Feedback Control of Dynamical Systems using Neuromorphic Vision Sensors

by

Erich Mueller

B.S. Aeronautical and Astronautical Engineering, MIT, 2009
M.S. Aeronautics and Astronautics, MIT, 2012

Submitted to the Department of Aeronautics and Astronautics in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2015

©2015 Massachusetts Institute of Technology. All rights reserved.

Signature of Author: ______________________

Department of Aeronautics and Astronautics
December 09, 2015

Certified by: ______________________________

Emilio Frazzoli
Professor of Aeronautics and Astronautics
Thesis Supervisor

Certified by: ______________________________

Berthold Klaus Paul Horn
Professor of Computer Science and Engineering

Certified by: ______________________________

Andrea Censi
Research Scientist, Laboratory for Information and Decision Systems

Accepted by: ______________________________

Paulo Lozano
Associate Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee
ABSTRACT

The recent development of neuromorphic vision sensors, which provide an asynchronous, high-speed alternative to conventional cameras has lead to a considerable amount of research into their applicability to robotic control systems. However, algorithms for onboard control of mobile robotic platforms such as automobiles or aircraft using these sensors are lacking and in fact almost all existing implementations keep the sensor stationary. This research has several objectives. First, to develop a rigorous understanding of how to use asynchronous temporal contrast vision sensors for heading regulation and tracking in such a way as to fully leverage the remarkable properties of these sensors including high bandwidth, low latency and low power consumption. Second, to provide a theoretical and experimental comparison between neuromorphic vision sensors and conventional cameras in the context of this problem. Finally, to describe and test algorithms for high-speed motion planning in cluttered environments using neuromorphic vision sensors.

Thesis Supervisor: Emilio Frazzoli
Title: Associate Professor of Aeronautics and Astronautics
"The natural state of the mind is to dream, and every once in a while we constrain the dream with some input."

- Jean Jacques Slotine
## Contents

1 Introduction .............................................. 11
   1.1 Overview .............................................. 11
   1.2 Motivation ........................................... 12
   1.3 Existing Methods and Related Work .................... 13
      1.3.1 Heading Regulation and Visual Servoing .......... 13
      1.3.2 Motion Planning in Cluttered Environments ....... 13
      1.3.3 Sensor Architectures .............................. 14
      1.3.4 Algorithms ....................................... 14
      1.3.5 Using Vision ..................................... 15
      1.3.6 Dynamic Vision Sensors ........................... 16
      1.3.7 Event Triggered Control ........................... 18
      1.3.8 Forest Flight Problem ............................ 19
   1.4 Contributions ........................................ 20
      1.4.1 Outline .......................................... 20
      1.4.2 DVS Modelling .................................... 20
      1.4.3 Events as Measurements ........................... 21
      1.4.4 Control Laws for Heading Regulation .............. 21
      1.4.5 DVS Versus Cameras .................................. 21
      1.4.6 Forest Flight ...................................... 21

2 Sensor Modelling ......................................... 22
   2.1 Dynamic Vision Sensors .................................. 22
      2.1.1 Conventional Cameras .............................. 23
      2.1.2 DVS .............................................. 23
   2.2 Noise Modelling ....................................... 24
   2.3 The Heading Regulation Problem ......................... 25
      2.3.1 Why low computation? .............................. 27
   2.4 DVS Model in the Heading Regulation Context .......... 28
      2.4.1 Pixel-level Output .................................. 29
      2.4.2 Sensor-level Output ................................ 30
      2.4.3 Stream Probabilities ................................ 35
   2.5 Events as Configuration and Rate Measurements .......... 36
      2.5.1 Event Location and Configuration .................. 36
2.5.2 Event Rate and Heading Rate .............................................. 38
2.5.3 Event Polarity and Motion Direction ................................. 39
2.5.4 Delay ................................................................. 40
2.5.5 Asynchronous Measurement Model ................................. 41
2.5.6 Continuous-Time Sensing ............................................ 41
2.5.7 Model Summary and Discussion ..................................... 42
2.6 Discussion ................................................................. 44

3 Algorithms and Analysis .................................................. 48
3.1 Experimental Setup ....................................................... 48
3.2 System Identification ..................................................... 51
  3.2.1 Plant Dynamics ....................................................... 51
  3.2.2 Identification ....................................................... 52
  3.2.3 Control Gains ....................................................... 54
  3.2.4 Sensor Lag .......................................................... 56
3.3 Local Estimation Algorithms for Complex Environments .......... 60
  3.3.1 Single-Edge Brightness Profile ................................ 61
  3.3.2 Piecewise Constant Brightness Profiles ......................... 62
3.4 Global Maximum Likelihood Estimation .............................. 65
  3.4.1 Estimation Algorithms ............................................ 66
  3.4.2 Computational Time and Space Complexity ...................... 69
  3.4.3 Calibration ........................................................ 69
  3.4.4 Experimental Verification ........................................ 70
3.5 Filtering to Reduce Delay ............................................... 72
  3.5.1 Fixed Lag Model .................................................. 83
3.6 Open-Loop Model Assessment ........................................... 84
  3.6.1 Heading Measurements .......................................... 85
  3.6.2 Heading Rate Measurements ..................................... 88
3.7 Feedback Control with DVS ............................................. 91
  3.7.1 Stability .......................................................... 92
  3.7.2 Proportional-Derivative Control ................................ 92
  3.7.3 Bounded Input Control .......................................... 94
  3.7.4 Switching Control .............................................. 95
  3.7.5 Experimental Results .......................................... 98
  3.7.6 Relation to Event Triggered Control ............................ 102
  3.7.7 Dithering and Microsaccades .................................. 103
  3.7.8 Summary ........................................................ 109
3.8 Applicability of Conventional Algorithms ............................ 110
  3.8.1 Framing Events ................................................ 110
  3.8.2 Incrementalized Algorithms .................................... 110
3.9 Discussion ............................................................... 111
# List of Figures

1.1 DVS128 sensor .......................................................... 16
1.2 DVS example data .......................................................... 17

2.1 Translational problem with brightness profile .......................... 25
2.2 Heading regulation problem .............................................. 27
2.3 DVS pixel output model .................................................. 30
2.4 DVS sensor output model ................................................ 31
2.5 Examples of the event rate multiplier $g_1(q,n)$ ......................... 33
2.6 Examples of the event rate multiplier $g_1(q,n)$ ......................... 34
2.7 Effect of edge smoothness on $g_0(q,n)$ ................................ 35
2.8 Spatial distribution of noise events across pixels ....................... 45
2.9 Pixel count distribution across number of noise events ............... 45
2.10 Noise event interarrival time distribution ............................ 46

3.1 Rotating drum testbed concept ......................................... 49
3.2 Rotating drum apparatus .................................................. 50
3.3 Rotating drum signals and delays ....................................... 51
3.4 Experimental plant frequency response ................................ 53
3.5 Observed and predicted static friction curve .......................... 54
3.6 Observed and simulated plant trajectories ............................. 56
3.7 The effect of sensor lag on measurements ................................ 58
3.8 Sensor lag determination .................................................. 59
3.9 Event locations vs heading angle ....................................... 60
3.10 “Cartoon Forest” test backdrop ........................................ 61
3.11 Example event map for local heading measurement .................... 63
3.12 Raw and mapped event locations ..................................... 64
3.13 Measurement error using mapped and raw event locations .......... 65
3.14 Global heading measurement calibration data ........................ 69
3.15 Typical estimator heading distributions ............................... 70
3.16 Global estimation - measured vs true heading ........................ 71
3.17 Double sided event wake trailing the edge ............................ 73
3.18 Event wake following an edge ......................................... 74
3.19 Expected lag as a function of filter parameters ........................ 79
3.20 Lag variance as a function of filter parameters ....................... 80
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.21</td>
<td>Expectation and variance of lag vs buffer size</td>
<td>81</td>
</tr>
<tr>
<td>3.22</td>
<td>Optimal frontier in filter parameter space</td>
<td>82</td>
</tr>
<tr>
<td>3.23</td>
<td>Maximum likelihood filter step response</td>
<td>83</td>
</tr>
<tr>
<td>3.24</td>
<td>Measurement error autocorrelation</td>
<td>86</td>
</tr>
<tr>
<td>3.25</td>
<td>High-speed measurement error distribution</td>
<td>87</td>
</tr>
<tr>
<td>3.26</td>
<td>The influence of heading rate on measurement error</td>
<td>88</td>
</tr>
<tr>
<td>3.27</td>
<td>Event polarity and sign of heading rate</td>
<td>89</td>
</tr>
<tr>
<td>3.28</td>
<td>Event rate used to measure heading rate</td>
<td>90</td>
</tr>
<tr>
<td>3.29</td>
<td>Event rate vs heading rate</td>
<td>91</td>
</tr>
<tr>
<td>3.30</td>
<td>Switching vs PD control simulated trajectories</td>
<td>96</td>
</tr>
<tr>
<td>3.31</td>
<td>Switching vs PD control steady state distributions</td>
<td>97</td>
</tr>
<tr>
<td>3.32</td>
<td>Closed-loop experimental electronic signals</td>
<td>99</td>
</tr>
<tr>
<td>3.33</td>
<td>Closed-loop step responses - PD and switching control</td>
<td>100</td>
</tr>
<tr>
<td>3.34</td>
<td>Closed-loop step responses - DVS and potentiometer</td>
<td>101</td>
</tr>
<tr>
<td>3.35</td>
<td>Closed-loop sine tracking - DVS and potentiometer</td>
<td>101</td>
</tr>
<tr>
<td>3.36</td>
<td>Closed-loop sine tracking with DVS</td>
<td>102</td>
</tr>
<tr>
<td>3.37</td>
<td>Dithering control stability proof diagram</td>
<td>105</td>
</tr>
<tr>
<td>3.38</td>
<td>Closed-loop responses with dithering</td>
<td>107</td>
</tr>
<tr>
<td>3.39</td>
<td>Closed-loop tracking error vs dithering frequency</td>
<td>108</td>
</tr>
<tr>
<td>3.40</td>
<td>Switching ZOH vs dithering steady state distributions</td>
<td>109</td>
</tr>
<tr>
<td>4.1</td>
<td>DVS and camera tracking error with optimal sensor parameters</td>
<td>123</td>
</tr>
<tr>
<td>4.2</td>
<td>DVS and camera optimal sensor parameters</td>
<td>124</td>
</tr>
<tr>
<td>4.3</td>
<td>DVS and camera optimal error with data rate constraint</td>
<td>126</td>
</tr>
<tr>
<td>4.4</td>
<td>DVS and camera optimal resolution with data rate constraint</td>
<td>127</td>
</tr>
<tr>
<td>4.5</td>
<td>Closed-loop sine tracking - DVS and camera</td>
<td>130</td>
</tr>
<tr>
<td>4.6</td>
<td>Closed-loop sine tracking - DVS and camera</td>
<td>133</td>
</tr>
<tr>
<td>4.7</td>
<td>The fundamental difference between DVS and cameras</td>
<td>135</td>
</tr>
<tr>
<td>5.1</td>
<td>Forest flight planning algorithm benchmarking</td>
<td>139</td>
</tr>
<tr>
<td>5.2</td>
<td>Using the absence of events to infer free space</td>
<td>143</td>
</tr>
<tr>
<td>5.3</td>
<td>Experimental free space estimation using absence of events</td>
<td>146</td>
</tr>
<tr>
<td>5.4</td>
<td>Screenshot from the forest driving simulation - LIDAR</td>
<td>149</td>
</tr>
<tr>
<td>5.5</td>
<td>Screenshot from the forest driving simulation - DVS</td>
<td>150</td>
</tr>
<tr>
<td>5.6</td>
<td>Forest flight collision data - sparse forest</td>
<td>153</td>
</tr>
<tr>
<td>5.7</td>
<td>Forest flight collision data - medium density forest</td>
<td>153</td>
</tr>
<tr>
<td>5.8</td>
<td>Forest flight collision data - dense forest</td>
<td>154</td>
</tr>
</tbody>
</table>
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Commonly used variables</td>
<td>28</td>
</tr>
<tr>
<td>3.1</td>
<td>Experimental parameters</td>
<td>98</td>
</tr>
<tr>
<td>3.2</td>
<td>Experimental components</td>
<td>98</td>
</tr>
<tr>
<td>4.1</td>
<td>Camera model parameters</td>
<td>115</td>
</tr>
<tr>
<td>4.2</td>
<td>Experimental parameters II</td>
<td>129</td>
</tr>
<tr>
<td>4.3</td>
<td>Experimental closed-loop performance metrics</td>
<td>131</td>
</tr>
<tr>
<td>5.1</td>
<td>Forest flight simulation parameters</td>
<td>151</td>
</tr>
<tr>
<td>5.2</td>
<td>Forest flight algorithms computation</td>
<td>154</td>
</tr>
<tr>
<td>6.1</td>
<td>Heading control summary of results</td>
<td>158</td>
</tr>
</tbody>
</table>
# List of Theorems

<table>
<thead>
<tr>
<th>Section</th>
<th>Theorem Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5.1</td>
<td>Model (Asynchronous DVS Measurement Model)</td>
<td>41</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Model (Continuous-Time DVS Measurement Model)</td>
<td>43</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Proposition (Lag Reduction for Stochastic Delay Systems)</td>
<td>75</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Model (Constant Lag DVS Measurement Model)</td>
<td>84</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Model (Zero Lag DVS Measurement Model)</td>
<td>84</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Model (Continuous-Time DVS and Noise-Free Rate Measurement Model)</td>
<td>92</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Definition (Stability)</td>
<td>92</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Theorem (Instability of PD Control)</td>
<td>92</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Lemma (Stability with Bounded Noise)</td>
<td>94</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Theorem (Stability of Switching Control)</td>
<td>95</td>
</tr>
<tr>
<td>3.7.3</td>
<td>Theorem (Stability of Dithering Control)</td>
<td>104</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Theorem (Optimal Sensor Performance)</td>
<td>118</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Overview

Neuromorphic vision sensors, of which dynamic vision sensors (DVS) are a particular type, have recently become available and been demonstrated to have properties that are beneficial in a number of applications, including automatic control systems. DVS, also known as silicon retina, are similar to conventional cameras in that they have optics which focus light onto an array of pixels, however unlike cameras they provide an asynchronous stream of data corresponding to changes in brightness rather than regular frames of brightness data. These sensors have yet to be used onboard for motion planning for autonomous vehicles in unknown, unstructured environments. This work focuses on the development of algorithms and theory to address the application of this new type of sensor to problems related to motion control. An abstract theoretical model of the operation of the sensor in the context of the heading regulation task, derived from pixel-level behaviour, is presented, analyzed and used to develop control laws that can deal with the unique noise properties of the sensor. Algorithms incorporated into the model that allow control laws to be implemented across a range of conditions are also presented. In order to assess the accuracy of the models presented and the efficacy of the algorithms and control laws developed, results from a series of experiments are presented and discussed. These tests include simulation and real-world testing, and allow for both open and closed-loop evaluation. As described, the first major question addressed by this work is:

How can DVS be used for heading regulation and tracking in known environments?

In addition to building an understanding of how we can use these remarkable sensors for the task of heading control, we would also like to understand whether doing so is a good idea. Intuitively DVS have nice properties, but a deeper analysis is required to support the claim that they can outperform conventional cameras at constant mean computation. Thus, the second major question addressed is:

Are DVS actually better than conventional cameras for the proposed tasks?
The objective of this thesis is to answer these two questions through a combination of modelling, novel algorithm development, control design, analysis and experimentation. In addition to exploring the heading regulation problem in depth, results related to the much more complex task of controlling an unmanned aerial vehicle to fly through a forest at high-speed are also presented.

Following the introduction, chapter 2 begins by detailing the notation and problems addressed in the remainder of the document, and uses the low-level principles of operation of an ideal DVS to derive a measurement model for the DVS in the context of heading regulation. Chapter 3 describes the experimental apparatus used for testing, presents algorithms that allow the measurement model to be extended to more general cases, and describes control laws used to stabilize the model, all while incorporating experimental results for verification. Chapter 4 details the theoretical and experimental comparison between DVS and conventional vision sensors, and finally chapter 5 describes the forest flight problem along with proposed DVS algorithms and simulation results. Chapter 6 summarizes the contributions of this work and provides a discussion of the results and future research directions.

1.2 Motivation

The motivation for this work comes from the incredible properties of DVS including extremely low latency and the possibility for very high bandwidth control when compared to commercially available cameras, combined with the fact that relatively little work has been done to develop rigorous methods for control based on these sensors.

Dynamic vision sensors have several characteristics that make them attractive as sensors for heading regulation or vehicle motion planning, as well as interesting from a theoretical perspective. Their output can be viewed as a highly compressed, asynchronous stream that excludes much of the less useful data produced by conventional cameras. In fact, this stream can in many cases be compared to the output of a conventional camera with a moving edge detection filter applied—an operation in general requiring substantial computation. In addition to providing a significant degree of hardware level data compression, the nature of their operation allows them to detect motion with very low latency and high effective frame rate, making them potentially useful for high bandwidth control problems such as helicopter control or rapid flight through a cluttered environment. Furthermore, when compared with conventional vision sensors (cameras), their output does not theoretically change with a scaling of time, so algorithms may be applicable over a wide range of operating speeds. High-speed feedback control of dynamic systems using DVS has been demonstrated for controlled situations in which the sensor remains fixed.

The problem of controlling the heading (angle) of the sensor has applications including star tracking for astronomical observation or navigation, aesthetic image stabilization, and missile guidance. In many cases, deployment on a power or mass constrained platform means that minimizing the computation required to achieve a given level of performance is important, and therefore a sensor and algorithms that can reduce necessary power consumption would be valuable.

High-speed autonomous flight through a forest could enable small unmanned aerial vehi-
cles to perform reconnaissance, capture footage for film-making, or execute search and rescue missions. Small aerial vehicles are subject to tight power and mass constraints that translate directly into computational limitations, and so using a sensor that requires much less computation to provide the same level of control would be extremely beneficial.

Numerous results from the field of event based control also suggest that event triggered sensing similar to that used by DVS is superior to the fixed interval discrete-time sensing used by cameras. Specifically, event triggered sensors are shown to require fewer measurements on average to provide the same control performance, or to provide superior control performance using the same mean sampling rate, when compared to periodic sampling.

1.3 Existing Methods and Related Work

1.3.1 Heading Regulation and Visual Servoing

One of the primary problems considered in this research is that of heading regulation and tracking using neuromorphic vision sensors. This is the task of actuating the sensor to follow a moving target or hold a particular view fixed as it is subject to unpredictable disturbances. In the literature this is often referred to as visual servoing, for which there is a large body of existing work [1, 2, 3]. These methods generally compute control commands directly from observation of image landmark features, or, alternatively, “moments”, without constructing a map or performing explicit state estimation. The identification of image features can be a computationally intensive task, and capturing synchronous frames leads to latency on the order of the frame period. In low light conditions, motion blur can also affect performance. Therefore, while parts of the formalization, specifically the mathematics of the image-formation process, are essentially the same, the fact that we use asynchronous events renders existing techniques not directly applicable.

1.3.2 Motion Planning in Cluttered Environments

Motion planning is a fundamental problem in robotics and has been extremely well studied from both theoretical and practical perspectives, and addressed in a wide variety of applications including walking robots [4], urban vehicles [5], helicopters [6] and others. It is the problem of trying to plan the motion of a dynamical system to reach some goal while avoiding undesirable configurations of the system. A common example of such a problem is the task of steering a robot through a cluttered environment, from its initial location to a specified goal location, while avoiding obstacles. If the state of the system is fully described by a finite-dimensional configuration and we are able to command directly the time derivative of the configuration, then this task can be formally specified as that of finding a continuous path from some initial configuration to any point in a specified set of goal configurations such that the configuration at no time enters a set of disallowed configurations. There are numerous algorithms that have been proposed, studied, and implemented for solving this problem. Randomized graph or tree methods such as probabilistic roadmaps (PRM) [7] and rapidly exploring randomized trees (RRT) [8] incrementally construct a graph connecting points in the free space which can then be used for motion planning. Potential field approaches such
as that described in [9] and [10] drive the system to descend an artificially constructed cost function that takes on high values inside obstacles and reaches a minimum at the goal. For low-dimensional systems, constructs such as Voronoi diagrams [11] and visibility graphs can be used given an explicit polygonal representation of the obstacle set. General cell decomposition methods work by dividing the free space into cells, computing connectivity relationships, and using some discrete space planner such as Dijkstra's algorithm or A*. For a survey of many of the methods that have been developed for motion planning in known environments, see [12].

In general the solution to such problems is not unique, and it is often desirable to find the optimal allowable path to the goal. In such a case, the path functional describing optimality is often the total time [13], the energy or fuel consumed [14], or some combination of these quantities [15]. Optimal motion planning for the purely kinematic case can be achieved using several of the methods described above, as well as several modifications that have been developed to support optimality. One important example that will be of use in this work is RRT* [16], which extends RRT with an additional optimization step that maintains asymptotic computational bounds while producing an asymptotically optimal solution. Furthermore, there are many systems for which such a simple motion model does not well apply, and additional dynamical constraints must be imposed upon the path in order to ensure that it is feasible. Motion planning with dynamical constraints and obstacles can also be achieved by modified versions of RRT or PRM such as LQR-trees [17] or LQR-RRT* [18], or simply constrained versions [19] [20].

### 1.3.3 Sensor Architectures

From a practical perspective motion planning problems in general rely on the ability of the system to observe its environment through some type of sensor architecture. There are a variety of sensors suited to robotic applications, with advantages and disadvantages. In implementation, systems have commonly made use of scanning laser range finders [21], wheel odometry [22], GPS [23], RSSI for localization [24], inertial sensors, and others. Cameras are a particularly lucrative type of sensor to be used for mobile robot applications, as they are small and inexpensive, and the ubiquitous use of vision by animals is a testament to the potential it has for motion control applications. As such, vision sensors have been frequently used in motion planning applications such as with mobile robots [25] [26] and robotic manipulators [27].

### 1.3.4 Algorithms

In many applications, it is unreasonable to assume a map of obstacles is known in advance, and motion planning algorithms that do not know the exact structure of the environment in advance are required. Rather, a more realistic model is one in which the vehicle is equipped with a set of onboard sensors that provide enough information to allow local collision avoidance, in addition to a GPS-like sensor that enables long distance waypoint tracking. Receding horizon control methods that assume a limited sensing radius have been developed to address the problem of planning in an initially unknown environment [28] [29] [30]. Randomized tree
planners have also been extended to consider limited sensing range and uncertainty in the model of the environment [31]. A rich family of algorithms have been developed for the problem of simultaneous localization and mapping (SLAM), using a variety of sensor systems [32], [33], which attempt to resolve the problem of limited sensor range by constructing a map of the environment during operation. While SLAM is a general procedure that can be and has been used under different sensor architectures, it is well suited to vision [34] [35] [36].

1.3.5 Using Vision

Cameras offer a number of advantages as sensors for mobile robotic applications, and many algorithms have been developed for motion planning and environment mapping using one or more cameras specifically.

The problem of reconstructing an initially unknown environment using camera data has been extensively studied, and can be addressed using several different methods. When multiple cameras are used, points or regions can be matched between frames from different cameras, and three-dimensional locations can be computed using triangulation [37] [38] [34]. With a single moving camera, a similar procedure can be performed where regions are matched between frames captured by the same device in order to estimate depth and camera motion simultaneously [39] [40]. When the motion of the camera through the environment is used for three-dimensional reconstruction this problem is generally known as structure from motion [39].

Methods for estimating depth from local texture cues in a single image have also been shown to be effective under some circumstances, and have seen some success with the application of supervised learning algorithms to predict depth given a training set with measured depths [41]. Depth can also be computed using image defocus [42] [43], provided the aperture size is relatively large so as to make the depth of field in the image sufficient for such purposes. Shape from shading [39] can be applied to estimate depth gradients from images with known lighting conditions. Combinations of cues such as monocular texture and stereo triangulation have been shown to provide superior performance to a single method approach [44]. One of the most common modern methods for environment estimation is to match specific points of interest (features), between frames captured by different cameras or at different times. These methods have often been applied to simultaneous localization and mapping problems similar to that considered in this work [34] [36] [35].

Feedback control methods based on cameras that do not rely on the intermediate step of estimating depth or three-dimensional obstacle locations have also been developed. One such approach to the problem of autonomous forest flight has been to use supervised learning to train a controller to emulate the action of an expert human pilot by providing it with a large set of training data matching camera frames to control actions and training it to generate an appropriate control given a new frame of data [45]. In general, supervised learning techniques are limited by the need to specify a set of features to be used for classification.
1.3.6 Dynamic Vision Sensors

Figure 1.1: DVS128 sensor by iniLabs - a commercially available dynamic vision sensor with 128x128 pixel resolution and USB interface.

Overview

While conventional cameras capture full frames at regular intervals (the frame period), outputting a raster of brightness data representing the total amount of light incident on each pixel over some fixed duration of time (the exposure time), temporal contrast sensors operate in an entirely different, asynchronous manner. Such sensors are structurally and optically identical to conventional cameras, but instead output a series of events at arbitrary times, with each event carrying a timestamp, two-dimensional pixel location, and polarity bit. An event is generated each time the brightness at a pixel changes, up or down, by some amount relative to the brightness observed at the time of the previous event from that pixel, and indicates the time, location, and direction (up or down) of the brightness change [46] [47] [48] [49] [50].

Recently developed dynamic vision sensors also provide the ability to sample the brightness level when an event is triggered, or interleave periodic frame-based brightness measurements with events using the same chip [51] [52] [53]. These sensors afford the additional possibility of incrementally reconstructing brightness levels from events, and are more accurately described by models from the theory of event triggered control.
Figure 1.2: Example data from a dynamic vision sensor, accumulated over a short period of time, and ignoring the polarity of events. The events were captured with the sensor fixed, looking at a rotating mirror. The DVS itself is seen as the large circle at the top, with a person on the right and windows and a bicycle on the left.

The development of dynamic vision sensors has been motivated to some degree by the functioning of biological vision systems, in particular the retina found in the eyes of many organisms, which contain photosensitive cells that respond to changes in illumination. Early tests with human subjects have been used to show the dependence on motion of the visual system, and to explain microsaccades - imperceptible motions of the eye executed while seeming looking at a fixed target [54] [55] [56]. Perhaps the capabilities of a control system based on dynamic vision sensors could more closely match the incredible capabilities of natural control systems.

Dynamic vision sensors have a number of distinct advantages over conventional cameras. The threshold imposed on brightness changes means that typically only the image edges are ever seen, vastly reducing the total amount of data output by the device and increasing the effective frame rate. In addition to high bandwidth, the asynchronous nature of the event generation allows for very low latency sensing, and their sensitivity to motion makes them particularly applicable to scenarios where only moving regions of the image are of interest. While in general the framed output of a moving conventional camera varies as the velocity of
the sensor changes, the sequence of events output by a dynamic vision sensor should simply be scaled in time, ensuring that algorithms effective at one velocity are effective across a wide range of velocities. The advantages of dynamic vision sensors over conventional cameras are summarized as,

- Hardware-level compression means high control bandwidth or low computation,
- Asynchronous event triggering means low latency reaction,
- Event driven approach means time scale invariance of sensor output,
- Low power consumption.

Control Applications

Though they have only recently been developed, the use of dynamic vision sensors in a variety of applications has been studied, including particle tracking [57], gesture recognition [58], and pedestrian and vehicle classification [59] [60]. They have also been used successfully for a number of motion control applications including pole balancing [61], microrobotic haptic feedback [62], ball interception [63], and robotic manipulation [64]. Algorithms for environment or state estimation based on dynamic vision sensors have also been explored, and include low latency off-board vehicle localization [65], visual odometry [66], stereo vision [67] [68] [69], optical flow [47] [70].

Analog neuromorphic sensors have been proposed for robotic control algorithms for some time [71] [72], however these simple sensors differ substantially from dynamic vision sensors in their operation.

Perhaps most pertinent to the application here under consideration are several recent algorithms for environment estimation. In [73], a system using laser pulses in combination with a dynamic vision sensor was used for terrain reconstruction, while [74] and [75] present work toward autonomous navigation in unknown environments. An offboard method for vehicle tracking is given in [65], but relies on the tracking of LEDs blinking at different frequencies. [76] and [77] describe the development of a system for autonomous vehicular lane following based on an event driven approach. Though [75] provides an algorithm for SLAM based on a single dynamic vision sensor, it is distinguished from this work by its reliance on the existence of a unique inversion to the optical projection (it uses features on a flat ceiling).

1.3.7 Event Triggered Control

Event triggered control refers to the control of systems based on measurements taken not at regular intervals or continuously in time, but at the occurrence of significant events. One of the most commonly studied scenarios is that where an event occurs whenever the error between the system state and the value of the state at the previous event is equal to a specified threshold value [78], [79], [80] [81], [82], [83]. The similarity of this formulation to that of using a dynamic vision sensor for control makes event based control an important notion for analyzing the behaviour of any control systems using these sensors. Event based control is often motivated by the desire to reduce communication requirements between the
sensors and the controller in a system [78], [81] - a goal also very much in line with that of using dynamic vision sensors.

Typical methods for controlling systems under such a sensor model use open-loop control signal generation between events in addition to corrections made due to events [80], and analysis of event based control systems indicate that they can often outperform periodic discrete-time controllers [84]. Experimentation with event based control schemes has been used to demonstrate their ability to handle substantially reduced data transmission while maintaining stability and robustness to model uncertainty and disturbances [85].

One interesting theorem for event based control systems states that for a system with continuous-time dynamics, given a stabilizing continuous-time controller for the system and the associated Lyapunov function proving global closed-loop asymptotic stability, an event based controller using the same feedback control law and updating the control signal upon the triggering of events will also be globally asymptotically stable if the events are triggered any time the rate of decrease of the Lyapunov function becomes zero [82], [86], [87], [88]. This approach has been used to derive analytic conditions for the maximal event trigger threshold that will maintain stability, and to show that this approach results in fewer sensor measurements than periodic sampling. Of particular interest to the application of event triggered control theory to DVS is the extension of such results to output feedback control systems [89], [90], [91], as well as distributed asynchronous systems in which different components of the state or output generate events asynchronously [92], [93], [94], [95], [96], [97]. Of particular relevance to DVS are [96] and [97], which consider the case of asynchronous output-feedback event triggering that models very well newly developed DVS which provide brightness measurements when events occur.

The results from event triggered control are largely motivational to the problem at hand, suggesting that improved performance is possible when compared to a periodic sensor like a camera. In [84], it is shown that for a simple tracking problem driven by noise, the mean squared tracking error achieved by an event triggered sensor is less than that of a periodic sensor using the same communication bandwidth (mean sampling rate). A similar analysis is performed in [98], yielding similar results. [99] and [100] formulate the problem of optimal threshold selection as the minimization of a weighted sum of mean squared tracking error and number of events generated, and shows that the optimum is lower for an event triggered sensor. In fact, the notion of event triggered control has been investigated as long ago as the 1960s, where the problem of adaptive timestepping for control systems [101], [102], [103] was investigated and shown to be qualitatively better than fixed timestep sampling on the basis of both tracking and event rate. The notion of trying to achieve equal changes in the signal level between samplings was described in [104], which is a similar concept to modern self-triggered sampling [105], [106].

### 1.3.8 Forest Flight Problem

A specific motion planning problem that will serve as the motivation for a large part of this work is that of flying an unmanned aerial vehicle (UAV) through a forest environment. This problem has applications to exploration and reconnaissance and has been the subject of a considerable amount of research spanning a number of different fields in recent years. [107]
considers the sub-problem of avoiding collisions with obstacles directly in the line of flight, while [45] considers a general supervised learning approach to motion planning in the forest and [108] proposes an algorithm based on a simplified dynamical model with limited sensing range. Motion primitives selected specifically for this problem are described in [109]. Many generalized motion planning algorithms using real sensors would be applicable to this problem as well.

The forest flight problem has been studied from a theoretical perspective, with results published regarding theoretical bounds on the performance of a vehicle flying through the forest [110]. These results suggest that, for a vehicle with idealized dynamics, in the space of forest density and forward velocity, there is a sharp threshold, below which it is very likely that the vehicle would be capable of flying indefinitely, and above which it is likely to crash in a short period of time.

1.4 Contributions

1.4.1 Outline

This thesis is intended to demonstrate several contributions to the existing body of literature surrounding the use of dynamic vision sensors for motion control. The primary contributions of this work are presentation of:

1. A model for the operation of the DVS in the context of heading regulation that incorporates the defining characteristics of the sensor including noise and is amenable to theoretical analysis. (Sections 2.1, 2.4, and 2.5)

2. Algorithms that allow the model to apply in a wide range of cases, specifically in a variety of environments. (Sections 3.3 and 3.4)

3. Control laws that stabilize the system under the sensor model given. (Section 3.7)

4. A theoretical and experimental comparison between DVS and conventional cameras as applied to heading regulation and tracking. (Chapter 4)

5. Algorithms for high-speed forest flight using DVS, along with simulation results comparing DVS and cameras. (Chapter 5)

1.4.2 DVS Modelling

Low level models of the pixel-by-pixel operation of dynamic vision sensors are not particularly useful as is for designing higher level control laws and analyzing their performance. One of the main contributions of this thesis is to present a model for the operation of the DVS in the context of heading regulation and tracking that can be used to design stabilizing feedback controllers. The model presented captures the defining characteristics of the DVS, specifically the idea that without motion the sensor cannot see, but is simple enough to be useful for control design.
1.4.3 Events as Measurements

Building on the model presented in earlier sections, which relied on extremely simplifying assumptions about the environment, the thesis also presents algorithms that allow events to be treated as measurements of heading (as described by the model) across a range of different environments or tracking targets. Extremely efficient local methods that require only a small constant factor more computation than in the trivial case are presented, followed by slightly more complex methods that allow global measurement and are still tractable enough to operate on an event-by-event basis. open-loop experimental results for model validation are also presented and discussed.

1.4.4 Control Laws for Heading Regulation

Using the models developed in previous sections, control laws for heading regulation are presented and analyzed. The major difficulty in designing control laws for use with the DVS is that while we generally want to stabilize a system to a stationary configuration, the lack of motion in such a state means that the DVS is unable to observe the configuration. In our models this appears as a configuration dependent noise whose variance becomes infinite when there is no motion and must be specially accounted for. In the context of double integrator plant dynamics, we discuss why proportional-derivative control is unstable, and present a stable alternative that provides much better performance in simulation and reality.

1.4.5 DVS Versus Cameras

A rigorous theoretical and empirical comparison between DVS and cameras is presented and provides support for the claim that DVS are superior for heading regulation in certain circumstances. Specifically a comparison is made between the control or tracking performance of the different sensor classes when mean computational load is fixed, and each sensor is optimized with respect to its own parameters such as resolution and frame rate. Both measurement-only and closed-loop comparisons are included, followed by a discussion of the tradeoffs between the two classes of sensor. Theoretical predictions for performance as a function of sensor parameters are compared to experimental observations, providing further support for the models used.

1.4.6 Forest Flight

Finally, the more complex task of high-speed forest flight that has originally served as motivation for this work and for the use of DVS for feedback control is addressed. While theoretical results in this area are not part of the contribution, algorithms are presented and simulation results comparing their performance to cameras and scanning laser rangefinders are given.
Chapter 2

Sensor Modelling

This chapter begins by describing the basic principles of operation and issues surrounding the use of dynamic vision sensors in feedback control systems, and gives notation surrounding DVS and cameras which will be used for the remainder of the document. Then, higher level models for the sensor are developed and discussed. These models will in section 3.7 be used for feedback control design.

2.1 Dynamic Vision Sensors

Dynamic vision sensors and cameras are similar in that they both consist of a lens which focuses light onto an array of pixels, however they differ significantly in the pixel-level operation as well as how data from the pixels is packaged and sent out of the sensor. Where a camera with a global shutter captures absolute brightness data from all pixels at the same time and packages it into frames for further processing, a DVS produces an asynchronous stream of events corresponding to relative brightness changes. This section gives ideal models for both sensors in the context of feedback control, which will be used for the remainder of the analysis and algorithms presented in this thesis.

Both types of sensors share a common image formation model, which describes the process by which light is focused onto the pixel array and thereby relates the environment to the sensor output. Let \( y(s, t) : \{1, 2, \ldots, n_x\} \times \{1, 2, \ldots, n_y\} \times \mathbb{R} \rightarrow \mathbb{R} \) denote the logarithm of brightness of the image at a pixel located at position \( s \in \{1, 2, \ldots, n_x\} \times \{1, 2, \ldots, n_y\} \) in the array and time \( t \in \mathbb{R} \).

While \( y(s, t) \) denotes the logarithm of brightness, for brevity we will refer to it simply as the brightness function and the quantity that it represents as brightness. The distinction between log brightness and brightness will not be important for our purposes and will generally be ignored.

This brightness function encapsulates any point spread function associated with the optics and finite pixel size to give the brightness level that would be measured by pixel \( s \) at time \( t \). While \( n_x \) and \( n_y \) denote the dimensions of the sensor's pixel array in pixels, we will often consider a one-dimensional sensor, in which case \( n \) will be used to denote the dimension and a subscript will be used to indicate the pixel, so that \( y(s, t) \) becomes,
\[ y(t) = [y_1(t), y_2(t), \ldots, y_n(t)]^T. \] (2.1)

2.1.1 Conventional Cameras

A camera will be modelled as measuring, for each pixel, the integrated brightness over a short period of time, and recording the result of doing so simultaneously across all pixels into a frame of data. We will denote the \(i\)th frame \(Y'(s) = \{1, 2, \ldots, n_x\} \times \{1, 2, \ldots, n_y\} \mapsto \mathbb{R}\), where

\[ Y'(s) = \int_{i\Delta}^{i\Delta + \nu} y(s, t) dt. \] (2.2)

The exposure time is \(\nu\) and the frame period is \(\Delta\). These frames, containing brightness data for all pixels in the array, are captured and sent to the controller at regular intervals of \(\Delta\) units of time. In practice they may be acquired by the controller (computer) synchronously, so that no image processing begins until the entire image is received, or asynchronously, so that the image may be processed line-by-line as it is received. We will assume that the time required to transmit the image to the computer is small compared to \(\Delta\), and therefore that this difference is negligible.

2.1.2 DVS

The output of a DVS is an asynchronous stream of events associated with changes in brightness, in which each pixel acts independently, and the events contain a timestamp, pixel location and polarity. For an idealized model of a DVS, each pixel with coordinate \(s\) is said to have internal state \(\hat{y}(s, t)\), and trigger an event whenever

\[ |\hat{y}(s, t) - y(s, t)| \geq h, \] (2.3)

where \(h\) denotes the event threshold, a fixed sensor parameter somewhat analogous to frame rate for a camera. If this occurs when \(y(s, t) > \hat{y}(s, t)\) it is said to have positive polarity, and conversely will be said to have negative polarity. When pixel \(s\) triggers an event, the internal state is updated according to

\[ \hat{y}(s, t) \leftarrow y(s, t) + v, \]

where \(v\) represents noise in the process which precludes reconstruction of the absolute brightness levels from the sequence of events. The internal state is initialized according to

\[ \hat{y}(s, 0) = y(s, 0) \quad \forall s. \]

Except at event updates, the internal state evolves according to the pixel adaptation equation,

\[ \dot{\hat{y}}(s, t) = \kappa(y(s, t) - \hat{y}(s, t)), \] (2.4)

where \(\kappa \geq 0\) is the pixel adaptation parameter that determines how quickly the internal state of the pixels track the brightness to which they are exposed. For \(\kappa = 0\), each pixel produces
an event when the brightness at that pixel deviates at least $h$ units from the brightness at the
time of the previous event, and the internal state of each pixel does not vary between events.
Increasing $\kappa$ results in a sort of high-pass filtering in which slow changes in brightness do not
trigger events while faster changes of the same magnitude do. While in practice these sensors
may be subject to this kind of pixel adaptation (i.e. $\kappa \neq 0$) which leads to the property
that sufficiently slow changes in brightness will not trigger events, we will generally restrict
ourselves to the consideration of ideal sensors for which the pixels do not adapt ($\kappa = 0$).

Events are defined by the Address-Event Representation, in which each event is a triple
containing the time it occurred, $t$, its pixel location, $s$, and its polarity (sign of brightness
change), $b$. Thus the $i^{th}$ event is defined,

$$e_i = (t_i, s_i, b_i) \in \mathcal{E},$$

where $\mathcal{E}$ denotes the set all possible events (data type of event).

At a sensor level, the events from all pixels are aggregated into a stream that is sent
asynchronously to the controller, so that the controller receives one event at a time in the
correct chronological order. Transmission of the events from the sensor to the computer may
require some time and therefore introduce some lag, and this topic will be discussed in greater
detail in chapter 3.

In practice, what this model means is that the sensor can only see motion in the scene,
and will not produce events in regions of the image will little or no motion. The thresholding
effect also means that only relatively high contrast regions will produce events, which can be
thought of as a low level form of detecting moving edges.

## 2.2 Noise Modelling

While an idealized dynamic vision sensor responds to changes in brightness by producing
events and does not produce events otherwise, in practice a considerable number of spurious
events are output by the sensor, and we would like to model the process by which such false
events are generated. This section presents a series of increasingly complex models that can
be used to represent the spurious event generation process with increasing fidelity, as well as
a procedure for estimating the model parameters.

The dynamic vision sensor, also known as the silicon retina, is a type of neuromorphic
vision sensor, and most of the models presented are based on ideas derived from the study of
biological neurons. Specifically, the event activity of neurons has been modelled as the result
of a level crossing condition for some diffusion process, as in [111] and [112]. These models
use an Ornstein-Uhlenbeck process, which is a modification of the well known Wiener process,
to model the diffusion that triggers a neuronal spike at the crossing of a threshold, and reset
the diffusion process when an event is produced. The statistics of the inter-event time are
studied in [113], and methods for parameter estimation of such a model are presented in [114],
[115] and [116]. It is also possible to use a similar model based on a Wiener process, which
has not been as extensively studied in the area of neuronal modelling, but has been used in
the problem of survival time estimation as in [117], [118] and [119]. Hitting times for Wiener
processes have also been studied from a theoretical perspective in [120] and [121].
While it is generally acknowledged that current state-of-the-art DVS are subject to noise in the form of spurious events, and several ad-hoc methods have been proposed (which are fundamentally similar) [122], [123], [47] for filtering these events, to the authors knowledge no rigorous modelling of the process behind this noise has been done. Nonetheless, this work will use a relatively simple noise model based on independent Poisson processes, and filtering will be achieved in a way that makes no distinction between noise and meaningful events.

2.3 The Heading Regulation Problem

The first problem considered is that of stabilizing the heading of a sensor attached to a moving vehicle with respect to objects in the field of view. Birds and other animals perform active stabilization of their view through motion of their head and eyes, and many algorithms for motion planning with vision can benefit from stabilization of the image on the field of view. Equivalently, the problem can be thought of as that of actuating the sensor to track a target following some unknown arbitrary reference trajectory.

For simplicity, we consider a one-dimensional sensor, and the case where the sensor is mounted to a translationally fixed, rotating platform in a static environment. It will also be assumed that the angular separation between pixels is equal for all pixels in the sensor. Based on this assumption, the problem is very similar to a translational problem with an orthographic projection and this is the formulation that will be used in order to avoid periodic conditions arising from rotation and work instead with variables taking on arbitrary real values. This transformation is depicted in figure 2.1.

Figure 2.1: For simplicity the rotational problem is converted into a translational problem with orthographic projection. An example continuous scene brightness profile $m(\cdot)$ is shown also.

A one-dimensional vision sensor moving along a single axis is studied, and the environment is defined by a static continuous brightness profile $m : \mathbb{R} \mapsto \mathbb{R}^+_0$ that gives the brightness observed at each location. Let the position of the sensor be denoted $q(t) \in \mathbb{R}$, the position
of pixel \(i\) relative to the position of the sensor be \(s_i \in \mathbb{R}\), and the point spread function for the pixel be \(\gamma: \mathbb{R} \mapsto \mathbb{R}_+^d\), so that for a continuous-time pixel, the brightness output at time \(t\), \(y_i(t)\) is,

\[
y_i(t) = \int_{-\infty}^{\infty} \gamma(\xi)m(q(t) + s_i - \xi)d\xi.
\] (2.5)

The point spread function \(\gamma(\cdot)\) defines how light is projected onto the sensor plane and how the sensor creates the discretized (pixelated) image from the light that reaches the sensor. Specifically it encodes the contribution of the underlying brightness at point \(\xi\) to the measurement made at \(0\). Summing these contributions leads to equation 2.5, where the brightness at pixel \(i\) is determined by convolving the underlying brightness \(m\) with the point spread function \(\gamma\). For the remainder of the discussion we will consider a sensor with perfect focus, so the point spread function will be considered a pulse of finite width \(2\rho\) so that,

\[
y_i(t) = \frac{1}{2\rho} \int_{-\rho}^{\rho} m(q(t) + s_i - \xi)d\xi \quad i \in \{1, 2, \ldots, n\}.
\] (2.6)

The value of \(\rho\) is half the pixel width, which is the angle subtended by the light sensitive area of the pixel and is distinct from the pixel spacing. The \(n\) pixels will be assumed to be equally spaced across the field of view \(L\), so that

\[
s_i = \frac{i - 1}{n - 1}L - \frac{L}{2},
\] (2.7)

and the pixel spacing is \(\Delta_s = \frac{L}{n} \geq 2\rho\). The fraction of the sensor pixel array consisting of light sensitive areas, known as the fill factor, is,

\[
\frac{2\rho}{\Delta_s} = \frac{2n\rho}{L}.
\] (2.8)

Now, the task is one of regulating the position to zero using a feedback control law computed incrementally whenever a new event is received. Because of the high total frequency of events seen in typical applications using DVS, the amount of computation performed for each new event must be minimal. Practically, event rates can exceed one million per second, so a controller requiring somewhat less than one microsecond of computation per event is desired.

We will consider a general dynamical system with \(p\)-dimensional state \(x(t) \in \mathbb{R}^p\), and assume that our state representation includes the position \(q\) as the first component (i.e. \(x_1(t) = q(t)\)). The additional components of the state will vary depending on the physical system considered, and in practice may represent the velocity \(\dot{q}\), motor current, rotor blade velocity, or any other quantities required to model the dynamics of the system. The dynamics of this system will be defined by the function \(f: \mathbb{R}^p \times \mathbb{R} \times \mathbb{R}^p \mapsto \mathbb{R}^p\) so that,

\[
\dot{x} = f(x, u, w),
\] (2.9)

where \(w(t) \in \mathbb{R}^p\) is an external process noise signal driving the system, \(u(t) \in \mathbb{R}\) is the control input chosen by the controller. Again since we consider a general dynamical system the precise nature of \(u(t)\) is unspecified, however in practice it may be the torque or voltage output of a
motor, the pressure in a pneumatic cylinder, the openness of a fuel valve, or whatever quantity is used as the control input in a dynamical model for the system. Recall the event triggering condition,

$$|y_i - \hat{y}_i| \geq h,$$  \hspace{1cm} (2.10)

indicating that pixel $i$ triggers an event when its brightness deviates more than some threshold $h$ from the level at the time of the previous event, so that when an event is triggered,

$$\hat{y}_i \leftarrow y_i + v,$$  \hspace{1cm} (2.11)

where $v$ is a random variable, identically distributed and independent across samples, modelling part of the sensor noise. The other part to the sensor noise is the presence of spurious events, which will be assumed to be generated by identical Poisson processes acting independently across pixels, as described in section 2.4. The objective is to design an event based controller that can regulate $x$ to $0$ under a variety of assumptions regarding the dynamics and the brightness function $m$.

Figure 2.2: Depiction of the DVS driven heading regulation problem. The image formation model is defined by equation 2.5.

Figure 2.2 shows the fundamental problem considered: controlling a known single-input single-output (SISO) dynamical system based on DVS feedback. In the absence of disturbances or reference tracking, with known initial conditions this problem can be solved by open-loop control, however closed-loop control allows an arbitrary unknown reference trajectory to be tracked and noise to be rejected. There are many possible approaches to this problem, and the emphasis will be on methods involving minimal computation per event.

2.3.1 Why low computation?

The algorithmic approaches described in this section are motivated by a desire to achieve the given control objectives with minimal computational load. Reduced computation means
reduced power consumption, which can benefit any mobile robotic system but is particularly pertinent to physically small systems such as micro air vehicles. Neuromorphic vision sensors can provide very high bandwidth and low latency control, and because it is passive the sensor itself consumes relatively little power. The DVS128 sensor chip itself consumes 23mW of power, compared for example to 2.5W for a small scanning laser rangefinder (Hokuyo URG-04LX-UG01, 5m range), and 150mW for a smartphone-type camera (FLIR Lepton).

If processing of the events can also be done using very little power, systems with high-speed control demands and stringent constraints on available power can be controlled. Agile micro air vehicles are a perfect example of such a system, requiring sensing capabilities beyond those of state-of-the-art conventional cameras to execute their most aggressive manoeuvres. As they continue to be miniaturized, control bandwidth requirements will increase while available power will decrease. Dynamic vision sensors, as they too become smaller and smaller, may provide a means by which such systems can autonomously navigate unknown environments while avoiding obstacles.

### 2.4 DVS Model in the Heading Regulation Context

The DVS model presented in section 2.1 does not account for the noise observed in practice by such sensors. While section 2.2 gives a brief overview of different ways in which noise acting on the system might be modelled, for the purposes of theoretical analysis we will focus on the relatively simple case in which noise is present in the form of fully exogenous spurious events added into the event stream for each pixel. This section will extend the model from 2.1 to account for noise of this form, and provide a higher level abstraction of the DVS in the...
context of heading regulation. In section 2.5, this model will be further extended to a linear sensor model with additive noise whose variance depends on the system configuration, and this model will be used in section 3.7 for feedback control design.

While we have characterized the noise present in dynamic vision sensors as coming from a spurious event process, observations from experimental data suggest that the spurious event generation process depends significantly on the history of legitimate events. As such, simple classification of events as “legitimate” or “spurious” is not accurate, as most events contain at least some information about changes in brightness.

In this section we describe the sensor output as a stream of “information-bearing” events merged with a stream of completely exogenous spurious events produced by a noise process independent of the brightness history. We will model the output of each pixel as the union of these two independent event streams.

### 2.4.1 Pixel-level Output

Recall the expression for the brightness output at pixel \( i \) and time \( t \):

\[
y_i(t) = \frac{1}{2\rho} \int_{-\rho}^{\rho} m(q(t) + s_i + \xi) d\xi.
\]  
(2.12)

Assuming \( m \) is continuous, the instantaneous rate of change of brightness at pixel \( i \) is

\[
\dot{y}_i(t) = \dot{q} \frac{\partial y_i}{\partial q} = \frac{\dot{q}(t)}{2\rho} (m(q(t) + s_i + \rho) - m(q(t) + s_i - \rho)).
\]  
(2.13)

In the absence of noise, we will model the rate of events produced by the pixel as being instantaneously proportional to this rate of change of brightness, scaled by the event threshold \( h \). This rate will be denoted \( \lambda_{IB} \), because these events are caused by brightness changes and therefore can be considered information bearing in the sense that they contain information about the scene.

\[
\lambda_{IB}(t) = \frac{c_0 |\dot{y}_i(t)|}{h} = \frac{c_0 |\dot{q}(t)| |g_i(q(t), \rho)|}{2\rho h}, \quad g_i(q(t), \rho) = |m(q(t) + s_i + \rho) - m(q(t) + s_i - \rho)|,
\]  
(2.14)

where \( c_0 \) is a constant factor scaling the rate of events independent of the environment, rate of motion, or threshold. The noise-free event generation process will be modelled as a Poisson process with time varying parameter \( \lambda_{IB} \) indicating the instantaneous rate of event production. While this model is not exact, since the events generated by this process are related through the image formation model and sensor motion, a Poisson process is used as an approximation in order to derive theoretical results.

Dynamic vision sensors are subject to a unique and interesting type of noise consisting of events generated spuriously in the absence of changes in brightness. This noise will be modelled as a completely exogenous Poisson process independent of the process generating the information bearing events. The background rate of spurious events can vary from pixel to pixel, and for pixel \( i \) the rate will be assumed to also be inversely proportional to the event threshold,
Figure 2.3: Pixel output model with spurious event noise. The event stream emanating from any pixel consists of the union of two independent Poisson processes: one containing information about the scene and the other pure exogenous noise.

Figure 2.3 depicts the stream of events generated by a single pixel as the union of the streams from the independent information bearing and spurious Poisson processes. As a result of their independence, the combined output is also a Poisson process with intensity parameter

$$\lambda^i(t) = \lambda^i_{IB}(t) + \lambda^i_S.$$  \hspace{1cm} (2.16)

### 2.4.2 Sensor-level Output

Now we will consider how the events from all pixels are aggregated to form the overall sensor output. We will assume that the Poisson processes describing the generation of events at each pixel are all independent, and therefore that the overall sensor output is a Poisson process with intensity $\Lambda(t)$,

$$\Lambda(t) = \sum_{i=1}^{n} \lambda^i(t) = \sum_{i=1}^{n} \lambda^i_{IB}(t) + \sum_{i=1}^{n} \lambda^i_S$$

$$= \frac{c_0|\dot{q}(t)|}{2\rho h} \sum_{i=1}^{n} |m(q(t) + s_i + \rho) - m(q(t) + s_i - \rho)| + \frac{1}{h} \sum_{i=1}^{n} \lambda^i_0$$ \hspace{1cm} (2.17)

$$= \frac{c_0|\dot{q}(t)|}{2\rho h} g(q(t), n, \rho) + \frac{\Lambda_0}{h},$$

where

$$g(q, n, \rho) = \sum_{i=1}^{n} |m(q + s_i + \rho) - m(q + s_i - \rho)|,$$ \hspace{1cm} (2.18)
Figure 2.4: DVS output model with spurious event noise. The output event stream consists of the union of two independent Poisson processes: one containing information about the scene and the other pure exogenous noise.

Figure 2.4 shows the sensor-level aggregated output as the union of an information bearing event process and a spurious event process. The function \( g(q, n, p) \) is dictated by the brightness profile (environment). Recalling from equation 2.8 that sensor fill factor is equal to \( 2n \rho L^{-1} \), we see that in the zero fill factor limit, \( \rho \to 0 \), and

\[
\frac{g(q, n, \rho)}{2 \rho} \to g_0(q, n) = \sum_{i=1}^{n} \nabla m(q + s_i),
\]

where \( g_0(q, n) \) represents the event rate multiplier in the zero fill factor case and \( \nabla \) represents the spatial gradient so that \( \nabla m(\xi) = \frac{\partial m}{\partial \xi} \). If the fill factor is 100\%, the pixel width is equal to the pixel spacing and the parameter \( \rho \) is eliminated using the relationship

\[
\rho = \frac{L}{2n},
\]

yielding

\[
\frac{g(q, n, \rho)}{2 \rho} = \frac{n}{L} g_1(q, n) = \frac{n}{L} \sum_{i=1}^{n} \left| m(q + s_i + \frac{L}{2n}) - m(q + s_i - \frac{L}{2n}) \right|
\]

where \( nL^{-1} g_1(q, n) \) is the event rate multiplier in the full fill factor (100\%) case. The relationship between the resolution \( n \) and the function \( g(q, n, \rho) \) for a fixed fill factor is complex and depends on the spatial scales of structure in the environment.

For the simple case of a single, perfectly sharp edge such that:

\[\Lambda_0 = \sum_{i=1}^{n} \lambda_i.\]
\begin{equation}
m(\xi) = \begin{cases} 
0 & \text{if } \xi < 0 \\
1 & \text{otherwise},
\end{cases} \tag{2.23}
\end{equation}

with 100% fill factor, \( g(\cdot) \) is a constant as long as the edge is within the field of view:

\begin{equation}
g_1(q, n) = \begin{cases} 
1 & \text{if } q \in (-\frac{L}{2}, \frac{L}{2}) \\
0 & \text{otherwise.}
\end{cases} \tag{2.24}
\end{equation}

This is also true for the case where the environment consists of a single point source of light such as a single star, and a good approximation for brightness profiles consisting of several widely spaced perfectly sharp edges or point light sources.

For a broader class of brightness profiles, \( g(q, n) \) may vary quite differently with \( n \). For binary images, if the topological dimension\[124\] of the set of edges in the continuous space image is equal to the fractal dimension\[125\] of the same set then,

\begin{equation}
g_1(q, n) \to g(q) \text{ as } n \to \infty, \tag{2.25}
\end{equation}

which follows directly from the definition of box counting fractal dimension \[126\]. If the required equivalence of dimensions does not hold, the brightness profile is fractal and \( g \) will increase without bound as \( n \) increases. Regardless, these are asymptotic results for very high resolution imaging, with which we are not overly concerned given the objective of operating with minimal computation.
Figure 2.5: Examples of the sum of absolute brightness differences $g_1(q, n)$ with full full factor, fixed $q$ and varying $n$, for a number of test images. The event rate is multiplied by $nL^{-1}g_1(q, n)$. Shows how event rate scales with resolution for different environments with 100% fill factor. Each plot shows the sum of absolute brightness differences between horizontally adjacent pixels, divided by vertical resolution, as a function of resolution for the image to its left.
Figure 2.6: Examples of the sum of absolute brightness differences $g_1(q, n)$ with full full factor, fixed $q$ and varying $n$, for a number of test images. Shows how event rate scales with resolution for different environments with 100% fill factor. Each plot shows the sum of absolute brightness differences between horizontally adjacent pixels, divided by vertical resolution, as a function of resolution for the image to its left.

The dependence of the event rate multiplier $g_1(q, n)$ on the resolution $n$ will become important in chapter 4 as we try to select the optimal choice of sensor resolution for a given task. Figures 2.5 and 2.6 give an idea of how this function changes with resolution for several example brightness profiles. As expected, profiles with a small number of sharp edges quickly approach a constant value as the resolution increases, while environments exhibiting structures at fine scales continue to increase even for considerable resolution.

Again considering the expression for total sensor event rate,

$$\Lambda(t) = \frac{\text{cong}(q(t), n, \rho)}{Lh} |\hat{q}(t)| + \frac{\Lambda_0}{h},$$

we see that for 100% fill factor, based on the empirical observations summarized in figures 2.5 and 2.6, that at low resolution the event rate will generally be super-linear in the resolution $n$, and for fractal environments will remain so for all $n$. 

34
For the zero fill factor case, wherein each pixel cone becomes a ray and the angle subtended by each pixel becomes zero, the dependence on \( n \) is quite different. For a piecewise constant environment with perfectly sharp edges, since the spatial brightness gradient is zero at all points except a degenerate set of edges on which it is infinite, we would expect \( g \) to be zero for all headings except a discrete set at which it is infinite, with the number of configurations yielding infinite \( g \) increasing linearly with \( n \). As a result, the information bearing event rate would at all times be zero except occasionally being infinite, which is not a particularly useful way of looking at the system.

If instead we consider a case in which the edges are not perfectly sharp but span a fixed angle, we would expect \( g_0(q, n) \) to increase approximately linearly with \( n \), as the number of pixel points inside the edge increases in proportion to resolution. So it is clear that unlike with full fill factor, it is possible for \( g_0(q, n) \) to increase linearly with \( n \) even in non-fractal environments. A diagram of this effect is shown in figure 2.7.

\[
\begin{align*}
g_0(q, n) &= 0 \text{ or } \infty \\
g_0(q, n) &\sim n
\end{align*}
\]

Figure 2.7: Depiction of why consideration of smooth edges and zero fill factor leads to an event rate multiplier \( g_0(q, n) \) that increases linearly in \( n \). For the sharp edge on the left, the multiplier is zero except when one of the pixels lands directly on the edge, while the smooth edge on the right causes \( g_0(q, n) \propto n \), so under this model event rate increases linearly in sensor resolution.

### 2.4.3 Stream Probabilities

Under the assumption that the information bearing and spurious event streams are independent, the probability that a given event carries information about the configuration is
\[ p(t) = \frac{\Lambda_{IB}(t)}{\Lambda(t)} = \frac{\frac{\partial}{\partial t} n g(q(t), n, \rho) |\dot{q}(t)|}{\frac{\partial}{\partial t} n g(q(t), n, \rho) |\dot{q}(t)| + \Lambda_0} \]  

(2.27)

Assuming that the average rate of spurious events across pixels remains constant with resolution, then \( \Lambda_0 \) can be rewritten as

\[ \Lambda_0 = n \bar{\lambda}_S, \quad \bar{\lambda}_S = \frac{1}{n} \sum_{i=1}^{n} \lambda^i_S, \]  

(2.28)

where \( \bar{\lambda}_S \) does not depend on \( n \). Using this the expressions for \( \Lambda \) and \( p \) can be rewritten as,

\[ \Lambda(t) = \frac{n}{\bar{\lambda}_S} \left( \frac{\frac{\partial}{\partial t} g(q, n, \rho) |\dot{q}|}{L} + \bar{\lambda}_S \right), \]  

(2.29)

\[ p(t) = \frac{\frac{\partial}{\partial t} g(q, n, \rho) |\dot{q}|}{\frac{\partial}{\partial t} g(q, n, \rho) |\dot{q}| + \bar{\lambda}_S}. \]  

(2.30)

In subsequent sections these quantities will form the basis for a measurement model for a dynamic vision sensor that relates event locations to system configuration (heading).

### 2.5 Events as Configuration and Rate Measurements

Previously in section 2.4 we presented a DVS sensor model in which the stream of events output consisted of the union of an information bearing and an exogenous noise stream, and derived expressions for the rates of events generated by each of the independent streams in the context of heading control. Now these ideas will be developed to show that DVS events can be thought of as measurements of both the configuration (heading) and its rate of change. In essence, event location provides information about the heading, the timing (rate) of events information about the magnitude of the angular rate, and the polarity of events information about the sign of the angular rate.

In this section our analysis will focus on the single-edge case, and in subsequent sections algorithms will be presented that extend these results to more general environments.

#### 2.5.1 Event Location and Configuration

In the single-edge case, with no persistent excitation of pixels, information bearing events should occur only in pixels subtending the edge, so that the event location gives a direct measurement of the heading. However in the presence of noise not all events contain information about the state. The probability that a given event does is denoted \( p \). Using this idea we can formulate an expression for the asynchronous discrete-time measurement \( z_0(t) \in \mathbb{R} \) which at this point is equivalent to event location:

\[ z_0(t) = \begin{cases} 
q(t) + v_0(t) & \text{with probability } p \\
v_1 & \text{with probability } 1 - p.
\end{cases} \]  

(2.31)
where $v_0$ and $v_1$ are independent random variables describing the noise:

$$v_0(t) \sim \mathcal{N}(0, \sigma_{v_0}(t)), \quad v_1 \sim \text{Unif} \left( -\frac{L}{2}, \frac{L}{2} \right). \quad (2.32)$$

This expression captures the idea that some fraction $(1 - p)$ of events do not contain any information regarding the configuration of the system, while the remaining events can be thought of as additively corrupted measurements of the configuration. While this is obvious only for the single-edge case, it will later be shown to be the same for more complex brightness profiles. The variance $\sigma_{v_0}$ represents the error associated with uncertainty about the sub-pixel location of the edge, and should therefore be proportional to the pixel spacing and inversely proportional to the sensor resolution $n$. This idea will be explored further in later sections as a tradeoff between resolution and event threshold is formulated.

The expectation of the measurement is,

$$E[z_0(t)] = p(t)(E[q(t)] + E[v_0(t)]) + (1 - p(t))E[v_1] = p(t)q(t),$$

and we see that the raw measurement is biased when $p < 1$, which is always the case as long as spurious events are present. The variance of the measurement is

$$\text{Var}[z_0] = \sigma_{v_0}^2 p + \frac{L^2}{12} (1 - p) + q^2 p(1 - p). \quad (2.33)$$

The variance of $z_0$ approaches the constant value $\frac{L^2}{12}$ as $p$ goes to 0, and the variance of $v_0$ as $p$ goes to 1. Thinking about the variance alone however is somewhat deceptive, as it would suggest that the measurement can only become so inaccurate. The measurement is also biased, with the expectation approaching 0 as $p \to 0$ and $q$ as $p \to 1$. Assuming we know $p$ exactly, we can transform the measurement into an unbiased form by dividing the raw measurement by $p$,

$$z_1(t) = \frac{z_0(t)}{p(t)}, \quad E[z_1(t)] = q(t), \quad \text{Var}[z_1(t)] = \frac{\sigma_{v_0}^2(t)}{p(t)} + \frac{L^2(1 - p(t))}{12 p(t)^2} + \frac{q(t)^2(1 - p(t))}{p(t)}. \quad (2.34)$$

This transformed measurement is unbiased, but its variance now grows without bound as $p$ goes to 0. Neglecting higher moments of the distribution, $z_1$ can now be thought of as a measurement of $q$ corrupted by additive noise with the given variance,

$$z_1(t) = q(t) + v(t), \quad E[v(t)] = 0$$

$$\text{Var}[v(t)] = \sigma_v^2 = \frac{\sigma_{v_0}^2}{p} + \frac{L^2(1 - p)}{12 p^2} + \frac{L^2(1 - p)}{p}. \quad (2.35)$$

Performing this measurement de-biasing requires us to know $p$, however since $p$ depends on the system state it is not known exactly and must be estimated. The assumption that $p$ is known precisely is justified in situations where we have access to an accurate rate measurement, but $p$ can also be estimated using the DVS itself. While this will introduce uncertainty
in p and render the moments in equation 2.35 inaccurate, for analytical tractability we will neglect this uncertainty. In order to estimate p from the DVS event stream, recall that p is a function of \( \dot{q} \) and q, as well as the resolution, event threshold, and fill factor,

\[
p(t) = \frac{\varphi \cdot g(q(t), n, \rho) |\dot{q}(t)|}{\varphi \cdot g(q(t), n, \rho) |\dot{q}(t)| + \lambda_S},
\]

where \( g \) is a multiplier on the rate of information bearing events that depends on the structure of the environment and is empirically observed to be increasing in n but possibly asymptotic. If \( g \) does not vary appreciably with q, as in the single-edge case, we see that knowing \( p \) exactly would allow us to compute \( |\dot{q}| \) exactly. For now assume that we have an accurate estimator of the overall event rate \( \Lambda \) that will allow us to compute \( p \) accurately using the relationship

\[
p(t) = 1 - \frac{\Lambda_0}{\Lambda(t)},
\]

since it is assumed that the background rate of spurious events \( \Lambda_S \) is known. From this we see that \( p \) can be estimated using only the total event rate \( \Lambda \), without consideration for the structure of the multiplier \( g \), so is in a sense easier to estimate the \( |\dot{q}| \).

### 2.5.2 Event Rate and Heading Rate

Now we will consider how the event rate can be used to produce a measurement of the magnitude of the sensor angular rate. Recall from equation 2.29 that the observed event rate is related to the configuration and configuration rate through,

\[
\Lambda(t) = \frac{n}{h} \left( \frac{c_0}{L} g(q(t), n, \rho) |\dot{q}(t)| + \dot{\lambda}_S \right),
\]

The total event rate \( \Lambda(t) \) is estimated using the minimum variance unbiased estimator [127] based on the timing of the past m events, i.e.

\[
\hat{\Lambda} = \frac{m - 1}{\sum_{i=0}^{m} \Delta_i}, \quad \Delta_i = t_{i+1} - t_i
\]

with \( t_i \) being the time of the \( i \)th most recent event. This event rate estimator, \( \hat{\Lambda} \), is unbiased and has asymptotic variance \( \Lambda^2 \),

\[
\mathbb{E}[\hat{\Lambda}] = \Lambda, \quad \lim_{m \to \infty} \text{Var}[\hat{\Lambda}] = \Lambda^2,
\]

if the event rate is constant. Given the very high (\( \approx 1 \text{Mpersecond}) \) event rates seen in practice, we will use the asympt.

and \( |\dot{q}| \) (equation 2.29) gives an expression for an additional component of the measurement, \( z_2 \),

\[
z_2(t) = \left( \frac{\dot{\Lambda}(t)}{n} - \dot{\lambda}_S \right) \frac{L}{c_0 g(q(t), n, \rho)},
\]

which has expectation,
\[ \mathbb{E}[z_2(t)] = |q(t)|, \]

and variance,

\[
\text{Var}[z_2(t)] = \left( \frac{Lh}{c_0g(q,n,\rho)\lambda_n} \right)^2 \text{Var}[\lambda] = \left( \frac{Lh}{c_0g(q,n,\rho)\lambda_n} \right)^2 \lambda^2 \] (2.44)

\[
= |q(t)|^2 + \frac{2Lh\lambda_s}{c_0g(q,n,\rho)\lambda_n} + \left( \frac{L\lambda_s}{c_0g(q,n,\rho)} \right)^2.
\]

For conciseness we will define,

\[
\Gamma(t) = \frac{2Lh\lambda_s}{c_0g(q,n,\rho)\lambda_n} + \left( \frac{L\lambda_s}{c_0g(q,n,\rho)} \right)^2,
\]

so that,

\[
\text{Var}[z_2(t)] = |q(t)|^2 + \Gamma(t). \] (2.46)

While the type of estimation described would give rise to measurements with errors correlated in time, we will neglect this for analytic simplicity and model the observation of time between events as a measurement of the absolute heading rate \( |q| \) corrupted by additive noise with variance as above.

This value can serve as a second component to the overall measurement, allowing each event to be interpreted as a measurement of both the heading and heading rate, \( q \) and \( |q| \). The standard deviation in the measurement \( z_2 \) increases linearly in \( |q| \), approaching a constant value as \( q \) approaches zero. This is as expected, since we have assumed that we have accumulated sufficient events to use the asymptotic result in equation 2.41. In reality, as the event rate decreases the time scale over which this assumption holds increases, eventually becoming infinite as the event rate reaches zero. Nonetheless the linear increase in standard deviation is predicted by an analytically tractable model and is well matched by empirical observations (see figure 3.28).

2.5.3 Event Polarity and Motion Direction

While the event timing alone can be used to produce event-by-event measurements of the absolute heading rate \( |q| \) through estimation of the total event rate, it carries no information about the direction of motion according to our model. However event polarity is highly correlated to the sign of the heading rate, and can be easily incorporated to produce measurements of the signed angular rate.

To understand how polarity is dictated by the direction of motion we again consider the single-edge case, and in later sections will extend the results to more general environments. In this simple case, the edge causes increases in brightness when it moves one way and decreases when it moves the other, and as a result the polarity of all information bearing event matches
exactly the direction of motion. For more complex brightness profiles, the polarity can be used with methods analogous to those described in sections 3.3 and 3.4 to produce measurements of the heading rate direction. In practice event polarity is a very good indicator of direction of motion even for relatively low heading rates (see figure 3.27), and we will therefore in our models neglect any uncertainty introduced by the estimation of the sign of heading rate.

Combining this with the estimate of the magnitude of heading rate we obtain an expression for the heading rate measurement,

$$z_2(t) = \dot{q}(t) + w(t) \quad \mathbb{E}[w(t)] = 0 \quad \text{Var}[w(t)] = |\dot{q}(t)|^2 + \Gamma(t). \quad (2.47)$$

Concatenating the heading and heading rate measurements gives,

$$z(t) = \left( \begin{array}{c} q(t) + v(t) \\ \dot{q}(t) + w(t) \end{array} \right), \quad (2.48)$$

where $v(t)$ and $w(t)$ are the additive, zero mean, noise terms applied to the heading and heading rate measurements respectively.

### 2.5.4 Delay

Observations of data from the sensor suggest that the model presented to this point fails to capture an essential aspect its operation: delay. While the ideal sensor may have zero measurement lag, with the pixels operating exactly according to the event generation model described, in reality pixels exhibit a low-pass behaviour in which they remain excited after experiencing a change in brightness. In other words the event rate depends not only on the instantaneous rate of change of brightness but also on past changes.

As we have shown, the data contained in the event can be used to construct an unbiased measurement of the configuration and configuration rate in the case with no persistent pixel excitation or delay. Based on this, an event produced by a pixel some time delayed from a brightness change but triggered by that change can be thought of as a delayed measurement of the same quantities. To formalize this we must specify exactly how the pixel excitation model is modified. With no delay we gave this expression (equation 2.14) for the rate of information bearing events produced by a given pixel $i$:

$$\lambda_{IB}^i(t) = \frac{c_0 |\dot{y}_i(t)|}{h}. \quad (2.49)$$

To account for persistent excitation, this is generalized to

$$\lambda_{IB}^i = \frac{c_0}{h} \int_0^\infty \phi(\tau)|\dot{y}_i(t-\tau)|d\tau, \quad (2.50)$$

where $\phi(\tau)$ is a weighting function defining the time scale of the persistent excitation and more specifically exactly how the event rate depends on past excitations. A lag-free model is recovered if $\phi(\tau)$ is a delta-function at the origin.

The output from this single pixel can now be thought of as a superposition of an infinite number of Poisson processes - one for each value of the delay $\tau$, and this reasoning extends
immediately to the sensor level aggregation of events. The probability of an event having come from the process with delay $\tau$ is proportional to the weighting coefficient $\phi(\tau)$. By this reasoning the measurements derived from event data are subject to a stochastic delay with continuous distribution function $\phi(\tau)$. So, modifying our previous measurement function 2.48 we get,

$$z(t) = \begin{pmatrix} q(t - \tau) + v(t) \\ \dot{q}(t - \tau) + w(t) \end{pmatrix}, \quad \tau \sim \phi(\tau). \quad (2.51)$$

Because stochastic delay is more difficult to deal with analytically than deterministic delay, in section 3.5 we show how filtering methods that allow us to extend these models to more general environments also drastically reduce the variance in delay, allowing us to more reasonably approximate the delay as fixed.

The function $\phi$ can be computed in practice using observations of the “wake” of events following the transit of an edge across the field of view. Such a wake is shown in figure 3.18.

### 2.5.5 Asynchronous Measurement Model

To summarize the discussion of the sensor model up to this point, we are able to make additively corrupted measurements of both the heading (using event location) and the heading rate (using event rate and polarity), but our measurements are subject to a stochastic delay and still arrive asynchronously according to a Poisson process.

<table>
<thead>
<tr>
<th>Model 2.5.1 (Asynchronous DVS Measurement Model).</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z(t_k) = \begin{pmatrix} q(t_k - \tau) + v_k \ \dot{q}(t_k - \tau) + w_k \end{pmatrix}, \quad \tau \sim \phi(\tau)$</td>
</tr>
<tr>
<td>$E[v_k] = 0, \quad Var[v_k] = \frac{\sigma_v^2}{p} + \frac{L^2(1-p)}{12p^2} + \frac{q^2(1-p)}{p}$</td>
</tr>
<tr>
<td>$E[w_k] = 0, \quad Var[w_k] =</td>
</tr>
<tr>
<td>$p(t_k) = \frac{\alpha_p}{h} g(q, n, \rho)</td>
</tr>
<tr>
<td>$\Lambda(t_k) = \frac{n}{h} \left( \frac{\alpha_0}{L} g(q, n, \rho)</td>
</tr>
<tr>
<td>$\Gamma(t_k) = \frac{2Lh\lambda_s}{c_0g(q, n, \rho)n} + \left( \frac{L\lambda_s}{c_0g(q, n, \rho)} \right)^2$</td>
</tr>
<tr>
<td>$(t_{k+1} - t_k) \sim \text{Exp}(\Lambda(t_k))$</td>
</tr>
</tbody>
</table>

### 2.5.6 Continuous-Time Sensing

Until now the models presented have retained the property that measurements occur asynchronously whenever an event occurs. Now we would like to approximate the sensor with a continuous-time model subject to additive noise that takes into account variation in event
rate. This continuous-time sensor model will be useful for feedback control design and analysis compared to an asynchronous discrete-time model. This is discussed in section 3.7.

We transform the asynchronous discrete time sensor with zero mean additive noise into a continuous-time sensor with continuous zero mean additive noise of the appropriate variance. Given the instantaneous total event rate \( \Lambda(t) \), the expected time between events is \( \Lambda^{-1}(t) \), and we would like to determine the power spectral density of the continuous-time noise signal corresponding to discrete noise at the associated rate. For discrete-time noise signal \( v(t) \) and continuous noise signal \( v_c(t) \) this means,

\[
\Lambda \int_0^{\Lambda^{-1}} v_c(t) dt = v(t), \quad \mathbb{E}[v_c(t)v_c(t-\tau)] = \delta(t-\tau)\sigma^2,
\]

i.e. the discrete-time noise is the average of the continuous-time noise over the expected time between measurements. The value \( \sigma \) is the continuous time “variance” of the noise signal. This corresponds to an estimation model in which discrete measurements are produced by using a simple moving average filter driven by a continuous-time measurement process. Since we know the variance of the discrete measurements \( v(t) \), we can solve for the zero-lag autocorrelation, \( \sigma^2 \) of the signal \( v_c(t) \) under the assumption that it has locally zero autocorrelation,

\[
\text{Var}[v(t)] = \Lambda^2\Lambda^{-1}\sigma^2. \tag{2.53}
\]

For the configuration measurement,

\[
\text{Var}[v(t)] = \Lambda^2\Lambda^{-1}\sigma^2 = \sigma_c^2p + \frac{L^2}{12}(1-p) + q^2p(1-p)
\]

\[
\Rightarrow \mathbb{E}[v_c(t)v_c(t-\tau)] = \delta(t-\tau)\Lambda^{-1}\left(\frac{\sigma_c^2}{p} + \frac{L^2(1-p)}{12p^2} + \frac{q^2(1-p)}{p}\right). \tag{2.55}
\]

Similarly for the continuous-time rate measurement,

\[
\mathbb{E}[w(t)w(t-\tau)] = \delta(t-\tau)\Lambda^{-1}(\lvert q(t) \rvert^2 + \Gamma(t)). \tag{2.56}
\]

This conversion to continuous-time changes slightly the behaviour of the noise variance and its dependence on heading rate, while retaining most of the interesting asymptotic behaviour. The heading measurement signal becomes arbitrarily noisy as the heading rate approaches zero, while heading rate can be measured with finite accuracy even when it is zero. One difference is in the asymptotic accuracy of heading measurements as rate becomes infinite. Previously heading measurements could only ever reach accuracy limited by pixel width, but under this model the averaging of arbitrarily many measurements at high rate means that the continuous time noise on heading measurements decreases to zero as rate becomes infinite.

### 2.5.7 Model Summary and Discussion

In this section and the previous chapter we have developed a model for the dynamic vision sensor in the context of heading regulation that incorporates some of the key features of its behaviour while being more amenable to analysis and more useful for control system design than
the raw pixel-level model. The key interesting feature of this model and defining characteristic of the DVS is that the signal-to-noise ratio approaches zero for heading measurements as the heading rate approaches zero. In other words the sensor cannot make meaningful observations of heading when it is stationary. Measurement lag caused by persistent pixel excitation is also part of the model and will be discussed in greater detail in subsequent sections. To summarize the results so far, we have a continuous-time sensor with 2 dimensional output \( z(t) \),

<table>
<thead>
<tr>
<th>Model 2.5.2 (Continuous-Time DVS Measurement Model).</th>
</tr>
</thead>
</table>
| \[
\begin{align*}
    z(t) &= \left( \frac{q(t - \tau) + v(t)}{\hat{q}(t - \tau) + w(t)} \right), \quad \tau \sim \phi(\tau) \\
    \mathbb{E}[v(t)] &= 0, \quad \mathbb{E}[v(t)v(t - \tau)] = \delta(t - \tau)\text{Var}[v(t)] \\
    \mathbb{E}[w(t)] &= 0, \quad \mathbb{E}[w(t)w(t - \tau)] = \delta(t - \tau)\text{Var}[w(t)] \\
    \text{Var}[v(t)] &= \Lambda^{-1} \left( \frac{\sigma_\theta^2}{p} + \frac{L^2(1 - p)}{12p^2} + \frac{q^2(1 - p)}{p} \right) \\
    \text{Var}[w(t)] &= \Lambda^{-1} \left( |\dot{q}(t)|^2 + \Gamma(t) \right) \\
    p(t) &= \frac{\alpha g(q, n, \rho)|\dot{q}|}{\frac{\partial g(q, n, \rho)}{\partial \dot{q}}} + \lambda_s \\
    \Lambda(t) &= \frac{n}{\hbar} \left( \frac{\alpha_0 g(q, n, \rho)[\dot{q} + \lambda_s]}{\frac{\partial g(q, n, \rho)}{\partial \dot{q}}} + \lambda_s \right), \quad \Lambda_0 = n\lambda_s \\
    \Gamma(t) &= \frac{2Lh\lambda_s}{\alpha_0 g(q, n, \rho)n} + \left( \frac{L\lambda_s}{\alpha_0 g(q, n, \rho)} \right)^2 
\end{align*}
| |

The measurements of configuration \( q \) and speed \( \dot{q} \) are corrupted by zero-mean additive Gaussian noise whose variance depends on the state. For simplicity we will refer to terms multiplying the delta functions in the autocorrelation functions for these quantities as variances, so that we can state the asymptotic measurement variances as,

\[
\begin{align*}
    |\dot{q}| \to 0 \\
    &\Rightarrow \Lambda \to \Lambda_0 \\
    &p \to 0 \\
    \text{Var}(v) \to \infty \\
    \text{Var}(w) \to \Gamma, \\
    |\dot{q}| \to \infty \\
    &\Rightarrow \Lambda \to \infty \\
    &p \to 1 \\
    \text{Var}(v) \to 0 \\
    \text{Var}(w) \to \infty.
\end{align*}
\]

(2.57)

(2.58)

The heading measurement error variance increases to infinity as the heading rate approaches zero, while the heading rate measurement remains bounded. However, as heading
rate approaches infinity, the heading measurement becomes arbitrarily accurate while the heading rate measurement grows without bound. This is a result of the fact that the heading rate measurement is produced using an estimate of the total event rate $\hat{A}$, which has standard deviation proportional to the event rate itself.

This model allows continuous-time control design and analysis to be applied to the problem, and provides an additive form for the noise that is also easier to deal with analytically. In subsequent sections the control design problem will be addressed and it will be seen that the explosion of the heading measurement variance as heading rate approaches zero precludes the direct application of a linear control methodology. Until now we have considered only a simple single-edge case in which it is obvious that the event location is a measurement of heading, and in the following sections we will show that even for more complex environments, with the use of efficient algorithms for signal estimation, this model still holds.

2.6 Discussion

In this chapter we have provided a model for the pixel-level operation of the DVS on its own (section 2.1), described the heading regulation problem in which we are interested (section 2.3), and presented derived models for the DVS in the context of one-dimensional heading regulation.

Our closed-loop model for operation of the DVS in a heading regulation setup describes the stream of events coming from the sensor as the union of information-bearing and spurious streams, and assumes that we cannot determine which stream each event is derived from. As a result, we are left only with a probability, $p$, giving the likelihood that some event was part of the information-bearing stream. This value is given in equation 2.30, and has several physically intuitive properties. Most importantly, as the angular rate of the sensor approaches zero, the probability becomes zero, and the signal-to-noise-ratio (SNR) approaches zero as well. Secondly, the factor weighting the angular rate is a function representing the total amount of brightness variation in the image, indicating that for a given heading rate the SNR is higher for images with greater total variation. This total variation in brightness may of course vary with heading, and also varies with sensor resolution depending on the "fractalness" of the scene. In fact this quantity could most likely be formally related to fractal dimension, but that is beyond the scope of this thesis. For simple scenes such as the cartoon forest backdrop used in experimental testing, it becomes roughly constant above about 50-pixel resolution, while for more complex scenes such as a cityscape from above it continues to increase even for high resolution. (Section 2.4)

This model has been derived under the assumption that noise is present in the form of an exogenous Poisson process producing events, and that events produced as a result of brightness changes are also the result of a Poisson process with time-varying intensity. Neither of these assumptions is exactly accurate, as even simple analysis of the event stream from a stationary sensor reveals that the noise events are not exactly Poisson, and certainly the determinism in the information-bearing event process means that they are not Poisson. For instance the distribution of event interarrival times when the sensor is held stationary facing a white wall does not match the exponential distribution of a Poisson process (Figure 2.10), and some
autocorrelation in spurious event interarrival times at a single pixel is also observed.

Figure 2.8: Spatial distribution of noise events across pixels, comparing experimental observations gathered from a stationary DVS pointed at a white wall to an example realization of a uniform distribution in which all pixels are equally likely to see noise events. Shows that some pixels consistently produce higher rates of noise events than others, but with no clear spatial pattern to these rates.

Figure 2.9: An alternative visualization of the non-uniformity in pixel noise event rates (similar to figure 2.8). The number of pixels producing each of a given number of noise events over the test period when a stationary DVS looks at a white wall is compared to the theoretical distribution for identical pixels.
Figure 2.10: Distribution of log interarrival times for events generated by a stationary DVS looking at a white wall (blue), compared to the predicted distribution resulting from a Poisson process (red). While the observed distribution shows relative conformity to the Poisson model, it does deviate slightly, and has the interesting property of being bimodal. This bimodality may be indicative of two different processes responsible for noise at the pixel level.

Figures 2.8, 2.9 and 2.10 illustrate some of the shortcomings of this model with respect to the noise generation process. Firstly, not all pixels produce noise at equal rates, and while this fact does not influence the model as presented, it does suggest that the SNR will differ across pixels, and therefore algorithms based on aggregate mean behaviour (using a single value for $p$) will fail to take advantage of this non-uniformity. Secondly, the distribution of noise event interarrival times does not exactly match that resulting from a Poisson process (2.10), and in fact is bimodal. However these inaccuracies are both believed to be negligible for our purposes, and the model still captures the key unique behavioural properties of the dynamic vision sensor in the heading regulation problem.

We have also presented a model for a dynamic vision sensor in the context of a robotic control problem as a configuration and configuration rate sensor with measurements corrupted by additive zero-mean noise. The measurement error variance depends on the configuration
and configuration rate and is singular where $\dot{q} = 0$. (Section 2.5)

One of the primary objectives of this analysis is to develop a model for the sensor that can be used for closed-loop control design. The greatest difficulty in applying existing standard techniques to control design given the sensor model is the fact that the noise variance in the model is dependent upon the configuration and configuration rate and has a singularity where configuration rate is zero. Because of the endogenous nature of the noise, we cannot apply tried and true optimal control and estimation techniques. With exogenous noise, the trajectory taken by the system does not impact the accuracy of the measurements, allowing the control task to be separated from the estimation task. In our case, it may be desirable to manipulate the trajectory in order to enter regions of the configuration or state space that allow more accurate measurements to be made. For example we may want to occasionally drive the system away from our target to decrease our estimate covariance, or even introduce artificial process noise into the system to ensure that we do not remain too long with the heading rate near zero. Computing optimal solutions given this coupling can be extremely expensive, so we have focused on simple low-computation methods for dealing with the particular behaviour of the DVS sensor model.
Chapter 3

Algorithms and Analysis

This chapter describes experiments conducted and the algorithms developed for heading control with DVS.

3.1 Experimental Setup

In order to make a convincing argument for the superiority of dynamic vision sensors over conventional cameras for heading regulation, and to demonstrate the efficacy of the novel algorithms presented, it is necessary to conduct controlled experimental tests comparing the performance of the different sensors in various respects. To this end, an apparatus wherein the sensor (either a camera or DVS) was actuated rotationally and contained within an automatically actuated drum rotating about the same axis was constructed. Figure 3.1 shows the concept of the rotating drum apparatus, while the fabricated apparatus is shown in figure 3.2.
Figure 3.1: Conceptual representation of the rotating drum apparatus used for testing and comparing heading regulation algorithms.
The apparatus constructed consists of two concentric brushed DC motors and high-speed motor controllers, one responsible for actuating the sensor, and another for actuating the backdrop. The angles of the motor shafts are measured by mechanical connection to two continuous rotation potentiometers connected to the analog-digital-converters of an ATMEGA328 microcontroller (on an Arduino board), which also generates the pulse width modulated (PWM) voltage signals sent to the motor controllers that dictate the duty cycles of the high voltage motor drive signals. The microcontroller is connected via USB to a personal computer (laptop), which is also connected by USB to the dynamic vision sensor. Each angle measurement, taken as the resistance across a potentiometer, is digitized into a 10-bit value, and therefore has a precision of approximately 1/3 degree. The ADC process requires approximately 0.1ms and the maximum sensing rate is therefore about 5kHz for both potentiometers. PWM signals generated by the microcontroller use a frequency of approximately 8kHz, which is passed through the motor controller, and communications between the microcontroller and laptop are through a virtual serial port operating at 115.2kbaud, or approximately 14kB/s. The key signal and electronic elements of the setup are shown in figure 3.3.
Figure 3.3: Electronic signal connections and key blocks in rotating drum setup. Important communication delays $\Delta t_1, \Delta t_2$ and $\Delta t_3$ are also shown.

While one of the objectives of this work is to analyze the behaviour of systems implementing event-by-event control signal adjustments, in practice the USB connection between the DVS and the laptop requires events to be buffered into packets before being sent to the laptop, where they are then received effectively simultaneously. Thus, results will be presented variably for buffer-wise data and event-wise data, and the significance of buffering will be later discussed in detail.

3.2 System Identification

This section describes procedures used to ascertain the dynamical nature of the experimental apparatus (figure 3.2), and gives the results of the system identification. Specifically we consider how to model the rotational dynamics of the plant and the lag associated with taking measurements from the DVS.

3.2.1 Plant Dynamics

DVS based feedback control must be compared to feedback control using a camera and potentiometers. To do this, a reasonable model for the dynamics of the plant should be found. Observed data suggests that the dynamics are well modelled by a second order system, and since we have a rigid apparatus and no reason to believe the torque exerted on the platform depends on heading, the plant alone was initially fit to a model of the form,

$$\ddot{q} = a\dot{q} + bu,$$  \hspace{1cm} (3.1)

where $u$ is the voltage applied to the motor actuating the sensor. This linear dynamical
system has the unknown parameters $a$ and $b$ to be determined by system identification, and has transfer function,

$$P(s) = \frac{b}{s(s - a)}. \quad (3.2)$$

While the linear model is useful for control design, the actual system was subject to static friction, so a model including static friction was also fit. Under this model the dynamics of the system become,

$$\ddot{q} = a\dot{q} + a_1 e^{-a_2|\dot{q}|}\text{sign}(\dot{q}) + bu. \quad (3.3)$$

This model contains two additional parameters $a_1$ and $a_2$ defining the static friction acting on the system, but is otherwise the same as the linear model.

### 3.2.2 Identification

For the linear dynamical model, the unknown parameters $a$ and $b$ were selected to minimize the mean squared error between the predicted and observed gain and phase lag as shown in figure 3.4. The resulting plant dynamics are:

$$\ddot{q} = -28\dot{q} + 3935u, \quad (3.4)$$

where $u$ is measured in volts and $\dot{q}$ is measured in degrees per second. Specifically,

$$b = 3935 \frac{\text{deg}}{\text{Vs}^2}, \quad a = -28 \text{ s}^{-1}. \quad (3.5)$$

The gain value of 3935 deg/Vs$^2$ is equivalent to 527 potentiometer least significant bits (pLSB) per second squared per PWM duty cycle LSBs (oLSB).
Figure 3.4: Frequency response of the plant alone, excited at an amplitude of 3.75V (80/255 PWM duty cycle), showing collected data and data from the fit of a 2nd order model.

The results shown in figure 3.4 are from a series of 20 trials, of 30 seconds each in duration, during which time an approximately sinusoidal voltage (PWM duty cycle and direction) signal was supplied to the actuator. Across trials, the frequency of the signal was varied logarithmically between 1 and 12Hz. The supplied signals could not be exactly sinusoidal, as this would lead to a drift in heading angle, so a small proportional feedback term was added to the signal to stabilize the heading during these tests. This contribution to the signal is considered to be negligible for the purposes described here.

For the non-linear model including the static friction term, fitting of test trajectories (sine
and step) resulted in a low discrepancy between simulated and actual data. Averaging the resulting parameter estimates across 15 closed-loop sinusoid tracking tests and 15 closed-loop step response tests, the fit parameters were,

\[ a = 45, \quad a_1 = 5500, \quad a_2 = 0.0015, \]

The parameter estimates for each trial were determined as those minimizing the mean squared error (MSE) between simulated and observed trajectories. The very low value for \( a_2 \) indicates that the exponential structure of the friction function is not significantly different than a simple signed constant term, where \( a_2 = 0 \). Figure 3.5 depicts the static friction nonlinearity along with the result of the parameter fit.

![Friction vs Rate](image)

Figure 3.5: Observed and predicted friction components of angular acceleration in the system, showing a strong non-linear friction curve with a slight directional asymmetry that is not captured by the symmetric model.

### 3.2.3 Control Gains

For the performance comparisons between DVS-based systems and other systems in subsequent sections, the same control gains must be used with the various sensing methods, and
will be compared on the basis of mean squared tracking error. Thus we would like to find a control law minimizing,

\[ J_0 = \lim_{T \to \infty} \frac{1}{T} \int_0^T q^2 dt. \]  (3.6)

Given this control objective \( J_0 \), the linear model for our testbed dynamics can be used to determine the optimal control law. However simply using \( J_0 \) would lead to an infinite gain solution, so the LQR problem must be modified to include a weight on the control input. This leads to the alternative cost \( J_1 \),

\[ J_1 = \lim_{T \to \infty} \frac{1}{T} \int_0^T q^2 + ru^2 dt. \]  (3.7)

The \( J_1 \) optimal control policy is a linear feedback law \( u = -k_1 q - k_2 \dot{q} \) with,

\[ k_1 = \sqrt{\frac{1}{r}}, \quad k_2 = \frac{a}{b} + \sqrt{\frac{a^2}{b^2} + 2k_1}. \]  (3.8)

The choice of \( r \) remains, and is not based on any concern for the inherent cost of control input, but on the fact that control saturation is possible. For a given choice of \( r \) and \( \dot{q} = 0 \), there is a tracking error \( q \) above which the control input is saturated. Therefore we can choose \( r \) by specifying the maximum tracking error that we expect to see in practice. Based on closed-loop sinusoid tracking tests in which the error remained below 5 degrees, we chose a tracking error of 10 degrees to correspond to input saturation. With the corresponding value of \( r \) the gain values in usable units are,

\[ k_1 = 10^\frac{\text{oLSB}}{\text{pLSB}}, \quad k_2 = 4.42 \frac{\text{oLSB}}{\text{pLSB/s}}. \]

This control law theoretically minimizes \( J_1 \) for the identified linear model of the system, and is effective qualitatively in practice, so will be used (with appropriate unit conversions) to compare the different sensors. A comparison between simulated and experimental closed-loop trajectories is shown in figure 3.6.
These results show that the plant can be reasonably well modelled in this way, to the extent that simulation results match experimental results to a significant degree when potentiometer based feedback is used. Nonetheless, when reference tracking problems are considered, the arbitrary motion of the reference will render knowledge of the dynamics much less useful and will serve as the motivation for an output feedback approach.

### 3.2.4 Sensor Lag

While the low level latency of the DVS is claimed to be 3 microseconds [46], [48], in practice, particularly due to its communication via USB, the latency between when a change in brightness occurs and when the computer becomes aware of the change ($\Delta t_3$ in figure 3.3), is much larger. This section describes the procedure used for determination of this lag.

First, the microcontroller of the rotating drum apparatus was programmed to execute a sinusoidal trajectory with the single-edge backdrop fixed, while the laptop queried DVS event data and potentiometer readings in parallel using a multi-threaded approach. The result was a sequence of DVS event buffers accompanied by laptop timestamps indicating the time at which each buffer was received, and a sequence of potentiometer readings indicating heading angles with laptop timestamps. This data can then be used to determine the relative delay between the time the potentiometer data was received and the time the equivalent DVS data was received, $\Delta t_3 - \Delta t_2$. In order to do this, each event was associated with a heading angle—the angle measured by the potentiometer at the time that event was received. If $\Delta t_2$ were 0, this would simply be the true angle of the potentiometer at the time the event was received by the computer.

To account for buffering, the event timestamps were used to associate each event with a laptop clock time by matching the highest timestamp in a buffer to the time the buffer
was received. Linear interpolation between potentiometer readings was used to compute the
heading at the non-synchronous event times.

An artificial lag can be inserted into these calculations by simply adding the lag to the
laptop clock times for every event, and this allows the lag identification to be formulated as an
optimization over the choice of this artificial lag parameter. The lag parameter was chosen so
that the direction of motion of the edge does not influence where the edge is observed. Figure
3.7 shows the effect of lag on the observations, and illustrates the optimization problem to
be solved. Specifically, the lag was chosen to minimize the mean squared error from linear
regression of the event location versus heading angle.
Figure 3.7: The effect of introducing artificial lag into the event measurements: the edge is observed at different locations depending on its direction of motion. The color of the points indicates the direction of motion of the sensor at the time each event buffer was received. With no lag the position at which the edge is measured should not depend on the motion.

This procedure can be conducted on an event-by-event basis or a buffer-wise basis, and because the buffers indicate the true time the events were received by the computer, rather than the theoretical time assuming fully asynchronous communication was possible, the buffers were used. An interesting consequence of this is the fact that different statistics of event locations within each buffer have an effectively different lag, as shown in figure 3.8.
Figure 3.8: Mean squared regression error in event location as a function of relative delay \( \Delta t_3 - \Delta t_2 \), for the mean, median and mode event location in each buffer. The minimum for each plot indicates the associated lag: -0.3ms for the mode, 0.0ms for the median and 2.0ms for the mean.

Based on the known behaviour of the microcontroller-computer interface, it is possible to estimate \( \Delta t_2 \) as 1.0ms, which would indicate that the buffer-wise mean event location serves as a measurement of heading angle with a lag of approximately 3ms, while the buffer-wise mode has a lag of only 0.7ms. The difference in these two quantities will be discussed at length in the following section.
Figure 3.9: The distribution of raw event locations and the distribution of buffer-wise median event locations for the same data. Color indicates the direction of motion at the time each event was produced (a) or buffer of events was received (b).

3.3 Local Estimation Algorithms for Complex Environments

So far we have developed a model of the DVS in the heading regulation and tracking problem as a sensor providing continuous-time measurement of the heading and heading rate corrupted by additive zero mean noise with variance dependent upon the state. The development of this model was based on the idea that each event effectively represents a heading measurement, which is obvious only in the case where the brightness profile is a single perfectly sharp edge. In this section we present efficient methods that allow the model to be used for more general classes of known brightness profiles, for small deviations from some heading (i.e. locally). In the next section more general methods are presented that under certain assumptions allow the model to apply globally. The DVS model also provides a heading rate measurement signal, the derivation of which did not depend on the specific brightness profile. Therefore, these sections will focus only on methods by which event location is translated into heading. Because we are interested in leveraging the unique properties of DVS, the objective is to provide algorithms efficient enough to be applied whenever a new event is received, and that also do not introduce significant sensor lag.

Experimental results from this and subsequent sections show heading tracking performance with the more complex “cartoon forest” backdrop shown in figure 3.10. This backdrop represents the kind of piecewise constant environment for which the efficient local and global methods have been designed, and will serve as the primary environment for testing of these algorithms.

Because we consider the trade-off between power and performance, in this section we describe low-computation, memoryless methods, that is those for which the heading signal produced depends only on the most recent event output by the DVS and not on any events...
before that. In the following section (3.4), we will see that using a short rolling history of events can enable global sensing, while section 3.5 will provide a theoretical basis for the concept that this technique can actually reduce sensor lag. As the methods of this section are local and require no memory or complex filtering, they embody the goal of developing control systems for low power, high performance systems. Specifically, the goal of the methods described in this section is to find a function $h: \mathbb{R} \rightarrow \mathbb{R}$ that maps event locations to local heading measurements.

The results of this section are also concisely outlined and presented in [128].

### 3.3.1 Single-Edge Brightness Profile

The simplest case we consider is that in which the brightness profile (backdrop or environment) consists of a single sharp edge as shown in figure 3.1. In this case,

$$m(s) = \begin{cases} 
0 & s < 0 \\
1 & s \geq 0
\end{cases} \quad (3.9)$$

and the event map $h(s) = s$ can be used to produce a signal effectively equivalent to a heading measurement that can be applied directly in a proportional feedback controller. Considering the results shown in figure 3.9, the relationship between heading angle and event location

Figure 3.10: “Cartoon forest” backdrop used for testing algorithms in a more complex environment, shown attached to the heading control and tracking test apparatus with actuated drum.
is linear as long as the velocity of the sensor is sufficient to produce enough events. The problem that the sensor produces no meaningful events when there is no motion is inherent to its operation and not a result of the presence of spurious events, but spurious events can also disguise a low rate of legitimate events, increasing the velocity required to achieve reliable sensing. For most of the discussion in this section it will be assumed that velocity is sufficient at all times so that our measurements of heading are not subject to noise (i.e. $z(t) = q(t)$), as our focus for this section is to extend previous analysis to a broader class of brightness profiles. However, according to the sensor model presented in section 2.5, the accuracy of the resulting measurements depends on the heading rate, and the methods for constructing a heading measurement signal will not change that.

Because of the linear relationship between event location and heading angle, it is possible to produce a feedback control signal proportional to the heading error in a memoryless way. However for reasons discussed in section 3.5 it may still be beneficial to maintain a short history of events in order to mitigate the effect of persistent excitation of pixels and reduce lag.

### 3.3.2 Piecewise Constant Brightness Profiles

The linear relationship between event location and heading seen in the single-edge case can be locally reproduced for backdrops with piecewise constant brightness using a computationally efficient map $h(s)$ that is piecewise linear. In this case there will be a region around the origin (zero heading) for which the distribution of mapped event locations resembles that seen with a single-edge. Figure 3.11 shows a piecewise constant brightness profile and the associated event map required to achieve local sensing linearity ($h(s) \propto q$) at the origin $q = 0$. This map can also be interpreted as first assuming the heading is zero, then assigning each event to the nearest edge given that assumption and using the distance between that event and its assigned edge as a measurement of the heading.
Using the simple event map \( h(s) = s + c \), where \( c \) is a constant offset value determined by the centroid of the visible edges in the brightness profile at \( q = 0 \) would lead to mapped events representing unbiased measurements of the heading, however the variance of such a measurement would be much larger than that resulting from the piecewise linear map depicted in figure 3.11. Figure 3.12 compares raw \( (h(s) = s) \) and mapped event locations as a function of heading angle for a simple two edge case where

\[
m(s) = \begin{cases} 
0 & -12 < s_x < 12 \\
1 & \text{otherwise,}
\end{cases}
\]

and the event map for this case, as pictured generally in figure 3.11, is

\[
h(s) = \begin{cases} 
s - 12 & s < 0 \\
0 & s = 0 \\
s + 12 & s > 0.
\end{cases}
\]
Figure 3.12: Event locations and mapped locations for a 2 edge backdrop test, showing that near the origin \((q = 0)\), the event map results in a signal resembling \(-q\) with lower variance than in the unmapped case. Solid lines indicate the expected event location as a function of heading angle.

It is clear that in a region around \(q = 0\), the mapped event locations provide a measurement with much lower variance than the raw event locations, though both have the correct expectation. Farther from the origin however, while the raw locations remain unbiased (except by noise events), the map introduces a bias.
Figure 3.13: Regression error (estimator variance) between desired and actual signal on an event-wise basis for the three different test cases. High variance near the edges is caused by low sensor velocities during observations at these points.

Figure 3.13 summarizes these results by showing the mean squared event-wise regression error as a function of heading angle for the three different scenarios: one edge, two edges raw, and two edges mapped, and indicates that the noise on the mapped two edge signal is comparable to that for the single edge in a region around the origin. Outside this region, the error rises and is comparable to the raw two edge case, however there is also a bias introduced as seen in figure 3.12.

3.4 Global Maximum Likelihood Estimation

While it has been shown that a signal locally resembling the heading can be produced on an event-by-event basis using a computationally cheap map applied to event locations for an arbitrary piecewise constant brightness profile, the local nature of the result leaves much to be desired. Furthermore, as will be shown in section 3.5, the nature of the sensor allows lag to be reduced by maintaining a rolling buffer of past events, and it is therefore worthwhile considering controllers with some memory. So long as the amount of memory required is
small, this will not compromise the goal of producing controllers that could be implemented at a very low level in practice.

The methods and results of this section are also described in [129], which includes experimental results from subsequent sections as well.

3.4.1 Estimation Algorithms

Here we present an efficient method based on modified Bayesian inference, for incrementally adjusting a maximum likelihood estimate of the heading angle given a single new event. This estimate can be used to compute a proportional control term and will be shown in later sections to be empirically effective.

The update in typical Bayesian inference, computing the posterior distribution for the heading \( q(t) \) given a new event \( e \) can be written in the form

\[
\text{Pr}[q|e] \propto \text{Pr}[e|q] \text{Pr}[q].
\]  

(3.10)

where \( \text{Pr}[e|q] \) is the likelihood of observing event \( e \) given a particular heading angle \( q \).

In addition to the measurement update, the distribution \( \text{Pr}[q] \) would typically be propagated between measurements, for example if a Kalman filter or particle filter was used. A low-dimensional representation of \( \text{Pr}[q] \) such as a Gaussian may fail to sufficiently capture our belief, while for a high-dimensional representation such as a histogram, both the propagation and update are computationally taxing given our strict time constraints. So, while we would like to use an accurate representation of the distribution, we would also like to reduce the computation required to do so. Propagation can be eliminated based on the assumption that \( q \) is approximately constant over the duration of the past \( n \) events. This requires the further assumption that \( n \) is less than the number of visual edge pixels in the image at any time. For instance if the edges in the image on the sensor's pixel array subtend a total of \( M \) pixels, the sensor must output at least \( M \) events by the time the heading error has changed by one pixel.

Writing the updates for the previous \( n \) events together, where \( e_i \) is the \( i \)th most recent event, we have,

\[
\text{Pr}[q|e_1, e_2, \ldots, e_n] \propto (\text{Pr}[e_1|q] \text{Pr}[e_2|q] \text{Pr}[e_3|q] \ldots \text{Pr}[e_n|q]) \text{Pr}[q].
\]  

(3.11)

Based on this, we propose a rolling update to \( \text{Pr}[q] \):

\[
\text{Pr}[q|e] \propto \frac{\text{Pr}[e|q]}{\text{Pr}[e_n|q]} \text{Pr}[q],
\]  

(3.12)

where \( e_n \) is the \( n \)th most recent event not including the new event. This update rule uses only the \( n \) most recent events to estimate the heading, ignoring all events before that. This eliminates the need to propagate the distribution based on the dynamics of the system, which in addition to requiring more computation would also require a model for the dynamics of the environment that we do not assume is available. The amount of additional computation required to propagate the estimate between measurements depends on the choice of representation for the probability distribution. For a linear system with Gaussian estimate, propaga-
tion requires only a few matrix multiplications, while for a more complex representation and nonlinear dynamics the computation could be much greater.

Now we will use the pixel-level measurement model presented in section 2.4 to derive an explicit expression for the measurement likelihood $Pr[e|q]$. This will allow us to address the task of computing the updated estimate when the distribution over $q$ is represented as a histogram. For the purposes of this discussion we will neglect spurious events, so that the rate of events produced by pixel $i$, from equation 2.14 is

$$\lambda^i(t) = \frac{c_0 |q(t)| g_i(q(t), \rho)}{2 \rho h}.$$  \hfill (3.13)

Considering the zero fill-factor limit,

$$\frac{g_i(q(t), \rho)}{2 \rho} = |\nabla m|,$$  \hfill (3.14)

so that

$$\lambda^i(t) = \frac{c_0 |q(t)||\nabla m|}{h}.$$  \hfill (3.15)

The measurement model in section 2.4 also assumes independence between pixels, so that the probability that the most recent event occurred at pixel $i$ is proportional to the rate of events generated by that pixel. Thus, the probability that an event with location $s$ is generated given $q$ is proportional to spatial gradient of brightness at that location,

$$Pr[s|q] \propto \xi_m(s, t) = \frac{\lambda^i(t)}{\lambda(t)} \propto |\nabla m(s, t)|.$$  \hfill (3.16)

If we make the approximation that pixels cannot generate events unless the rate of brightness change exceeds some threshold value $h_r$, this model becomes

$$Pr[s|q] \propto \begin{cases} |\nabla m(s, t)| & |q| \nabla m(s, t) | \geq h_r \\ 0, & \text{otherwise} \end{cases}.$$  \hfill (3.17)

This approximation is based on empirical observations, and will allow for computational savings in updating the distribution $Pr[q]$. Because we are essentially filtering out events caused by smooth changes in brightness, the filtered event stream can be approximated as arising from a relatively small set of sharp edges in the image. This implies that events at a given pixel should occur only for a discrete set of heading angles, so that

$$Pr[s|q] \propto \sum_{i=1}^{k(s)} \delta(q - q_i) + p_0,$$  \hfill (3.18)

where $\delta(\cdot)$ denotes the delta function. The constant offset term $p_0$ is added to model the possibility that an event is generated for reasons beyond our model—random noise, small changes in the environment, etc. The set of heading angles for which events may be triggered at location $s$ is defined by the $q_i$ terms, while $k(s)$ denotes the number of such headings for
a given pixel location and will be important in determining the computational complexity of
the algorithm. In other words, \( q_i \) are the locations of the sharp edges in the image. Since
\( q \in \mathbb{R} \), it is reasonable to assume that none of the delta functions in the above equation lie
at exactly the same point and the probability distribution across heading angles given the \( n \)
most recent events \( E^n \) is,

\[
Pr[q|E^n] \propto \sum_{j=1}^{n} \sum_{i=1}^{k(s_j)} \delta(q - q^j_i) + p_0, \quad (3.19)
\]

where \( Q^j = \{q^j_i\} \) is the set of heading errors at which events are expected to occur for pixel
\( j \). This distribution is a constant offset plus a collection of delta functions representing the
unions of all such sets,

\[
Q = \bigcup_{j=1}^{n} Q^j. \quad (3.20)
\]

where \( Q \) represents a lookup table used to map an event location to a discrete set of potential
heading errors, and can be computed using prior knowledge of the brightness profile function
and an accurate calibration of the sensor optics, or an automatic calibration procedure as
described in section 3.4.3. One obvious way to estimate the heading error is to maintain a
histogram of the elements of \( Q \) and use it to approximately determine the maximum likelihood
estimate for \( q \). This can be done efficiently as new events are received by:

1. Given the new event location \( s \), look up the corresponding set of errors \( Q^j \)
2. Insert each element of \( Q^j \) into the histogram
3. Insert a reference to \( Q^j \) into an \( n \)-length FIFO queue
4. Remove the terminal element from the same FIFO queue, let it be called \( Q^f \)
5. Remove each element of \( Q^f \) from the histogram
6. Determine the new maximum occupancy bin of the histogram

This procedure is an implementation of the rolling update given in equation 3.12, under
the approximation that events are only generated at a discrete set of points for which the
brightness gradient is sufficiently high. Insertion of an element of \( Q^j \) into the histogram means
incrementing the associated bin while removal corresponds to decrementing the associated bin.
The insertion of all elements of \( Q^j \) is equivalent to multiplying the prior by the measurement
term \( Pr[e|q] \) (a constant value plus delta functions) in equation 3.12, and the division by
\( Pr[e_n|q] \) is equivalent to removing the elements of \( Q^f \) from the bins.

The heading value corresponding to the bin of maximum occupancy in the histogram is
used as the heading measurement, and lends this method the name Maximum Likelihood
estimation.
3.4.2 Computational Time and Space Complexity

Assuming basic operations such as data lookup, array element incrementation, and queue end insertion and removal all require computational time independent of \( n \) and \( k \), where \( k \) is the mean value for the number of heading errors triggering events at a given pixel, steps 1-5 require \( O(k) \) operations. The efficiency of step 6 depends on how the new maximum of the histogram is computed. Naively it requires \( O(N) \) operations, where \( N \) is the number of bins in the histogram, however if a max-heap is used the entire procedure is achievable in \( O(k\log(N)) \) operations, requiring \( k \) inserts and \( k \) removals at a cost of \( \log(N) \) each.

The memory required for this algorithm is also minimal, at \( O(nk + N) \) space required to store \( n \) lists of \( k \) errors and \( N \) bins storing small integer values in the histogram. The lookup table mapping event locations to lists of potential heading errors must also be stored, and requires \( O(kL) \) space, where \( L \) is the total number of pixels in the sensor. This gives an overall space complexity of \( O(nk + kL + N) \), which is very small in practice.

There is no reason why step 6 must be performed every time a new event is received, and indeed significant computational savings may be seen by re-computing the maximum likelihood estimate only every \( n \) events, since the assumption that \( \hat{q} \) changes by less than a single pixel over the course of \( n \) has already been used. If \( N \) is chosen proportional to \( n \), this would allow the amortized computational time of the final step to be \( O(1) \) and the total computational time complexity to be \( O(k) \), where \( k \) will typically be a small integer.

3.4.3 Calibration

The table of potential heading values indexed by event location, \( Q \), can be computed using knowledge of the function \( m \) and the parameters of the sensor optics. However \( Q \) can be constructed without the need for explicit optical calibration given a short sequence of event observations labelled with associated heading angles. In our case the experimental apparatus allowed for easy collection of event data with synchronized heading angle data while the sensor was driven along a sinusoidal trajectory with the backdrop fixed. Figure 3.14 shows the resulting distribution of heading angles for which events were generated at two different pixels.

![Figure 3.14](image)

Figure 3.14: Calibration histograms for two different pixels computed from 20s calibration data with the cartoon forest backdrop. Shows the count of events produced as a function of true heading angle as a blue histogram, overlaid with a smoothed version in red and identified peaks in green.
Given the raw histogram data shown in figure 3.14, a simple peak detection procedure was used to compute the small discrete set of heading errors for which events are expected at each pixel. This vastly reduces the computation required to update the maximum likelihood estimate of heading error, as detailed in subsection 3.4.1. The peaks computed are shown as green vertical lines in figure 3.14, and constitute $Q_i$ for the pixels shown. These are typical pixels, and the value of $k$ is 3.2 for the cartoon forest backdrop.

3.4.4 Experimental Verification

Before studying the behaviour of the closed-loop system controlled using the DVS128, tests were performed using open-loop data to verify correct operation and determine what to expect in closed-loop. Figure 3.15 shows the histogram over heading angles at two different times and depicts the typical performance of the sensor in estimating heading as it is actuated along a sinusoidal trajectory with fixed backdrop.

![Buffer-wise Heading Likelihood](image)

Figure 3.15: Typical sensor heading histograms with the cartoon forest backdrop, produced from the 256 most recent events and showing a robust peak when the sensor is moving that disappears when there is no motion of the sensor or backdrop.

Here we see that when the sensor is moving the maximum likelihood estimate is clearly discernable as a peak typically 3-4 times the height of the surrounding noise. However when the heading error is not changing, in this case when the sensor reaches the limits of its sinusoidal path, the events observed are attributable only to noise and the heading error cannot be measured. This is to be expected as the sensor sees only changes in brightness, however it does illustrate the most critical issue in using these sensors for tracking, which is that controllers must be able to deal with the fact that the system is unobservable when the error velocity is below some threshold. The controller can more easily be adjusted if it is known when this is the case, and there are several ways to determine whether the sensor is moving fast enough to see. As suggested by figure 3.15, one way is to use some measure.
of the certainty of the maximum likelihood estimate. Another is to use the overall event rate as a proxy for velocity, which was our experimental approach. Dealing with such a sensor theoretically is a rich problem beyond the scope of this paper, however we will see that empirically good tracking can be achieved by simply maintaining a zero-order-hold on the control input while the event rate is below a chosen threshold.

Figure 3.16: Distribution of DVS measured headings versus true headings measured by a potentiometer, for sinusoidal sensor trajectory with the cartoon backdrop. Field of view is 45 degrees.

The accuracy and reliability of the maximum likelihood estimate across a relatively wide range of angles is illustrated in figure 3.16, which plots true heading/estimated heading pairs for all event buffers received over the course of a 30 second sinusoidal trajectory. We see here also that at the limits of the sinusoid, occurring around +/-40 degrees, the estimation breaks down as expected. For both figures 3.15 and 3.16, events are sent from the sensor to the laptop in packets of up to 255 events, and estimates were computed on a packet-wise basis. This is simply a practical issue arising from the use of USB for communication, and the same methods could easily be used on an event-wise basis as previously described if low
level communication were available.

### 3.5 Filtering to Reduce Delay

Given the objective of using the lag identification results to understand the latency in the event-by-event processing case, it is important to ask how the buffer-wise lag characterization can provide insight into the latency of individual events. While it is not easy to draw conclusions about the event-wise sensor delay because the USB communication interface entails a significant overhead that renders buffering possibly more efficient than not, it is reasonable to think of the effective event-wise lag if all events are processed individually without memory as being much more like the buffer-wise mean than the buffer-wise mode. This fact leads to the question: is it possible to reduce the effective lag in the system by buffering events?

Typically, filtering measurements from a sensor involves a tradeoff between accuracy and latency, but in this section it will be seen that under a well motivated model for the dynamic vision sensor in the context of heading regulation with a single-edge, there is theoretical support for the claim that filtering events in a particular way can reduce lag compared to the unfiltered case. First considering figure 3.17, as well as qualitative observations, it is seen that during tests in which the sensor was actuated along a sinusoidal heading trajectory with a fixed, single-edge backdrop, a trail or “wake” of events follows the edge. This is consistent with persistent excitation spurious event models presented in previous sections, and with the conclusion illustrated in figure 3.8 as the mean of this distribution is -1.11 pixels, which corresponds to 2.1ms, while the lag between the mode and mean previously indicated was 2.3ms.
Figure 3.17: The distribution of event locations relative to the mode, corrected for the direction of motion so that negative relative location trails the mode and positive relative location leads it. Represents buffers captured during execution of a sinusoidal trajectory when the sensor velocity is sufficiently high for the mode to be a good indicator of position.

Note that there is also an apparent increase in the incidence of events in advance of the edge, which is defined as the mode event location. A short range apparent non-causal influence can be explained by the fact that buffering the events causes them to be spread out, though this should only account for a peak 3 pixels wide rather than 1. Additional effects that may give rise to this perplexing result are lens blur or flare issues, however the continued long range decrease of event rate in front of the edge is primarily attributed to the sinusoidal trajectory followed, and is consistent with the notion of persistent excitation when considering the time since traversal of the edge given the trajectory. Nonetheless, the curious effect of an apparently non-causal increase in event rate preceding the passage of the edge is eliminated in figure 3.18, which shows the difference between the left and right halves of the distribution shown in figure 3.17, flipped to indicate lag rather than relative time. This characterizes the wake of events trailing the edge.
Figure 3.18: Average "wake" of events following the edge, minus the distribution of non-mode events in front of the edge and expressed in terms of time rather than pixels. An exponential distribution with the same expectation is also shown (solid line) for comparison.

Comparing the distribution in figure 3.18 with the exponential of the same mean shown in the same figure explains the behaviour of the median in figure 3.8, which is much closer to the mode than the mean than would be expected of an exponential tail. The distribution is poorly modelled by a single exponential, and can be considered to have a "fat tail" that has led to higher lag in the expectation.

Stochastic delay of this kind was discussed in section 2.5.4, and has been incorporated into DVS measurement model 2.5.2. Figure 3.18 shows the empirically determined lag distribution $\phi(t)$ from section 2.5.4.

Given these empirical results, the DVS in the context of the single-edge heading regulation problem can be modelled as a unique kind of sensor that samples a signal subject to a stochastic lag. We will consider a continuous time signal $q(t)$, and a sensor which produces noise-free measurements of $q(t)$ at discrete moments in time $t_1, t_2, t_3, \ldots$. Let the corresponding measurements be denoted $z_1, z_2, z_3, \ldots$ respectively. The stochastic lag sensor model proposed is:
\[ z_i = q(t_i - \tau_i), \quad (3.21) \]

where \( \tau_i \) for \( i \in \{1, 2, 3, \ldots \} \) are a sequence of IID random variables with probability distribution \( f_\tau \) satisfying causality, i.e.

\[ f_\tau(\xi) = 0 \quad \forall \xi < 0, \quad (3.22) \]

and the conditions,

\[ \frac{df_\tau(\xi)}{d\xi} < 0, \quad E[\tau] = \int_{-\infty}^{\infty} \xi f_\tau(\xi) d\xi = \lambda^{-1}, \quad (3.23) \]

requiring it to be decreasing and have a well defined expectation \( \lambda^{-1} \). This measurement model is meant to capture the stochastic lag component of model 2.5.2 with respect to heading measurements produced by the DVS. We will also assume that the measured signal \( q(t) \) is piecewise linear and continuous.

**Proposition 3.5.1** (Lag Reduction for Stochastic Delay Systems). *Given the measurement model defined by equations 3.21, 3.22 and 3.23, maximum likelihood filtering using a rolling history of measurements of a piecewise linear signal produces filtered measurements with decreased expected delay and decreased variance in delay compared to the raw measurement signal.*

**Proof.** Consider at time \( t \) the previous \( n \) measurements \( Z^n(t) = \{z_1, z_2, \ldots, z_n\} \), made with a sensor subject to stochastic delay as described, and the associated empirical PDF \( \hat{f}_z \) and CDF \( \hat{F}_z \),

\[ \hat{F}_{z}^{a,b} = \hat{F}_z(b) - \hat{F}_z(a) = \int_a^b \hat{f}_z(\xi) d\xi = \frac{|Z^n(t) \cap [a, b]|}{n}. \quad (3.24) \]

First we will consider the case that all measurements were actually made simultaneously, so that the elements of \( Z^n(t) \) are identically distributed. The true CDF of \( z, F_z \) is related to the PDF of \( f_\tau \) through convolution with an indicator function,

\[ F_{z}^{a,b} = \int_0^\infty f_\tau(\xi) I_{a,b}(t - \xi) d\xi, \quad (3.25) \]

\[ I_{a,b}(t - \xi) = \begin{cases} 1 & q(t - \xi) \in [a, b], \\ 0 & \text{otherwise}. \end{cases} \quad (3.26) \]

The function \( I_{a,b} \) indicates whether the underlying signal being sampled, \( q(t) \), lies within \([a, b]\) at a given time. Since we are considering identically distributed measurements, from Borel's version of the Law of Large Numbers, we know that the empirical distribution approaches the true distribution as the number of samples approaches infinity,

\[ \hat{F}_{z}^{a,b} \rightarrow \int_0^\infty f_\tau(\xi) I_q(t - \xi) d\xi \quad \text{as} \quad n \rightarrow \infty, \quad (3.27) \]
i.e. the number of samples lying in any region \([a, b]\) approaches the true probability of a sample landing in that region. The idea is that using the maximum of the empirical distribution (computed for example by binning the distribution into a histogram) as the filtered signal can improve the performance of the estimate. While deterministic measurement delay in most systems cannot be reduced by filtering, systems with stochastic delay are an important exception, and the reasoning is based on the law of large numbers. In the interest of analyzing how the filter influences delay, we consider the case where the underlying signal \(q(t)\) is piecewise linear, so that measurements of \(q(\tau)\) can be treated as measurements of \(r\). This allows us to make claims in terms of the measurement delay, rather than estimation accuracy. In this situation \(F^\prime\) becomes \(F^\tau\),

\[
F^a,b = \int_a^b f_r(\xi) d\xi = \lim_{n \to \infty} F'^a,b.
\]

Given our conditions on \(f_r\) (equations 3.22 and 3.23),

\[
\arg \max_{\xi} f_r(\xi) = 0.
\]

Based on this we would like to use the argument of maximum of the empirical PDF \(\tilde{f}_r = \tilde{f}_z\) as the filtered signal. However the maximum of an empirical distribution for IID samples from a continuous PDF is not a well defined quantity, as the empirical distribution consists of a sequence of delta-functions at the sample locations. Let \(\tilde{f}_r\) be a piecewise constant approximation of \(f_r\) that approaches the true distribution in the limit of infinite samples,

\[
\lim_{n \to \infty} \tilde{f}_r = f_r,
\]

and maintains the property

\[
\tilde{f}_r(\xi) = 0 \quad \forall \xi < 0.
\]

Then

\[
\lim_{n \to \infty} \arg \max_{\xi} \tilde{f}_r(\xi) = 0,
\]

and using the argument of maximum of \(\tilde{f}_r\) as the filtered signal would yield zero lag. Again, this relies on the assumption that measurements are simultaneous, and is only true in the limit as \(n\) goes to infinity. Considering the case where the measurements are distributed in time and \(n\) is finite gives the intuition behind the fundamental tradeoff of this type of filtering. Let the expected filter delay for finite \(n\) be

\[
\beta(n) = E_Z[n \arg \max_{\xi} \tilde{f}_r(\xi)] \to 0 \quad \text{as} \quad n \to \infty,
\]

and \(T^n(t) = \{t_1 - t, t_2 - t, \ldots, t_n - t\}\) be the relative times at which the previous \(n\) measurements \(Z^n(t)\) were made - the times the sensor was queried, not including the random delay. Filtering using the empirical distribution of the measurements introduces a delay of \(\langle T^n(t) \rangle\) into the filtered signal resulting in expected filtered delay,
\[ E[\tau] = \langle T^n \rangle + \beta(n) \quad , \quad \langle T^n \rangle = \frac{1}{n} \sum T^n. \] (3.34)

If we assume a minimum inter-sample time of \( \epsilon \), then from causality

\[ \langle T^n \rangle < \langle T^{n+1} \rangle, \] (3.35)

and we see that \( T^n \) is increasing in \( n \) for any given time \( t \). Furthermore, it will be assumed that for a single sample \( (n = 1) \), the maximum of the approximated empirical distribution corresponds to the mean, so that

\[ \beta(1) = \lambda^{-1}, \] (3.36)

and again based on the Law of Large Numbers, the expected maximum of the approximated empirical distribution is non-increasing in \( n \),

\[ \frac{\partial \beta}{\partial n} \leq 0. \] (3.37)

So we see that expected delay in the filtered signal is the sum of a term increasing in \( n \), \( \langle T^n(t) \rangle \), and a bounded, non-increasing term that asymptotes to 0, \( \beta(n) \). If the maximum inter-sample time is also specified,

\[ \Delta = \max_i (t_{i+1} - t_i), \] (3.38)

the introduced lag is bounded,

\[ \langle T^n(t) \rangle \leq \frac{n \Delta}{2}, \] (3.39)

leading to the condition,

\[ \forall \lambda \exists n, \Delta : E[\tau] < \lambda^{-1}. \] (3.40)

This shows that for any stochastic delay model with \( f_\tau \) satisfying the conditions given, there is some choice of sampling period \( \Delta \) and filter buffer size \( n \) that leads to reduced lag compared to the raw signal input to the sensor.

Returning to the neuromorphic sensor case, each event can be thought of as an observation of the position of the single-edge and therefore the heading angle, and the wake shown in figure 3.18 serves as reasonable justification for a stochastic delay model for the sensor in which the wake defines \( f_\tau \). It is reasonable to assume, given the microsecond precision of the event timestamps, that the wake satisfies the required conditions 3.22 and 3.23, and therefore that the result in statement 3.40 holds. In this case however, because the inter-event time is not under our control, the result only suggests that it may be possible to improve lag by filtering. Based on wake observations, we will use \( \lambda^{-1} = 2.5 \text{ms} \) and an inter-event time \( \Delta = 2 \mu s \), and the analysis will be based on measurements spaced uniformly in time, which is approximately

77
accurate for an edge with sufficient contrast spanning many rows of pixels. Given equally spaced measurements,

\[ \mathbb{E}[\tau] = \frac{n\Delta}{2} + \beta(n), \quad (3.41) \]

and

\[ \text{Var}[\tau] = \frac{(n\Delta)^2}{12} + \beta_v(n), \quad (3.42) \]

where \( \beta(n) \) and \( \beta_v(n) \) are the expectation and variance of the delay if all measurements were made simultaneously, assuming no correlation between \( \tau \) and the time of the query. Now, because our approximation to the empirical distribution is a histogram with fixed bin size \( w \),

\[ \lim_{n \to \infty} \beta(n) = \frac{w}{2}, \quad \lim_{n \to \infty} \beta_v(n) = 0 \quad (3.43) \]

While this might suggest using a very small \( w \), doing so would result in a histogram that more accurately captures the highly discontinuous nature of the empirical distribution and therefore does not resolve the problem of a highly ambiguous maximum. Thus, there should be an optimal choice of \( w \) for any given \( n \) that balances smoothing the empirical distribution with reducing the lag introduced by binning. This optimization can be formulated as,

\[ J^* = \min_{n,w} \mathbb{E}[\tau]. \quad (3.44) \]

Here we wish to minimize the expected filtered measurement delay over the number of recent measurements used and the histogram bin size. Because the measurement delay is stochastic, the expectation alone does not fully describe it, and we are faced with the more general problem of determining a preferred distribution for \( \tau \) as a function of \( n \) and \( w \). That is to say we would like to determine which choice of \( n \) and \( w \) leads to the “best” distribution of \( \tau \). Above we have used expectation as a measure of optimality, but there is no single way to incorporate higher moments of the distribution into the cost function. One common approach is to consider two moments, as in,

\[ J^*(v) = \min_{n,w} \mathbb{E}[\tau] \quad \text{s.t.} \quad \text{Var}[\tau] = v, \quad (3.45) \]

where a solution is found for every value \( v \) which minimizes the expectation given fixed variance. In this case it would be reasonable to consider lower variance in sensor delay superior, so this formulation gives a means of trading off low expectation with low variance.

In order to solve these optimization problems numerically and determine the optimal filter parameters \( n \) and \( w \) for the DVS, the functions \( \beta(n) \) and \( \beta_v(n) \) must be evaluated. Both quantities depend on the distribution of the location of the bin of maximum occupancy in the histogram, which can be computed exactly using computationally tractable stochastic matrix methods described in [130].

Modelling the event wake distribution as exponential so that \( f_e(t) = \lambda e^{-\lambda t} \), and using a histogram with \( N \) non-overlapping, equal-width bins spanning a finite interval, the distribution of the maximum occupancy bin was computed for a range of values of \( n \) and \( N \). Level
sets of the mean and variance of filtered delay in filter parameter space are shown in figures 3.19 and 3.20 respectively.

Figure 3.19: Iso lines of expected lag in the space of histogram bins $N$ and sample history size $n$, for parameter values $k_0 = 2\mu s$ and $\lambda = 1\mu s$ approximately consistent with DVS performance data. The minimum expected lag of 385 $\mu s$ occurs at $N = 14$ and $n = 58$. 
Figure 3.20: Iso lines of variance in lag in the space of histogram bins $N$ and sample history size $n$, for parameter values $k_0 = 2\mu s$ and $\lambda = 1\text{ms}$ approximately consistent with DVS performance data.

These figures show that not only can filtering events in this way lead to dramatically reduced expected latency, it reduces the variance in latency even more, and while the merit of reducing variance is generally regarded a subjective matter, a preference for lower variance is widely considered rational. Figure 3.21 shows the mean and standard deviation along a slice of constant $N = 14$. 
Considering the mean and standard deviation together as in equation 3.45 results in an optimal frontier in filter parameter space. This frontier describes a set of optimal filter parameters to be chosen from based on preferences regarding mean and variance, and is shown in figure 3.22.
Figure 3.22: Optimal frontier in filter parameter space, labelled with expected delay in milliseconds. Curve gives the buffer parameters resulting in minimum variance in delay for a given expected delay.

The result of this method of filtering measurements can be seen in the case of a step change in the signal to be sampled, \( q(t) \). In this case, only two histogram bins are required, as each measurement is either 0 or 1 depending on the realization of \( \tau \) for that measurement. Figure 3.23 shows the result of this type of filtering on a signal undergoing a step change at \( t = 0 \), and indicates that in this case there is no clear best choice, since the probability that the output is correct is initially higher for the buffer-free case, while filtering causes a later, more rapid improvement in performance. If we were to describe the performance of the filter step response as the amount of time until the output matches the input with probability greater than some level, the chosen level would influence the optimal choice of buffer size \( n \). For levels much greater than 50%, with the test parameters used the optimal \( n \) would be much greater than 1.
Figure 3.23: Filter responses to a step change in the sampled signal, with random measurement delay. The probability that the output of the filter matches the true signal is shown for a number of rolling buffer sizes. Shows that increasing the buffer size decreases the time required for the output signal to be correct with high certainty, for example to achieve correct measurement with 90 percent chance a buffer of size 40 is best.

These results have shown that in contrast to the common scenario where filtering measurements cannot decrease lag and instead is based on a tradeoff between accuracy and latency, the presence of a random measurement delay allows the type of filtering described to reduce the expected latency and variance of latency simultaneously. The stochastic delay measurement model presented here is empirically consistent with the operation of the dynamic vision sensor in a heading regulation setup with a fixed backdrop or environment, and the result can therefore be applied to the use of DVS for heading regulation. In fact it has been experimentally demonstrated that DVS heading measurement lag is reduced using histogram mode filtering compared to mean event location, which is a reasonable proxy for event-by-event estimation. Furthermore, tractable computational procedures for determining the optimal buffer size and histogram resolution given sensor parameters have been presented.

3.5.1 Fixed Lag Model

Based on this analysis, we can reasonably approximate the post-filtering measurement model as having fixed constant lag $\tau$ or even zero lag. These modifications to the continuous-time measurement model given in Model 2.5.2 are shown here,
Model 3.5.1 (Constant Lag DVS Measurement Model).

\[
    z(t) = \left( q(t - \tau) + v(t) \right) \left( \dot{q}(t - \tau) + w(t) \right), \quad \tau \text{ constant}
\]

\[
    \mathbb{E}[v(t)] = 0, \quad \mathbb{E}[v(t)v(t - \tau)] = \delta(t - \tau) \mathbb{V}[v(t)] \\
    \mathbb{E}[w(t)] = 0, \quad \mathbb{E}[w(t)w(t - \tau)] = \delta(t - \tau) \mathbb{V}[w(t)]
\]

\[
    \mathbb{V}[v(t)] = \Lambda^{-1} \left( \frac{\sigma_v^2}{p} + \frac{L^2(1 - p)}{12p^2} + \frac{q^2(1 - p)}{p} \right)
\]

\[
    \mathbb{V}[w(t)] = \Lambda^{-1} (|\dot{q}(t)|^2 + \Gamma(t))
\]

\[
    p(t) = -\frac{\partial g(q, n, \rho)}{\partial q} \dot{q} + \lambda_S \\
    \Lambda(t) = \frac{n}{h} \left( \frac{c_0}{L} g(q, n, \rho) |\dot{q}| + \lambda_S \right), \quad \Lambda_0 = n \lambda_S \\
    \Gamma(t) = \frac{2Lh \lambda_s}{c_0 g(q, n, \rho) n} + \left( \frac{L \lambda_s}{c_0 g(q, n, \rho)} \right)^2
\]

Model 3.5.2 (Zero Lag DVS Measurement Model).

\[
    z(t) = \left( q(t) + v(t) \right) \left( \dot{q}(t) + w(t) \right)
\]

\[
    \mathbb{E}[v(t)] = 0, \quad \mathbb{E}[v(t)v(t)] = \delta(t) \mathbb{V}[v(t)] \\
    \mathbb{E}[w(t)] = 0, \quad \mathbb{E}[w(t)w(t)] = \delta(t) \mathbb{V}[w(t)]
\]

\[
    \mathbb{V}[v(t)] = \Lambda^{-1} \left( \frac{\sigma_v^2}{p} + \frac{L^2(1 - p)}{12p^2} + \frac{q^2(1 - p)}{p} \right)
\]

\[
    \mathbb{V}[w(t)] = \Lambda^{-1} (|\dot{q}(t)|^2 + \Gamma(t))
\]

\[
    p(t) = -\frac{\partial g(q, n, \rho)}{\partial q} |\dot{q}| + \lambda_S \\
    \Lambda(t) = \frac{n}{h} \left( \frac{c_0}{L} g(q, n, \rho) |\dot{q}| + \lambda_S \right), \quad \Lambda_0 = n \lambda_S \\
    \Gamma(t) = \frac{2Lh \lambda_s}{c_0 g(q, n, \rho) n} + \left( \frac{L \lambda_s}{c_0 g(q, n, \rho)} \right)^2
\]

3.6 Open-Loop Model Assessment

Previously in section 2.5 we extended the low level model of the dynamic vision sensor using a series of approximations to produce a sensor model whose configuration sensing component has the form,

\[
    z(t) = q(t - \tau) + v(t), \quad (3.46)
\]
where \( z(t) \) is a transformed event location serving as an additively corrupted measurement of the system configuration, \( \tau \) is a random measurement delay, and \( v(t) \) is additive noise whose variance goes inversely with the magnitude of the heading rate \(|\dot{q}|\) and approaches infinity as \(|\dot{q}|\) approaches 0. Both an asynchronous discrete-time measurement model (model 2.5.1) and a derived continuous-time model (model 2.5.2) have been presented, with the former being useful for algorithm design and experimental evaluation, and the latter being useful for control design and analysis. In this section experimental results are presented and compared to the predictions of the model, whose discrete form is used. Under this model, measurements arrive asynchronously according to a Poisson process, with an intensity that depends on the heading rate as outlined in section 2.5.

The previous sections described algorithms for performing high-speed heading estimation on an event-by-event basis in environments with arbitrary piecewise constant brightness profiles, and presented experimental results from actual DVS data confirming that using the prescribed estimation procedure events can still be treated as measurements of the heading.

In this section we present open loop sensor data to support the models given in section 2.5, and compare the predicted and observed properties of the measurements generated by the sensor.

### 3.6.1 Heading Measurements

We have so far derived from a pixel-level model of the DVS a higher-level model for the DVS as a configuration and configuration-rate sensor corrupted by additive noise for general piecewise constant brightness profiles. The objective of this derivation is to provide a model for the DVS that can be used for control design and comparison to conventional vision sensors, and it must therefore capture the important distinctions in behaviour between a DVS and a camera.

Figure 3.24 shows the measurement error autocorrelation both on an event-wise basis (i.e. when each event is taken as a measurement of heading), and on a 256-event buffer-wise basis (i.e. where the mode location of the 256 most recent events is taken as a heading measurement), for events captured while the sensor followed an approximately sinusoidal trajectory. The low autocorrelation values support the claim that the noise is well modelled as white.
Figure 3.24: Observed autocorrelation in measurement error for the linearized sensor model with the single-edge backdrop. Shows that at both an event and 256-event buffer level, autocorrelation is low and the assumption of rate-conditional independence of measurements is appropriate.

In the limit where the heading rate approaches infinity $\dot{\theta} \rightarrow \infty$, the heading measurement error variance approaches the constant value $\sigma_{\theta_0}^2$, which corresponds roughly to the width of a single pixel. Figure 3.25 verifies this empirically by showing the distribution of observed measurement errors compared to a predicted distribution with a standard deviation of 1 pixel, for heading rates in excess of 120 degrees per second.
Another important aspect of the derived model is the dependence of measurement error variance (and bias for uncorrected measurements) on the magnitude of the heading rate. Figure 3.26 provides experimental verification of this phenomenon, showing measurement error as a function of heading rate. It can be seen from these figures that the direction of motion is not important, and that error variance increases as the magnitude of the rate decreases, for both the single-edge backdrop and cartoon forest backdrop with maximum likelihood estimation. Furthermore, the error bias is shown to increase as heading rate decreases, as predicted by the sensor model, and as is also clear from figures 3.9 and 3.16 When this bias is cancelled out, it leads to variance increasing without bound as heading rate approaches zero, as previously discussed.
Figure 3.26: Illustration of the influence of heading rate on measurement error before bias correction. Comparison between observed measurement variance and that predicted by the model showing that the model captures the general trend but fails to predict the observed sharp drop-off in variance above 40 deg/s.

In order to correct for the bias in measurements predicted by the model, a feedback linearization procedure was proposed in which a transformed measurement with no bias was obtained by dividing the raw measurement by the calculated probability that the event was caused by brightness changes and not exogenous noise (equation 2.35). This procedure relies on the prediction from the model that the overall event rate increases predictably with the heading rate and can therefore be used to compute heading rate.

### 3.6.2 Heading Rate Measurements

To this point we have largely neglected considerations related to event polarity. However the polarity of events contains information about the scene and platform motion, specifically event polarity is highly correlated with the direction of motion. In combination with a measurement of the instantaneous event rate, which can be used to estimate the magnitude of the heading rate, the polarity can be used to produce a measurement of the signed heading rate.

While we have not presented a formal model for how event polarity is determined, roughly speaking increases in brightness lead to events with an “up” polarity while decreases in bright-
ness lead to “down” events. For a single-edge environment this implies that event polarity should be a very good indicator of the sign of heading rate, which indeed it is (figure 3.27). In this case the most common event polarity among the 256 most recent events agreed for 97 percent of events with the sign of the heading rate.

Using polarity to measure the sign of heading rate is easily extended from the single-edge case to arbitrary piecewise constant brightness profiles through a procedure similar to that presented in section 3.4.

Figure 3.27: Depiction of the relationship between event polarity and the sign of heading rate, $\dot{\theta}$, for a single-edge backdrop. On an event-wise basis event polarity matches motion direction for 97 percent of events. Color indicates most common event polarity among the 256 most recent events.

Figure 3.28 shows how the measurement error variance in the heading rate measurement increases with heading rate, verifying the prediction of the model. Particularly for heading rates above 100 degrees per second the model predicts the behaviour of the system well.
Figure 3.28: Event rate versus absolute velocity, observations and model. Shows that for angular rates above 100 deg/s the model predicts the measurement variance accurately, while for lower speeds the variance in rate measurement is slightly underestimated.
Figure 3.29: The relationship between absolute heading rate and event rate, for two different brightness profiles, showing a strong linear trend in both. Supports the idea that heading rate measurements based on event rate are similar across different environments.

Figure 3.29 shows the relationship between event rate (as calculated from a buffer of the 256 most recent events) and absolute heading rate, for the single-edge and cartoon forest environments. In both cases a reasonable positive correlation is seen, indicating that the event rate could potentially be used to compute heading rate.

### 3.7 Feedback Control with DVS

In this section we design control laws based on the continuous-time sensor model for the DVS detailed in section 2.5, and present experimental results comparing their performance to that of low-level control based on feedback from rotary potentiometers.

Given an assumption of linear dynamics, and our apparently linear sensor model, it would seem that control design is now a textbook task. This is not true however due to the unique nature of the additive noise in the model. In this section we present and analyze a number of approaches to stabilizing the system, where we refer to stability as the convergence of the expectation and variance of the steady state distribution over system states.

We will consider a model similar to Model 3.5.2, wherein we assume perfect measurement of heading rate. This measurement model is given as Model 3.7.1.
Model 3.7.1 (Continuous-Time DVS and Noise-Free Rate Measurement Model).

\[
\begin{align*}
    z(t) &= \left( q(t) + v(t) \right) \\
    \mathbb{E}[v(t)] &= 0, \quad \mathbb{E}[v(t)v(t)] = \delta(t) \text{Var}[v(t)], \quad v(t) \text{ Normal} \\
    \text{Var}[v(t)] &= \Lambda^{-1} \left( \sigma_v^2 p + \frac{L^2}{12} (1 - p) + q^2 p (1 - p) \right) \\
    p(t) &= \frac{\rho \varphi \varphi g(q, n, \rho) |\dot{q}|}{\rho \varphi \varphi g(q, n, \rho) |\dot{q}| + \Lambda} \\
    \Lambda(t) &= \frac{n}{h} \left( \frac{\rho \varphi \varphi g(q, n, \rho) |\dot{q}| + \Lambda_s}{\rho \varphi \varphi g(q, n, \rho) |\dot{q}| + \Lambda_s} \right), \quad \Lambda_0 = n \lambda_s
\end{align*}
\]

3.7.1 Stability

We would like to design controllers that by some reasonable definition stabilize the system, so that the closed-loop dynamics satisfy some notion of stability. For our purposes we will define stability as the existence of a steady-state distribution across states of the system whose mean is zero and whose variance is finite.

Definition 3.7.1 (Stability). A system with state \( x \) is stable if, as time approaches infinity the expectation of \( x \) approaches zero and the covariance of \( x \) approaches a finite quantity, i.e.

\[
t \to \infty \Rightarrow \mathbb{E}[x] \to 0, \quad \mathbb{E}[xx^T] \to C < \infty.
\]

3.7.2 Proportional-Derivative Control

Because the sensor model provides us with measurement signals for both the configuration \( q \) and the configuration rate \( \dot{q} \), it is natural to think of proportional-derivative (PD) control if the dynamics of the plant are second order and linear. However this is problematic because of the nature of the noise. Specifically, since the variance in the noise added to the heading measurement goes to infinity as the heading rate goes to zero, using a global PD controller means feeding back a signal with potentially arbitrarily high variance. To make things worse, we are trying to stabilize the system to zero heading rate, meaning that at least being near states for which measurement error variance is infinite is inevitable.

Theorem 3.7.1 (Instability of PD Control). A single input LTI dynamical system with measurement model 3.7.1 is unstable under the action of LTI output feedback control (PD).

Proof. Consider a LTI system with internal state \( x \), defined by

\[
\dot{x} = Ax + Bu, \quad x_1 = q, \quad x_2 = \dot{q},
\]

combined with measurement model 3.7.1 in closed-loop using a LTI controller, i.e.

\[
u = Kz = k_1 q + k_2 \dot{q} + k_1 v,
\]

92
to give the closed-loop dynamics,

\[ \dot{x} = Ax + B(k_1q + k_2\dot{q}) + Bk_1v. \]  

(3.50)

Because \( v \) has infinite variance at \( \dot{q} = 0 \), there cannot exist a well-defined steady state distribution for \( x \). This is because the noise is Gaussian and therefore has full support, which means that the steady state distribution must have full support. So, if a steady state distribution with finite variance were to exist, it would have nonzero probability mass at \( \dot{q} = 0 \), meaning that the distribution for \( \dot{q} \) would be undefined as a result of the expression for the non-conditional variance of the heading measurement noise \( v \), which is,

\[ \text{Var}[v] = E[v^2] = \int_{-\infty}^{\infty} \text{Pr}[v = z]z^2dz, \]  

(3.51)

\[ \text{Pr}[v = z] = \int_{-\infty}^{\infty} \text{Pr}[v = z|\dot{q} = y] \text{Pr}[\dot{q} = y]dy, \]  

(3.52)

\[ \text{Var}[v] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \text{Pr}[v = z|\dot{q} = y] \text{Pr}[\dot{q} = y]z^2dydz \]  

\[ = \int_{-\infty}^{\infty} \text{Pr}[\dot{q} = y] \int_{-\infty}^{\infty} \text{Pr}[v = z|\dot{q} = y]z^2dzdy, \]  

(3.53)

and from the sensor model,

\[ \int_{-\infty}^{\infty} \text{Pr}[v = z|\dot{q} = y]z^2dz \sim \frac{1}{|y|}, \]  

(3.54)

so the variance of the measurement noise, not conditioned upon the state is,

\[ \text{Var}[v] \sim \int_{-\infty}^{\infty} \frac{\text{Pr}[\dot{q} = y]}{|y|}dy, \]  

(3.55)

and we see that a candidate steady state distribution where \( \text{Pr}[\dot{q} = y] \) does not equal zero at \( y = 0 \) is not valid. Furthermore, since the measurement noise has full support for all values of \( \dot{q} \), any candidate distribution satisfying \( \text{Pr}[\dot{q} = 0] = 0 \) would immediately violate this condition as a result of the dynamics. Thus, using PD control cannot lead to a well defined steady state distribution, although it may in practice provide reasonable performance.

\[ \square \]

Generalizing the analysis, we see that this problem with lack of convergence of the distribution of states will occur for a large class of control laws including those where the control signal increases without bound in \( q \) for some cases where \( \dot{q} = 0 \). In other words, the controller cannot feed back noise with infinite variance.
3.7.3 Bounded Input Control

The discussion of PD control has elucidated the idea that controlling a system with this type of noise is not as straightforward as the notation of the model might suggest. One obvious solution to the problem of feeding back noise with infinite variance is to bound the control input. In practice all systems face input saturation at some level, and introducing artificial input saturation can also ensure stability, as long as the saturation does not cause instability of the noise-free system.

However for systems with unstable dynamics, control with input saturation will result in a loss of global stability, which is not desirable and certainly represents a shortcoming if the input saturation is artificial.

In general, stability of the noisy system can be proven if the closed-loop dynamics can be divided into a deterministic term and an exogenous noise term by proving stability of the deterministic (noise-free) system and boundedness of the noise. Let \( G \) and \( G_{NF} \) be two similar second order LTI systems,

\[
G : \dot{q} = a_1 q + a_2 \dot{q} + r \quad r \sim \mathcal{N}(0, \sigma^2(q, \dot{q}))
\]

\[
G_{NF} : \dot{q} = a_1 q + a_2 \dot{q}.
\] (3.56)

**Lemma 3.7.1 (Stability with Bounded Noise).** For a second order LTI system with additive zero-mean Gaussian noise, if the noise-free system is asymptotically stable and the noise variance is bounded in magnitude, then the system is stable. i.e.

\[
(G_{NF} : t \to \infty \Rightarrow (q, \dot{q}) \to 0), \quad (\exists c_0 \in \mathbb{R} : |r| \leq c_0 < \infty),
\]

\[\Rightarrow (G : t \to \infty \Rightarrow \mathbb{E}[q, \dot{q}] \to 0, \text{Var}[q, \dot{q}] \to c_1 < \infty).\] (3.57)

**Proof.** Let \( x = [q, \dot{q}]^T \) denote the state of the system, and let the dynamics from equation 3.56 be written,

\[
\dot{x} = Ax + R
\] (3.58)

First we show that the expectation \( \mathbb{E}[x] \) tends to zero. The evolution of the expectation is dictated by,

\[
\frac{d}{dt} \mathbb{E}[x] = \mathbb{E}[\dot{x}] = A \mathbb{E}[x],
\] (3.59)

because the noise \( R \) has zero mean. Since the noise-free system is stable, this means that the first condition for stability is met, i.e.

\[
\mathbb{E}[x(t)] \to 0 \text{ as } t \to \infty.
\] (3.60)

Since we have also assumed that the noise \( r \) is bounded, we can consider all probability distributions for \( R \) allowable under this constraint. Since \( r \) is Gaussian with bounded variance dependent on the state, we can simply consider its maximum attainable variance across all states. If the system is stable under the action of the noise of maximum variance, it must be stable, and from textbook control theory we know that the steady state distribution when the
additive noise has fixed variance is a well-defined Gaussian, the system must be stable even subject to noise.

3.7.4 Switching Control

A simple alternative to artificial input saturation is to use open-loop control when $\dot{q}$ is less than some threshold level and closed-loop control otherwise. Here we will consider the example of a double integrator plant such that,

$$\ddot{q} = u,$$

with the control law,

$$u(t) = \begin{cases} -z_1(t) - z_2(t) & \text{if } |z_2(t)| > c \\ -z_1(t_{cross}) - z_2(t_{cross}) & \text{if } |z_2(t)| < c, \end{cases}$$

(3.62)

where $t_{cross}$ is the most recent time the threshold $|z^2| = c$ was crossed. Inside the threshold region, $|z^2| < c$, the controller holds the input at what it was when it last entered the region. For now we will assume zero noise on the rate measurements, so that $\dot{z}^2 = \ddot{q}$.

**Theorem 3.7.2** (Stability of Switching Control). A system with double integrator dynamics and heading measurement model 3.7.1 is stabilized by the control law given in equation 3.62.

**Proof.** Since we have assumed that the rate measurements are perfect, it is immediately clear that this control law satisfies the bounded disturbance condition, so all that remains is to prove stability of the deterministic noise-free system.

Consider the Lyapunov function candidate $V(q, \dot{q})$,

$$V(q, \dot{q}) = \frac{1}{2} \dot{q}^2 + \frac{1}{2} q^2.$$  

(3.63)

Clearly this function is bounded from below and increases without bound in all directions as $|q, \dot{q}|$ goes to infinity. Furthermore, when the first case in the control law is satisfied, the system is effectively under PD control, and $V$ is decreasing,

$$|\dot{q}| > c \Rightarrow u = -q - \dot{q} \Rightarrow \dot{V} = -\dot{q}^2 \leq 0.$$  

(3.64)

When the system reaches the threshold $|\dot{q}| = c$, it switches to open-loop control, during which time the function $V$ is not necessarily decreasing. However, it is guaranteed to reach the threshold again in a finite time, at which point the value of $V$ is the same as it was when the threshold was first crossed. This is a consequence of the symmetry of phase plane paths for the double integrator under constant control input - starting at $\dot{q} = c$, the system will follow a parabolic trajectory that reaches $\dot{q} = -c$ at the same value of $q$.

Since the discrete-time function $V_k = V(t_k)$ where $t_k$ is the $k^{th}$ crossing of the threshold $|\dot{q}| = c$ is decreasing,

$$V_{k+1} \leq V_k,$$  

(3.65)
and the time between successive crossings is bounded, the noise-free system is stable, and therefore the noisy system is stable as well.

Figure 3.30: Simulation results showing heading and control signals for feedback control of a double integrator using either PD control or the switching control described here and the continuous-time DVS sensor model. Shows a considerable improvement in steady state tracking with the switching controller.
Figure 3.31: Comparison of steady state tracking errors for PD control versus switching control for a double integrator with the continuous-time DVS sensor model. Shows the distribution of long-time tracking errors computed by simulation for the two different control laws, and indicates considerably better performance from the switching controller.

In addition to being provably stable this controller gives a significant improvement in behaviour compared to PD control. Figures 3.30 and 3.31 show results from a simulation comparison of the two controllers driving a double integrator plant subject to additive measurement noise consistent with that of the DVS model,

\[ v \sim \mathcal{N}(0, \frac{1}{q^2}), \]  

and with proportional and derivative gains both equal to one \((k_1, k_2 = 1)\) and the switching threshold set at \(c = 0.1\). The figures illustrate the fact that the switching controller yields far superior steady state performance, without sacrificing step response time. Specifically, the mean steady state error is about one order of magnitude less, and the worst case heading error across 20k sample points is 148 times less. The theory and simulation together suggest that this switching control scheme should not sacrifice response time, since it is essentially similar to PD for large displacements, and is greatly superior to PD control at rejecting the unique type of noise associated with DVS.
3.7.5 Experimental Results

A number of experimental tests were conducted to verify the efficacy of the proposed control methods on a real system. Shown in figure 3.10 is the cartoon forest backdrop used for testing, and the measurement signals \( z^1 \) and \( z^2 \) corresponding to the heading \( q \) and heading rate \( \dot{q} \) respectively are derived using the efficient global measurement techniques described in section 3.4. A PD control scheme with switching as described in the previous subsection was used to control the DVS, while comparison tests with the potentiometer used pure PD control with equivalent gains. The control gains were selected using an LQR design process based on the plant model described in section 3.2 and are tabulated along with other relevant parameters in table 3.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>P Control Gain</td>
<td>( k_1 )</td>
<td>1.3 V/deg</td>
</tr>
<tr>
<td>D Control Gain</td>
<td>( k_2 )</td>
<td>0.013 V/(deg/s)</td>
</tr>
<tr>
<td>DVS Switching Threshold</td>
<td>( c )</td>
<td>10 deg/s</td>
</tr>
<tr>
<td>DVS Buffer Size</td>
<td></td>
<td>256 events</td>
</tr>
<tr>
<td>ML Filter Histogram Bins</td>
<td>( N )</td>
<td>180</td>
</tr>
<tr>
<td>ML Filter Histogram Bin size</td>
<td></td>
<td>1.5 deg</td>
</tr>
</tbody>
</table>

Table 3.1: List of physical parameters used in experimental tests.

Table 3.2 lists the

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor and Backdrop Motors</td>
<td>60:1 Pololu metal gearmotors</td>
</tr>
<tr>
<td>Motor Controller Boards</td>
<td>Arduino Pro 5V w/ ATMEGA328</td>
</tr>
<tr>
<td>Microcontroller Board</td>
<td>360deg continuous rotation</td>
</tr>
<tr>
<td>Potentiometers</td>
<td>iniLabs DVS128 2013</td>
</tr>
<tr>
<td>Dynamic Vision Sensor</td>
<td>Matrix Vision mvBlueFox</td>
</tr>
<tr>
<td>Conventional Camera</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.2: List of physical components used in experimental tests.

The signal connections in the closed-loop experimental tests with actuated backdrop are shown in figure 3.32. The backdrop is actuated independently of the sensor, and has its angle measured at high rate by a potentiometer connected to the ADC of a microcontroller. Both the heading angle and the reference (backdrop) angle can be read at approximately 5kHz with approximately 200\( \mu \)s of delay using this apparatus. As in tests without an actuated backdrop, the DVS communicates via USB with a laptop computer, which sends control commands to a microcontroller over USB as well.
Figure 3.32: Schematic of electronic signals in the closed loop experimental apparatus, showing the independent actuation of the backdrop and plant (sensor). In several of the tests the backdrop is actuated along an approximately sinusoidal trajectory, while in others it remains fixed.

Figure 3.33 extends the simulation comparison between the PD controller and our switching controller to the real world, showing step responses for both control laws in the experimental apparatus. As in the simulation results, the steady state tracking performance is superior for the switching control law, with no decrease in step response performance.
Figure 3.33: Comparison of step responses using PD control and the switching control described in this section, on the rotating drum experimental apparatus. Agrees with simulation results to show that the switching controller can be more effective in practice.

Figures 3.34 and 3.35 compare the performance of the DVS- and potentiometer-based control schemes, with a step response shown in 3.34 and tracking of a sinusoidal reference shown in 3.35. While the performance of the potentiometer is clearly better, the DVS provides comparable rise time in the step response with only slightly more overshoot and settling time. It is also important to note that the step height is approximately 40 degrees, which is nearly as large as the 45 degree field of view of the DVS. This indicates that the global measurement algorithm given in section 3.4 is effective at providing global measurements of heading, confirming the open-loop results. The sinusoidal tracking performance is also slightly worse for the DVS than the potentiometer in terms of overshoot, however the phase lag is comparably small.
Figure 3.34: Comparison of step responses between the DVS with cartoon forest backdrop and an onboard potentiometer-based control loop. Controllers with equivalent gains were used. Shows the capability of the DVS to match an embedded system at control speed with only slightly more overshoot and initial heading greater than 40 degrees (compared to 45 degree total FOV for the DVS).

Figure 3.35: Comparison of sinusoidal reference tracking between the DVS with single-edge backdrop and an onboard potentiometer-based control loop. The reference signal (backdrop heading) is approximately a 2Hz sinusoid with 20 degree amplitude. Controllers with equivalent gains were used. Shows that the performance of the DVS is considerably worse than that of the potentiometer with respect to overshoot but comparable in terms of lag.

Figure 3.36 shows results for the DVS alone tracking a lower frequency sinusoidal reference (1Hz), for which the overshoot is relatively less than in the faster case, but comparable in overall magnitude at around 5 degrees or 16 pixels. The high overshoot observed when using the DVS to track a sinusoidal reference trajectory is explained partially by the fact that when the backdrop slows down and changes directions the tracking error rate must pass through
zero, causing a situation where the switching controller leaves the control input in a zero order hold and leading to overshoot. This effect is mitigated at lower reference signal frequencies because relative to the frequency the time during which the DVS is unable to make reliable measurements is shorter.

![](image)

Figure 3.36: Sinusoidal reference tracking test results using the DVS and cartoon forest backdrop, with backdrop actuated at 1Hz across 70 degrees. Shows the ability of the DVS controller to track a moving reference with reasonable accuracy.

These experiments overall demonstrate the efficacy of the signal estimation and control algorithms presented in this chapter and show that it is possible to achieve high-speed visual servoing using dynamic vision sensor feedback that is comparable in tracking performance, particularly lag, to low level embedded control.

### 3.7.6 Relation to Event Triggered Control

The asynchronous measurement model that we have discussed is in some ways essentially similar to the event-triggered measurement model presented in the literature, for example in [80]. In these models, measurements are triggered whenever the configuration (in our case heading) deviates by some threshold amount from the origin. This is analogous to the way the DVS produces events asynchronously, and we can think of the width of a pixel as being the maximum event threshold from event triggered control. Returning to our assumption that the number of visible edge pixels in the image is greater than \( n / \alpha \) at any time, for some small constant \( \alpha \geq 1 \), it is not possible for the system to deviate by more than the angle subtended by \( \alpha \) pixels before \( n \) events are produced, and we see that sensor model maps well to the theoretical event-based framework. In [131], Heemels et al. show how a controller can be designed for a system with arbitrary perturbed linear dynamics under such a measurement model in which the full state is measured whenever the system deviates by some threshold value from the origin, and present theoretical results showing "practical stability", meaning long-time bounded deviation from the origin. These results were extended in [89] to deal with
the output feedback case, and are thus directly applicable to our problem. In the specific problem of interest, mechanical control of sensor heading, it is necessary to introduce process noise in order for the system to avoid becoming stuck at positions more than $\alpha$ pixels from the origin, and this is consistent with the requirement of perturbations in [131].

One important distinguishing characteristic of our model is that it accounts for spurious events, which cannot be easily dealt with under the framework of event-triggered control. Another distinction comes into play if we include pixel adaptation in the DVS model, which acts as a high pass filter on brightness changes. If this phenomenon is modelled, fast changes in brightness produce more events than equivalent slow changes, and it is no longer possible to say with certainty that at least $n$ events must be produced for each pixel worth of rotation. The DVS model presented does not explicitly account for this, but it also carries no guarantees about the number of events produced for a given rotation and is therefore distinct from models based on event triggered control.

3.7.7 Dithering and Microsaccades

Given the fact that the noise in our DVS model has variance that depends on the state of the system, it is natural to ask whether it may be beneficial to manipulate the state in a way that improves the accuracy of measurements. In general, separation of estimation and control will not be optimal when noise variance depends on state, and the optimal controller may cause behaviour intended to improve measurement accuracy at the cost of short term tracking accuracy. In our case, measurement accuracy increases with the configuration rate $\dot{q}$, so we must face a tradeoff between tracking accuracy, for which the ideal state state rate is zero, and measurement accuracy.

One of the simplest ways to manipulate the state to increase accuracy is dithering, which can be thought of as the addition of a periodic or random signal to the base control input for the purposes of increasing the magnitude of steady state configuration rate. Seemingly paradoxically this may lead to improved tracking performance due to increased measurement accuracy. It is interesting to note that many organisms in nature dither their eyes (or heads where their eyes cannot move), with motions called microsaccades. Since we are dealing with neuromorphic vision sensors whose operation mimics biological retina, it is fitting that introducing microsaccades may serve a purpose for us as well.

An alternative explanation for biological microsaccades based on our neuromorphic sensor model is that they are an unintended consequence of trying to stabilize such a sensor using a simple control scheme, as seen for PD control in figure 3.30. In fact, the switching controller also results in steady state noise that could be described as microsaccades, and the question becomes: are natural microsaccades added to improve behaviour, or a consequence of the nature of biological vision? Since zero heading rate means the system is unobservable, and since any unstable or marginally stable dynamics would require observability to be stabilized, the steady state distribution for such a system under the action of a stabilizing control must have mass for some nonzero heading rates.

By feeding back noisy measurements, we are already introducing closed-loop process noise, but it may be beneficial to introduce additional process noise, particularly if the added dithering has some temporal structure. Consider the control law,
\[ u = -b_p z_1(t) - z_2(t) + \text{sign}(\sin(\omega_d t)), \quad (3.67) \]

where \( b_p \) is an indicator used to switch the proportional feedback term on and off and is given by,

\[ b_p = \begin{cases} 
0 & \text{if } |z_2(t)| < \omega_d^{-1} \\
1 & \text{otherwise}. 
\end{cases} \quad (3.68) \]

In this case the parameter \( \omega_d \) is the dithering frequency. This controller attempts to stabilize the heading rate \( \dot{q} \) to 1 and -1 alternately, with period defined by \( \omega_d \). In the absence of the proportional control term \((b_p = 0)\), this leads to oscillations in heading rate whose amplitude depends on the frequency response properties of the system. For a simple double integrator as we have previously considered (single integrator from input to rate), the resulting rate oscillation amplitude will be approximately inversely proportional to the dithering frequency. This is why \( \omega_d^{-1} \) appears in the switching condition for the proportional control term.

**Theorem 3.7.3** (Stability of Dithering Control). A system with double integrator dynamics and heading measurement model 3.7.1 is stabilized by the control law given in equations 3.67 and 3.68.

**Proof.** From Lemma 3.7.1 we know that stability is proven if we can show that the additive noise acting on the closed-loop system is bounded and that the noise-free system is stable. Since we have assumed perfect rate measurements, our thresholding condition defining \( b_p \) (equation 3.68) ensures that the noise disturbance is bounded. Thus, we must only show stability of the noise-free system.

When \( b_p = 1 \), the proportional term is turned on and the linearity of the dynamics implies that the resulting trajectory will be a linear combination of the homogeneous solution (undriven, no dithering), and the response to the dithering input. Since the dithering input has no constant offset, by symmetry the system will converge to follow an oscillatory trajectory with mean location \((q, \dot{q}) = 0\). This is also true of the heading rate only when \( b_p = 0 \). In other words,

\[ b_p = 1 \Rightarrow E[q, \dot{q}] \to 0 \text{ as } t \to \infty \]
\[ b_p = 0 \Rightarrow E[q] \to 0 \text{ as } t \to \infty. \quad (3.69) \]

If \( b_p = 0 \), the system will asymptote to following an oscillatory trajectory centered on \( \dot{q} = 0 \). If the amplitude of these oscillations strictly exceed the chosen dithering threshold \( \omega_d^{-1} \), then the proportional term is guaranteed to be activated within at most one dithering period and remain activated for a finite period of time. Therefore the system must spend a finite fraction of its time with \( b_p = 1 \). Consider again the Lyapunov function candidate \( V(q, \dot{q}) \),

\[ V(q, \dot{q}) = \frac{1}{2} q^2 + \frac{1}{2} \dot{q}^2. \quad (3.70) \]

Clearly this function is bounded from below and increases without bound in all directions as \([q, \dot{q}] \) goes to infinity. If \( b_p = 1 \),
\[ \dot{V} = -q^2 \leq 0, \]  
(3.71)

and this is a valid Lyapunov function indicating that the system is stable. If \( b_p = 0 \) the Lyapunov function candidate is not valid as the sign of \( \dot{V} \) is uncertain. However, if we consider the value of \( V \) at successive times entering the threshold region \( |\dot{q}| < \omega_q^{-1} \), we can obtain a discrete time Lyapunov function.

Under the action of the controller with \( b_p = 0 \), the asymptotic behaviour of the system is as depicted in figure 3.37. We will consider the Lyapunov function candidate \( V \) at the points labelled 1, 2, 3 and 4 in the diagram, \( V_1, V_2, V_3, V_4 \), as well as the corresponding heading and heading rate labelled with the associated subscripts.

![Diagram](image)

Figure 3.37: closed-loop trajectory diagram used in the proof of stability of a double integrator under the action of the dithering controller.

If we introduce a switching threshold on \( \dot{q} \) as we have described (switching on the dotted lines in the figure), from our previous observation that the function is decreasing when \( b_p = 1 \),

\[ V_2 - V_1 < 0, \quad V_4 - V_3 < 0. \]

(3.72)

Furthermore, we observe by symmetry that

\[ q_1 - q_4 = q_2 - q_3, \]

(3.73)

and

\[ q_2 > q_3 \Rightarrow q_2^2 - q_3^2 > q_1^2 - q_4^2. \]

(3.74)

So if we consider the change in \( V \) around a single cycle, \( \Delta V \),

105
\[ \Delta V = (V_2 - V_1) + (V_3 - V_2) + (V_4 - V_3) + (V_1 - V_4) \]
\[ = ( < 0 ) + (V_3 - V_2) + ( < 0 ) + (V_1 - V_4) \]
\[ = ( < 0 ) + \frac{1}{2}(q_1^2 - q_2^2 + q_3^2 - q_4^2) < 0 \]  
\[ (3.75) \]

and we see that the function \( V \) must decrease between successive cycles, so that if we consider \( V_k \) to be the value of \( V \) the \( k^{th} \) time the system exits the threshold region with \( \dot{q} > 0 \) (point 1 on the diagram) then,

\[ V_{k+1} \leq V_k, \]  
\[ (3.76) \]

and the system converges to an oscillatory trajectory centered at \((q, \dot{q}) = 0\). Since the noise-free system is stable and the additive noise is bounded, the noisy system is also stable.

While the control law given is designed for a double integrator plant, the concept is easily generalized: use an oscillating input term to drive the heading rate to oscillate, and switch on the proportional feedback term when the heading rate is above some threshold level. The latter condition resolves concerns about the existence of a well defined steady state distribution (stability) under the assumption that heading rate is measured accurately, by preventing feedback of a measurement corrupted by noise of arbitrary variance. Thus, to prove stability of the system in the presence of DVS measurement noise, only stability of the noise-free system must be proven.

For the double integrator, we see that when \( b_p = 0 \), the system is stable in rate \( \dot{q} \) in the sense that the rate will converge to a finite set due to the bounded disturbance (dithering) applied to the control input. Because the control term in this case in no way depends on the heading itself, the heading retains marginal stability in this case. Similarly, when \( b_p = 1 \), the system is stable in both rate and heading. Therefore as long as the system cannot maintain a steady state condition where \( b_p = 0 \) for all time, it will be stable. This requirement is enforced by selection of the switching threshold - we must ensure that the steady state oscillations caused by the dithering are sufficient to trigger periodic activation of the proportional control term. Based on the frequency response for a single integrator plant, we can use \( \omega_d^{-1} \) for this threshold.
Figure 3.38: Simulated system trajectories using the dithering feedback control law with a double integrator plant and DVS sensor model, for different values of the dithering frequency \( \omega_d \).

Figure 3.38 shows simulated closed-loop trajectories resulting from using this type of dithering control law with a double integrator system and our DVS measurement model, for a selection of different dithering frequencies. This result shows that for low frequency dithering, the proportional control switching threshold is set relatively high and the resulting process noise is low, but the error caused by the dithering is high. Conversely, as the dithering frequency becomes high, the error is attributable primarily to process noise caused by allowing noisier heading measurements at lower heading rates to be used.

This tradeoff between the reduction in noise caused by dithering and the error introduced by the dithering is depicted in figure 3.39, which shows the mean squared steady-state tracking
error computed by simulation for control of a double integrator using dithering controllers with varying frequencies. In this case the optimal dithering frequency is approximately 6.5. However at this point the mean squared tracking error is still approximately 4 times higher than that achieved by the ZOH switching controller of section 3.7.4.

![Optimal Dithering for Double Integrator](image)

Figure 3.39: Mean squared tracking error from simulated control of a double integrator plant with DVS sensor model, using the dithering control law, as a function of dithering frequency. Indicates that for the parameters used the optimal dithering frequency is approximately 6.5.

The ZOH switching controller from section 3.7.4 is compared to the dithering controller in terms of steady state performance in figure 3.40, which depicts the steady state error distribution for the two controllers when used on a double integrator plant. Both methods vastly outperform PD control, for which the steady state mean squared tracking error is undefined.
Figure 3.40: Comparison of steady state tracking errors for the switching control law described in section 3.7.4 versus dithering control law with optimal dithering frequency, for a double integrator with the continuous-time DVS sensor model. Shows the distribution of long-time tracking errors computed by simulation for the two different control laws. While the ZOH switching controller performs noticeably better, both vastly outperform standard PD.

3.7.8 Summary

We have presented dithering as an alternative to zero order hold switching for control of a simple dynamical system using our neuromorphic vision sensor model. Both methods account for the fact that the heading measurement noise variance diverges as the heading rate approaches zero by switching off proportional control terms when the heading rate is below some threshold. In the former case the control is held constant at its value when the threshold was crossed, while in the latter case the control continues to vary according to an oscillatory dithering signal that ensures the heading rate will again reach an acceptable level for making measurements.

Arguments have been provided in support of the claim that both of these methods are stable (for the double integrator) in the sense that they lead to well defined steady state distributions for the heading and heading rate, with expectation equal to zero. In that sense they are found to be strictly superior to all linear control laws (PD control), because such laws in the presence of our measurement model lead to divergence of the heading distribution. Simulation and experimental results also support the conclusion that these methods provide better performance than simple PD control.

The results of this section are interesting also in their pertinence to biological microsac-
cades. Analysis of the measurement model derived from low-level principles of operation of the neuromorphic vision sensor shows that in closed-loop the lack of observability when the heading rate is near zero necessitates steady state motions that may be interpreted as microsaccades. The process of adding motion to improve performance was made explicit in the dithering case, but both of our proposed controllers lead to motions in the steady state as a direct result of the measurement model. From this we might suggest that in nature microsaccades are actually required to fixate gaze on an object due to the way retinal neurons work.

3.8 Applicability of Conventional Algorithms

Most algorithms designed for use with conventional vision sensors (cameras) are not directly applicable given the asynchronous frame-free nature of the output from a DVS. While it may be possible to accumulate events into frames in order to approximate a frame based sensor, such a procedure does not take advantage of the remarkable properties of the DVS such as very high-speed and low latency. In this section we will discuss a general methodology for adapting existing algorithms for use with DVS in a way that does not negate the benefits of event-based sensing.

3.8.1 Framing Events

The most obvious way to apply existing vision algorithms to DVS event streams is to implement a procedure that accumulates events over short periods of time to produce frames like the one shown in figure 1.2. There is no reason for the frames to be regularly periodic, and in fact capturing a fixed number of events rather than accumulating for a fixed duration retains some of the advantages of the sensor, specifically its ability to adapt the data rate to meet the demands of the problem. In such an implementation, the sensor operates somewhat like a camera with edge detection, however because it can only perceive moving parts of the scene, the edges observed depend on the direction of motion. Because our objective is to fully leverage the properties of the sensor, we will not focus on framing methods but rather on methods in which each event is used as it comes in.

3.8.2 Incrementalized Algorithms

Many conventional vision algorithms execute steps on a pixel-by-pixel or region-by-region basis that depend only on local brightness information and not on information from distant parts of the image. Feature detection, optical flow, calculating texture transforms, and matching in stereo vision are just a few examples of tasks for which cutting edge algorithms use information only locally. This ubiquitous property of vision algorithms suggests that many of them may be readily incrementalized, or converted into a form capable of handling events such as those produced by DVS as they arrive.

In this context, the general idea is that each DVS event can be thought of as defining a new frame differing from the previous one by a known amount at only a single pixel. This is not strictly true, since the threshold-based event generation process means that brightness
fluctuations less than the threshold cannot be detected as they would be by a camera with infinite color (brightness) depth. However, for a camera where brightness measurements are of finite precision (i.e. any digital camera) it is more accurate, and brightness fluctuations less than the threshold may be insignificant anyways. Given that each event defines a new frame, a frame based algorithm can be used. However most such algorithms would be vastly too computationally intensive to execute for every event, given that existing DVS can produce up to 1M events per second, unless they could be effectively incrementalized. Dependence of the algorithm only on local information should allow an incremental form to be derived straightforwardly from the original form. For example feature detection algorithms would only have to re-check for features including the location of the event, as checking any other region would be redundant. Incrementalization would also require a DVS that provided actual brightness data along with the events, and in fact the most recent models do have this capability.

In chapter 4, the idea of incrementalization is revisited and applied to the problem of determining whether DVS or cameras are superior for a given task at fixed mean computational allocation.

3.9 Discussion

In this chapter we have detailed the experimental apparatus used for testing, and described the system identification process used to determine its dynamics. A simple PD controller for the system was also determined using LQR techniques applied to the linear dynamical model from the identification. (Sections 3.1 and 3.2)

Additionally, we have presented algorithms extending intuitive results from a simple single-edge brightness profile (environment) to more complex brightness profiles, particularly those that are well modelled as piecewise constant. The algorithms described are efficient enough to be implemented for event-by-event configuration estimation with the capabilities of current computer technology, and provide a transformed sensor output with reduced lag that serves well as a configuration measurement.(Sections 3.3 and 3.4)

We have analyzed our maximum likelihood procedure for producing event-by-event heading measurements and shown that given a stochastic lag model for the system based on the persistent excitation of pixels in the event generation model it is possible to reduce sensor lag using the type of filters we describe. (Section 3.5)

Experimental results have been presented to support the claim that the key elements of the behaviour of the DVS are well captured by this model in spite of the approximations and assumptions made, and that the heading estimation and filtering algorithms perform as expected in open-loop. (Section 3.6)

Finally, the models developed in section 2.5 were used to show that conventional PD control is theoretically ineffective for heading control with DVS (corroborated by empirical tests), in the sense that it leads to a lack of convergence of the state distribution. Alternative control laws that lead to a stable steady state distribution were presented and shown to give superior performance in simulation and experiment. (Section 3.7)

Together these results provide theoretical and empirical support for the claim that DVS
can be used for heading control with event-by-event algorithms that do not compromise the low latency of the sensor in a variety of environments. Furthermore, specific methods for achieving performance that is empirically comparable to that achieved by low-level potentiometer control have been presented for our one-dimensional sensor case.
Chapter 4

Optimal Sensor Choice: DVS or Camera?

This chapter is dedicated to the exploration of the optimal sensor selection problem. We place ourselves in the position of designing a robotic system for a generalized task, and consider whether a DVS or camera should provide superior performance. We also ask: if we can choose between cameras of different resolution and frame rate, and DVS of different resolution and event threshold, what is the optimal sensor from each class. Given the optimal sensor of each type, we can compare performance across classes. Because it is expected that performance would strictly increase with increasing resolution, frame rate and event sensitivity, we will introduce computational constraints and optimize the sensor with respect to parameters and class for a given mean computational allocation.

Some of the ideas presented in this chapter are also described in [132], which gives a slightly different formulation but is consistent with the results shown here.

4.1 Linearized Sensor Model: Camera

In chapter 2 we introduced a high level DVS model that considered each event as a delayed measurement of the configuration of the system $q$, with additive noise whose variance was dependent on $q$ and $\dot{q}$. Now we will discuss a comparable model for the operation of a conventional camera, which will be used to compare the performance of the two sensor classes.

Following from section 2.1, and equation 2.2, a camera will be approximated as a sensor that makes periodic, instantaneous measurements of the brightness across all pixels, such that the $k^{\text{th}}$ frame is,

$$y_k(s) = m(q(t_k - \tau) + s),$$

(4.1)

where again $s$ indicates the pixel and ranges across integers from 0 to the resolution $n$ in one dimension. This model neglects motion blur but does account for communication lag via the parameter $\tau$, and assumes effectively zero fill factor ($\nu \to 0$ in equation 2.2. The measurements will be assumed perfectly periodic, so that
If we now consider the simple single-edge brightness profile, it is clear that the configuration $q$ cannot be measured to greater accuracy than a pixel given a single frame. This results from the zero fill factor and perfect focus condition. To see this, consider the task of finding a configuration for which the observed brightness matches the predicted brightness exactly at all pixels. If the brightness profile is

$$m(\xi) = \begin{cases} 0 & \text{if } \xi < 0 \\ 1 & \text{if } \xi \geq 0, \end{cases}$$

then a perfect match is obtained if

$$y(s) = \begin{cases} 0 & \text{if } s < 0 \\ 1 & \text{if } s \geq 0. \end{cases}$$

Since $s$ is integer, this happens as long as $q \in (0, 1)$, meaning that the uncertainty in the heading angle determined in this way is the angle corresponding to a single pixel in the field of view (again under the orthographic projection model all pixels subtend the same angle). While additional edges or smooth brightness gradients may allow for measurements with sub-pixel accuracy, we will continue under the assumption that the uncertainty in position measurements obtained from frames will be on the order of a single pixel.

There are many methods by which the most likely configuration for a given frame can be obtained, all of which can be broadly described as matching the captured image to the most similar region in the known brightness profile. In subsequent sections we will consider the implications for optimality of the computation used by various methods, but for now the specific method is not important, as we will assume that the heading angle can be determined to the nearest pixel given any frame captured from the camera. While self-similar environments may pose a problem to this assumption, generally prior information about the heading will be available for disambiguation. Thus, we have a canonical discrete-time sensor that makes periodic, noisy, delayed measurements of the configuration,

$$z_k = q(t_k - \tau) + v_k, \quad v_k \sim \mathcal{N}(0, \frac{L}{n}).$$

Recall that $L$ is the angular field of view, and $n$ is the number of pixels in the sensor so the measurement noise has standard deviation proportional to the angle subtended by a single pixel. The noise will be modelled as Gaussian and white for simplicity. Table 4.1 lists the parameters of this model.

### 4.2 Optimal Sensor Selection Problem

We now present a method, based on work in [133] and [134], for selecting the optimal sensor for the problem of estimating heading. The optimal sensor choice problem studied here is one of selecting the optimal choice of parameters, in this case resolution, and frame rate or event
threshold, for the sensor, and as such is similar to work done in [135] and [136] which study the optimization of these parameters with respect to a human defined value function based on perceived video quality. Optimal camera parameter selection has also been considered in the context of robotics, for example in [137] or [138] where the computation-optimal frame rate for the task of tracking a moving object is computed, or [139] where the performance-optimal focal length for object tracking is studied. Optimal selection of other parameters including exposure time and gain [140] and image blur parameters [141] has been used in robotics as well.

To make the comparison between DVS and cameras as meaningful as possible we would like to compare the optimal sensor of each type, where optimal here means over resolution \( n \) and frame period \( \Delta \) for the camera, and resolution \( n \) and event threshold \( h \) for the DVS. We would like to understand what the optimal choice of sensor parameters are, and hence the optimal type of sensor, for a given mean computational allocation. We will assume that the mean computation performed for each camera frame or DVS event is the same for the entire duration of operation.

Let the mean computation per frame for a camera with \( n \) pixels be \( f_c(n) \) and the mean computation per event for a DVS with \( n \) pixels be \( f_d(n) \). The mean computational load for the camera system then becomes,

\[
W_c = \frac{f_c(n)}{\Delta}, \quad (4.6)
\]

while the load for the DVS depends on the mean event rate, which according to our model is inversely proportional to \( h \),

\[
W_d \approx \frac{c_0 f_d(n)}{h}, \quad (4.7)
\]

where \( c_0 \) is a constant factor in the mean event rate. Constraining the sensors to operate at the same mean computational load leads to the condition,

\[
W_c = W_d = W = \frac{f_c(n)}{\Delta} = \frac{c_0 f_d(n)}{h}, \quad (4.8)
\]

and the optimal sensor of each class may be determined as that which minimizes some cost function \( J \). For the class of cameras,

\[
J^*_c = \min_{n,\Delta} J(n, \Delta)
\]

\[\text{s.t.} \quad \Delta = \frac{f_c(n)}{W}. \quad (4.9)\]
For the class of DVS,

\[ J_d^* = \min_{n,h} J(n, h) \]

s.t. \( h = \frac{c_0 n f_d(n)}{W} \). \hspace{1cm} (4.10)

This problem allows for the comparison of the optimal camera for the task with the optimal DVS, given the computational cost functions \( f_c(n) \) and \( f_d(n) \). The cost functions are as yet unspecified, but will represent trajectory tracking error in an open-loop estimation system or closed-loop control system.

### 4.3 Open-Loop Sensor Comparison

Now a problem similar to that considered in section 2.3 is studied. Section 2.3 deals with the problem of heading regulation, while now the objective is accurate estimation of the heading without feedback control. Recall the observation model from section 2.3, which simplifies the problem as translational with an orthographic projection. The position of the sensor (heading), is \( q(t) \), while the position of each pixel relative to the sensor is \( s_i \). The known position dependent brightness profile is \( m(\cdot) \) so that the brightness observed by each pixel is \( y_i(t) = m(x + s_i) \), and the field of view is \( L \).

The two types of sensors previously defined (cameras, DVS) are compared, where the objective is to estimate the configuration \( q \) based solely on the measurements from the sensor, where the true position \( q(t) \) follows a completely unknown trajectory that is constrained only to be continuous. The performance metric will be the mean squared estimation error over a finite time horizon,

\[ J = \int_0^1 (q(t) - \hat{q}(t))^2 dt. \] \hspace{1cm} (4.11)

#### 4.3.1 Comparison at Fixed Computation

In order to estimate the position with a conventional camera one might solve the following optimization problem whenever a new frame is received,

\[ \min_q \left( H(q) = \sum_{i=1}^n (y_i - m(q + s_i))^2 \right), \] \hspace{1cm} (4.12)

and update the estimated angle, \( \hat{q} \), to the argument of minimum, which amounts to finding the angle at which the observed scene brightness profile most closely matches the predicted brightness profile.

For the camera, this problem is solved each time a new frame is received, and will be assumed to require \( n^{1+\beta} \) operations, where \( \beta \in [0,2] \), and that this operation will find the optimal position to the nearest pixel. For \( \beta = 2 \), this is consistent with checking \( n \) distinct positions and picking the best.
Assuming the initial position of the sensor is known, the DVS can be used to solve this brightness matchup problem incrementally. If the value function before an event is received is,

$$H(q) = \sum_{i=1, i \neq j}^{n} (y(s_i) - m(q + s_i))^2 + (y(s_j) - m(q + s_j))^2,$$

then the value function subsequent to an event at pixel $j$ is,

$$H'(q) = \sum_{i=1, i \neq j}^{n} (y(s_i) - m(q + s_i))^2 + (y'(s_j) - m(q + s_j))^2$$

$$= H(q) + (y(s_j) - m(q + s_j))^2$$

$$= H(q) + h^2 (y(s_j) - m(q + s_j)),$$

so the value function may be incrementally updated when an event is received, and it will be assumed to require $n$ operations to localize the angle to the nearest pixel. This is consistent with testing $n$ distinct angles and selecting the best. A superior computation bound can be achieved using the histogram methods described in section 3.4, wherein the maximum occupancy bin in an $n$-bin histogram can be recomputed in $O(\log n)$ time assuming the use of a balanced binary tree to store the histogram bins.

Based on the assumed number of operations required to update the estimate given new data, which neglect constant factors that differ between the camera and the DVS, the mean computational bandwidth required by the camera and the dynamic vision sensor are given by $W_c$ and $W_d$ respectively,

$$W_c = \frac{n^{1+\frac{g}{2}}}{\Delta}$$

$$W_d = |g(t)| \frac{n^2 g}{h},$$

where $g$ is the pixel-wise mean absolute spatial gradient of brightness, assumed to be approximately constant across all possible views. The DVS computational load $W_d$ is the product of the noise-free event rate (see section 2.4) and the mean computation per event, $n$.

$$g = \frac{1}{n} \sum_{i=1}^{n} |\nabla m(\xi_i)|.$$

We will also assume that $g$ is independent of $n$ and therefore represents a constant multiplier in the event rate not influenced by resolution.

We will now assume that the heading rate is sufficient that the DVS measurement noise is not significantly effected by rate and that for both sensors the measurement is modelled as a noisy update in which the posterior value of the estimate is a uniformly distributed random variable centered at the true value of the angle and with support over a one pixel wide interval, so that immediately following an update,
\[ \dot{q} = q + \varepsilon_0 \quad \varepsilon_0 \sim \text{Unif}(-\frac{L}{2n}, \frac{L}{2n}). \] 

(4.17)

This differs slightly from our previous model using a Gaussian error distribution, but the concept is essentially the same so a uniform distribution is used for simplicity.

Now it can be asked, given the camera model described, what are the optimal choice of sensor parameters \( n \) and \( \Delta \) for a given mean computational allocation? Similarly for the dynamic vision sensor, what is the optimal choice of \( n \) and \( h \)? Formally, the solution to the following optimization problems is sought, for the camera and DVS respectively,

\[
\begin{align*}
J_c^*(W) &= \min_{n, \Delta} \mathbb{E} \left[ \int_0^1 \frac{1}{M} \sum_{i=1}^M (q(t) - \hat{q}_c(t))^2 dt \right] \\
\text{s.t.} & \quad \frac{n^{1+\frac{g}{h}}}{\Delta} = W
\end{align*}
\]

(4.18)

\[
\begin{align*}
J_d^*(W) &= \min_{n, h} \mathbb{E} \left[ \int_0^1 \frac{1}{M} \sum_{i=1}^M (q(t) - \hat{q}_d(t))^2 dt \right] \\
\text{s.t.} & \quad \frac{n^2 g}{h} \frac{1}{M} \sum_{i=1}^M |\dot{q}_i(t)| = W,
\end{align*}
\]

where the set of functions \( \{q_i(t)\} \) represents \( M \) distinct test trajectories where the performance is measured as the average mean squared tracking error across the different trajectories. The quantities \( \hat{q}_c \) and \( \hat{q}_d \) represent the signal estimates computed using the camera and the DVS respectively. Note that the optimal solutions \( J_c^* \) and \( J_d^* \) depend on the computational allocation \( W \), and we are generally interested in analyzing the behaviour of the optimum across a wide range of available computation.

A set of linear test trajectories, where \( q_i(t) = a_i t \) with \( a_i > 0 \forall i \) is considered. Defining the mean slope and mean squared slope of the test trajectories,

\[
\begin{align*}
< a > &= \frac{1}{M} \sum_{i=1}^M M a_i \\
< a^2 > &= \frac{1}{M} \sum_{i=1}^M a_i^2.
\end{align*}
\]

(4.19)

**Theorem 4.3.1** (Optimal Sensor Performance). For a piecewise linear trajectory \( q(t) \) consisting of \( M \) linear segments of equal duration, assuming that the number of sensor pixels \( n \in \mathbb{R}^+ \), that the DVS is not subject to spurious event noise, and that the number of events produced by the DVS during each of the \( M \) tests equals the expected number of events, the optimal performance costs are,
Proof. For a given value of \( a_i = a \), the performance of each sensor can be derived in closed form using only some mild approximations. Assume that the first frame of the camera occurs exactly at \( t = 0 \) and that the frame rate is an integer \( (\frac{1}{\Delta} \in \mathbb{Z}) \) so that there is a frame exactly at \( t = 1 \). Then, for the camera, the cost function can be written,

\[
J_c = E \left[ \sum_{i=1}^{1/\Delta} \int_0^\Delta (e_{0,i} + at)^2 dt \right]
\]

\[
= E \left[ \sum_{i=1}^{1/\Delta} (e_{0,i}^2 + \frac{a^2 \Delta^3}{3} + a e_{0,i} \Delta^2) dt \right]
\]

\[
= E[e_0^2] \sum_{i=1}^{1/\Delta} \Delta + \frac{a^2}{3} \sum_{i=1}^{1/\Delta} \Delta^3 + a E[e_0] \sum_{i=1}^{1/\Delta} \Delta^2
\]

\[
= \frac{3L^2}{16n^2} + \frac{a^2 \Delta^2}{3}.
\]

For the full set of test functions \( \{q_i(t)\} \), this result becomes,

\[
J_c = \frac{3L^2}{16n^2} + \frac{<a^2> \Delta^2}{3}.
\]

(4.22)

Enforcing the constraint on bandwidth allocation, the problem is transformed into an unconstrained one dimensional optimization in \( n \),

\[
J_c = \frac{3L^2}{16n^2} + \frac{<a^2> n^{2+\beta}}{3W^2},
\]

(4.23)

The optimal solution can be found for \( n \in \mathbb{R} \) by setting the gradient of \( J_c \) to zero and solving for \( n_c \). This leads to,

\[
n_c^* = \left( \frac{9L^2 W^2}{8(2 + \beta)} <a^2> \right)^{1/(\beta+4)}
\]

(4.24)

\[
J_c^* = k_0 \left( \frac{L^2(\beta+2)}{W^4} <a^2> \right)^{\frac{1}{\beta+4}}
\]

\[
k_0 = \left( \frac{3}{16} \left( \frac{8(2 + \beta)}{9} \right)^{\frac{2}{\beta+4}} + \frac{1}{3} \left( \frac{9}{8(2 + \beta)} \right)^{\frac{2+\beta}{4+\beta}} \right).
\]

(4.25)
Note that we also make the approximation that \( n \) can take on arbitrary real values, though in reality is must be integer.

The solutions for the dynamic vision sensor can be similarly computed. While the exact event timings could be computed given a specific function \( m(\cdot) \), it is instead assumed that the events are generated via a Poisson process with parameter \( \lambda \), which gives the number of events generated across the entire field of view during the \( t \in (0, 1) \) test interval. Specifically, this leads to the event interarrival times being modelled as I.I.D. random variables with an exponential probability distribution defined by the same parameter \( \lambda \). This is consistent with the higher level modelling presented in previous sections, however in this case we neglect the effect of exogenous spurious events. The variables \( \{\Delta_i\} \) will now be used to denote the \( i^{\text{th}} \) event interarrival time.

The expectation in the cost function can be written,

\[
J_d = \mathbb{E} \left[ \sum_{i=1}^{N} \int_{t=0}^{\Delta_i} (e_{0,i} + at)^2 dt \right] \\
= \mathbb{E} \left[ \sum_{i=1}^{N} (e_{0,i}^2 \Delta_i + \frac{a^2 \Delta_i^3}{3} + a e_{0,i} \Delta_i^2) dt \right] \\
= \mathbb{E} [e_{0,i}^2] \sum_{i=1}^{N} E[\Delta] + \frac{a^2}{3} \sum_{i=1}^{N} E[\Delta^3] + a \mathbb{E} [e_0] \sum_{i=1}^{N} E[\Delta^2],
\]

with

\[
\Delta_i \sim \text{Exp}(\lambda) \quad \lambda = \frac{|a_i| g_n}{h}, 
\]

and \( N \) being the number of events occurring over the test interval, which is approximated as a constant \( N = \lambda \) due to the fact that treating \( N \) properly as a random variable with a Poisson distribution leads to a series without known closed form solution. Typically the rate of events will be very high and it is reasonable to assume that any effects resulting from overall variation in the number of events will be small next to the variation in their timing.

Using the formula

\[
\mathbb{E}[\Delta_i^2] = \frac{\xi^2}{\lambda^2},
\]

gives,

\[
J_d = \frac{3L^2}{16n^2} + \frac{2h^2}{n^2 g^2},
\]

which holds also for the full set of test trajectories \( \{q_i(t)\} \). Enforcing the constant bandwidth allocation condition,

\[
h = \frac{<a> g n^2}{W},
\]

gives,
\[ J_d = \frac{3L^2}{16n^2} + \frac{2n^2}{W^2}. \] (4.31)

Finding the optimum for \( n \in \mathbb{R} \),

\[ n_d^* = \left( \frac{3W^2L^2}{32 < a^2 >} \right)^{\frac{1}{4}}. \] (4.32)

\[ J_d^* = k_1 \frac{< a > L}{W} \] (4.33)

\[ k_1 = \frac{3}{2\sqrt{6}} + \frac{\sqrt{6}}{4}. \]

For a given bandwidth allocation \( W \), the optimal choice of parameters has been found for both classes of sensor, and the corresponding minimum mean squared estimation error is given in equations 4.25 and 4.33 for the camera and DVS respectively. Both solutions are the result of a relaxation of the condition that \( n \) be integer, and admit any value \( n > 0 \). The ratio of optimal costs is,

\[ \frac{J_d^*}{J_c^*} = \frac{k_1}{k_0} \left( \frac{< a >}{L^{\beta}W^{\beta}. < a^2 >^{\frac{1}{2}}} \right)^{\frac{1}{\beta+1}}. \] (4.34)

Note that,

\[ \beta > 0 \Rightarrow \lim_{W \to \infty} \frac{J_d^*}{J_c^*} = 0. \] (4.35)

This indicates that as long as the estimation algorithm used by the camera requires strictly more than \( \mathcal{O}(n) \) operations to perform an estimation update given a new frame, as the computational allocation approaches infinity, the mean squared estimation error achieved by a DVS will become an arbitrarily small fraction of that achieved by a camera, if both sensors have their parameters selected optimally for the task. The value of \( \beta \) depends upon the efficiency of the algorithm used by camera, and should range between 0 and 2, which corresponds to the computation required to update the estimate ranging between \( \mathcal{O}(n) \) and \( \mathcal{O}(n^2) \). For \( \beta = 0 \), the camera requires only \( \mathcal{O}(n) \) operations to update the estimate with a new frame and

\[ \beta = 0 \Rightarrow \frac{J_d^*}{J_c^*} = \sqrt{6} \frac{< a >}{< a^2 >^{1/2}}. \] (4.36)

In this case, the asymptotic behaviour is the same, and the performance ratio depends on the tasks considered. While it is not necessarily meaningful to analyze this case, because constant factors multiplying computation time have been neglected, it is seen that the relative performance of the DVS improves if the mean squared rotational rate is high compared to the squared mean rotational rate. This suggests that the DVS would be a better choice for estimation tasks in which the absolute rotational rate is small most of the time, but
occasionally very high for brief periods. To see this, consider the case where $a_i = 1 \; \forall i < M$ and $a_M = M$, then,

$$\left( \frac{J_d^*}{J_e^*} \right)^2 = 6 \frac{<a^2>}{<a^2>}$$

$$= 6 \frac{(2M-1)^2}{M(M^2 + M - 1)}$$

$$\approx 6 \frac{4M^2}{M^2 + M^3} \text{ for } M >> 1,$$

and,

$$\lim_{M \to \infty} \frac{J_d^*}{J_e^*} = 0. \quad (4.38)$$

This gives the optimal performance ratio where the estimation task is characterized by unity angular rate under nominal conditions, with a one in $M$ chance of experiencing an angular rate of $M$, and shows that as the chance $\frac{1}{M}$ goes to zero, with the corresponding rate $M$ going to infinity, the optimal mean squared estimation error for the DVS becomes an arbitrarily small fraction of that for the conventional camera. This is consistent with the intuition that the DVS is superior at adapting to large variations in rate.

However, if the task includes operating at a range of angular rates, such that $a_i = c_i$, then,

$$\left( \frac{J_d^*}{J_e^*} \right)^2 = 6 \frac{<a^2>}{<a^2>}$$

$$= 9 \frac{(M+1)}{(2M+1)} \geq \frac{9}{2},$$

and,

$$\lim_{M \to \infty} \frac{J_d^*}{J_e^*} = \sqrt{\frac{9}{2}}. \quad (4.40)$$

The fact that the performance ratio is a constant value indicates that neither sensor is preferred because constant runtime factors have been neglected. So, regardless of the number of distinctly sloped test trajectories, as long as the slopes are uniformly spaced neither sensor is preferred.

The value of $n$ used in the optimizations to this point was constrained neither by a minimum nor to integer values. In reality, it is unreasonable to expect that a sensor with very few pixels would be capable of providing the assumed localization accuracy, so a minimum $n \geq n_{\text{min}}$ should be imposed. In this case, the unconstrained value functions both have a single global optimum, so the optimal choice of $n$ for either sensor is simply $\text{max}\{n_{\text{min}}, n^*\}$. 

122
Figure 4.1: Optimal mean squared estimation error for the DVS and cameras with several different computational use parameter values ($\beta$), indicating the time complexity of computing a measurement from a frame. Two different types of operating conditions are shown: heading rate selected uniformly from some range (a) and heading rate nominally 1 with rare sharp spikes to 1000 (b). Since constant factors have been neglected the emphasis is on the difference between the two plots and the relative slopes of the different curves - the constant factors are the same for both.
The results from this section are depicted graphically in figures 4.1 and 4.2, which show the optimal cost and sensor resolution respectively as functions of computational allocation. Both of these plots represent the proportionality relation in equation 4.20, and since these relationships neglect constant factors the absolute magnitude of the values is not significant. However these figures do illustrate the computation-asymptotic behaviour and the change in performance as the operating conditions are varied.
As in the analysis, two different test conditions are shown, one wherein the heading rate to be tracked is selected uniformly at random from a prescribed range, and another wherein the heading rate is nominally small but occasionally sees a very large jump. The key result is the same: that the DVS is asymptotically (as computational allocation approaches infinity) better for a given level of computation as long as the camera requires more than $O(n)$ operations for each measurement, and may be better by a constant factor if the camera requires exactly $O(n)$, depending on the task. The figures also show that for a camera with a given localization algorithm defining the computational requirement $\beta$, the DVS may be worse for low computational allocation and better for high computational allocation. However, as can be seen from the optimal resolution plots (figure 4.2), the optimal number of pixels is often quite low, and it is likely that this model would break down for very low resolution. Nonetheless, as $\beta$ goes to 0, this transition from camera to DVS superiority can occur at arbitrarily high computational allocations and therefore arbitrarily high optimal resolutions.

### 4.3.2 Comparison at Fixed Data Rate

We can repeat the previous analysis using a constraint on data rate rather than computation, so that we are choosing the tracking-optimal sensor for a fixed sensor output bitrate. This is equivalent to assuming that the DVS performs a constant number of operations per event and the camera performs $O(n)$ operations per frame. Let $o_c$ be the number of bytes of data used to represent each pixel in each frame of the camera, and $o_d$ be the number of bytes used to represent each event for the DVS. For the camera, using the previous result for $\beta = 0$,

$$J^*_c = \frac{cL < a^2>^{\frac{1}{2}}}{2W}.$$  \hfill (4.41)

The DVS result can now be determined using the same methods as before, with,

$$W_d = \frac{o_d < a > g}{h},$$  \hfill (4.42)

$$J_d = \frac{3L^2}{16n^2} + \frac{2 < a >^2}{W^2},$$  \hfill (4.43)

and leads to,

$$n^*_d = \left( \frac{3W^2L^2}{16o_d < a >^2} \right)^{\frac{1}{3}},$$  \hfill (4.44)

$$J^*_d = \left( \frac{16}{3} \right)^{\frac{1}{3}} \left( \frac{L o^2_d < a >^2}{W^2} \right)^{\frac{2}{3}} + \frac{2o^2_d < a >^2}{W^2},$$  \hfill (4.45)

where $W$ now denotes the mean data rate.
Figure 4.3: Root mean squared estimation error provided by the optimal DVS and camera at fixed mean data rate, as a function of available communication bandwidth (data rate). Shows that the DVS will eventually become superior as the available bandwidth increases. Units indicate that constant factors have not been neglected as in the computation case.

This result is depicted graphically along with the result for the camera in figure 4.3, which shows that as expected we again observe an inversion where regardless of constant factors, there exists some data rate allocation above which the DVS provides superior performance to the camera. In fact, for data rate the performance ratio becomes,

\[
\frac{J_d^*}{J_c^*} = \begin{cases} 
2 \left( \frac{16}{3} \right)^{\frac{1}{3}} o_d^{\frac{2}{3}} & a > \frac{3}{2} \\
L_0^2 W_0^2 a_c < a^2 > \frac{1}{2} + 4o_d^2 < a^2 > \frac{1}{2},
\end{cases}
\]

In the high bandwidth limit \( W \to \infty \), this looks like,

\[
\frac{J_d^*}{J_c^*} \propto \frac{1}{W},
\]
indicating that the performance of the DVS again becomes arbitrarily better than that of the camera for a fixed mean data rate as the data rate goes to infinity.

Unlike in the computation-constrained case where the specific algorithms used lead to changes in constant factors which were therefore neglected, when dealing with data rate alone it is more reasonable to consider constant factors. To assess the validity of our results we will consider a camera outputs all frames as full rasters of brightness data with each pixel represented by 3 bytes \((o_c = 3)\), and a DVS whose events each require 6 bytes \((o_d = 6)\). We will also consider that we are trying to track a target moving at 45 degrees per second with a field of view of 90 degrees. The results in figure 4.3 use these values, showing that for these operating conditions the DVS begins to outperform the camera at around 22 bytes per second data rate allocation, and that by 100kB/s data rate, the DVS provides over 40 times less tracking error than the camera.
The results in figure 4.3 show that for very low mean data rate allocation (less than 22B/s) the camera provides better performance than the DVS. However considering figure 4.4, which shows the optimal sensor resolution for the same operating conditions and range of data rate allocations, we see that the optimal sensor resolution for both classes of sensor is less than 5 pixels at that data rate, and this represents a somewhat unrealistic situation. Depending on the structure of the environment (brightness profile), our assumption that it is possible for both sensors to localize the heading to the nearest pixel with each measurement may not be valid for such low resolution. So while the performance inversion is present, in most practical situations the DVS is superior for all realistic resolution options.

4.3.3 Discussion

It has been shown that for the heading estimation problem, the performance of the DVS (as captured by our models) is superior to that of the camera for certain tasks at fixed mean computational allocation. It is seen that the DVS is superior for two reasons: first, it is able to adapt its sampling rate to the rate of motion of the system while the camera must use a fixed rate at all times, even when it wastes computation. This benefit is essentially that seen by event-triggered controllers and is well known in the event based control literature. It is captured by the $\frac{\omega^2}{a^2}$ factor in the results for heading estimation. Secondly, the DVS receives frame data and performs computations incrementally, meaning that with the same amount of computation it can process more frames than the camera and the effective sampling rate is higher. This benefit increases with the computational allocation, and is captured by the $\frac{1}{W}$ factor in the results.

This analysis is in the same spirit as that in [84], except that computation is considered in
addition to data rate and the effect of sensor resolution is also captured. In [84], it is shown that using asynchronous measurements can lead to a constant factor improvement at fixed mean communication bandwidth, while our results focus on how the relative performance varies with computational allocation. Our results show that under certain conditions we expect an inversion of sensor class superiority as allocation increases, where the camera is initially better than the DVS but becomes worse as resource availability improves. This stems partially from constant factors in computation, as the camera is initially better only because for lower allocation the constant factor may dominate, where the DVS is better asymptotically. In the case where we consider computation, because constant factors depending on the details of the environment (brightness profile) and algorithms have been neglected, this inversion result may not actually apply in reality. In the case where we consider only data rate, constant factors have not been neglected and the inversion is observed, however the optimal sensor resolutions are very low at low data rate and call into question our assumption that localization is possible.

It is interesting to note that in this analysis the performance of the optimal sensor did not depend on the structure of the environment, so long as the mean brightness gradient across the field of view could be well approximated as a constant \(g\) and did not vary with the task. If this approximation holds, we see that the event threshold is optimally selected in such a way as to cancel the effect of \(g\) on the tracking error, and neither optimal performance nor the optimal choice of sensor resolution is influenced by the environment. In many environments, \(g\) is not constant, in which case the relative performance of the camera will improve because we cannot ensure that the event threshold for the DVS is always optimal.

The fact that the sensor FOV \(L\) appears in the final expressions for optimal parameter choice and performance is interesting, as one might expect the end result to be independent of this parameter. However this is really a consequence of the fact that the target heading rates \(\{a_i\}\) are in absolute terms, and in all cases if they are expressed in terms of "fields of view per second" rather than degrees per second for example the dependence on \(L\) disappears and the symmetry we would expect to observe is present.

These results are all based on models for the sensors involving a degree of approximation. We have assumed that it is possible to localize the sensor when a new DVS event or camera frame is received, and provided support for this assumption in the form of a matching method that can be incrementalized to work efficiently with the DVS. This method is somewhat similar to the maximum likelihood algorithm from section 3.4, which has been demonstrated effective under some practical operating conditions. These results are also based on the assumption that the camera must output uncompressed image data, which does not account for the possibility of low-level compression or feature matching as could be implemented with a field programmable gate array (FPGA) or application specific integrated circuit (ASIC) at substantial power and time savings compared to using a general purpose computer. However we have provided results for a range of computational time complexities, some of which \((\beta = 0)\) correspond to very efficient camera performance.
4.4 Closed-Loop Heading Control

4.4.1 Angle Regulation and Target Tracking

In this section we present closed-loop experimental results demonstrating the performance of the DVS128 in the angular tracking problem, and compare the closed-loop performance parameters, as well as some closed-loop trajectories, with those obtained using a conventional camera. The purpose of this comparison is not necessarily to show that the trajectories are superior for the neuromorphic sensor in this particular case, but to demonstrate that it can effectively close the loop at a much higher rate, lower latency, and using much less computation than a camera.

For comparison, the sensor inside the rotating drum apparatus described in section 3.1 was swapped out with a conventional camera of a similar overall size and using a similar lens. The camera used was a Matrix Vision mvBlueFox, which also communicates via USB, and can capture frames of 752x480 pixels at a rate of 60 frames per second. Camera tests were conducted with a different, much simpler backdrop consisting of a single sharp edge delineating a light region from a dark region, and used an extremely simple algorithm whose main step is the calculation of horizontal adjacent pixel brightness differences. Nonetheless, in spite of the much simpler problem of tracking a single-edge, the camera used significantly more computation, even accounting for the difference in sensor resolution.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>P Control Gain (DVS)</td>
<td>$k_1$</td>
<td>1.3 V/deg</td>
</tr>
<tr>
<td>D Control Gain (DVS)</td>
<td>$k_2$</td>
<td>0.013 V/(deg/s)</td>
</tr>
<tr>
<td>P Control Gain (Camera)</td>
<td>$k_1$</td>
<td>0.6 V/deg</td>
</tr>
<tr>
<td>D Control Gain (DVS)</td>
<td>$k_2$</td>
<td>0.01 V/(deg/s)</td>
</tr>
<tr>
<td>DVS Switching Threshold</td>
<td>$c$</td>
<td>10 deg/s</td>
</tr>
<tr>
<td>Camera Frame Rate</td>
<td>$\Delta^{-1}$</td>
<td>60 Hz</td>
</tr>
</tbody>
</table>

Table 4.2: List of physical parameters used in experimental tests.

4.4.2 Closed-Loop Performance

Figure 4.5 shows the closed-loop heading trajectory along with the sinusoidal reference trajectory for the DVS and camera. Because the camera is only capable of making measurements at 60Hz, and with the latency required to send an entire frame over USB, the PD controller was made less aggressive by reducing the overall gain in order to maintain stability. As a result, the tracking performance of the camera is noticeably worse than that of the DVS in this case.
Figure 4.5: Closed-loop trajectories with a sinusoidally varying reference heading, comparing the performance of the camera with that of the DVS. Shows that it is possible to achieve good tracking control using the DVS.

### 4.4.3 Performance Parameters

Looking at closed-loop trajectories for the system affords only a limited ability to compare the performance of the different sensors because the optimality of the implementation is uncertain and the possibility that improvements may be made remains. Additionally, such results are only indicative of performance in the specific tests conducted, and may not be representative of relative performance across a wider range of problems. Nonetheless, the characteristics of the controllers presented in this section make a strong argument for the superiority of neuromorphic sensors for the heading tracking problem studied. Table 4.3 shows the key performance parameters of the different sensors used in implementation, as well as some
Table 4.3: Closed-loop performance metrics for the sensors used in the experimental comparison, as well as hypothetical parameters for an embedded neuromorphic sensor and a high-speed camera of the same resolution. All computation times are for a single core of an Intel core i5 running at 3.1GHz.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Frequency (Hz)</th>
<th>Lag (µs)</th>
<th>CPU use (ms/s)</th>
<th>Data (MB/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potentiometer</td>
<td>5000</td>
<td>100</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DVS128 (USB)</td>
<td>1200</td>
<td>700</td>
<td>24</td>
<td>1.2</td>
</tr>
<tr>
<td>Camera (mvBlueFox)</td>
<td>60</td>
<td>2000</td>
<td>95</td>
<td>21.7</td>
</tr>
<tr>
<td>DVS128 (embedded)</td>
<td>280000</td>
<td>15</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td>Fast Camera (128x128)</td>
<td>1200</td>
<td>1000</td>
<td>86</td>
<td>19.7</td>
</tr>
</tbody>
</table>

These results show that even a high-speed, low resolution camera operating at the same resolution and frequency at which the DVS effectively operates would require approximately 3.6 times as much computation for an environment with only a single-edge than the DVS would for the much more complex cartoon forest environment. Furthermore, the data rate from the sensor would be approximately 16 times higher, resulting in greater communication delay and most likely increased power consumption. Using such a simple environment for the camera means that the computational load is essentially a lower bound, and it is reasonable to expect an additional factor of 3-4 increase in computation required for the camera to deal with an environment such as the cartoon forest. The frequency values for the DVS128 represent mean update frequency across an entire sinusoidal reference tracking trial, but the true updates are aperiodic and the rate adapts automatically to the heading error rate as in event based control, leading to further computational savings if the system is subject to varying operating conditions.

Also shown in table 4.3 is the hypothetical performance of a DVS128 sensor embedded tightly in a low level control loop that recomputes the control input for every new event, as proposed in section 3.4.1. The measurement latency of 15 µs is taken from [142], and the value of 280kHz update frequency is the observed mean total event rate for the sensor. Given these values, we conclude that the DVS128 is capable of providing sensing for actuated tracking of extremely fast systems using very limited computational resources.

### 4.5 Closed-Loop Sensor Comparison

We have presented a theoretical comparison of DVS and cameras for the task of heading estimation, and now we would like to return to our original task of heading regulation and tracking to compare the two classes of sensors in a closed-loop setting. While it has been demonstrated that the DVS can be actuated and used for heading control, providing similar
performance to an embedded potentiometer feedback loop in experimental tests, we would like to explore the relationships between sensor parameters and performance in a manner similar to section 4.3.

### 4.5.1 Problem Formulation

Similarly to the open-loop case, we would like to compare the performance of the optimal sensor from each class when mean computational load is fixed. For the camera this means solving the minimization,

\[
J_c^* = \min_{n,\Delta,u(t)} \mathbb{E} \left[ \int_0^\infty q(t)^2 + r u(t)^2 dt \right] \quad \text{s.t.} \quad W = \frac{f_c(n)}{\Delta},
\]

given the continuous-time dynamics describing the evolution of \(q(t)\) under the action of the control input \(u(t)\), and the discrete-time measurement process corresponding to the capture of camera frames and localization to the nearest pixel,

\[
z_k = q(t_k) + v_k, \quad v_k \sim \mathcal{N}(0, \sigma_v = \frac{L}{n}), \quad t_{k+1} - t_k = \Delta.
\]

This problem is solved using the certainty equivalence principle which indicates that it can be separated into an estimation problem and a control problem without sacrificing optimality. In other words the optimal solution is to use a Kalman filter for estimating the state of the dynamical system given the observations \(z_k\), then compute the control input as a linear function of the state estimate with gains determined using a linear-quadratic regulator (LQR) formulation.

For the DVS, the optimization problem is similar,

\[
J_d^* = \min_{n,\Delta,u(t)} \mathbb{E} \left[ \int_0^\infty q(t)^2 + r u(t)^2 dt \right] \quad \text{s.t.} \quad W = f_d(n)R,
\]

where \(R\) is the asymptotic mean event rate. Recalling the continuous-time measurement model presented in chapter 2, neglecting delay,

\[
z(t) = \begin{pmatrix}
q(t) + v(t) \\
q(t - \tau) + w(t)
\end{pmatrix},
\]

\[
\mathbb{E}[v(t)v(t - \tau)] = \delta(t - \tau)\Lambda^{-1}(\sigma_v^2 p + \frac{L^2}{12} (1 - p) + q^2 p (1 - p)),
\]

\[
\mathbb{E}[w(t)w(t - \tau)] = \delta(t - \tau)\Lambda^{-1}(\frac{|q| + \Lambda_0^2}{h}),
\]

\[
p = \frac{\alpha q g(q, n, \rho) |q|}{\alpha \tilde{L} g(q, n, \rho) |q| + \tilde{\lambda}_s},
\]

\[
\Lambda = \frac{n}{\tilde{h}} \left( \frac{\alpha g(q, n, \rho) |q| + \tilde{\lambda}_s}{\tilde{L}} \right), \quad \Lambda_0 = n \tilde{\lambda}_s.
\]
Because of the dependence of noise variance on the state of the system, the certainty equivalence principle no longer holds, and the combination of a Kalman filter and LQR is not necessarily optimal.

4.5.2 Experimental Results

The key closed-loop sensor class comparison and model verification results are shown in figure 4.6, which gives the performance, under the optimal choice of parameters, as a function of computational allocation for the two sensors. To generate these results, the vertical resolution of both sensors was kept fixed, while the horizontal resolution was decreased by integer factors. The camera frame rate was also varied by integer factors between 400Hz and 30Hz (i.e. 400Hz, 200Hz, 133Hz, 100Hz, 80Hz, 67Hz, ...), and the DVS threshold was artificially manipulated by rejecting events at random with a probability representing the threshold.

![Figure 4.6: A comparison of the closed-loop tracking performance (sinusoidal reference, 2Hz, 70 degrees amplitude), between the camera and DVS with optimal choice of parameters. The results predicted by the model are also shown. Shows that similarly to the open-loop case, the camera outperforms at low computation while the DVS becomes superior at high computational availability, though they both asymptote to similar performance.](image)

Sensor resolution can be artificially reduced in at least two ways: sub-sampling and area averaging, both of which are perfectly reasonable methods of achieving reduced resolution. Until now our discussion of fill factor has been relatively limited, but it becomes important when we consider how to simulate reduced sensor resolution. Reducing resolution by sub-
sampling causes the fill factor to decrease with the resolution, while using area averaging maintains a constant fill factor. However with area averaging the simulated larger pixels contain an array of disjoint light sensitive regions, so the fill geometry is more complex than can be described with a single value.

This distinction between de-resolving methods is important when it comes to performance. Low fill factor effectively gives very high precision and the measurement error looks much more like quantization noise than additive normal noise. For example consider the case in which you have only a single pixel with near zero fill factor and the single-edge backdrop. The pixel samples the brightness of the backdrop at a single location, and the brightness will be either high or low and can easily be used to determine which side of the edge the camera sees. For many physical systems including our experimental apparatus, knowing only the sign of the tracking error is sufficient to provide reasonable performance, so in this specific test case the camera performs very well at low resolution. In the actual experimental tests, a control law based on identifying the location of the edge using gradient information was implemented, which requires 3 pixels instead of 1 but is subject to the same reasoning.

As seen in figure 4.6, using area averaging as a means of artificially reducing the camera resolution results in performance much more in line with the model. This is because when area averaging is used, the fill factor remains constant and the location of the edge is much more difficult to ascertain. Again considering the single pixel case, the brightness observed would equal the mean brightness across the field of view, which should be proportional to the heading, but slight variations in lighting and difficulty in calibrating zero-heading brightness give rise to error, the nature of which is much more similar to additive noise than in the low fill factor case. With the area averaging method, the gradient-based edge localization used by the camera breaks down at very low resolution (below around 6 pixels). While it is definitely possible to do sub-pixel localization, the camera sensor model presented does not account for this so it is not useful for model validation.

It is interesting to observe that in this case the low fill factor, sub-sampling method yields superior performance to the high fill factor, area averaging method for the camera. While higher fill factor is typically associated with better performance in machine vision, in this example it is not the case. This phenomenon most likely extends to other piecewise constant brightness profiles in which it is beneficial to be able to easily determine which of the constant regions each pixel lies within.

4.6 Discussion

In this chapter we have presented a formal theoretical comparison between the two sensor classes, DVS and cameras, in both open- and closed-loop. For each class of sensor, the optimal choice of sensor parameters for a particular task or set of tasks was found, and the optimal sensors of each class were compared in terms of estimation or tracking performance subject to constraints on computational or data rate allocation. Under a reasonable model for the computation required per DVS event and per camera frame, the relative performance of the different sensors was compared for different tasks.

Our analysis supports several conclusions regarding the relative performance of the two
sensors. Firstly, we observe that the optimal choice of sensor depends on the task. For example as we have seen, the DVS is favoured if the estimation task requires tracking of a signal that nominally changes slowly but experiences infrequent short, rapid changes, while the camera may be favoured (depending on algorithmic efficiency), when the distribution of motion in the signal to be tracked is more uniform. This phenomenon highlights one of the great benefits of the DVS compared to cameras - adaptability. Conceptually DVS vary their computational usage to maintain roughly constant performance, while cameras do the converse, maintaining computational usage while allowing performance to vary. This idea is depicted graphically in figure 4.7.

Figure 4.7: Depiction of the fundamental difference between DVS and cameras. DVS can adapt their computational demands to maintain constant performance while cameras allow performance to vary but maintain constant computational load.

As an example of the behaviour depicted in figure 4.7, consider the task of open-loop heading tracking where task difficulty indicates the heading rate to be tracked, and computation is approximately proportional to data output rate. The data rate from the DVS is directly proportional to heading rate, as seen in a number of the models presented, while the camera outputs data at a fixed rate. Similarly, the performance cost, in this case the mean absolute tracking error, increases linearly with heading rate for the camera if a zero order hold is used for signal estimation, while the DVS increases the event rate to match the heading rate and maintains constant measurement error. This assumes noise-free operation, and the addition of noise actually causes the DVS to perform more poorly at low heading rates (see figure 3.26). While for this example figure 4.7 happens to be very accurate, this trend should also be observed to some extent in a variety of situations including complex feedback control tasks.

In addition to being better able to adapt computational demand to meet the needs of the task, DVS are also seen to be superior due to their asynchronous and incremental nature. The asynchronicity of measurements results in decreased latency and therefore superior performance with no increase in average computational usage. We have however not considered peak computational load, which would yield a comparison much more favourable to the camera. Given that our primary motivation is the reduction of power consumption and that our proposed methods are still implementable in practice on very small and lightweight computers, focusing on the mean computation is justified.

Secondly, though our discussion to this point has mostly neglected constant factors in the computation, we can return to our original DVS model as presented in section 2.5, to
understand how the relative performance of the different sensor classes might vary with environment. Intuitively, much of the advantage of the DVS is derived from the hardware-level compression of the video data stream that it provides, and therefore we might expect that environments less amenable to this compression are worse for the DVS. Under the models presented in this chapter however this is not the case. Returning to the expression for the overall DVS event rate $\Lambda$,

$$\Lambda = \frac{n}{h} \left( \frac{c_0}{L} g(q, n, \rho) \left| \hat{\dot{q}} \right| + \bar{\lambda} \right),$$

we see that the structure of the environment is captured in $g(q, n, \rho)$, which was assumed to be constant in the analysis of this chapter. Essentially, the effect of $g$ is cancelled out by the optimal choice of event threshold for the DVS and does not appear in the final cost function capturing DVS tracking error. This is also the case if $g$ is not assumed constant but instead depends on $n$. The assumption that $g$ is constant in $q$ is not entirely accurate, and may be detrimental to performance since the event threshold is held fixed during operation. Fixing the event threshold despite variations in $g$ with $q$ means that the sensor is not always operating with optimal parameters and the performance may suffer.

Perhaps the most interesting aspect of the results presented in this chapter is the potential for an inversion of optimal sensor choice as the available computation increases. This phenomenon is predicted by both the theoretical estimation results and the numerical closed-loop results, and is corroborated by the closed-loop experimental results (for fixed camera fill factor). In all cases the prediction and observation is that for low available computation, the camera is superior, and as computational allocation increases the DVS eventually becomes superior. This is somewhat surprising given that the DVS is meant to reduce computational load. However if we instead consider fixing performance and allowing computational usage to vary, the equivalent results would show that below a certain level of performance the camera is superior, while for high performance applications the DVS would require less computation, and this is exactly what was expected to begin with. Therefore the results of this chapter overall confirm the expectation that the DVS can provide high level performance, in open and closed-loop, with computational savings compared to a camera. Nonetheless, the camera is better when performance requirements are not as strict or alternatively when computational allocation is lower.

One explanation for the fact that the camera outperforms at low computational allocation is the nature of the noise in the DVS. When the sensor resolution is very low, the effect of spurious events can be much more detrimental, and in effect the DVS cannot successfully operate at the low resolutions that the camera can. Below 12 pixel resolution the DVS could never successfully stabilize the system, most likely due to the presence of spurious events and the difficulty in estimating heading rate at such low resolution. However this does not explain the theoretical prediction in the estimation-only case. Another explanation that also accounts for the prediction is the constant factor in the computation required per event. For low computational allocation, the constant factors dominate and give superiority to the camera, while high available computation allows asymptotic effects to become more significant, giving the DVS the advantage.
Chapter 5

High-Speed Forest Flight

The overarching goal of this work is to advance understanding of the use of dynamic vision sensors for control of robotic systems. Until now we have restricted ourselves to a limited class of rotational problems, and in this final chapter we return to the original goal of using the DVS for motion planning with a mobile robotic platform. In this case we are faced with the additional complexities of translational motion which preclude the use of most of the methods described so far. Specifically, pixel brightness is no longer so simply related to configuration as in the rotational case,

$$ y(s, t) = m(q + s). $$

(5.1)

Instead, brightness depends on the unknown three-dimensional locations of objects and the lens projection, and it is no longer possible to use the simple techniques previously described to infer configuration from event location or configuration rate from event rate. As such, completely different algorithms will be used and this chapter will deviate from the models and methodology presented in previous chapters.

This chapter describes algorithms, of varying computational complexity, that can utilize the unique properties of the DVS to perform event-by-event navigation for mobile robots in unknown, cluttered environments. While a detailed simulation has been developed to test the efficacy of these algorithms and compare their performance to planners based on camera or scanning laser rangefinder feedback, much of this work should still be considered preliminary and a start to future research.

5.1 Forest Flight Problem

Our specific problem of interest when it comes to motion planning is the forest flight problem, as discussed in the introduction. Previous sections have dealt with heading regulation for a sensor mounted on a moving vehicle, and this section will attempt to address the other aspect of the motion: translation. We consider a simplified, translation-only model of vehicle dynamics,
\[
\begin{align*}
\dot{x}_1 &= v_f \\
\dot{x}_2 &= u \\
|u| &\leq 1.
\end{align*}
\] (5.2)

Under these simplistic dynamics, the vehicle moves forward at a constant speed \(v_f\) and the control input \(u = \dot{x}_2\) is the lateral velocity, which is bounded in magnitude by 1. The two-dimensional position of the vehicle is denoted \(x = [x_1, x_2]^T\). There is no heading angle in the state of the system, and the sensor is assumed to point at all times in the same direction. As in previous sections, a one-dimensional sensor is considered, in this two-dimensional world.

The environment contains a number of obstacles (in our case trees) that must be avoided, and the objective is for the vehicle to survive as long as possible moving forward at constant speed without colliding with any trees. In order to do this the vehicle uses sensor information related to the relative positions of the trees, and we will be considering three types of sensors. First, a scanning laser rangefinder (LIDAR), capable of determining the distance to the nearest obstacle in a discrete set of directions. Second, a single conventional camera, which when velocity is known can use correspondences between subsequent frames to compute distances. Last the DVS will be considered, whose event output is in some way related to the 3D structure of the environment.

Each of the sensors will be assumed to point in the forward \((x_1)\) direction, and for comparison all sensors will be assumed to have the same field of view equal to the cone of possible paths that the vehicle can take. Since the lateral velocity is bounded by 1 but the forward velocity is a parameter of the problem, a reasonable design choice would be to use a narrower field of view for a faster vehicle, as the cone of feasible paths is narrower.

### 5.2 Algorithm Benchmarking

One interesting aspect of the forest flight problem as formulated is the fact that full knowledge of the forest has an empirically negligible effect on the efficacy of motion planning algorithms compared to only local range information. This is a consequence both of the simplicity of the vehicle dynamics and the structure of the obstacles in the forest, and can be seen in figure 5.1, which compares the performance of two omniscient planners and one with only local range information.
Figure 5.1: Comparison of the performance of different algorithms on an idealized forest flight problem. Shows that planning using only local range information can be nearly as effective as omniscient planners and more effective if computation is constrained.

The simulation results presented in figure 5.1 compare three different planners: RRT*, an optimal randomized tree planner using kinematic constraints to ensure path feasibility [16], a lattice based planner that steers in the direction connected to the most reachable leaf nodes, and the visible distance planner which at all times moves in the direction in which it can go.
the farthest without colliding. In figure 5.1, the planners are depicted on the left, with the corresponding performance plots on the right indicating mean survival time as a function of forward velocity and forest density.

For our purposes, the main message to take away from these results is that omniscience and complex planning are overrated in the forest, and a very simple memoryless planner using only instantaneous range information to determine the control input can perform well. This is good because in a real-world application such as an autonomous air vehicle designed for sub-canopy forest reconnaissance power and therefore computation is very limited, and because it gives us hope that a simple but fast algorithm based on DVS events may be effective.

5.3 DVS Algorithms

5.3.1 A Simplistic Planner

The first class of algorithms described is based on methods like that in [143], which gives a method for constructing a polar histogram of obstacle density indicating the desirability of each possible direction of travel at each moment in time. While this method originally assumes depth values are included in the measurements, and is designed for sensors with high angular uncertainty, the principle can be applied to the opposite case, where depth is unknown but angular resolution is high, and so can be effective for obstacle avoidance using dynamic vision sensors. Our method is motivated by the need to conserve computational resources and perform only a small number of operations for each event received while updating control signals on an event-by-event basis in order to utilize the low sensor latency. The general idea is to maintain an estimate of the “quality” of each of a discrete set of directions, and to update this estimate incrementally with each event received.

In a uniformly textured environment, the rate of DVS events at a particular pixel is proportional to the magnitude of optical flow at that pixel, which can be used to calculate depth assuming that the velocity of the vehicle is known. If we return conceptually to the piecewise constant brightness profiles considered in the rotational problems of previous chapters, events are thought of as being triggered by an image edge traversing a pixel, and one way of trying to generalize this reasoning to translational problems is to consider edge points. Each such point lies in the 2D plane in which our vehicle travels, and triggers a DVS event whenever it crosses the ray emanating from a given pixel. This is consistent with many existing camera based navigation algorithms that rely on the association of features to 3D locations for building a map of the environment.

At pixel $i$, the frequency of events, $f_i$, is proportional to the local density of edges in the direction of that pixel, $\gamma_i^{-1}$, according to,

$$f_i(t) = \gamma_i^{-1}(t)|v(t) \times c_i(t)| = \frac{|v_1(t)(s_i + \frac{\rho}{2}) - v_2(t)|}{\gamma_i(t)\sqrt{1 + (s_i + \frac{\rho}{2})^2}},$$  \hspace{1cm} (5.3)$$

where $v = [\dot{x}_1, \dot{x}_2]^T$ is the two-dimensional velocity of the sensor, $\rho$ is the pixel width, and
\[ c_i(t) = \begin{cases} [1, s_i + \frac{\gamma}{2}]^T & v_1(t)s_i - v_2(t) < 0 \\ [1, s_i - \frac{\gamma}{2}]^T & \text{else} \end{cases} \] (5.4)

is the direction along the leading edge of pixel \( i \) - the boundary such that for the motion of the sensor any point must pass through it before passing through the other. Expressing the event frequency as the inverse of the event interarrival period \( \tau_i(t) \), the inverse of the density can be expressed in terms of the period,

\[ \gamma_i(t) = \tau_i(t) \frac{|v_1(t)(s_i + \frac{\gamma}{2}) - v_2(t)|}{\sqrt{1 + (s_i + \frac{\gamma}{2})^2}} , \] (5.5)

showing that the density is inversely proportional to the mean time between events multiplied by the component of the sensor velocity in the direction of the leading edge of the pixel. This implies a basic incremental algorithm for estimating \( \gamma_i(t) \):

1. When a new event is observed at pixel \( i \), \( \gamma_i \leftarrow 0 \).
2. At all other times, let \( \gamma_i = |v \times c_i| \).

Because the goal is to use the estimated edge density in each direction for the purposes of motion planning with the objective of avoiding edge points, and the size of the vehicle is non-negligible, some form of "smoothing" must be applied to the edge density estimate in order to ensure the vehicle is guided to sufficiently wide corridors. Though it may not necessarily be considered smoothing, an operation that has produced good results in simulation is to compute the corrected inverse density estimate, \( \gamma'_i \) as the minimum of the surrounding raw density estimates,

\[ \gamma'_i(t) = \min_{j \in (-m, m)} \gamma_{i+j}(t). \] (5.6)

with the parameter \( m \) indicating the width of the region over which the minimum is computed, and must be chosen heuristically to obtain good performance.

Then, a simple motion controller that steers the vehicle at all times in the direction of maximum \( \gamma'_i \) is used:

\[ i^*(t) = \arg\min_i \gamma'_i(t) \]

\[ u(t) = u_{i^*(t)}. \] (5.7)

The value \( u_i \) is the control input required to steer the system in the direction of the \( i^\text{th} \) pixel. This assumes that the cone of view of the sensor matches exactly or lies within the cone of points reachable under the input constraint \(|u| < 1\), which is reasonable given that observing unreachable parts of the space is not useful.

One interesting thing to note about this algorithm is that it utilizes a continuous-time estimation procedure, which is unique to DVS and does not make sense for conventional cameras. This illustrates the fact that the DVS is truly a continuous-time sensor, as the absence of events carries information. Focusing on the spaces between events rather than just on the events themselves is a common theme among the algorithms of this chapter.
5.3.2 Minimum Risk Planners

Now we will consider a more complex method of computing optimal paths based on estimating collision probabilities along edges in a planning tree or trajectory library. This algorithm is loosely based on probabilistic reasoning in occupancy grid methods [144] [145], which represent obstacles by dividing the environment into a grid and estimating the probability that each cell contains an obstacle, and is broadly similar to the algorithms in [146] and [147] which maintain a probabilistic model of the environment which is used to compute collision probabilities.

Our approach differs from many of the existing methods in the literature that use conventional cameras in the sense that it does not rely on the association of points in the image (events) to edge points in two dimensions. It is much more related to methods that try to directly estimate a dense map, such as in the form of an occupancy grid, representing the probability that there is an obstacle at each cell in the grid. However, it is generally unnecessary and potentially inconvenient to estimate probabilities for all parts of the space, and the proposed method would only estimate the collision probabilities along a set of candidate trajectories.

One key idea used in the proposed method is that of estimation based on a lack of events rather than the events themselves. Using the events to estimate collision probabilities along path segments is problematic for multiple reasons. Firstly, to speak abstractly, an event is caused either by one point or another, while the absence of events is the result of no events being triggered by either point. Thus, estimating collision probabilities from events requires considering a disjunctive combination of possibilities, while estimating non-collision probabilities from an absence can be done using a conjunctive combination, which is much easier. The concept of using the absence of events for estimation in an event-triggered control system is not new, and has been proposed and studied in [148] and [149]. Secondly, when using the absence of events it is possible to forego updating estimates each time there is a new event, and instead update them either with a fixed period or each time a pixel traverses part of a possible trajectory. Figure 5.2 shows the set of points that traverse pixel i over some period of time.
Consider the case where edge points can only exist at a discrete set of two-dimensional locations $\mathcal{V} = \{v_i = (x_i, y_i) \in \mathbb{R}^2\}$, and the probability that each such point $v_i$ is an edge point must be estimated. This can be done using Bayesian reasoning each time a pixel is traversed by one of the points $v_i$. Let $M$ be the stochastic event that no DVS events are produced by a given pixel during the traversal of a point in the set $\mathcal{V}$, which has duration $\Delta t$, and the event that there is an edge at point $v_i$ be $A_i$. We wish to compute the Bayesian update for the estimated probability that there is an edge a point $v_i$, $Pr[A_i]$:

$$Pr[A_i|M] = \frac{Pr[M|A_i]Pr[A_i]}{Pr[M]}.$$  \hspace{1cm} (5.8)

Defining $M_i$ as the event that no sensor events are produced by point $v_i$ during the given time interval,
\[
Pr[M] = Pr[\text{no spurious events generated}] \prod_j (Pr[\text{no event generated by point j}])
\]

\[
= (1 - p_s(\Delta t)) \prod_j Pr[M_j]
\]

\[
Pr[M|A_i] = (1 - p_s(\Delta t))Pr[M_i|A_i] \prod_{j \neq i} Pr[M_j]
\]

\[
= (1 - p_s(\Delta t))p_m \prod_{j \neq i} Pr[M_j]
\]

\[
\Rightarrow Pr[A_i|M] = \frac{(1 - p_s(\Delta t))p_m \prod_{j \neq i} Pr[M_j]}{(1 - p_s(\Delta t)) \prod_j Pr[M_j]} Pr[A_i]
\]

\[
= \frac{Pr[M_i|A_i]Pr[A_i]}{Pr[M_i]}
\]

\[
= \frac{p_m Pr[A_i]}{(Pr[M_i|A_i]Pr[A_i] + Pr[M_i|\neg A_i]Pr[A_i])}
\]

\[
= \frac{p_m Pr[A_i]}{p_m Pr[A_i] + 1 - Pr[A_i]}. \tag{5.9}
\]

where \( p_s(\Delta t) : \mathbb{R}^+ \rightarrow (0, 1) \) is a function that gives the probability that at least one spurious event is generated in the time interval \( \Delta t \), and \( p_m \) is the "miss probability"—the probability that an edge point traverses a pixel without triggering an event.

This provides a simple rule for updating the estimated occupancy probability, \( Pr[A_i] \) of a point based on the observation that a pixel is traversed by that point without producing any events. If \( k \) is the number of such traversals of some point, the probability estimate for that point, \( p(k) : \mathbb{Z}_0^+ \rightarrow (0, 1) \) is,

\[
p(k + 1) = \frac{p_m p(k)}{p(k)(p_m - 1) + 1}. \tag{5.10}
\]

It follows that in the limit as \( k \to \infty \), the update rule becomes a multiplication by \( p_m \),

\[
p(k) \in (0, 1), \quad p_m \in (0, 1)
\]

\[
\Rightarrow p_m(1 - p(k)) \leq 1 - p(k)
\]

\[
\Rightarrow p_m \leq 1 - p(k) + p_m p(k)
\]

\[
\Rightarrow \frac{p(k + 1)}{p(k)} < 1 \quad \forall k \tag{5.11}
\]

\[
\Rightarrow \lim_{k \to \infty} p(k) = 0
\]

\[
\Rightarrow \lim_{k \to \infty} p(k + 1) = p_m p(k),
\]

confirming the intuition that it is decreasing and tends toward zero as more and more measurements are made. While the update rule in 5.10 allows us to maintain occupancy probabilities
on a grid, it may be more efficient to use an alternative representation of the environment, for example as collision probabilities along continuous trajectories for our vehicle.

In order to extend this reasoning to continuous space, where each edge point \( e_j \in \mathbb{R}^2 \), we see that the number of points in the set \( \{x_i\} \) in no way affects the result, and so the density of points may be increased indefinitely. The density of points would however influence the prior estimate of edge probabilities, \( p(0) \), which, assuming a prior estimate for the mean density of edge points in space \( \rho \), and that the number of points of \( \mathcal{V} \) inside some area \( A \) is \( |\mathcal{V}| = N \), is,

\[
p(0) = 1 - e^{\frac{\rho A}{N}}.
\]  

(5.12)

This is the prior probability, based on the model that edge points are distributed in continuous space according to a Poisson process, that there is one or more edge points in the neighbourhood of a given point. \( p(0) \to 0 \) as \( N \to \infty \), and in the limit the approximate update rule is exact,

\[
p(x_i, y_i) \to p^m(x_i, y_i)(1 - e^{\frac{\rho A}{N}}) \text{ as } N \to \infty,
\]  

(5.13)

and gives the probability, for some point \((x_i, y_i)\), that there is an edge point at that location, given the density of the point grid, the mean density of edge points, the probability of missing an event \( p_m \), and the observation that \( k \) pixels have been traversed by the point without producing events during their traversal.

This procedure for estimating edge probability has been tested using open-loop data from the actual dynamic vision sensor. The sensor was mounted on a linear slide and actuated in a direction orthogonal to its optical axis, while viewing a simple planar object with several sharp edges on it. Under the assumption that the motion of the sensor is known, estimates of the structure of the edges in space as the sensor were computed and are shown in figure 5.3. These results are quite promising, particularly as they demonstrate the ability to quickly identify free space near the sensor and that it is possible to effectively estimate the location of edges using this method. In this particular example the probabilities are not represented on a discrete grid of points, but by the set of triangular regions that each pixel sweeps through between events as in figure 5.2.
Figure 5.3: Estimation of edge point probabilities using data from iniLabs dynamic vision sensor on a linear slide. Framed DVS events are shown on the left, with the estimated edge probability (black = prior, white = 0) on the right, for four different points in time. Sensor motion is assumed to be known.

Now consider calculating the probability of collision along some trajectory, which, for a finite sized vehicle is the probability that there is one or more edge points inside the region swept out by the robot along its path. Since the update rule in equation 5.13 holds in the limit $N \to \infty$, it will be used to derive the continuum limit for the trajectory collision probability. Suppose the trajectory in question causes the vehicle to sweep out an area $a$, and that there is a discrete set of possible edge points $V = \{v_i\}$ inside that area, then the probability of colliding is one minus the probability that none of the traversed cells contain obstacles,

\[
Pr[\text{collide along trajectory}] = P(N) = 1 - \prod_{i=0}^{N} \left(1 - p_{m}^{k(x_i,y_i)}(1 - e^{-\frac{a}{N}})\right). \tag{5.14}
\]

In order to compute the continuum limit, for $N \to \infty$, consider the case where $k(x,y)$, the number of event free traversals of some point $(x,y) \in \mathbb{R}^2$, is constant $(k(x,y) = k)$ inside the area swept out by the trajectory. Then,
\[
P(N) = 1 - \prod_{i=0}^{N}(1 - p_m^k(1 - e^{-\frac{e^N}{m}})) = 1 - (1 - p_m^k(1 - e^{-\frac{e^N}{m}}))^N = 1 - P(N)
\]

\[
\ln(-P(N)) = N \ln(1 - p_m^k + p_m e^{-\frac{e^N}{m}})
\]

\[
\ln(\lim_{N \to \infty} -P(N)) = \lim_{N \to \infty} \ln(-P(N)) = \frac{1}{d} \ln(1 - p_m^k + p_m e^{-\rho a_d}).
\]

Using l'Hopital's rule to calculate the limit,

\[
\ln(-P) = \lim_{N \to \infty} \frac{d}{1}\ln(1 - p_m^k + p_m e^{-\rho a_d}) = -\rho a_p^k
\]

\[
P = \lim_{N \to \infty} P(N) = 1 - e^{-\rho a_p^k}.
\]

Now, in order to extend this to the case where \( k = k(x, y) \), consider the probability of colliding when passing through \( M \) such regions with areas \( a_i \) and \( k = k_i \). It is again one minus the probability of no collisions,

\[
Pr[\text{collide along trajectory}] = 1 - \prod_{i=0}^{M} e^{-\rho a_i^k i} = 1 - e^{-\rho \sum_{i=0}^{M} a_i p_i^k},
\]

which in the limit \( M \to \infty, a_i \sim \frac{1}{M} \), becomes,

\[
Pr[\text{collide along trajectory}] = P = 1 - e^{-\rho \int \int a_i^k(x,y) dx dy}.
\]

Equation 5.18 above gives the probability that there is an edge point inside the area over which the integral is defined, given the function \( k(x, y) : \mathbb{R}^2 \to \mathbb{Z} \), the number of pixels that have been traversed by the point \( (x, y) \) without producing any events during their traversal. Returning to equation 5.17, the change in non-collision probability following a unit increase in \( k \) across the whole area is given by the ratio,

\[
\frac{\rho P^+}{\rho P} = \frac{e^{-\rho \sum_{i=0}^{M} a_i p_i^{k+1}}}{e^{-\rho \sum_{i=0}^{M} a_i p_i^k}} = \prod_{i=0}^{M} \frac{e^{-\rho a_i^{k+1} p_i^m - p_i^m}}{e^{-\rho a_i^k p_i^m}}
\]

\[
= \prod_{i=0}^{M} e^{-\rho a_i^{k+1} p_i^m p_i^m - p_i^m} = e^{-(p_m - 1)\rho \sum_{i=0}^{M} a_i p_i^k}
\]

\[
= (e^{-\rho \sum_{i=0}^{M} a_i p_i^k})^{p_m - 1} = (-P)^{p_m - 1}.
\]

So, the incremental change in the probability that a given area contains at least one edge point, given the new observation that a pixel has been swept across the entire area without producing any events, is defined by the relationship,

\[
P^+ = 1 - (1 - P)^{p_m}.
\]

Equations 5.18 and 5.20 provide the basis for estimating collision probabilities along paths using event timing data from a dynamic vision sensor and known motion of the sensor, and
planning a path of minimum collision probability. The general form of \( k(x, y) \) in equation 5.18 does not require that it take on integer values, and in fact it may be represented for computational purposes by any set of local or non-local basis functions. One of the primary challenges to implementing a minimum risk planner is developing an efficient procedure for exactly or approximately evaluating the integral in equation 5.18, or incrementally updating probabilities using equation 5.20.

The general objective within this approach is to either maintain a representation of \( k(x, y) \) that can be used to efficiently evaluate equation 5.18 and is efficiently updated based on new data, or to provide an incremental computation that is efficient for updating the probabilities directly. The nature of using absence of events for estimation means that updates need not be made only when new events are observed, as the sensor continuously provides new information. All such methods can be applied within the framework of a number of optimal motion planning algorithms, including trajectory libraries, randomized tree planners, dynamic programming and others.

5.4 Simulation

In order to test our proposed DVS based navigation algorithms, a simple simulation was written using the free open-source Java graphics library JMonkeyEngine. Output from the 3D computer graphics engine was used to compare our DVS driven vehicles to those driven using simulated long range scanning laser rangefinders (LIDAR) and conventional monocular methods based on dense cross-correlation calculations. Screenshots from the simulation are shown in figures 5.4 and 5.5. While we model DVS events as having arbitrary temporal precision and being generated instantaneously when the brightness crosses the associated threshold, in simulation the condition for event generation (threshold crossing) is evaluated only at a discrete set of times - the simulation timestep. In order to effectively simulate the DVS, a small timestep was used, and the exact timing of each event determined by linear interpolation of brightness between steps.
Figure 5.4: Screenshot from the forest driving simulation, showing the on-board camera view in the top left and a depiction of the estimation and planning under LIDAR control on the bottom. The top right view is for qualitative assessment only.
Figure 5.5: Screenshot from the forest driving simulation, showing the on-board camera view in the top left and the associated DVS output in the bottom left. The directional density estimates and smoothed estimates are shown in the bottom right, along with the current preferred steering.

Five different control schemes were implemented in simulation:

1. No Control: The vehicle traveled in a straight line, minimizing its cross-section with the oncoming forest.

2. Visible Distance Planner with LIDAR: A simulated LIDAR was used to plan according to the maximum visible distance rule described in section 5.2. Delay between measurements at different angles was not included in the simulation model, and all measurements were made simultaneously.

3. Visible Distance Planner with Monocular Vision: The same planner as above was used with point cloud obstacle data generated by a dense patch matching method and known ego-motion.

4. DVS Directional Density Planner: A simulated DVS was used with the very efficient methods described in section 5.3.1, which essentially steers in the direction of lowest estimated forest density.

5. DVS Minimum Risk Planner: A simulated DVS was used with the methods given in section 5.3.2, which uses absence of events to estimate collision probabilities along straight line trajectories beginning at the current location.
Parameter & Value
--- & ---
Sensor Horizontal FOV (same for all) & 110 degrees
Collision Detection Rate & 180 fps
LIDAR Measurement Directions & 100 directions
LIDAR Maximum Range & 80 m
Camera Horizontal Resolution & 200 pixels
Camera Matching Patch Size & 10x10 pixels
Camera Frame Rate & 180 fps
DVS Horizontal Resolution & 100 pixels
DVS Event Threshold & 5.8%
DVS Event Simulation Rate & 500 fps
Car Width & 2.9 m
Car Height & 1.7 m
Car Length & 6.2 m
Car Forward Speed & 50 m/s

Table 5.1: Simulation parameters used for forest flight tests.

The simulations used the very simple translation-only dynamics given in equation 5.2, and visually represented the vehicle as a car rather than an aircraft to emphasize the two-dimensional nature of the motion. LIDAR sensor data was generated directly from the simulation rather than from the known locations of the trees, and therefore truly represents the distance to the nearest obstacle in each direction.

A list of simulation parameters fixed across all trials is given in 5.1. Distance and time parameters were constructed to roughly correspond to metres and seconds to allow the data to be interpreted more naturally, and as such parameters are listed with these units.

The forest environment used in the simulation consisted of a set of 9 unique 512x512m patches of forest, generated as a Poisson forest of the specified density prior to each test. To avoid the expensive operation of online re-randomization of the entire forest, patches out of view were randomly rearranged whenever the boundary of the current forest was approached. This approach works with Dubbins car or other dynamics to give the effect of a seamless endless forest that is effectively randomized, and is also useful for human operation although the forest densities tested prohibit realtime performance.

5.5 Performance Comparison

The performance of the various sensor-algorithm pairs was evaluated using collision data obtained from the simulation using the polygon collision methods provided by JMonkeyEngine. Physical collisions were not implemented in the simulation, so the motion of the vehicle is completely unaffected by the collision condition itself. For our purposes a collision is defined as a simulation frame (captured at 180fps) in which some part of the vehicle is in contact with some part of any tree in the environment. Subsequent frames which are both in collision will be referred to as separate collisions, though they may be part of the same collision event.
Because not all collisions are equally severe, three different types of collisions were logged:

1. **Off-screen Collisions**: All points of collision between the car and the tree lie outside of the cone of view of the sensor. This distinguishes side-swipe collisions common with memoryless planners from more severe head-on collisions.

2. **Front-end Collisions**: Some part of the car that is within the cone of view of the sensor is collided. These are generally considered worse as they are theoretically avoidable by a memoryless planner and often represent a qualitatively more severe form of collision.

3. **Camera Collisions**: A very small (0.1m) cube surrounding the actual camera is in collision with an obstacle. While not related to collision severity, as a performance metric this type of collision represents the failure of the planner to avoid obstacles even with a vehicle of negligible size.

The distinction between different types of collisions was made in order to approximately classify them according to severity or importance, as well as to provide performance metrics that represent different camera placements or vehicle sizes. While the actual severity of a collision event is much more difficult to quantify and would require a much more complex categorization, these types of collision should give some idea of the difference in behaviour of the various methods.

All simulation tests consisted of the vehicle operating for a simulated 300s, with collisions logs being the only captured data used to evaluate performance. This corresponds to a simulated travel distance of 15km, due to the extremely high forward velocity of 180km/hr.

Figures 5.6, 5.7, and 5.8 show the collision rates using the various methods of control in simulation for a sparse forest (1329 trees/km$^2$), a medium density forest (2659 trees/km$^2$) and a dense forest (5319 trees/km$^2$). These results compare the performance of LIDAR- and camera-based maximum visible distance controllers with the minimum risk (free space) and directional density DVS-based controllers, as well as a no control condition as a baseline.
Figure 5.6: Collision data from simulation tests using the different simulated sensors and associated control algorithms, for a sparse forest. Collision rate is expressed as a fraction of simulation time, and total collisions are broken down into the three types: off-screen, front-end and camera.

Figure 5.7: Collision data from simulation tests using the different simulated sensors and associated control algorithms, for a medium density forest.
Figure 5.8: Collision data from simulation tests using the different simulated sensors and associated control algorithms, for a dense forest.

Examining these results, we see that the performance of all methods relative to the control-free baseline decreases as the forest density increases, with the total collision rate for the monocular camera controller increasing from 3.9% of the no-control rate in the sparse forest to 13.2% in the dense forest. This is to be expected, as in the limit of infinite forest density the trees become at all times unavoidable and no method could be strictly better than no-control.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Sparse Forest</th>
<th>Mid-Density Forest</th>
<th>Dense Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIDAR</td>
<td>3.2</td>
<td>4.0</td>
<td>6.1</td>
</tr>
<tr>
<td>Mono Camera</td>
<td>22.8</td>
<td>23.9</td>
<td>26.2</td>
</tr>
<tr>
<td>DVS - Risk</td>
<td>16.1</td>
<td>18.3</td>
<td>18.6</td>
</tr>
<tr>
<td>DVS - Density</td>
<td>0.012</td>
<td>0.013</td>
<td>0.015</td>
</tr>
</tbody>
</table>

Table 5.2: Computational requirements for executing the various algorithms tested in simulation, in seconds of computation per second of simulated time. As expected the DVS density planner requires far less computation, since it does not store or use a complex representation of the environment.

Overall these results also show that LIDAR and a monocular vision method using dense cross correlations between subsequent frames to generate an obstacle point cloud provide superior performance to the DVS-based controllers described in this work. However considering the extremely low computational requirements of the DVS-based directional density planner, its performance is exceptional. In all forest densities it provided a front-end collision rate approximately twice that of the camera, and in the sparse forest yielded a 6-fold decrease in
total collision rate compared to no control, and an 11-fold decrease in front-end collisions. On the other hand, the free-space DVS planner performed very poorly, giving total collision rates approximately equal to the much cheaper directional density planner and front end collision rates significantly higher.

While the directional density DVS planner was able to reduce the rate of front-end collisions substantially, particularly in the low density forest, it suffered from a relatively high total collision rate resulting from a large number of off-screen collisions. This is a consequence of the memorylessness of the planner - the fact that it forgets obstacles as soon as they exit view, and can be seen also in the reduction in camera collisions compared to no-control, which is greater even than the front-end collision reduction, at nearly 13 times for the sparse forest. The monocular camera and LIDAR planners were not subject to the same susceptibility to off-screen collisions as they maintained a representation of the obstacles in the environment and were therefore able to remember obstacles after they exited the field of view.

5.6 Discussion

While neither of the DVS driven planners was able to outperform the dense monocular camera method in any of the forests, using any of the collision types to measure performance, the performance of the low computation directional density DVS planner was very good given its efficiency. This method effectively requires only the incrementation of a single value in a one-dimensional array for each new event, along with periodic adjustment of the entire array to implement the open-loop density estimation. While we have not presented a formal comparison of the required computation for the various methods, because there exist many much more efficient camera-based methods, the directional density planner tested here in simulation could certainly be implemented in real-time on a small embedded system using only a fraction of available computation.

In reality, a single front-end collision and most collisions in general may be enough to render a small air vehicle inoperable, so we must ask whether simply achieving a substantial reduction in collision rate is a meaningful result, regardless of how computationally efficient it is. It seems that with the possible exception of the LIDAR, any of the sensors tested here would not alone be able to achieve the high level of reliability required for a real world mission. Furthermore, the LIDAR used in these simulation tests had a range of 80m, which is very high by current standards and would require substantial power. One possible solution would be to couple the DVS directional density planner with a short range scanning rangefinder. In this setup, the DVS would be used for longer range planning to steer the vehicle in a direction believed to contain a lower density of obstacles, and the short range proximity sensor would allow the vehicle to avoid obstacles that the DVS planner did not successfully steer away from.

Overall these results are promising, since the efficiency of the directional density planner means that it could be implemented easily at a very low level, and following miniaturization of DVS, may be able to drive the behaviour of extremely small aerial vehicles such as robotic insects. The method is actually so simple that it may be reasonable to implement in an entirely analog way, further reducing power and size constraints, and may be highly suited to systems where collisions are not fatal.
Chapter 6

Conclusion

6.1 Summary of Contributions

In this work we have developed a rigorous understanding of how to use dynamic vision sensors for certain high-speed robotics applications, with algorithms that make use of their remarkable properties to provide low latency, high bandwidth performance. Simulation and experimental tests were conducted that corroborate our theoretical results. Additionally, theoretical and empirical evidence has been presented to support the argument that in some control settings DVS are superior to cameras in the sense that they can provide better performance for a given mean computational load or data output rate. Specifically, the contributions of this work can be divided into four parts, embodied in the associated publications [128] [129] [132]:

1. Development of algorithms for local event-wise heading signal estimation for the heading control problem, where events are efficiently mapped to heading measurements.

2. Development of approximated inference methods to extend heading estimation techniques to a global domain, where events are mapped to heading measurements using only slightly more computation than in the local case.

3. Modelling of the event-to-measurement process in the context of heading regulation, and the resulting transformation of spurious event noise into additive measurement noise with state-dependent variance. Design of feedback control laws informed by this model and shown to be theoretically and empirically superior to linear control for a class of linear dynamical systems.

4. Comparison of DVS to cameras for the heading control problem, with theoretical results indicating that when computational resources are large, DVS are superior to cameras, particularly in cases where task variation is high. Support of these results through experimental testing.

In addition to these core contributions, a number of unpublished ideas have also been presented, including a theoretical argument for the notion that in systems with stochastic delay maximum likelihood filtering can actually reduce both the expectation and variance of
measurement lag. This idea is related to the notion of persistent excitation in DVS, wherein pixels produce events at an increased rate during the period following a brightness change - an observed phenomenon that is not fully captured in most of our models. The theoretical result therefore is used to justify treating the stochastic delay as deterministic in order to simplify much of the discussion related to heading estimation and control.

Several unpublished algorithms for sub-canopy forest flight navigation using dynamic vision sensors have also been presented and tested in simulation. This problem provided much of the original impetus for this work, and while most of the emphasis of the thesis has been on the heading control problem (which also has a number of important practical applications), the final chapter returned to the original problem to apply some of the concepts from the much simpler heading regulation task to that of forest flight. Extremely simplistic methods for avoiding trees using the DVS were presented, and simulation results suggest that while such minimalistic algorithms are likely not useable on their own, they can provide a substantial improvement over no control, and may therefore be useful in combination with other more powerful sensors. More complex but still relatively simplistic approaches yielded better results, but are still inferior in simulation to a computationally intensive camera-based approach.

6.2 DVS for Heading Regulation

If there is one message to take away from this work it is that a dynamic vision sensor can be effectively used for heading control in a way that does not compromise the latency or bandwidth of the sensor and which is more effective than a camera with the same available computation. DVS also eliminate some of the other problems of conventional cameras, including motion blur or poor dynamic range.

Table 6.1 summarizes the main heading regulation results given in this thesis, sorted by the type of brightness profile to which they apply and the type of result. Some entries in the table are absent because the 1-edge and 2-edge brightness profiles are both types of piecewise constant brightness profile, and tests were conducted using the cartoon forest environment, though the results should apply to the simpler cases as well. From this we see that the theory and algorithms related to arbitrary brightness profiles that are presented are generally untested and additional work is required to verify the practical efficacy of the methods proposed.

Much of this work has also focused on analyzing the relative performance of DVS and cameras and understanding the fundamental differences between the two classes of sensors. Theoretical results from the heading tracking problem show that under our idealized model of sensor operation in the absence of noise and at a fixed data output rate, DVS are superior to cameras above a certain transition data rate. Similar results hold when computational usage is fixed, assuming that the algorithms used with the camera require sufficiently more operations per frame than DVS operations per event. These results also indicate that while camera performance degrades when the task is highly varied (heading trajectory to be tracked has high variance in slew rate), the DVS is able to adapt and does not suffer the same loss. In other words, the relative performance of the DVS compared to the camera increases as the
variation in the task increases.

Generalizing this concept we describe one of the core differences between the two classes of sensor as the fact that the DVS is able to adapt the data rate and therefore computational usage as the problem varies, automatically increasing event rate if motion is high and decreasing it if motion is low. This adaptability is in contrast to the operation of a conventional camera where the output data rate is constant regardless of scene motion but as a result the performance must suffer if the task varies. It is this versatility, along with the asynchronicity of measurement and corresponding reduction in latency, that allows the DVS to outperform - a result consistent with established results from event-based control.

Experimental results generated using our rotating drum apparatus provide some evidence for our theoretical claims and support for the use of our sensor models. While we have demonstrated the efficacy of our methods for heading signal construction in piecewise constant brightness environments and our feedback control methodology derived from the unique sensor noise model presented, experimental tests were limited by the apparatus. The relatively large radius of the backdrop meant that it could not be actuated at bandwidths that would push the sensor to its limits, and the same is true for the sensor itself. Furthermore, friction in the apparatus, particularly static friction, meant that it was inherently very stable and did not represent the spacecraft dynamics for which we originally motivated the problem. Nonetheless, it did allow for the verification of a number of elements of our derived sensor model, which was then used in simulation to validate our control design techniques and compare the result with conventional PD control for a system with dynamics approximating those of a satellite. A large improvement in regulation performance was observed with a provably stable switching control scheme compared to PD control, which, under our sensing model was seen to be unstable.

Observations from the various feedback control tests conducted during this work have also established connections between basic feedback control and biological microsaccades. Under our linear continuous-time sensor model, noise variance diverges to infinity as the heading rate approaches zero, rendering it impossible to fixate precisely on a static target. As a result, control methods that attempt to regulate the heading are subject to persistent small steady

<table>
<thead>
<tr>
<th>Brightness Profile</th>
<th>1 Edge</th>
<th>2 Edges</th>
<th>Piecewise Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>OL Theory</td>
<td>2.1, 4.3</td>
<td>3.3</td>
<td>3.4, 2.4, 2.5</td>
</tr>
<tr>
<td>OL Algorithms</td>
<td>2.5</td>
<td>3.3</td>
<td>3.4</td>
</tr>
<tr>
<td>OL Simulation</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OL Experiments</td>
<td>3.2, 3.3, 3.6</td>
<td>3.3</td>
<td>3.4, 3.5, 3.6</td>
</tr>
<tr>
<td>CL Theory</td>
<td>-</td>
<td>-</td>
<td>3.7</td>
</tr>
<tr>
<td>CL Algorithms</td>
<td>2.5</td>
<td>3.3</td>
<td>3.4, 3.7</td>
</tr>
<tr>
<td>CL Simulation</td>
<td>-</td>
<td>-</td>
<td>3.7</td>
</tr>
<tr>
<td>CL Experiments</td>
<td>-</td>
<td>-</td>
<td>3.4, 3.7, 4.5</td>
</tr>
</tbody>
</table>

Table 6.1: A summary of the various types of results relevant to the heading control problem presented in this thesis, with the associated sections listed (OL = open-loop, CL = closed-loop).
state motions that could theoretically be eliminated if a sensor without our peculiar noise was used. While our switching controller mitigates these machine microsaccades compared to PD control, it cannot eliminate them completely, as the motion is required for observability of the system.

6.3 DVS for Forest Flight

The methods presented for DVS-based navigation in the forest are not particularly effective, though they are very computationally efficient. As an alternative, conventional vision has the opposite problem - though it can be relatively effective it can also be extremely computational inefficient. This is likely due in large part to the fact that cameras were developed mainly to capture images or video for the purposes of human consumption, and not for robotic applications. Humans are interested in the content of the image, and generally want most of the information to be saved, while an autonomous vehicle flying through the forest may not care about the details and would generally have a very different notion of what information is important. In a sense the DVS attempts to address this issue by providing a stream of information seemingly more amenable to use in navigation and less appealing to human aesthetic tastes. As we have seen, for certain problems compressing the video stream into thresholded changes delivered asynchronously can be beneficial.

6.4 Analog Vision Sensors

I believe many of the limitations of current vision-based robotics arise from forcing this type of highly parallel perception into a digital, serial computation framework. Parallel computing platforms such as graphics processing units (GPUs) can resolve the problem of serialization and give vast improvements in performance when it comes to speed, however such devices are often very power-hungry and are not suitable for small agile robots for which power constraints are drivers of design.

Perhaps it is digital computation that restricts machine vision from achieving what its biological counterpart has. DVS are analog sensors (the absence of events carries information) that use spikes to bridge the gap between their fundamental continuous-time operation and the discrete world of digital computation. What if sensors could be manufactured to, for example, compute depth through dense cross correlation methods with stereo vision or compute optical flow with a single lens, in an entirely analog way? What if such a sensor could be integrated with analog inertial sensors and undergo further analog computation to produce a feedback control signal? Such a sensor might have bandwidth on the order of megahertz and latency on the order of microseconds, while consuming only minimal power. Such a sensor might allow optical flow to be calculated using classical continuous-space continuous-time sensor models. With a stereo pair such a sensor may provide a passive sensing alternative to LIDAR that consumes less power and provides vastly faster sensing.

In the design of many machines including those made of words (algorithms), there is often a tradeoff between generality and efficiency. The more tasks a machine can do, the worse it is at any particular task. Conventional cameras are very general purpose devices and as such they
are not especially efficient for any particular class of robotic tasks. As production of analog or digital application specific integrated circuits (ASICs) becomes more economical and more widely available, roboticists may be able to use these tools to build machine vision systems tailor-made to their particular applications, whose efficiency vastly suprasses that of general purpose cameras. For now, design of vision sensors with novel pixels and an asynchronous communication interface is cutting-edge, however in the future-perhaps having custom ICs made will be as easy and common as having custom circuit boards made is now, and a wide variety of vision sensors suited to robotics will flourish.

6.5 Future Work

The study of neuromorphic vision sensors as applied to motion control problems is very much still open and certainly has vast potential for exploration. Based on the trend in camera-based approaches to some of the most difficult tasks in robotics it seems likely that provable performance guarantees will be very difficult to obtain in many cases. Like cameras, these are sensor interfacing to reality through an extremely complex image formation process and as such the exact relationship between sensor output and the a particular representation of the state of reality will be difficult to nail down. For example feature matching methods used with conventional vision rely on the concept that regions of multiple images matched to each other correspond to “the same” location in three-dimensional space. Nonetheless, I believe that sensors such as DVS may prove to be empirically very effective for a variety of robotic tasks and I would be happy to see future work pursue practically effective use of these sensors for high-speed motion control.

More directly related to the results presented in this thesis is future work pertaining to their continued empirical evaluation. Are the control methods proposed actually useful for satellite pointing? Do they actually require less power than camera-based control systems providing the same level of performance? How do the obstacle avoidance methods fare in real forest flight?

Extending the stability results presented for the double integrator to a broader class of dynamical systems would also be an valuable contribution. This may require additional control design, but further results in this direction may be able to utilize our measurement model to provide a general-purpose method for controlling systems with DVS.

While we have presented theoretical results for open-loop superiority of DVS over cameras under certain assumptions and with fixed computational or bandwidth allocation, the extension to closed-loop has been only empirical and in the form of a problem statement. Therefore an interesting future contribution would be to provide a similar theoretical analysis in the closed-loop case with the aim of making claims regarding optimal sensor choice.

With regards to autonomous navigation problems such as that considered in chapter 5, the possibility for future work is vast. Though we have presented a highly efficient method that in simulation is effective at guiding the vehicle to areas of lower tree density and reducing the number and severity of collisions, more sophisticated methods may be able to solve the problems associated with this approach to provide performance comparable to cutting-edge camera-based methods with minimal computational load. The exploration of stereo methods
for DVS could also be a rich area for future research, with relatively little work having been done in that direction.

Beyond robotic control, DVS are unique sensors that may have a wide range of applications, many of which are only beginning to be understood. In addition these sensors may be only the first in a series of vision sensors with unique pixels offering continuous-time measurements. Ongoing development of neuromorphic vision sensors and other types of vision sensors may be highly beneficial in a variety of settings. For example development of DVS that provide regular frames of brightness data as well as events is underway, and with such a sensor conventional vision algorithms could easily be adapted to gain the high-speed properties of DVS. Similarly, sensors whose pixels are able to respond in a continuous-time way to spatial brightness gradient rather than or in addition to temporal brightness gradient could also be useful, for example for optical flow computation. The possibilities beyond non-conventional vision are immense and may prove indispensable to future progress in the field of robotics.

6.6 Thanks

I would like to thank the members of my thesis committee: Emilio Frazzoli, Berthold Horn and Andrea Censi, for their guidance and support in completing this work. I would also like to thank Dmitry Yershov and Luca Carlone for their assistance in revising my thesis, and Sertac Karaman for serving as the external evaluator of my thesis proposal. Finally I would like to thank the MIT aero-astro support staff: the administrators Lisa Gaumond and Beth Marois for helping me acquire materials, obtain reimbursements and file paperwork on time, and the machinist Todd Billings for assisting me with machining experimental equipment.
Bibliography


