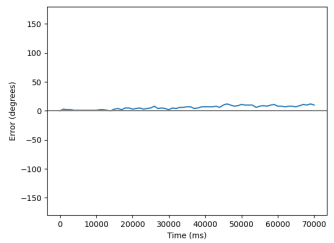
(d)



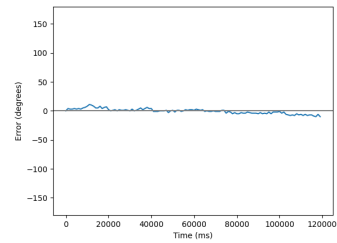
(e)



(f)

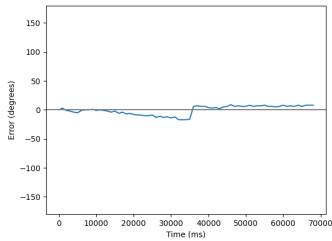


(g)

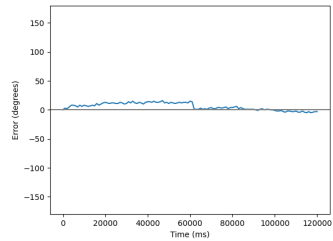


(h)

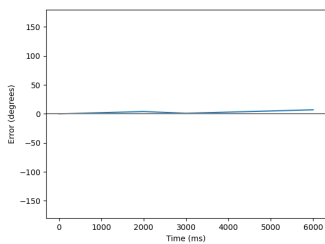
Figure A.77: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.



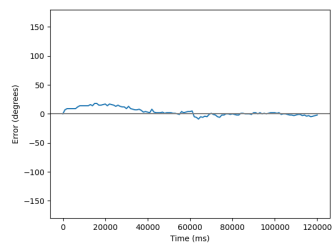
(a)



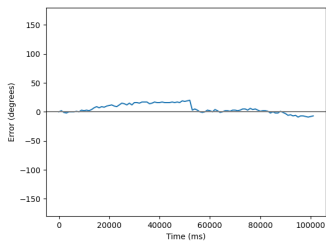
(b)



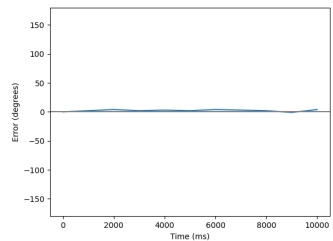
(c)



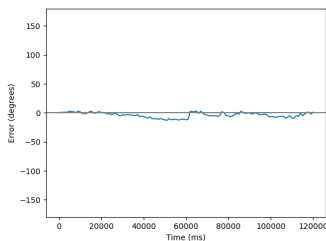
(d)



(e)



(f)



(g)

Figure A.78: Direction errors recorded in the Head Direction network when run on the ECMD dataset with a reset at the midpoint of each sequence.

Appendix B

Submitted Manuscripts

The following manuscripts were submitted for publication during the course of this PhD.

B.1 Manuscript 1: Event-Based Direction Tracking Using Neuromorphic Methods

Submitted to NeurIPS 2024. Following mixed feedback this paper was rejected.

Event-Based Direction Tracking Using Neuromorphic Methods

Gregor Mackenzie

School of Physics, Engineering and Technology
University of York
York, UK, YO10 5DD
gregor.mackenzie@york.ac.uk

Abstract

The ability to track direction is a key challenge for animals and in engineering applications when navigation is concerned. With the growth of neuromorphic computing and the prominence of neural networks it is common to look to biological systems for inspiration. Neuroscience provides evidence of head direction neurons, which specialise in tracking the direction of the agent. A head direction network which features a robust loop closure mechanism applied to complex data, such as driving scenarios, is yet to be demonstrated. Here an event-based head direction system is applied to real-world driving scenarios and combined with a visual reset which is triggered by an event camera. This develops on previous work, where there has been a lack of reset mechanisms for loop closure or the reset has been triggered by a simple feature, requiring no complex visual system. It also aims to offer a basis for creating a visual inertial odometry system requiring only data acquired by an event-camera and its integrated Inertial Measurement Unit (IMU).

1 Introduction

A crucial component of any system aiming to tackle odometry is the ability to accurately and reliably track the direction of an agent at any given time. While there are many approaches for this, the emerging field of neuromorphic computing is one which is largely unexplored for this application despite the biological models available [1]. Inspired by biological neural networks, neuromorphic computing uses Spiking Neural Networks (SNNs) for computation, a type of neural network which has been described as the third generation of neural networks [2].

SNNs differ from traditional neural networks by encoding information in the timing of spikes. They can be both faster and more efficient than traditional neural networks [3], since the neuron network is only active, and thus drawing power, during events compared to the continuous output of a traditional analog neuron. Such behaviour allows spiking neurons to intrinsically act as low latency, coincidence detectors [4], a property that can be exploited for operation in dynamic environments.

In addition to novel computing paradigms Neuromorphic computing has also seen the emergence of event-based sensors [5], with the most applicable to this work being event-cameras. Where a frame-based camera captures the absolute intensity for all of the pixels at regular intervals, in an event-camera the individual pixels operate independently and asynchronously to detect increases (decreases) in intensity and updates with a positive (negative) event whenever this change exceeds a threshold [6]. The result of this is that in the absence of relative motion nothing will be detected, giving low power operation, but in fast-paced environments information is captured at a much higher rate, eradicating motion blur and minimising latency [7, 8].

By taking inspiration from biological systems for tracking direction it is possible to utilise the intrinsic property of event-based operation exhibited by event-cameras and SNNs to create a homogeneous head direction system which is end-to-end event-based.

In this work the concept of a visual reset has been combined with a head direction network capable of accurately tracking direction across the range of angular velocities found in the ECMD driving dataset [9]. Beyond this a vision network is used to detect complex scenes and inter-network synapses are trained to build associations between scenes and specific directions.

2 Related work

The concept of specialist cell types in the brain is not a new one, with recording and analysis of direction and visual processing having been well studied in neuroscience. As a result, models for the operations of these systems have been long established [1, 10–13]. However with the emergence of neuromorphic computing, concepts and models from neuroscience are now more prevalent in engineering applications, with head direction and vision networks being no exception to this [14–17].

Krieser *et al.* [15] implemented a head direction network capable of a visual reset on a neuromorphic platform. Initially the network operated at a single angular velocity and reset to a single position indicated by a flashing diode. Following this a more complex system capable of operating at four angular velocities was shown [18]. For mobile robotics applications either one or a limited number of rates of rotation could and were shown to offer reasonable accuracy, however for an agent with an unconstrained angular velocity it can be assumed that errors would build quickly. The visual reset is proven to be functional but for more complex scenarios there is a need to distinguish scenes and landmarks which is not present.

More recently a head direction network using a ring structure was shown to be accurate over a range of angular velocities and implemented across multiple simulated and real-world scenarios [14]. This implementation is however purely a head direction network without any loop closure mechanism such as a visual reset. As such accuracy would be expected to degrade over time due to drift, however minimal it may be, or due to erroneous or noisy inputs.

There have been many implementations concerning event-based vision and navigation however the focus, typically, has either been to seek a target [19] or avoid an obstacle [20]. Such approaches however, don't translate to the task of scene detection required for a visual reset. Instead inspiration can be drawn from biological visual processing where lines are detected and used for cognitive processing [21].

3 System overview

The system consists of two distinct networks, a head direction network, for continuously tracking direction, and a vision network for detecting known scenes and thus triggering a visual reset. Both networks are run entirely using inputs from an event-camera (specifically a DVXplorer) with the angular velocity data coming from the event-camera's built in IMU.

3.1 Head direction network

The head direction network is formed using a ring network structure consisting of four layers (shown in Fig. 1), with each layer made up of 360 neurons, giving 1 degree increments. These layers are the head direction layer, two shift layers and a reset layer. The head direction layer uses a self-sustaining 'hill of activity' [22], to represent the current direction of the system. The two shift layers are used to stimulate a movement of the network left and right respectively, and the integrated head direction layer provides a mechanism through which the head direction layers activity can be reset to a new position.

In the absence of any inputs the head direction layer must sustain its activity for the representation of direction to persist. This is achieved by connecting neurons with a close proximity to one another with excitatory connections and neurons that are distantly spaced with inhibitory connections (specific weights and distances can be seen in Fig. 2). Excitation in the shift layers is driven entirely by angular velocity inputs from the IMU to provide stimulation. The head direction layer makes no

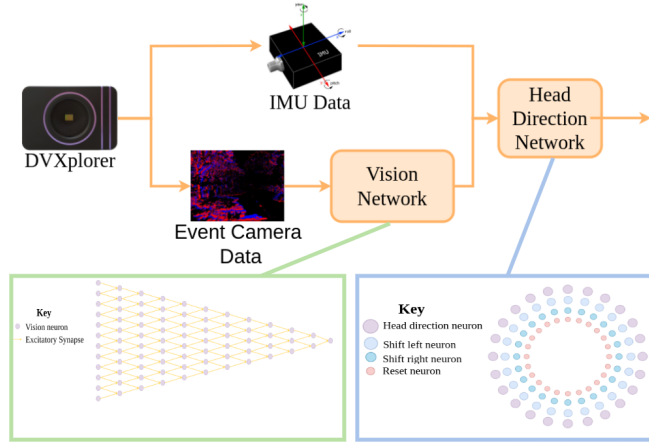


Figure 1: An overview of the system used to track the orientation of the vehicle. The angular velocity recorded from an IMU integrated into the event-camera is input into the head direction network. The events from the event-camera are input into the vision network which connects to the head direction network using synapses which are trained using Spike-Timing-Dependent Plasticity (STDP) to trigger visual resets. Including a view of the head direction network topology showing how direction can be continuously represented and shifted and the vision network showing the triangular structure and connectivity for one column of pooled input pixels.

excitatory connections to the shift layer but instead makes inhibitory connections to shift neurons which represent a significant difference of direction whether left or right. This is because combining inhibition with excitation is more robust [23] to shifting caused erroneously by noise than relying on summing excitatory connections to cause a shift. Any given neuron in the shift layer is connected to a small cluster of neurons in the head direction layer to move the 'hill of activity' to this new location. To enable the shifting mechanism to move activity around the network, an offset of one neuron (1 degree) in the respective direction of the shifting layer is applied when connecting the shifting neuron to the head direction layer. This can all be seen in Fig. 2.

While this network is capable of integrating angular velocity to track direction there is the risk of errors accumulating. This could be through drift over time or erroneous readings, both of which would result in a persistent offset relative to the true direction. By implementing a visual reset mechanism the network is capable of correcting such errors. The reset layer provides a mechanism for updating the location of the network, stimulating activity at a desired location in this layer results in a reset of the network, however landmarks at known directions must be identified as a trigger and thus a vision network is required.

3.2 Vision network

The vision network uses a triangular structure to detect vertical lines within scenes to create a distinct pattern of activity. Initially the pixels from the event camera are uniformly pooled to reduce the size from 640x480 pixels down to 16x12 pools to reduce the number of neurons required and thus the computational load.

The neurons within the vision network are organised into columns, with each column not sharing any connection with another column making them entirely independent. The pooled inputs connect to 12 neurons creating the 'base' of the triangle with each consecutive layer containing one less neuron than the one previous. Each neuron in the layers beyond the base are connected to the two adjacent neurons in the previous layer with a connection strength that is sufficiently weak that should a single presynaptic neuron spike then the postsynaptic neuron won't, but strong enough that should

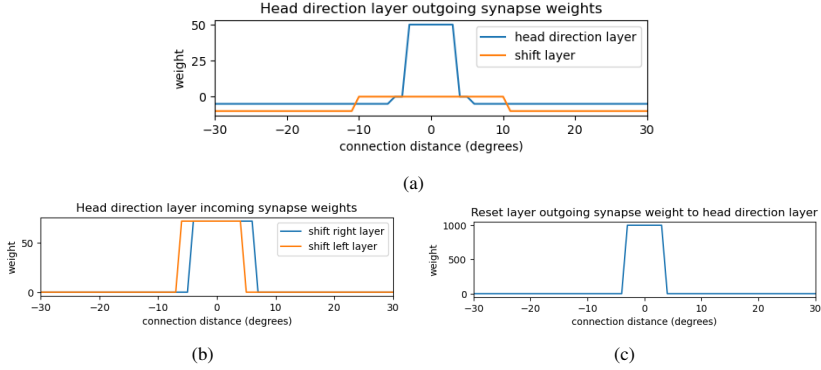


Figure 2: The strength of synaptic connections between the layers of the head direction network. a) The synapse weights connecting the head direction layer of the head direction network to itself and the shift layers. The head direction layer excites the neurons in close proximity but inhibits the rest of the layer. The shift neurons are inhibited except for the neurons with similar preferred directions to the active head direction neuron. b) The synapse weights connecting the shift layers of the head direction network to the head direction layer. This shows the respective right and left offsets of the connections. c) The synaptic strength from the reset layer to the head direction layer is significantly stronger than other connections as to overcome any inhibition as this may be used to trigger large shifts of activity.

both presynaptic neurons spike then the postsynaptic neuron will. This behaviour can be considered analogous to that of a digital AND gate. This behaviour leads to the detection of vertical lines as any activity in adjacently pool inputs will be passed 'up' through the triangle network indicating a longer line.

3.3 Training method

To build an association between the visual events and a direction at a given timestamp, a combination of synaptic time dependent plasticity (STDP) [24] synapses and reinforcement learning signal are used during the training phase. Following training the reinforcement signals aren't connected to the network and the learned weights from the STDP synapses are applied to static synapses.

Every neuron in the vision network is connected to every neuron in the reset layer of the head direction network, with all of these connections being trained. When training, a strong inhibitory current is also applied to all neurons in the reset layer, preventing unwanted activity triggered by the vision network. Instead a strong excitatory signal is used to induce individual spikes at the desired times at the neuronal index equivalent to the ground truth direction at that time point. With the vision network spiking according to the event camera data and the reset neurons spiking only at the desired times and directions the network is then able to build the weight associations between the pattern of the vision network and the desired resets to cause them to be triggered by the detected scene.

4 Results

To verify that the network operates as intended, it was run on the Highway sequences from the ECMD dataset [9] with no reset and with a reset at the halfway point. The error from these runs can be seen in Table 1. Across all of the sequences in the dataset it was found that the error was most visible in the latter stages of the run while accuracy was better initially as seen in Fig. 3. From this it can be seen that the network is experiencing drift, most likely due to the non-linearity of spiking neurons.

The accuracy of all of the sequences improved with the introduction of a reset at the halfway point. The range of improvement is however, large indicating that this is not always an effective strategy. It is hypothesised that identifying when an error has accumulated to trigger a reset would lead to

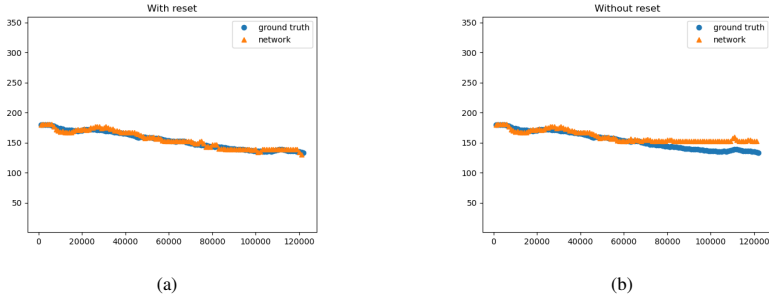


Figure 3: A comparison of the networks direction tracking with (a) and without (b) a reset at the midpoint. The respective accuracies are detailed in Table 1 under 'Medium 3'.

Table 1: ECMD highway errors

Sequence	Error (degrees)	
	W/o reset	Reset
Day		
Easy 1	8.53	5.92
Easy 2	6.47	5.77
Medium 1	13.11	8.27
Medium 2	9.60	5.92
Medium 3	9.69	2.88
Difficult 1	3.98	3.92
Night		
Medium 1	8.06	3.99
Medium 2	10.24	8.23
Medium 3	9.03	5.84
Difficult 1	9.79	9.51
Difficult 2	5.91	3.98

more accurate dataset results, while identifying and recognising reliable landmarks would be the best approach for real world applications.

The network was simulated using NEST [25] ran at speeds in excess of 4x real time without the reset. Due to the vision network being similarly sized to the head direction network, when the reset was introduced this was reduced to approximately 2x real time speed.

5 Discussion

The head direction network, without a reset, displayed acceptable levels of error, typically in the single digits, and the introduction of a reset proved effective, reducing the error in all cases. To expand upon the initial results presented here, followed up experimentation from ECMD [9] and alternate datasets [26–29] should be performed.

It was found that the shifting response was not entirely linearly aligned with the angular velocity, likely due to the intrinsic, nonlinear behaviour of spiking neurons. This has been minimised by reducing the refractory period of the neuron and making the spike rate with no inputs approximately 9Hz as the point at which a neuron starts to spike is it's least linear.

The range of improvement in accuracy caused by the reset is dependant on the timing of the reset, which can be controlled when using a dataset however for real world deployment a reset should occur

whenever a landmark is detected. Therefore an element of segmentation and key landmark detection ought to be introduced into the vision network to ensure only permanent, identifiable landmarks are learned.

References

- [1] K. Zhang, "Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory," *The Journal of Neuroscience*, vol. 16, pp. 2112–2126, 3 1996. [Online]. Available: <https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.16-06-02112.1996>
- [2] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, pp. 1659–1671, 12 1997, sNN review / intro. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0893608097000117>
- [3] B. Sengupta and M. B. Stemmler, "Power consumption during neuronal computation," *Proceedings of the IEEE*, vol. 102, pp. 738–750, 5 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6797884/>
- [4] P. König, A. K. Engel, and W. Singer, "Integrator or coincidence detector? the role of the cortical neuron revisited," *Trends in Neurosciences*, vol. 19, pp. 130–137, 4 1996. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0166223696800191>
- [5] A. Vanarse, A. Osseiran, and A. Rassau, "A review of current neuromorphic approaches for vision, auditory, and olfactory sensors," *Frontiers in Neuroscience*, vol. 10, 3 2016. [Online]. Available: <http://journal.frontiersin.org/Article/10.3389/fnins.2016.00115/abstract>
- [6] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorph event-based vision sensors: Bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, pp. 1470–1484, 10 2014. [Online]. Available: <http://ieeexplore.ieee.org/document/6887319/>
- [7] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120 db 15 us latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, pp. 566–576, 2008, dVS release. [Online]. Available: <http://ieeexplore.ieee.org/document/4444573/>
- [8] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, pp. 2333–2341, 10 2014, presenting DAVIS. [Online]. Available: <https://ieeexplore.ieee.org/document/6889103>
- [9] P. Chen, W. Guan, F. Huang, Y. Zhong, W. Wen, L.-T. Hsu, and P. Lu, "Ecmd: An event-centric multisensory driving dataset for slam," *IEEE Transactions on Intelligent Vehicles*, vol. 9, pp. 407–416, 1 2024, eCMD Dataset. [Online]. Available: <https://ieeexplore.ieee.org/document/10342726/>
- [10] J. Taube, R. Muller, and J. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis," *The Journal of Neuroscience*, vol. 10, pp. 420–435, 2 1990. [Online]. Available: <https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.10-02-00420.1990>
- [11] X. Xie, R. H. R. Hahnloser, and H. S. Seung, "Double-ring network model of the head-direction system," *Physical Review E*, vol. 66, p. 041902, 10 2002. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevE.66.041902>
- [12] A. Angelucci, J. B. Levitt, E. J. Walton, J. M. Hupé, J. Bullier, and J. S. Lund, "Circuits for local and global signal integration in primary visual cortex," *Journal of Neuroscience*, vol. 22, pp. 8633–8646, 10 2002. [Online]. Available: <https://www.jneurosci.org/content/22/19/8633https://www.jneurosci.org/content/22/19/8633.abstract>
- [13] S. Zeki and A. Bartels, "Toward a theory of visual consciousness," *Consciousness and Cognition*, vol. 8, pp. 225–259, 6 1999.
- [14] Z. Bing, A. E. Sewisy, G. Zhuang, F. Walter, F. O. Morin, K. Huang, and A. Knoll, "Toward cognitive navigation: Design and implementation of a biologically inspired head direction cell network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, pp. 2147–2158, 5 2022, head direction network, same as mine, no IH layer. [Online]. Available: <https://ieeexplore.ieee.org/document/9634854/>
- [15] R. Kreiser, M. Cartiglia, J. N. Martel, J. Conradt, and Y. Sandamirskaya, "A neuromorphic approach to path integration: A head-direction spiking neural network with vision-driven reset," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2018-May. IEEE, 4 2018, pp. 1–5, head direction network and visual reset mechanism. [Online]. Available: <https://ieeexplore.ieee.org/document/8351509/>
- [16] B. Chakraborty, U. Kamal, X. She, S. Dash, and S. Mukhopadhyay, "Brain-inspired spatiotemporal processing algorithms for efficient event-based perception," in *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*. IEEE, 4 2023, pp. 1–6, general SNN fo processing Event Camera data. [Online]. Available: <https://ieeexplore.ieee.org/document/10136914/>

- [17] M. Osswald, S.-H. Ieng, R. Benosman, and G. Indiveri, "A spiking neural network model of 3d perception for event-based neuromorphic stereo vision systems," *Scientific Reports*, vol. 7, p. 40703, 1 2017, depth estimation? using stereo event cameras and snns. [Online]. Available: <https://www.nature.com/articles/srep40703>
- [18] R. Kreiser, A. Renner, Y. Sandamirskaya, and P. Pienroj, "Pose estimation and map formation with spiking neural networks: towards neuromorphic slam," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10 2018, pp. 2159–2166, position and direction using snn, inputs unsure? [Online]. Available: <https://ieeexplore.ieee.org/document/8594228/>
- [19] M. B. Milde, H. Blum, A. Dietmüller, D. Sumislawska, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system," *Frontiers in Neurobotics*, vol. 11, 7 2017, dVS and neuromorphic computing obstacle avoidance and target acquisition. [Online]. Available: <http://journal.frontiersin.org/article/10.3389/fnbot.2017.00028/full>
- [20] Z. Bing, C. Meschede, K. Huang, G. Chen, F. Rohrbein, M. Akl, and A. Knoll, "End to end learning of spiking neural network based on r-stdp for a lane keeping vehicle," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2018, pp. 4725–4732, lane keeping robot using a DVS and a new learning method, R-STDP. [Online]. Available: <https://ieeexplore.ieee.org/document/8460482/>
- [21] E. Peterhans, B. Heider, and R. Baumann, "Neurons in monkey visual cortex detect lines defined by coherent motion of dots," *European Journal of Neuroscience*, vol. 21, pp. 1091–1100, 2 2005. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1460-9568.2005.03919.xhttps://onlinelibrary.wiley.com/doi/abs/10.1111/j.1460-9568.2005.03919.xhttps://onlinelibrary.wiley.com/doi/10.1111/j.1460-9568.2005.03919.x>
- [22] P. Song and X.-J. Wang, "Angular path integration by moving "hill of activity": A spiking neuron model without recurrent excitation of the head-direction system," *The Journal of Neuroscience*, vol. 25, pp. 1002–1014, 1 2005. [Online]. Available: <https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.4172-04.2005>
- [23] X. J. Wang, "Synaptic basis of cortical persistent activity: the importance of nmda receptors to working memory," *Journal of Neuroscience*, vol. 19, pp. 9587–9603, 11 1999. [Online]. Available: <https://www.jneurosci.org/content/19/21/9587https://www.jneurosci.org/content/19/21/9587.abstract>
- [24] D. E. Feldman, "The spike-timing dependence of plasticity," *Neuron*, vol. 75, pp. 556–571, 8 2012, review STDP and variations. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0896627312007039>
- [25] M.-O. Gewaltig and M. Diesmann, "Nest (neural simulation tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.
- [26] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2032–2039, 7 2018. [Online]. Available: <http://ieeexplore.ieee.org/document/8288670/>
- [27] K. Chaney, F. Cladera, Z. Wang, A. Bisulco, M. A. Hsieh, C. Korpela, V. Kumar, C. J. Taylor, and K. Daniilidis, "M3ed: Multi-robot, multi-sensor, multi-environment event dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023, pp. 4015–4022.
- [28] L. Gao, Y. Liang, J. Yang, S. Wu, C. Wang, J. Chen, L. Kneip, and S. Member, "Vector: A versatile event-centric benchmark for multi-sensor slam," *IEEE ROBOTICS AND AUTOMATION LETTERS*, vol. 7, 2022. [Online]. Available: <https://www.ieee.org/publications/rights/index.html>
- [29] S. Klenk, J. Chui, N. Demmel, and D. Cremers, "Tum-vie: The tum stereo visual-inertial event dataset," *IEEE International Conference on Intelligent Robots and Systems*, pp. 8601–8608, 2021.

B.2 Manuscript 2: Event-Based Orientation Tracking Using Neuromorphic Methods

Submitted to WACV 2025. Being a poor fit for the targetted conference this paper was rejected.

Event-Based Orientation Tracking Using Neuromorphic Methods

Abstract

The ability to track orientation is a key challenge for autonomous driving and mobile robotics applications. Especially now with the growth of neuromorphic computing and the prominence of neural networks it is common to look to biological systems for inspiration. From neuroscience there is empirical evidence of so called head direction neurons, the purpose of which is to track the direction of the biological agent. An application of this head direction system to autonomous driving applications with a loop closure mechanism to account for errors is yet to be done. Here we present a neuromorphic head direction system, applied to real-world driving scenarios with a visual reset for loop closure, triggered by stereo event cameras. This develops upon previous work, where there has been a lack of reset mechanisms for error correction. It also aims to offer a basis to build upon to create a reliable visual inertial odometry system requiring only data acquired by an event-camera and its integrated Inertial Measurement Unit (IMU).

1. Introduction

A crucial component of any system aiming to tackle the challenge of autonomous driving is the ability to accurately and reliably track the orientation of the vehicle at any given time. While there are many approaches that can, and have been taken for this, the emerging field of neuromorphic computing is one which is largely unexplored for this application. Inspired by biological neural networks, neuromorphic computing uses Spiking Neural Networks (SNNs) for computation, a type of neural network which has been described as the third generation of neural networks [9].

SNNs differ from traditional neural networks by encoding information in the timing of spikes, a method which can be both faster and more efficient than the operation of traditional neural networks [12]. The reason for this is due to activity only occurring within the neural network during events compared to more power intensive continuous outputs. Such behaviour allows spiking neurons to intrinsically act as low latency, coincidence detectors [7], a beneficial property when operating in dynamic environments.

In addition to novel computing paradigms Neuromorphic

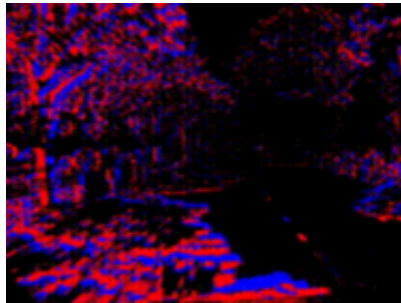


Figure 1. A 'frame' of accumulated positive and negative events sourced from the ECMD Dataset [3].

computing has also introduced event-based sensors [14], with the most applicable to this area of research being event-cameras. Where a frame-based camera captures the absolute intensity for all of the pixels at regular intervals, in an event-camera the individual pixels operate independently and asynchronously to detect increases (decreases) in intensity and updates with a positive (negative) event whenever this change exceeds a threshold [10] as shown in Fig. 1. The result of this is that in the absence of relative motion nothing will be detected, giving low power operation, but in fast-paced environments information is captured at a much higher rate, eradicating motion blur and minimising latency [2, 8].

By taking inspiration from biological systems for tracking direction it is possible to utilise the intrinsic property of event-based operation exhibited by event-cameras and SNNs to create a homogeneous head direction system which is end-to-end event-based.

The key contribution of this work are:

1. To the author's knowledge, the first neuromorphic head direction network with a loop closure mechanism applied to a real-world driving dataset.
2. A stereo event-camera driven visual reset for a head direction network.
3. A head direction network operating solely on inputs

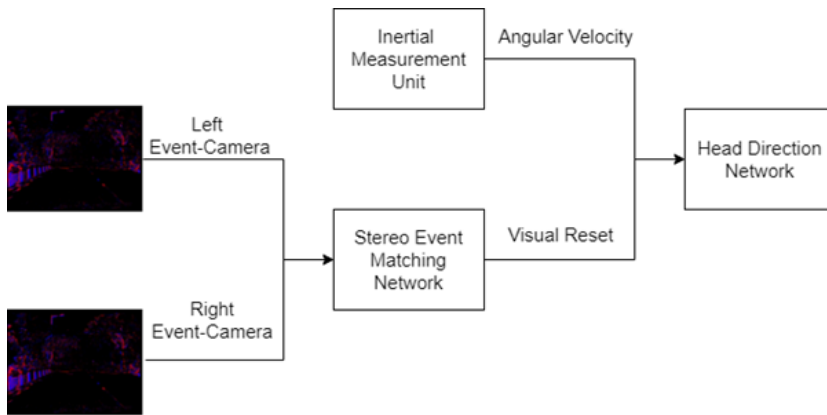


Figure 2. An overview of the system used to track the orientation of the vehicle. The angular velocity recorded from an IMU integrated into the left event-camera is input into the Head Direction network to track orientation. The events from two event cameras are input into the Stereo Matching network. The Stereo Matching network connects to the Head Direction network using synapses which are trained using Spike-Timing-Dependent Plasticity (STDP) to trigger visual reset.

from an event-camera through the integration of an Inertial Measurement Unit (IMU) into the DVXplorer.

2. Related Work

The concept of a head direction cell is not a new one, having been recorded and analysed in neuroscience for decades [13, 17]. As a result of this research, models for the head direction system have also been long established [16]. However with the emerging field of neuromorphic computing, concepts and models from neuroscience are now more prevalent in engineering applications and head direction networks are no exception to this [1, 5].

2.1. Neuromorphic Implementations of Head Direction Models

Krieser *et al.* [5] implemented a head direction network capable of a visual reset on a neuromorphic platform. The limitations of this implementation however are that the network operated at a single angular velocity (per direction) and reset to a single reset position indicated by a flashing diode. When applied to autonomous driving the limitation of operating at a single angular velocity is evident due to the variability of turning speed experienced by vehicles. A more complex system capable of operating at four angular velocities [6] has also been shown and while more capable the same limitation must be considered when applied to real-world driving. While the reset mechanism is highly applicable the use of a flashing diode also doesn't translate to real-world driving scenarios, thus a more complex visual system is required to identify unique scenarios for trigger-

ing a reset.

More recently a head direction network also using a ring structure was shown to be accurate over a range of angular velocities and implemented across multiple simulated and real-world scenarios [1]. This implementation is however purely a head direction network without any loop closure mechanism such as a visual reset. As such it can be assumed that it would suffer a loss of accuracy over time due to drift, however minimal it may be, or the accumulation of erroneous inputs.

3. System Overview

The system mainly consists of two distinct networks, a head direction network, for tracking orientation at any given time, and a stereo matching network for detecting scenes and triggering a visual reset. Both networks are run entirely on inputs from two event-cameras (specifically DVXplorers) with the angular velocity data coming from the IMU integrated into the left hand event-camera.

3.1. Head Direction Network

The head direction network is formed using a ring network structure consisting of four layers (shown in Fig. 4), with each layer being made up of 360 neurons, providing 1 degree increments. These layers are the head direction layer, two shift layers and an integrated head direction layer. The head direction layer uses a self-sustaining 'hill of activity' (see Fig. 3), to represent the current direction of the system. The two shift layers are used to stimulate a movement of the network left and right (anticlockwise and clockwise

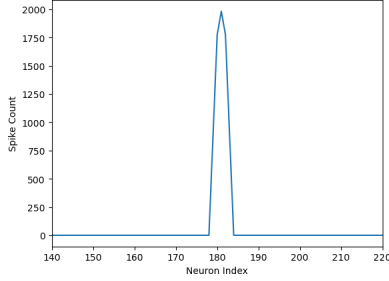


Figure 3. The representation of direction within the head direction neurons.

respectively) and the integrated head direction layer represents the position of the network at the next time step provided the position of the network is to change.

In the absence of any inputs the head direction layer must sustain its activity for the representation of direction to persist. This is achieved by connecting neighbouring neurons with excitatory connections and inhibitory connections for neurons representing a greater than 3 degree shift in either direction. Excitation in the shift layers is driven entirely by angular velocity inputs from the IMU to provide stimulation. The head direction layer makes no excitatory connections to the shift layer but instead makes inhibitory connections to shift neurons representing a greater than 3 degree shift in either direction as this is shown to be more robust to shifting caused erroneously by noise. Connections from the shift layers to the integrated head direction layer use a one-to-one connection meaning, each shift neuron is only connected to the corresponding integrated head direction neuron. However, to enable the shifting mechanism to move activity around the network, an offset of one neuron (1 degree) is applied to the left (right) for the shift left (right) layer. The integrated head direction layer makes a strong excitatory connection to its respective head direction layer to inhibit previous activity within the head direction layer and move the ‘hill of activity’ to the new direction. This can all be seen in Fig. 4.

While this network is capable of integrating angular velocity there is the risk of errors accumulating, either through drift over time or through erroneous readings resulting in a persistent offset relative to the true system orientation. In both cases, implementing a visual reset mechanism offers the capability of accounting for such errors. As the integrated head direction layer already provides a mechanism for updating the location of the network, stimulating activity at a desired location in this layer results in a reset of the network, however landmarks at known directions must be identified as a trigger and thus a vision network is required.

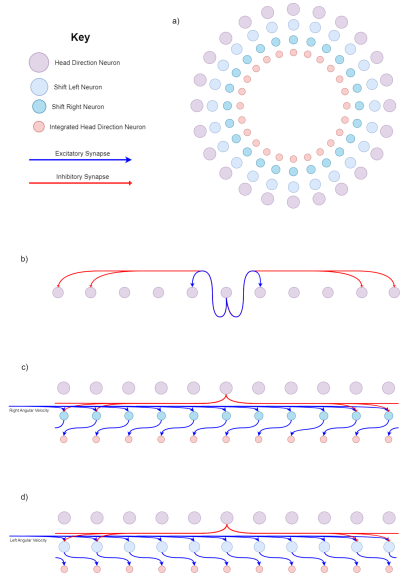


Figure 4. An overview of the Head Direction network layout and its connectivity to display desired behaviours. a) The structure of the Head Direction network reduces from 360 neurons per layer to 24 neurons per layer for simplicity. b) The connectivity of the head direction layer to create a ‘hill of activity’ to represent the current direction. Each neuron makes excitatory connections to its direct neighbours and inhibitory connections to all neurons that represent a direction greater than 3 degrees away. c) and d) The shift right and shift left mechanisms respectively. The head direction neurons inhibit all of the neurons that represent a direction greater than 3 degrees away. The right (left) angular velocity makes excitatory connections to all of the shift right (left) neurons. The shift right (left) neurons make excitatory connections to the integrated head direction neurons, offset by 1 degree to the right (left) of the current position causing the shift mechanisms. Not shown, the integrated head direction neurons connect to the corresponding head direction neurons to update the position in the head direction layer.

3.2. Stereo Matching Network

Due to the nature of event-camera operation, detecting changes in intensity, typically the boundaries or edges between relative light and dark are the cause for events. The polarity of this event however depends on the relative motion of this boundary i.e. the same boundary would trigger the same events but with inverted polarities if the motion was reversed. As such all events are made positive as to ensure that the same scene recreates the same event stream, regardless of relative motion

For visual processing the pixels from each event-camera are pooled into a 16x12 grid of 40x40 pixels. The neurons

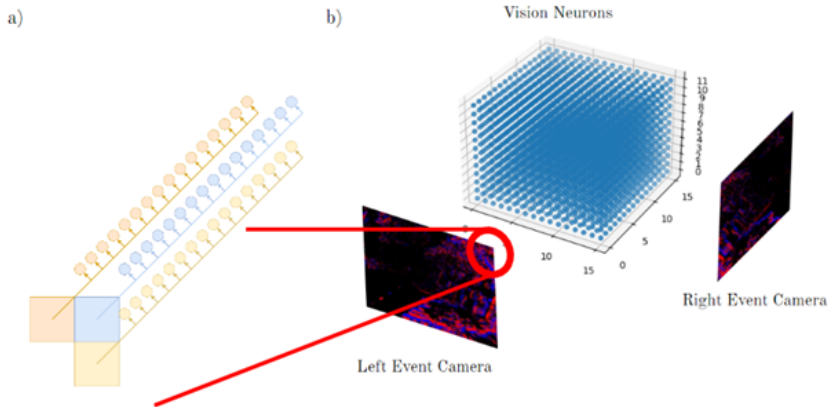


Figure 5. An overview of the structure and connectivity of the stereo matching network. b) The network is a $16 \times 16 \times 12$ cube of neurons, none of which are connected to one another. The left event-camera being aligned parallel to the x axis and the right event-camera being aligned parallel to the y axis. a) Each event camera is pooled into 16×12 grids of 40×40 pixels. The outputs of these 40×40 pixels are the inputs to all of the neurons running perpendicular. *e.g.* pixels (0:40, 0:40) from the left event-camera input into neurons (0, 0:16, 0) and pixels (0:40, 0:40) from the right event-camera input into neurons (0:16, 0, 0)

are organised into an unconnected cube with dimensions of $16 \times 16 \times 12$ meaning that when aligning an event-camera with the neurons each pool aligns with a row in the neuronal grid. The outputs from each pool are then connected to each individual neuron within its corresponding row.

By aligning the left event-camera perpendicular to the y axis and the right event-camera perpendicular to the x axis the ability of spiking neurons to act as coincidence detectors can be used to match the occurrence of events between cameras. With the events between cameras matched a distinct firing pattern is created that can then be used, with training to reset the Head Direction network.

3.3. Landmark Recognition Training

For the system to learn the association between a scene and a specific orientation STDP synapses are used during a training period. The STDP synapses are combined with a ground truth signal at given times so that only accurate associations are made through a reinforcement learning method.

The vision network is connected to the integrated head direction neurons using an all-to-all connection scheme with weights initialized to zero. During the training phase the integrated head direction neurons are globally inhibited by an offset current and no angular velocity inputs are used. The integrated head direction neurons are instead only stimulated by signals at predefined from the ground truth readings. As a result of this the vision neurons which fire immediately preceding the induced integrated head direction

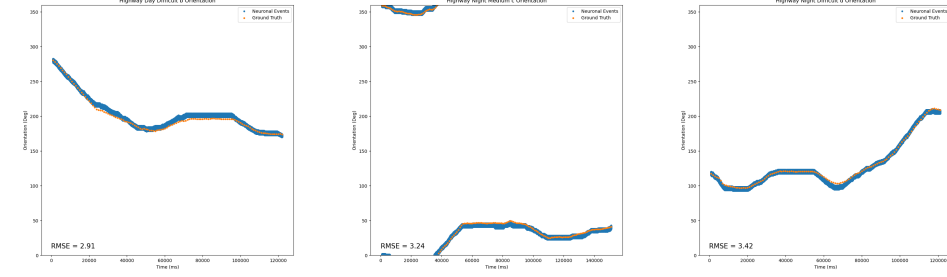
activity will have these synaptic weights increased, while the rest will remain zero. Following the synaptic weight increases any erroneous firing from the vision neurons doesn't induce activity in the integrated head direction layer due to the inhibitory offset current meaning that these synaptic weights will decrease due to being seemingly less 'selective'.

4. Results

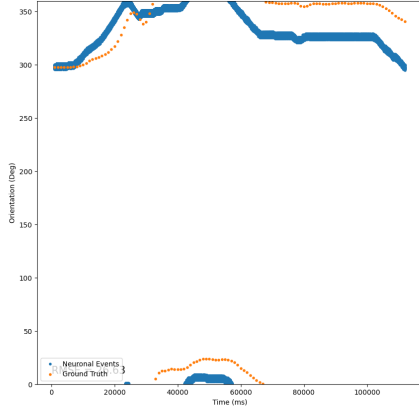
Sequence	Error w/o Reset	Error w Reset
Highway day easy a	26.63	17.07
Highway day easy b	3.7	N/A
Highway day medium a	5.66	N/A
Highway day medium b	3.69	N/A
Highway day medium c	3.79	N/A
Highway day difficult a	31.36	N/A
Highway day difficult b	2.91	N/A
Highway night medium b	5.90	N/A
Highway night medium c	3.24	N/A
Highway night medium d	5.79	N/A
Highway night difficult c	4.30	N/A
Highway night difficult d	3.42	N/A

Table 1. The root mean square error in degrees for the 12 sequences from the highway set of the ECMD dataset [3].

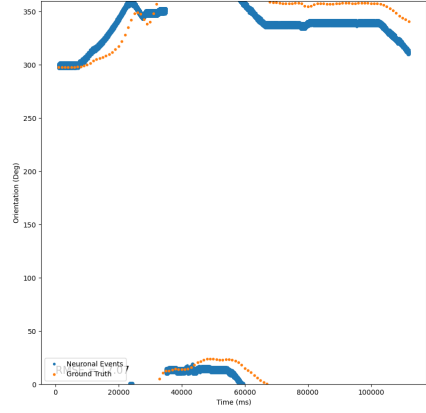
To verify the effectiveness of the head direction network



(a) A sample of accurate sequences.



(b) No visual reset



(c) With a visual reset at 35000ms.

Figure 6. The head direction neuron activity (blue) plotted against the ground truth (orange) for a sequence displaying notable error and a sample of accurate sequences.

it was run on the highway sequences of the ECMD dataset [3]. With the absence of any resets the root mean square error for 10 of the 12 run sequences ranged from 2.91 degrees to 5.90 degrees, as shown in Tab. 1 and Fig. 6a. When considering the function of head direction cells, a common purpose of their behaviour is to modulate the activity of grid cells [4, 11, 15], which are selective to six directions, and therefore 60 degree increments. As such operation with low single digit error can be considered well within a functional operating range and the application of a visual reset is unnecessary for function.

Of the two sequences which experienced significant errors, a visual inspection of Fig. 6b shows that from 20,000-30,000ms a turn in the ground truth plot is not present in the neuronal. This causes the entire neuronal representation from this point onwards to maintain this error. By training

the visual reset mechanism to trigger a reset at 35,000ms (as seen in Fig. 6c) the ongoing effects of this error are minimised, reducing the root mean square error by approximately 10 degrees.

5. Discussion

The applicability of neuromorphic computing combined with event-based sensing has been introduced to autonomous driving scenarios. A model demonstrating general accuracy has been presented for tracking orientation with error correction using a visual reset triggered by stereo event cameras. The error correction mechanism has been proven to account for both large errors caused by erroneous readings and smaller errors caused by drift over time.

Due to this work fitting within the emerging field of neuromorphic computing there is scope for it to be extended

upon in a number of ways. Due to the application of autonomous driving, applying the same principles to path integration would make the system of considerably more use and notably closer to a system which is usable for autonomous driving. Extending this work while still focusing on direction also offers numerous possibilities such as increasing the dimensionality to represent 3D direction; improving the accuracy of the network for a greater range of angular velocities; and adding autonomy to the training process for building visual reset associations.

References

- [1] Zhenshan Bing, Amir El Sewisy, Genghang Zhuang, Florian Walter, Fabrice O. Morin, Kai Huang, and Alois Knoll. Toward cognitive navigation: Design and implementation of a biologically inspired head direction cell network. *IEEE Transactions on Neural Networks and Learning Systems*, 33:2147–2158, 5 2022. Head direction network, same as mine, no IHD layer. [2](#)
- [2] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240×180 130 db 3 μ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49:2333–2341, 10 2014. Presenting DAVIS. [1](#)
- [3] Peiyu Chen, Weipeng Guan, Feng Huang, Yihan Zhong, Weisong Wen, Li-Ta Hsu, and Peng Lu. Ecmd: An event-centric multisensory driving dataset for slam. *IEEE Transactions on Intelligent Vehicles*, 9:407–416, 1 2024. ECMD Dataset. [1](#), [4](#), [5](#)
- [4] Klara Gerlei, Jessica Passlack, Ian Hawes, Brianna Vandrey, Holly Stevens, Ioannis Papastathopoulos, and Matthew F. Nolan. Grid cells are modulated by local head direction. *Nature Communications*, 11:4228, 8 2020. grid cell firing rate can depend upon head direction. [5](#)
- [5] Raphaela Kreiser, Matteo Cartiglia, Julien N.P. Martel, Jorg Conradt, and Yulia Sandamirskaya. A neuromorphic approach to path integration: A head-direction spiking neural network with vision-driven reset. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2018-May, pages 1–5. IEEE, 4 2018. Head direction network and visual reset mechanism. [2](#)
- [6] Raphaela Kreiser, Alpha Renner, Yulia Sandamirskaya, and Panin Pienroj. Pose estimation and map formation with spiking neural networks: towards neuromorphic slam. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2159–2166. IEEE, 10 2018. position and direction using snn, inputs unsure? [2](#)
- [7] Peter König, Andreas K. Engel, and Wolf Singer. Integrator or coincidence detector? the role of the cortical neuron revisited. *Trends in Neurosciences*, 19:130–137, 4 1996. [1](#)
- [8] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128×128 120 db 15 μ s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, 43:566–576, 2008. DVS release. [1](#)
- [9] Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10:1659–1671, 12 1997. SNN review / intro. [1](#)
- [10] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck. Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102:1470–1484, 10 2014. [1](#)
- [11] David C. Rowland, Yasser Roudi, May-Britt Moser, and Edvard I. Moser. Ten years of grid cells. *Annual Review of Neuroscience*, 39:19–40, 7 2016. [5](#)
- [12] Biswa Sengupta and Martin B. Stemmler. Power consumption during neuronal computation. *Proceedings of the IEEE*, 102:738–750, 5 2014. [1](#)

- [13] JS Taube, RU Muller, and JB Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *The Journal of Neuroscience*, 10:420–435, 2 1990. [2](#)
- [14] Anup Vanarse, Adam Osseiran, and Alexander Rassau. A review of current neuromorphic approaches for vision, auditory, and olfactory sensors. *Frontiers in Neuroscience*, 10, 3 2016. [1](#)
- [15] Simon Nikolaus Weber and Henning Sprekeler. Learning place cells, grid cells and invariances with excitatory and inhibitory plasticity. *eLife*, 7, 2 2018. [5](#)
- [16] Xiaohui Xie, Richard H. R. Hahnloser, and H. Sebastian Seung. Double-ring network model of the head-direction system. *Physical Review E*, 66:041902, 10 2002. [2](#)
- [17] K Zhang. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *The Journal of Neuroscience*, 16:2112–2126, 3 1996. [2](#)