

---

# **Analogue VLSI for Temporal Frequency Analysis of Visual Data**

---

*Alasdair Sutherland*



A thesis submitted for the degree of Doctor of Philosophy.  
**The University of Edinburgh.**  
December 2003





---

# Abstract

---

When viewed with an electronic imager, any variation in light intensity over time can provide valuable information regarding the nature of the object or light source causing the intensity change. By estimating the frequency of such light intensity variations, temporal frequencies can be extracted from the visual data, which may prove useful in a variety of applications. For instance, certain objects exhibit unique temporal frequencies, which could facilitate identification or classification. Other potential applications include remote, early failure detection for rotating machinery, as well as the possible detection of cancerous breasts using infra-red imaging techniques.

The aim of the research reported in this thesis is the development of a CMOS image-processor, capable of extracting such temporal frequencies from any scene it is exposed to. In addition to finding the fundamental frequency, the sensor aims to extract the relative strength of upto the first four harmonics, producing a Fourier style decomposition of the incident light intensity into a *temporal frequency signature*.

A heavy emphasis was placed on low power operation, leading to an investigation of analogue signal processing techniques with transistors biased in the subthreshold region of operation. The parallel processing advantages of combining light sensitive elements with signal processing elements in each pixel were also investigated, resulting in a system incorporating focal-plane computation.

Software simulations of various novel system level algorithms are reported, with the successful approach used to create fundamental frequency maps of test data. The approach was also simulated to prove its robustness to noise commonly found in CMOS imager implementations.

Circuits are presented which accurately extract the fundamental frequency of variations in light intensity, while benefiting from the low power consumption of subthreshold analogue circuitry. A novel algorithm which places a band pass filter onto the fundamental frequency of any incident light intensity with an accuracy of 3 % is also presented. The system can tune from 20 Hz to 10 kHz at a maximum rate of 9 kHz/s, and can be considered the first step in the creation of a single-chip pseudo-Fourier light intensity processing unit.



---

# Acknowledgements

---

Firstly, I would like to thank my supervisors, Dr Alister Hamilton and Dr David Renshaw, for their advice and guidance over the past few years. Particular thanks to Alister for providing such an interesting topic to research, as well as the freedom to investigate interesting approaches while ensuring the research remained focused.

I would also like to thank Patrice Fleury, Adria Bofill, Matthew Purcell, Robin Woodburn, Shinichiro Matsunaga and Keith Findlater as well as Alister and David for their invaluable contributions to my understanding of chip design, simulation, layout and testing.

Thanks to Marcus Alphey, Andrew Peacock, Chris Haworth and Peter Hillman for their assistance in the finer points of the many software tools used throughout this research.

Thanks also to Mark Glover at QinetiQ, along with his colleagues David Lees and Ron Ballingall for industrially sponsoring this research. Particular thanks to Mark for technical discussions and providing the test data sets used in the algorithm development.

Thanks must also go to all members of the old ISG and the new IMNS, both past and present, for making my time in the department such an enjoyable and rewarding experience.

Thanks to EPSRC, who co-sponsored this research under student award number 99302864.

Finally, I would like to thank my Mother and Father for their support and encouragement throughout my studies.



---

# Contents

---

Declaration of originality . . . . .	iii
Acknowledgements . . . . .	iv
Contents . . . . .	v
List of figures . . . . .	ix
List of tables . . . . .	xiii
Acronyms and abbreviations . . . . .	xiv
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Motivation . . . . .	1
1.3 Implementation Issues . . . . .	7
1.3.1 System Level Design Decisions . . . . .	7
1.3.2 Circuit Level Design Decisions . . . . .	9
1.4 Neuromorphic Approach to Image Processing . . . . .	11
1.5 Engineering Solution within Neuromorphic Framework . . . . .	14
1.6 Contributions . . . . .	15
1.7 Structure . . . . .	15
<b>2 Smart Sensors Incorporating Focal Plane Processing: Literature Review . . . . .</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Spatial Processors . . . . .	17
2.2.1 Silicon Retina: 'Scientific' Implementations . . . . .	18
2.2.2 Silicon Retina: 'Engineering' Approaches . . . . .	23
2.2.3 Vision Sensors for Object Orientation/Selection . . . . .	24
2.2.4 Other interesting Spatial Processing Performed on the Focal-Plane . . . . .	25
2.2.5 Comments on Focal-Plane Approaches to Spatial Processing . . . . .	26
2.3 Temporal Processors . . . . .	26
2.3.1 Temporal Processing: 'Scientific' Approaches . . . . .	27
2.3.2 Temporal Processing: 'Engineering' Approaches . . . . .	29
2.3.3 Comments on Focal-Plane Approaches to Temporal Processing . . . . .	29
2.4 Spatio-Temporal Processors: Motion/Velocity Estimation . . . . .	30
2.4.1 'Scientific' Motion Detection: Reichardt Correlation Algorithms . . . . .	32
2.4.2 'Scientific' Motion Detection: Alternative Algorithms . . . . .	35
2.4.3 'Engineering' or Computational Motion Detection: Token-Based Algorithms . . . . .	38
2.4.4 'Engineering' or Computational Motion Detection: Gradient-Based Algorithms . . . . .	41
2.4.5 Programmable Focal-Plane Image Processing . . . . .	43
2.4.6 Comments on Focal-Plane Approaches to Spatio-Temporal Processing . . . . .	45
2.5 Summary . . . . .	45
<b>3 Software Development of Temporal Frequency Analysis Algorithm . . . . .</b>	<b>48</b>
3.1 Test Data . . . . .	48



3.1.1	Fan Data Sequence . . . . .	49
3.1.2	Propeller Plane Data Sequence . . . . .	50
3.1.3	Helicopter Data Sequence . . . . .	50
3.2	Dyadic Tree Algorithm . . . . .	50
3.3	Focal Plane Extraction of Fundamental Temporal Frequency . . . . .	51
3.4	The <i>Average vs Active</i> Algorithm . . . . .	53
3.4.1	Software Simulation of the <i>Average vs Active</i> Algorithm . . . . .	54
3.4.2	Comments on the <i>Average vs Active</i> Algorithm . . . . .	55
3.5	The <i>Flashing Pixel</i> Algorithm . . . . .	56
3.5.1	Laplacian Mask vs Half-Laplacian Mask . . . . .	57
3.5.2	Development of the <i>Flashing Pixel</i> Algorithm . . . . .	60
3.5.3	Noise Analysis of the <i>Flashing Pixel</i> Algorithm . . . . .	65
3.5.4	Whole Image Analysis of the <i>Flashing Pixel</i> Algorithm . . . . .	78
3.5.5	Circuit-Level Implementation of the <i>Flashing Pixel</i> Algorithm . . . . .	86
3.6	Conclusions . . . . .	86
<b>4</b>	<b>Test IC One: Fundamental Frequency Extraction</b>	<b>89</b>
4.1	System-Level Design . . . . .	89
4.2	Circuit-Level Design . . . . .	90
4.3	Photoelement . . . . .	91
4.3.1	Photodiode . . . . .	91
4.3.2	Phototransistor . . . . .	91
4.4	Photocircuit: Logarithmic Compression Photocircuit . . . . .	92
4.4.1	Large-Signal Characteristics . . . . .	92
4.4.2	Small-Signal Characteristics . . . . .	94
4.4.3	Implementation of the Logarithmic Compression Photocircuit . . . . .	95
4.4.4	IC Test Results: Logarithmic Compression Photocircuit . . . . .	96
4.4.5	Comments on the Logarithmic Compression Photocircuit . . . . .	98
4.5	High Pass Filter: Gm-C First Order Filter . . . . .	100
4.5.1	Low Frequency Filtering with Gm-C Circuit Structures . . . . .	100
4.5.2	Realising Transconductance Elements with Operational Transconductance Amplifiers . . . . .	102
4.5.3	OTA-C First Order High Pass Filter . . . . .	105
4.5.4	OTA-C High Pass Filter: Power Consumption . . . . .	107
4.5.5	IC Test Results: OTA-C First Order High Pass Filter . . . . .	109
4.5.6	Comments on the OTA-C First Order High Pass Filter . . . . .	112
4.6	Comparator . . . . .	112
4.6.1	Comparator with Positive Feedback and Optional Hysteresis . . . . .	114
4.6.2	Comparators Implemented on Test IC One . . . . .	116
4.6.3	Comparator Power Consumption . . . . .	117
4.6.4	IC Test Results: Comparator . . . . .	119
4.6.5	Comments on the Comparator . . . . .	121
4.7	System Level Test IC Results . . . . .	123
4.7.1	Pixels One and Two: Frequency Accuracy . . . . .	124
4.7.2	Pixels Five and Six: Frequency Accuracy . . . . .	127
4.8	Comments on Test IC One . . . . .	127
4.8.1	Conclusions on Photoelement . . . . .	128



4.8.2	Conclusions on Photocircuit . . . . .	129
4.8.3	Conclusions on High Pass Filter . . . . .	130
4.8.4	Conclusions on Comparator . . . . .	131
4.8.5	Conclusions on Pixel Processing Cells . . . . .	132
<b>5</b>	<b>Test IC Two: Self-Referencing Fundamental Extraction</b>	<b>134</b>
5.1	System Level Approach . . . . .	134
5.2	Low Frequency OTA-C Low Pass Filter . . . . .	135
5.2.1	Operational Transconductance Amplifiers on IC Two . . . . .	135
5.2.2	OTA structures: Input Common Mode Range . . . . .	137
5.2.3	Low Pass Filter's Implemented on IC Two . . . . .	138
5.2.4	Low Pass Filter: Power Consumption . . . . .	139
5.2.5	IC Test Results: OTA-C First Order Low Pass Filter . . . . .	139
5.2.6	Comments on the OTA-C First Order Low Pass Filter . . . . .	145
5.3	System Level Test IC Results: Self-Referencing Pixel Processing Units . . . . .	146
5.3.1	Frequency Accuracy Tests . . . . .	147
5.3.2	Comparison of Pixels with and without Hysteresis in the Comparator . . . . .	149
5.3.3	Comparison of Different Illumination Levels . . . . .	151
5.3.4	Comparison of Pixels with Varying Filter Control Values . . . . .	152
5.3.5	Comparison of Pixels with and without Cascode OTAs . . . . .	154
5.3.6	Comments on Self-Referencing Pixel Processing Units . . . . .	155
5.4	Comments on Test IC Two . . . . .	156
5.4.1	Conclusions on OTA-C Low Pass Filters . . . . .	156
5.4.2	Conclusions on Self-Referencing Pixel Processing Cells . . . . .	157
<b>6</b>	<b>Test IC Three: <i>Minipix</i> Self-Referencing Pixel Processing Unit</b>	<b>159</b>
6.1	<i>Minipix</i> : Physical Layout . . . . .	159
6.2	<i>Minipix</i> : Simulated Current Consumption . . . . .	159
6.3	<i>Minipix</i> : Effect of Increased Variation in Filter Cutoff Due to Subthreshold Current Mismatch . . . . .	162
6.4	<i>Minipix</i> : Measured IC Test Results . . . . .	164
6.4.1	<i>Minipix</i> : Frequency Response . . . . .	164
6.4.2	<i>Minipix</i> : Frequency Accuracy . . . . .	164
6.5	Conclusions on the <i>Minipix</i> Algorithm . . . . .	165
<b>7</b>	<b>Test IC Three: Automatically Tuned Band Pass Filter with Phase Derived Feed- back</b>	<b>167</b>
7.1	Automatically Tuning BPF: System Level Approach . . . . .	167
7.2	OTA-C 4th-Order Band Pass Filter . . . . .	169
7.2.1	OTA-C 4th-Order BPF: Theory . . . . .	170
7.2.2	Operational Transconductance Amplifiers on Test IC Three . . . . .	172
7.2.3	Physical realisation of the 4th-Order OTA-C BPF . . . . .	172
7.2.4	OTA-C 4th-Order BPF: Power Consumption . . . . .	173
7.2.5	OTA-C 4th-Order BPF: Measured Test IC Results . . . . .	173
7.2.6	Comments on the OTA-C 4th-Order BPF . . . . .	175
7.3	Digital Phase Detector . . . . .	177
7.3.1	Digital Gates with Current Limiting Transistors . . . . .	178



7.3.2	Digital Phase Detector: Physical Realisation . . . . .	179
7.3.3	Digital Phase Detector: Simulated Test Results . . . . .	180
7.3.4	Comments on the Digital Phase Detector Circuit . . . . .	180
7.4	Charge Pump . . . . .	181
7.4.1	Charge Pump: Power Consumption . . . . .	183
7.4.2	Charge Pump: Physical Realisation . . . . .	183
7.4.3	Charge Pump: Simulated Results . . . . .	183
7.4.4	Comments on the Charge Pump Circuit . . . . .	184
7.5	System-Level Test IC Results: Automatically Tuned BPF . . . . .	184
7.5.1	Automatically Tuned BPF: Simulated Current Consumption . . . . .	185
7.5.2	Automatically Tuned BPF: Tuning Range . . . . .	187
7.5.3	Automatically Tuned BPF: Tuning Speed . . . . .	193
7.5.4	Automatically Tuned BPF: Tuning Accuracy . . . . .	196
7.6	System-Level Test IC Results: Automatically Tuned BPF with Visual Stimulus	202
7.6.1	Automatically Tuned BPF: Tuning Range with Visual Stimulus . . . .	202
7.7	Conclusions on the Performance of the Automatically Tuned BPF Algorithm	203
<b>8</b>	<b>Summary and Conclusions</b>	<b>206</b>
8.1	Summary . . . . .	206
8.2	Conclusions . . . . .	208
8.3	Contributions . . . . .	211
8.4	Critical Evaluation . . . . .	211
8.5	Future Work . . . . .	213
8.6	Final Comments . . . . .	215
<b>A</b>	<b>Dyadic Tree Algorithm</b>	<b>216</b>
A.1	Wavelet Transforms . . . . .	216
A.2	Wavelet Processing with the Dyadic Tree Filterbank . . . . .	217
A.3	Software Simulation of the Dyadic Tree Filterbank . . . . .	219
A.3.1	Dyadic Tree Sim Results: Luminescence <i>Flashing</i> at 20 Hz. . . . .	221
A.3.2	Dyadic Tree Sim Results: Luminescence <i>Flashing</i> at 90 Hz. . . . .	223
A.3.3	Dyadic Tree Simulations: Frequency Band Energy Content for Luminescence Device <i>Flashing</i> at 20 Hz and 90 Hz . . . . .	224
A.3.4	Dyadic Tree Simulations: Frequency Band Energy Content for All Available Luminescence Device Frequencies, 10 Hz to 90 Hz . . . . .	225
A.4	Comments on the Dyadic Tree Algorithm . . . . .	226
<b>B</b>	<b>Analogue Buffer Circuitry</b>	<b>229</b>
B.1	Differential Stage . . . . .	229
B.2	DC Offset . . . . .	229
	<b>References</b>	<b>232</b>



---

## List of figures

---

1.1	Definition of Temporal Frequency . . . . .	2
1.2	Temporal and Frequency Domain representations of Temporal Frequencies . .	4
1.3	Comparison of Object Frequency Signatures . . . . .	5
1.4	Pseudo Fourier Processor . . . . .	9
2.1	Retinal Approach to Spatial Filtering . . . . .	19
2.2	Retinal Processing for Edge Extraction . . . . .	22
2.3	Delbruck's Adaptive Photoreceptor . . . . .	27
2.4	Hassenstein-Reichardt Biological Motion Detection Algorithm . . . . .	31
2.5	Delbruck's Velocity Tuned Pixel . . . . .	34
2.6	Barlow and Levick's Model of Direction Selectivity in the Rabbit Retina . . . .	36
2.7	Etienne-Cummings et al Correlation/Token Motion Detector Hybrid . . . . .	37
2.8	Kramer et al's Token Based Velocity Sensor . . . . .	40
2.9	Spatial Processing Using Convolution Kernels . . . . .	43
3.1	Selected Consecutive Frames from Fan Data Sequence . . . . .	49
3.2	Selected Consecutive Frames from Propeller Plane Data Sequence . . . . .	50
3.3	Selected Consecutive Frames from Helicopter Data Sequence . . . . .	51
3.4	Temporal and Frequency Domain Representations of 100 Hz Square Wave, with and without 200 Hz Sine Wave Noise Source. . . . .	52
3.5	Tuning Band Pass Filters to Integer Multiples of the Fundamental Frequency . .	53
3.6	Operation of the <i>Average vs Active</i> Algorithm . . . . .	54
3.7	Selected Pixels from the 'Fan' Data Sequence used to test the <i>Average vs Active</i> Algorithm . . . . .	55
3.8	<i>Average vs Active</i> Algorithm Applied to Luminescence Device Flashing at 20 Hz	56
3.9	<i>Average vs Active</i> Algorithm Applied to Fan . . . . .	57
3.10	Selected Frames from the Propeller Plane and Helicopter Data Sequences . . .	58
3.11	Edge-Enhancement with the Laplacian Convolution Kernel . . . . .	59
3.12	Edge-Enhancement with the <i>Half-Laplacian</i> Convolution Kernel . . . . .	59
3.13	Comparison of Edge-Enhancement Convolution Kernels . . . . .	61
3.14	Comparison of Half-Laplacian and Laplacian Convolution Kernels on Single Pixel Data . . . . .	62
3.15	Selected Frames from the Plane and Helicopter Data Sequences, Highlighting the Chosen Pixel Locations . . . . .	63
3.16	Operation of the <i>Flashing Pixel</i> Algorithm Applied to Pixel 100,72 in the Plane Data Sequence . . . . .	64
3.17	Operation of the <i>Flashing Pixel</i> Algorithm Applied to Pixel 45,53 in the Heli- copter Data Sequence . . . . .	65
3.18	Band Pass Filter Positioning with the Plane Data-Set, using the half-Laplacian Mask algorithm . . . . .	66
3.19	Band Pass Filter Positioning with the Helicopter Data-Set, using the half- Laplacian Mask algorithm . . . . .	66



3.20	Effect of Fixed Pattern Noise on the <i>Flashing Pixel</i> Algorithm: Plane Pixel (100,72) . . . . .	67
3.21	DC Offset caused by Fixed Pattern Noise with the <i>Flashing Pixel</i> Algorithm . . . . .	69
3.22	Effect of Fixed Pattern Noise on the <i>Flashing Pixel</i> Algorithm: Helicopter Pixel (45,53) . . . . .	70
3.23	Differences Between the <i>Flashing Pixel Algorithm</i> , the Half-Laplacian HPF Algorithm and the No-Mask Algorithm. . . . .	71
3.24	Effect of Fixed Pattern Noise on the <i>Half-Laplacian HPF Algorithm</i> . . . . .	72
3.25	Effect of Fixed Pattern Noise on the <i>No-Mask Algorithm</i> . . . . .	74
3.26	Effect of Random Transient Noise on the <i>Flashing Pixel Algorithm</i> . . . . .	76
3.27	Effect of Random Transient Noise on the <i>Half-Laplacian HPF Algorithm</i> . . . . .	77
3.28	Effect of Random Transient Noise on the <i>No-Mask Algorithm</i> . . . . .	79
3.29	Selected Frames from the Test Data Sequences . . . . .	81
3.30	Fundamental Frequency Maps Generated with the <i>Flashing Pixel Algorithm</i> . . . . .	82
3.31	Fundamental Frequency Maps Generated with the <i>Half-Laplacian HPF Algorithm</i> . . . . .	84
3.32	Fundamental Frequency Maps Generated with the <i>No-Mask Algorithm</i> . . . . .	85
3.33	Circuit-Level Implementation of the <i>No-Mask Algorithm</i> . . . . .	87
4.1	Circuit Level Implementation for the <i>No-Mask Algorithm</i> . . . . .	90
4.2	Logarithmic Compression Photocircuit . . . . .	93
4.3	Simulation of the Logarithmic Photoreceptor's DC Response . . . . .	94
4.4	Logarithmic Compression Photocircuit: Small Signal Model . . . . .	95
4.5	Simulation of the Logarithmic Photoreceptor's AC Response . . . . .	96
4.6	Logarithmic Compression Photocircuit: Implementation on Test Chip One . . . . .	97
4.7	Measured Test IC Results- Comparison of Logarithmic Photoreceptor with Photodiode and Phototransistor Loads . . . . .	98
4.8	Measured Test IC Results- Frequency Response of the Logarithmic Compression Photocircuit with Phototransistor . . . . .	99
4.9	First Order Integrator- The Basis for many Gm-C Filter Structures . . . . .	101
4.10	Operational Transconductance Amplifier: NMOS Differential Pair with Current Mirror Load . . . . .	102
4.11	Simulated Comparison of Transconductance versus Tail Current for Simple OTA . . . . .	104
4.12	OTA Structures Implemented on Test Chip One . . . . .	105
4.13	1st Order Gm-C High Pass Filter . . . . .	106
4.14	Physical Layout of the High Pass Filters on Test IC One . . . . .	108
4.15	Measured Test IC Results-Frequency Response of the Two OTA-C High Pass Filter Structures . . . . .	110
4.16	Measured Test IC Results-DC Offset of the Two OTA-C High-Pass Filter Structures . . . . .	113
4.17	Comparator with Optional Hysteresis . . . . .	114
4.18	Effect of a Comparator with Hysteresis on a Noisy Signal . . . . .	115
4.19	Physical Layout of the Standard Comparator (Without Hysteresis) . . . . .	118
4.20	Measured Test IC Results-Comparator Switching Frequency . . . . .	120
4.21	Measured Test IC Results-Direct Comparison of Switching for Comparator with and without Hysteresis . . . . .	122
4.22	Signal Flow for Fundamental Frequency Extraction Algorithm One: The No-Mask Algorithm . . . . .	124



4.23	Physical Layout of Fundamental Frequency Extraction Algorithm: Pixel Two . . .	125
4.24	Measured Test IC Results-Frequency Accuracy of Pixel Cells One and Two . . .	126
4.25	Measured Test IC Results-Combined Frequency Response of the Circuit Elements Combined to Produce Pixel Processing Cells One and Two . . . . .	127
4.26	Measured Test IC Results-Combined Frequency Response of the Circuit Elements Combined to Produce Pixel Processing Cells Five and Six . . . . .	128
4.27	Measured Test IC Results-Frequency Accuracy of Pixel Cells Five and Six . . .	129
5.1	Circuit Level Implementation of the Self-Referencing Algorithm . . . . .	135
5.2	1st Order Gm-C Low Pass Filter . . . . .	136
5.3	Operational Transconductance Amplifiers on Test IC Two . . . . .	137
5.4	Capacitor created with MOS Transistor . . . . .	139
5.5	Physical Layout of the Three OTA-C Low Pass Filter Structures . . . . .	140
5.6	Measured Test IC Results: Frequency Response of OTA-C Low Pass Filter One	141
5.7	Measured Test IC Results: Frequency Response of OTA-C Low Pass Filter Two	142
5.8	Measured Test IC Results: Frequency Response of OTA-C Low Pass Filter Three	143
5.9	Measured Test IC Results: Input Common Mode Range of OTA-C Low Pass Filter Three . . . . .	144
5.10	Physical Layout of Pixel Processing Cell Three . . . . .	147
5.11	Measured Test IC Results: Frequency Accuracy of the Pixel Processing Elements	148
6.1	<i>Minipix</i> Self Referencing Pixel Processing Unit . . . . .	160
6.2	Effect of Subthreshold Current Mismatch on the <i>Minipix</i> Algorithm . . . . .	163
6.3	<i>Minipix</i> Sub-Circuit Frequency Response . . . . .	165
6.4	<i>Minipix</i> Frequency Accuracy . . . . .	166
7.1	System Level Algorithm Implemented on Test IC Three . . . . .	168
7.2	Second Order OTA-C Band Pass Biquad Filter . . . . .	171
7.3	Simulated Frequency Response for the 4th-Order OTA-C Biquad Band Pass Filter	172
7.4	Tunable PMOS Differential Pair OTA Circuit Implemented on Test IC Three . .	173
7.5	Physical Layout of the OTA-C 4th-Order Band Pass Filter . . . . .	174
7.6	Measured Test IC Results: Band Pass Filter Frequency Response and the Effect of Varying the Control Voltage . . . . .	176
7.7	Measured Test IC Results: Response of the BPF to a Variation in $Q$ Control . .	177
7.8	Asynchronous Digital Phase Detector Circuit . . . . .	178
7.9	Comparison of Inverter Circuits with and without Current Limiting Transistors	179
7.10	Physical Layout of the Asynchronous Digital Phase Detector Circuit . . . . .	180
7.11	Simulated Operation of the Digital Phase Detector . . . . .	181
7.12	Charge Pump Circuit . . . . .	182
7.13	Physical Layout of the Charge Pump Circuit . . . . .	184
7.14	Simulated Operation of the Charge Pump Circuit . . . . .	185
7.15	Layout of the Phase Derived BPF Tuning Algorithm . . . . .	186
7.16	Simulated Analogue and Digital Current Consumption when Tuning to 1 kHz .	188
7.17	Measured Test IC Results: Algorithm Tuning Range as a Function of BPF Output Magnitude . . . . .	190
7.18	Measured Test IC Results: Algorithm Tuning Range as a Function of BPF Variable Control Voltage . . . . .	192



7.19	Measured Test IC Results: Algorithm Phase Difference vs Frequency . . . . .	198
7.20	Measured Test IC Results: Algorithm Phase Difference vs Frequency . . . . .	199
7.21	Measured Test IC Results: Algorithm Tuning Accuracy for 1 kHz, 500 Hz, 100 Hz and 50 Hz . . . . .	201
7.22	Measured Test IC Results: Algorithm Tuning Range with Visual Stimulae . . . .	204
A.1	Time-Frequency Plots for Fourier and Wavelet Transforms . . . . .	217
A.2	Comparison of Fourier and Wavelet Transform Frequency Domain Division . .	218
A.3	Dyadic Tree Filterbank . . . . .	219
A.4	Comparison of Dyadic Tree Filterbank with and without Downsampling . . . .	220
A.5	Selected Pixels from the 'Fan' Data Sequence used to Test the Dyadic Tree Algorithm . . . . .	221
A.6	Selected Time and Frequency Domain Signals from Three Stage Dyadic Tree Simulation when Tested with the 20 Hz Negative Luminescence Device . . . .	222
A.7	Selected Time and Frequency Domain Signals from Three Stage Dyadic Tree Simulation when Tested with the 90 Hz Negative Luminescence Device . . . .	223
A.8	Energy within Dyadic Tree Output Bands for Luminescence Flashing at: (a) 20Hz, (b) 90Hz . . . . .	225
A.9	Energy within Dyadic Tree Output Bands for Fan at a Luminescence Frequency of: (a) 20Hz, (b) 90Hz . . . . .	226
A.10	Comparison of Dyadic Tree Output Integrals: (a) Luminescence Device, (b) Fan.	227
A.11	Comparison of Full Tree Output Integrals: (a) Luminescence Device, (b) Fan. .	228
B.1	CMOS Buffer Implementation: Circuit Topology and Measured IC Test Results	230
B.2	Measured Test IC Results-DC Offset of the CMOS Buffer . . . . .	231



---

## List of tables

---

4.1	Transistor Dimensions for Differential Pair OTA . . . . .	105
4.2	Transistor Dimensions for Mirrored OTA . . . . .	106
4.3	Simulated Bias Current Consumption of OTA-C High Pass Filter Structures . .	107
4.4	Transistor Dimensions for the Comparator With and Without Hysteresis . . . .	117
4.5	Simulated Bias Current Consumption of Standard Comparator . . . . .	118
4.6	Pixel Processing Cells On Test IC One . . . . .	124
5.1	Transistor Dimensions for the OTA Structures on IC Two . . . . .	136
5.2	Simulated Current Consumption for the PMOS OTA-C Low Pass Filter Structures	139
5.3	Simulated Current Consumption for the NMOS OTA-C Low Pass Filter Structures	139
5.4	Measured Test IC Results: DC Offset for OTA-C Low Pass Filters . . . . .	145
5.5	Pixel Processing Cells On Test IC One . . . . .	147
5.6	Frequency Accuracy Testing: Parameter Values . . . . .	149
6.1	Transistor Dimensions for the Minipix Algorithm . . . . .	161
6.2	Simulated Current Consumption of the Minipix Algorithm . . . . .	161
6.3	Minipix: Simulated Variation in LPF Cutoff Frequency . . . . .	163
6.4	Simulated Minipix Output Frequency Accuracy with Variable LPF Cutoff . . .	163
6.5	Minipix Frequency Accuracy Testing: Parameter Values . . . . .	164
7.1	Simulated Current Consumption for the 4th-Order OTA-C Band Pass Filter . .	174
7.2	Simulated Current Consumption for Digital Inverter Circuits with and without Current Limiting Transistor . . . . .	180
7.3	Transistor Dimensions for the Charge Pump . . . . .	182
7.4	Simulated Charge Pump Current Consumption, for Typical values of Control Voltage . . . . .	183
7.5	Simulated Average Current Consumption for the Automatically Tuned BPF Al- gorithm, Tuning to 250 Hz . . . . .	187
7.6	Simulated Average Current Consumption for the Automatically Tuned BPF Al- gorithm, Tuning to 1 kHz . . . . .	187
7.7	Parameter Values for the Nine Different Frequency Tuning Range Tests . . . .	188
7.8	Algorithm Tuning Time: Effect of Different Charge Pump Initialisation Voltages	194
7.9	Algorithm Tuning Time: Effect of Charge Pump Control Voltage . . . . .	194
7.10	Algorithm Response to Changes in Input Frequency: Charge Pump Control Equal to 4.39 V . . . . .	195
7.11	Algorithm Response to Changes in Input Frequency: Charge Pump Control Equal to 4.3 V . . . . .	196
7.12	Algorithm Tuning Accuracy . . . . .	200
7.13	Parameter Values for the Five Different LED Frequency Tuning Range Tests . .	203
8.1	Properties of the Automatically Tuned BPF Algorithm . . . . .	208
A.1	Mapping of Pseudo-Frequencies to Real Frequencies . . . . .	221



---

## Acronyms and abbreviations

---

BiCMOS	Bipolar and Complementary Metal Oxide Semiconductor
BPF	Band Pass Filter
AC	Alternating Current
ADC	Analogue to Digital Converter
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CMR	Common Mode Range
DAT	Digital Area Telethermometry
DC	Direct Current
DSP	Digital Signal Processing
FA1-FA4	Frequency Accuracy Tests 1-4
FFT	Fast Fourier Transform
FG	Floating Gate
Gm-C	Transconductance-Capacitor Element
HPF	High Pass Filter
IC	Integrated Circuit
ICMR	Input Common Mode Range
IR	Infra Red
LC	Inductor-Capacitor
LED	Light Emitting Diode
LPF	Low Pass Filter
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
NMOS	N-type Metal Oxide Semiconductor
NO	Nitric Oxide
NTSC	National Television System Committee
OTA	Operational Transconductance Amplifier
OTA-C	Operational Transconductance Amplifier-Capacitor
PC	Personal Computer
PMOS	P-type Metal Oxide Semiconductor



Q	Pole Quality Factor
RC	Resistor-Capacitor
SI	Switched Current
VLSI	Very Large Scale Integration
WFT	Windowed Fourier Transform



---

# Chapter 1

## Introduction

---

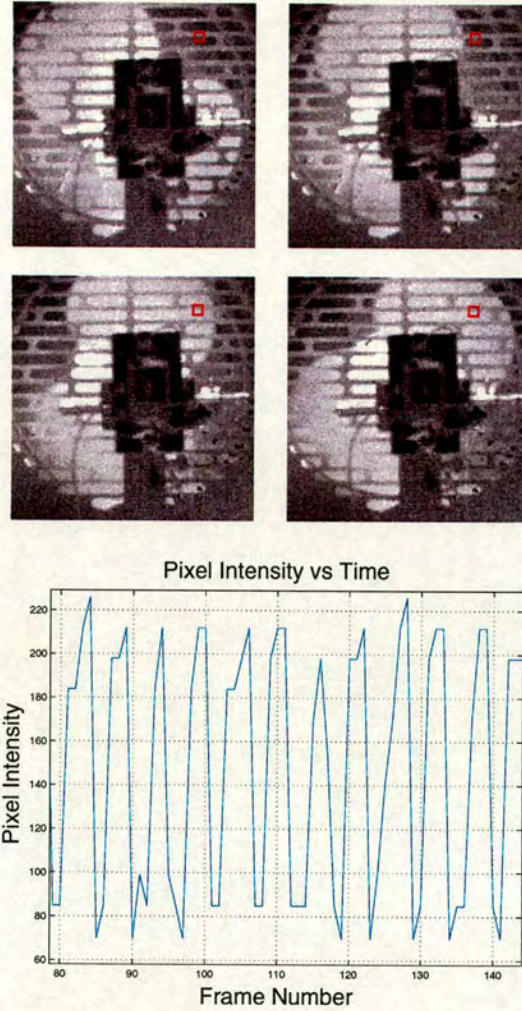
### 1.1 Introduction

The advent of electronic vision systems has recently produced a rival for traditional optical camera systems in the form of digital photography. The ability to capture images digitally and then store them using on-board memory has revolutionised the consumer photography market. However, another advantage of electronic imaging is the ability to process the images dynamically using signal processing techniques. This image processing ability allows signal processing algorithms to be integrated along with the image capturing hardware, producing dedicated image processors. The aim of the research reported in this thesis is the design of a new type of image processor. Instead of capturing image data in the form of a static image or photograph, the idea is to analyse any temporal frequencies present in the scene. Temporal frequencies refer to the time domain variation in pixel intensity, possibly caused by the presence of an object. As an example of a temporal frequency, consider figure 1.1. The diagram contains four consecutive frames of an image sequence depicting a rotating fan blade. Also included is the intensity change versus time for a single pixel from the image sequence as highlighted in red. As the blade appears and disappears from the selected pixel, the intensity increases and decreases accordingly, producing a temporal frequency that corresponds to the rotational frequency of the fan. The function of the image processor described in this thesis is to ascertain not only the fundamental frequency of such temporal variations, but the relative strength of up to the first four harmonics. Effectively, each pixel in the two dimensional image processor array should perform a Fourier style decomposition of the incident light intensity into its constituent frequency components. For the purposes of this thesis, such a Fourier decomposition will be referred to as a *frequency signature*.

### 1.2 Motivation

The main emphasis for the research reported in this thesis stemmed from a collaboration with QinetiQ UK Ltd, formerly DERA. While the final applications are classified, research per-





**Figure 1.1:** *Temporal Frequency: Four consecutive data frames and the corresponding time domain intensity variation from the highlighted pixel location.*

formed by QinetiQ pointed towards the use of temporal frequency signatures for the detection and classification of objects present in a scene. As such, QinetiQ provided financial assistance to investigate methods of developing a system capable of extracting frequency signatures from visual data.

The sponsor's application for the system involved the extraction of frequency signatures for the detection and classification of objects within the field of vision. As depicted in figure 1.1, the fundamental frequency of an object can be calculated from the intensity variation. However, if two objects share similar fundamental frequencies then more detail is required to distinguish between them. For this reason, the frequency signature of the object must be

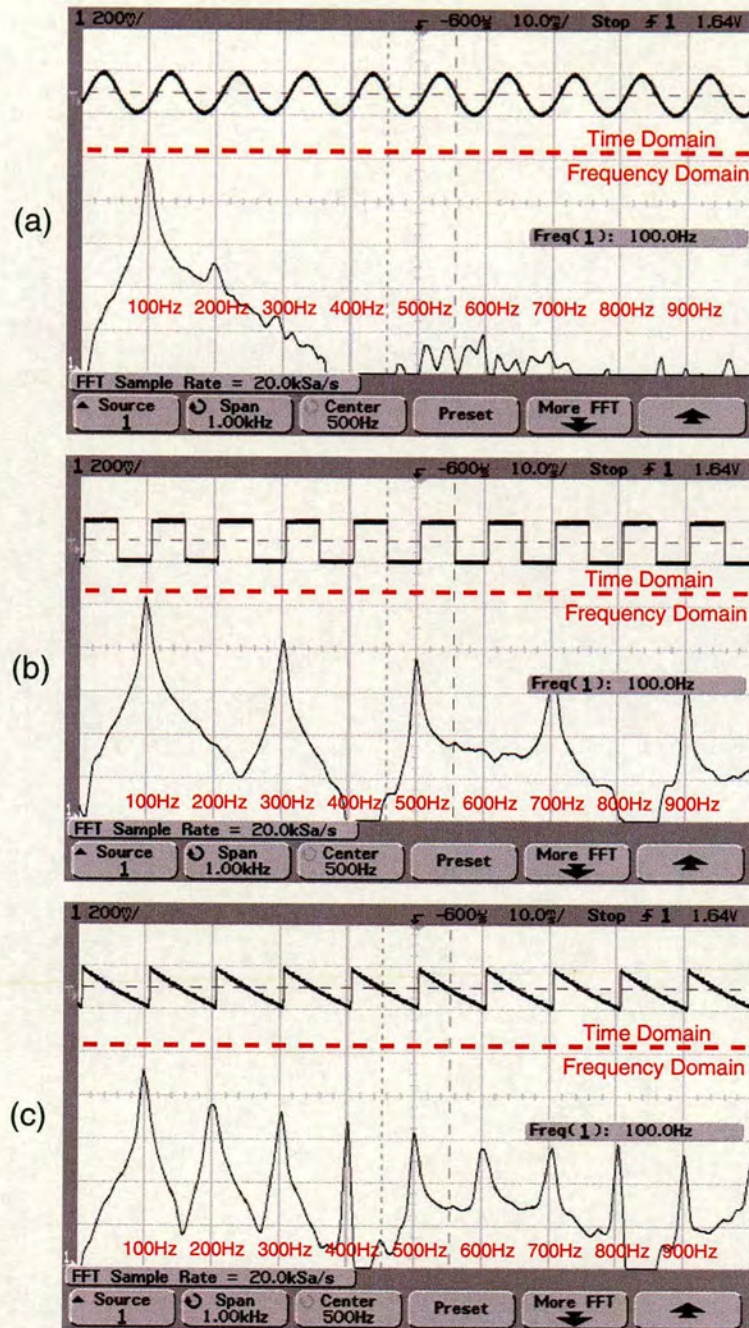


extracted. As an example, consider the three screen-shots depicted in figure 1.2. Each screen-shot shows the time and frequency domain representations from the output of single logarithmic photocircuit, designed as part of this research. Such a circuit converts incident light intensity into a corresponding voltage signal. The pixel was exposed to a near IR light emitting diode controlled by a function generator, allowing different signals to modulate the intensity of the illumination. It is clear from figure 1.2 that despite having similar fundamental frequencies of 100 Hz, the sine wave, square wave and sawtooth modulating signals can be distinguished between by analysing the frequency domain representations. Finding only the fundamental frequency would not provide sufficient information to tell the three modulating signals apart. However, a Fourier decomposition of the intensity variation highlights the differences clearly. As such, if the three different modulating signals represent different objects then it is clear that extracting the underlying frequency signature aids successful discrimination, despite the similarity in fundamental frequency.

As a more practically relevant example, consider figure 1.3 which contains individual frames from two different image sequences. Figure 1.3(a) depicts a fan while (b) shows a propeller plane with two engines. In each data sequence, two pixels were selected and the variation in intensity transformed into the frequency domain using a software-based fast Fourier transform. The pixels were selected on the basis that they experience a temporal frequency of interest and are highlighted with red and green blocks in figure 1.3(a) and (b). The frequency domain representation of the intensity variation for the fan's red and green pixels can be seen in figure 1.3(c) and (e) respectively. Similarly, those for the propeller plane's red and green pixels can be seen in figure 1.3(d) and (f). It is clear that the same object produces the same underlying frequency content, no matter which pixel is analysed. The two pixels corresponding to the fan data sequence show very similar frequency signatures, while the same is true for the propeller plane sequence. The sponsor company asserts that by analysing any frequency signatures present in a scene, it is possible to ascertain which objects are producing them. As such, it is the extraction of these underlying frequency signatures that is the ultimate aim for the system described in this thesis.

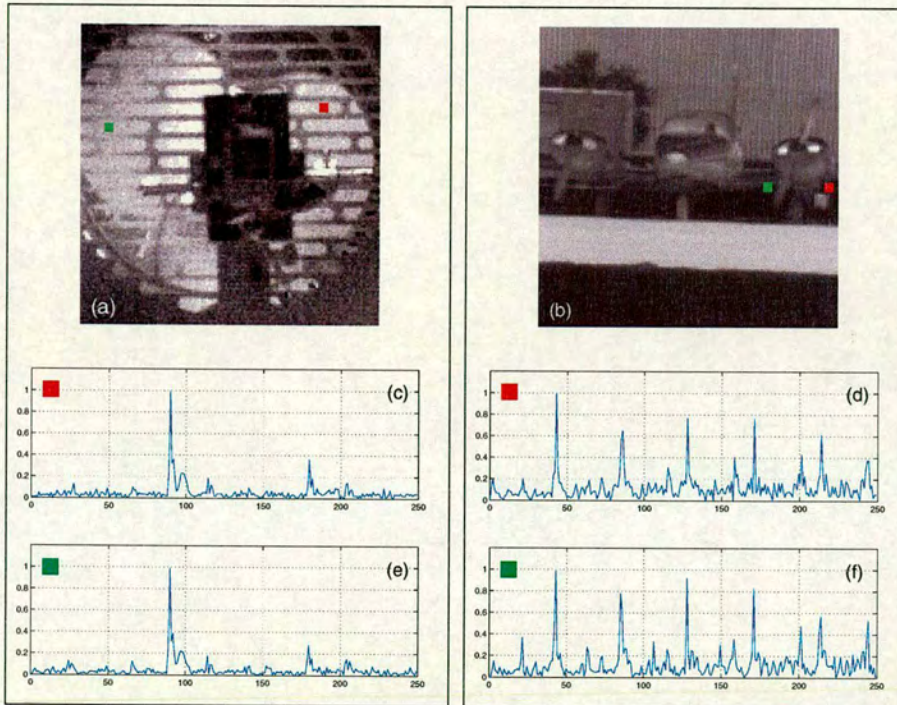
In addition to the detection and classification of objects in the field of vision, other applications for such an image processor include the potential detection of cancerous breasts. A recent paper by Anbar et al[1], investigated the use of dynamic area telethermometry for the objective diagnosis of breast cancer. The technique involves computing the fast Fourier transform (FFT) of





**Figure 1.2:** Temporal and Frequency Domain representations of Temporal Frequencies: (a) Sine Wave, (b) Square Wave, (c) Sawtooth Wave. Despite having similar fundamental frequencies, the three different stimuli can be distinguished by analysing the frequency domain representations.





**Figure 1.3:** *Comparison of Object Frequency Signatures: (a) Single frame from the 'fan' data sequence, (b) Single frame from the 'propeller plane' data sequence, Frequency domain representation of intensity variation for; (c) fan sequence - red pixel, (d) propeller plane - red pixel, (e) fan sequence - green pixel, (f) propeller plane - green pixel. Despite the differing locations of the chosen pixel, the underlying frequency signatures remain similar.*



time series of the average temperatures of areas of the breast. One of the effects of a cancerous tumour is to produce nitric oxide (NO), which among other things causes excessive widening of the blood vessels in the region. Under normal conditions, blood flow is modulated by neurohumoral control which produces temporally varying picomolar concentrations of NO inside blood vessels[1]. Vascular tone is also modulated by hydrodynamic cardiogenic pulses. The two effects when combined produce a frequency spectrum of blood flow in the healthy breast in the range of low mHz to  $>10$  Hz. The presence of excess concentrations of NO caused by the formation of a cancerous tumour causes blood vessels to dilate, thus stopping their response to neurohumoral modulation and effectively changing the frequency response of the blood flow. The effect of these changes can be quantitatively assessed by dynamic infrared imaging, with the results converted to the frequency domain for analysis. The research reported the use of a technique termed digital area telethermometry (DAT), basically a series of infra-red images taken at a frequency of 100 images per second to avoid aliasing at the frequencies of interest. When sampling signals, the frequency content of the signal is present at its original location in the frequency domain, but also repeats at integer multiples of the sampling frequency. If the sampling frequency is too close to the maximum frequency present in the original signal, the 'copy' of its frequency response may *fold-down* into its frequency spectrum, violating the integrity of the signal. This is termed *aliasing*, and can be removed by sampling according to Nyquist's criterion, at no less than twice the largest frequency present in the original signal[2].

The time domain signals were then converted to the frequency domain using a computer based FFT. The sensor described in this thesis operates in the visible light spectrum, yet it could be attached to an infra-red detector array using flip chip bonding techniques[3]. The sensor would then accept signals corresponding to the infra-red data it is exposed to, before performing the frequency signature extraction. Anbar et al[1] suggest that their analysis techniques provide impressive sensitivity and specificity regarding the diagnosis of breast cancer. The authors claim that the technique could be a forerunner of 21st century medical diagnostic devices, when data is remotely sensed before being analysed by computers. The sensor described in this thesis could be integrated with an IR detector to produce a step in this direction, with a real time analysis of the frequency spectrum of blood flow in the breast.

Another potential application for such an image-processor include non-invasive fault analysis for rotating machinery, where the harmonic content conveys the condition of the appliance. Although no research has been performed on this subject, analysing the harmonic frequency



content of a rotating machine may give some insight into the presence or future likelihood of a mechanical failure. Current techniques for fault analysis include monitoring the frequency response of vibrations[4, 5].

### **1.3 Implementation Issues**

While not providing a detailed list of specifications, the sponsor company had several general design criterion which served to shape the development of the research described in this thesis. An emphasis was placed on a low power implementation, with the proposed system being able to operate for reasonable periods of time from a battery supply. A compact, low area solution was also favoured, with a low area implementation having a positive effect on overall economic cost. Another important consideration was the desire for real time processing from the system. The time delay between the appearance of an object and the capturing of its frequency signature had to be kept to a minimum. In addition, the system had to be able to operate at low frequency, due to the nature of the objects being observed with the sensor. A figure of between 1 Hz and 10 Hz was specified as the minimum range of fundamental frequencies that had to be detectable, with 10 kHz the maximum. For all these reasons, a decision to concentrate on hardware realisation of both the image sensor and the signal processing was made by the sponsoring company.

The next two sections detail the system and circuit level design decisions that were made and the reasoning behind those decisions. Although they are separated here, in reality both were considered simultaneously to produce the simplest and most elegant solution to the problem.

#### **1.3.1 System Level Design Decisions**

Essentially given a free reign at both system and circuit level, the initial phase of the research described in this thesis was to identify system level algorithms to perform the necessary function while still being practically realisable in hardware. An obvious solution to the problem involves the coupling of a commercially available image sensor with some form of digital signal processing circuitry. Signals from each pixel would be sampled and stored in memory before being multiplexed in time to the DSP block. Such a brute-strength approach may produce excellent results, but will be both area and power intensive, as well as failing the real-time processing requirement. An alternative approach to hardware based image processing that has



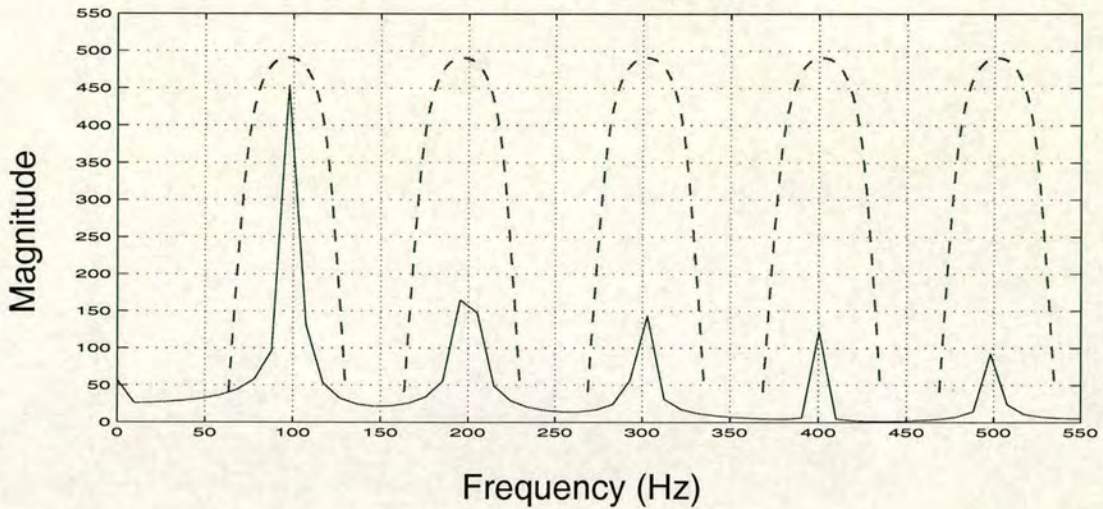
attracted recent academic interest involves integrating imaging and signal processing circuitry on the same substrate. From an engineering view-point, the integration of light sensitive elements with dedicated processing may provide elegant solutions to practical problems. Such *focal-plane processing* techniques employ pixels that include not only structures to convert the incident illumination to an electrical signal, but processors to enhance/suppress certain elements of that signal. Contrasting this with a standard engineering approach, where there is no interaction between image capture (CMOS camera) and image processing (DSP/PC), suggests a number of possible advantages for the implementation of such *smart sensors*[6]:

- Speed: the ability to process in parallel and remove unwanted data at the pixel level reduces processing and communication bottlenecks
- Size: focal plane processing may allow for elegant, single-chip solutions to image processing problems, where the alternative involves separate, bulky, power intensive processing steps
- Power consumption: Due to the large number of pixel level processors, many of the focal plane processors reported in the literature employ circuits biased in the weak inversion region of operation. This allows for extremely small bias currents which in turn manifests as low power consumption

For the application and design criterion of the research described in this thesis, a focal plane processing technique seems particularly relevant. By incorporating some signal processing at the pixel level, the potential advantages of a focal plane approach are closely matched to the needs of the project.

Having decided on implementing some form of pixel level pre-processing, development of a novel algorithm to extract the temporal frequency signature could begin. Several alternatives were developed and simulated in software, the details of which can be found in chapter three. All algorithms were conceived with circuit level realisations for each processing step, to allow simple conversion into hardware. The chosen algorithm relied on first finding the fundamental frequency of the intensity variation using pixel level processing. This information is then used to place tunable band pass filters at integer multiples of the fundamental, creating a pseudo Fourier decomposition of the incident light intensity. The process is depicted pictorially in figure 1.4. The output from each of the band pass filters gives some marker of the energy within that particular frequency band.





**Figure 1.4:** *Pseudo Fourier Processor: The adopted algorithm first finds the fundamental frequency of any intensity variation. A tunable band pass filter is then placed at this frequency and the first four integer multiples, extracting the frequency signature.*

### 1.3.2 Circuit Level Design Decisions

The adopted algorithm was selected on the strength of its performance when compared to the alternatives, the details of which can be found in chapter three. However, another advantage was the ease with which each processing step could be translated to a circuit level equivalent. However, there were still a number of circuit level decisions to be made, based on both the original criteria specified by QinetiQ and the requirements of the algorithm.

Focal plane processing techniques require that the light sensitive elements are integrated on the same substrate as subsequent signal processing circuitry. The two dominant types of electronic imager systems are charge coupled device (CCD) and CMOS. Vision systems implemented in CMOS technology have created recent interest both industrially and academically[7, 8]. Despite the maturity of CCD imagers, an advantage of CMOS in this application is the ability to integrate sensors and processing on the same silicon substrate. Vision sensors implemented in CMOS processes can also be cheaper than CCD equivalents, and can offer advantages in power consumption[7].

The first decision to be made was the choice between an analogue or digital CMOS implementation. With focal plane processing, the area available for each pixel processor has to be minimised to allow practically useful resolutions. A digital implementation would require ana-



logue to digital converters, either in each pixel or at the side of the array. Placing a sufficiently accurate data converter in each pixel while producing a realistically sized pixel cell seemed infeasible. A single ADC for each column or row of the imager may be more realistic, but would require sampling of the signals from the pixel array, possibly producing a data bottleneck. In addition, such sampling would require a clock signal to be supplied to each pixel, producing a possible inter-connect problem. Such a clock may also introduce sampling noise to the system. The majority of the focal plane image processing systems described in the literature make use of analogue circuit techniques to avoid cumbersome data conversion circuitry. In addition, the extremely low bias currents required when biasing transistors in the weak inversion region of operation allow designers to develop low power image processors. Subthreshold transistors suffer from poor matching characteristics, which limits their usefulness in certain applications. However, it was felt that such problems could be addressed at the algorithm development stage, by choosing circuit structures and techniques for which mismatch posed fewer problems. QinetiQ's requirement for a low power system coupled with the potential disadvantages of a digital implementation led the research to focus on analogue signal processing techniques using transistors biased in the weak inversion region of operation.

Analogue signal processing techniques can be split into two separate categories, continuous time and discrete time. Discrete time techniques such as switched capacitor and switched current allow the design of accurate filter time constants, as they are controlled by a digital clock. This also means that tunable filters can be implemented, simply by varying the clocking frequency. However, as previously mentioned, supplying clocks to each pixel could prove costly in terms of area, particularly with the two-phase, non-overlapping variety required for switched capacitor circuitry. Another potential disadvantage of sampled data analogue signal processing techniques is the need for accurate analogue memory. The signals would require to be sampled and stored before processing, adding to the size of the system. In addition, storing analogue values accurately is difficult, which may produce problems regarding the robustness of the algorithm. A continuous time approach to the problem may suffer from poorly controlled filter time constants, particularly at the low frequencies of interest. However, it was felt that it may offer some advantage in terms of area of implementation as there is no need for memory elements. Another potential advantage of adopting a continuous time approach stems from the very function of the image processor. The aim is to extract frequency signatures as accurately as possible. Any sampling of the pixel's photocurrent or photovoltage may introduce aliased frequencies, which could conceivably fold down into the frequency band of interest unless re-



moved with an anti-aliasing filter. The inclusion of such a filter would add to the silicon area consumed by a discrete-time implementation.

For all these reasons, emphasis was placed on developing an analogue, continuous time focal plane processor with circuits biased in the weak-inversion region of operation.

## 1.4 Neuromorphic Approach to Image Processing

Real-time image processing is a computationally intensive task. The high-density of visual information in a typical scene, coupled with the large range of illumination levels produces huge amounts of data. As an example, a one second long uncompressed NTSC video stream creates approximately 22 MB of data[9]. A conventional approach to the implementation of real-time image processing algorithms may involve coupling a CCD camera, to capture the image data, with a high-end computer to perform the image processing. Such a technique produces data rates that only the most advanced computer systems can process[10], producing an expensive solution in terms of physical space, power consumption and economic cost. A standard video camera captures an image approximately 30 times in a second, which in itself may be unacceptable delay for some motion control algorithms[11].

However, the simplest insects, with brains the size of grains of rice, can successfully analyse visual data extremely rapidly to avoid obstacles[9]. The reasons for this lie in the differing architectures employed by the 'engineering' solution and its biological equivalent.

The conventional approach described above uses a camera to convert the inherently analogue signals into a digital equivalent. These are then passed to a computer in serial format for processing. In contrast, a biological motion processing algorithm utilises massive parallelism, such that data acquisition and processing can be performed continuously and that the whole scene/image can be continuously monitored for events. There is also much closer integration of light sensitive elements with image processing elements in the form of local operations. This allows such 'early-vision' tasks as adaptation to ambient light levels (temporal processing)[12], edge enhancement (spatial filtering)[13, 14] and motion detection (spatio-temporal processing)[15]. Such techniques produce pre-processed images, thus reducing the amount of information required by further stages and speeding up the overall processing time.

This ambiguity between the complexity of state of the art engineering approaches to motion



processing compared with biological alternatives has created academic interest in modelling the latter.

Although researchers had previously attempted to model biological vision algorithms using discrete electronics[6], Carver Mead at Caltech was among the original pioneers when he realised the potential in using analogue CMOS VLSI to model biological data processing[13]. The disparity in processing power between digital computers and even the simplest animal brains led him to the conclusion that a new, more powerful and efficient form of computation can be instigated from the study of biology[16]. In doing so, he created a new paradigm for analogue circuit design, termed *neuromorphic engineering*. The idea was to create CMOS chips that took biological processing structures as inspiration, in the process both learning more about biology as well as creating radical new structures for the solution of common engineering problems. The rationale behind this new approach was the similarity between neuronal wetware in the brain and CMOS hardware when transistors are biased in the subthreshold mode of operation[17]. Both utilise continuous time variables to convey information from one processing stage to the next. The advent of CMOS VLSI provides a two dimensional substrate onto which millions of transistors can be integrated, similar to the massive parallelisation present in neural structures. In addition, this substrate is as limited by the constraints of power consumption, inter-connectivity and precision as biological wetware, thus allowing the design of realistic models which may provide extra insight into the actual workings of the biological reality[18]. The fact that the underlying computational elements in neural computation exhibit low precision, poor reliability and low noise maps directly to the problems of matching and noise with transistors. Yet, despite this inherent poor quality, the structures and processing techniques used to organise these low level operators in the brain perform processing orders of magnitude more efficiently than current digital computation[16]. Essentially, the study of neuromorphics suggests a completely new approach to computation. Using the analogue computation in the brain as inspiration, it may be possible to develop extremely efficient algorithms despite the inherent limitations and lack of precision in CMOS transistors[19].

However, there are underlying problems involved in modelling the complexity of biological wetware with CMOS hardware. While it is true that the imprecision in matching of transistors is similar to that of individual neurons, the huge number of inter-connections within the brain allows complex data to be smoothed and averaged, compensating for the imprecision in the processing neurons. Indeed, possibly the greatest barrier to successful neuromorphic im-



plementations is in modelling the inter-connect present in the brain. According to Koch[20], in each cubic centimeter of the brain there are 100,000 cells and two kilometres of wiring, allowing each neuron to communicate with up to 10,000 others. Such dense inter-connect is not yet possible in CMOS, meaning the imprecision of matching in subthreshold transistors is more prevalent from a system level perspective. The fact that biological neurons exist in a three dimensional framework, while standard CMOS processes are strictly two dimensional is another severe limitation to neuromorphic implementations. Another potential problem with the use of CMOS technology in implementing neural structures is the need to store analogue values or weights for extended periods of time. One relatively recent approach involves the use of floating gate devices. FG CMOS devices have an electrically isolated gate electrode which is manipulated to either add or subtract electrons, effectively changing the threshold voltage of the device. It can be considered as an additional voltage source, capable of adding to or subtracting from the existing overdrive voltage, allowing manipulation of the  $I_{ds}$ - $V_{gs}$  curves. As the gate is floating, the charge on the gate-source junction should remain constant, meaning such a device can be used as non-volatile analogue memory, holding its value for long periods of time[16, 21, 22]. However, the use of floating gate CMOS devices to store analogue values for long periods of time is a contentious issue, with some research suggesting that the stored charge degrades over time, with threshold voltages varying by as much as 1V over a four month period[23].

It is clear that there are many obstacles to be overcome before the potential benefits of adopting a neuromorphic approach to circuit design are realised. Despite almost 15 years of research, very few commercial products based on a neuromorphic approach have been released, with one notable exception being Logitech's optical mouse, based on a design by Arreguit and van Schaik[24]. While it is true that the lack of commercial success may be due to the maturity and therefore bankability of standard engineering techniques, some of the blame has to be aimed at the limited performance of neuromorphics reported in the literature. As the Logitech example highlights, the best applications for neuromorphic design principles are low precision tasks where the inherent problems of offset and noise are less important than power consumption or area of implementation. It is clear that neuromorphics will probably not compete with digital techniques in most applications. However, niche markets such as 'bionic' implants or low power, low cost image processors may yet benefit from such an approach. Recent efforts by Toumaz electronics in implantable electronic cochlea and Iguana Robotics collaboration with Ralph-Etienne Cummings suggest that some areas of industry are beginning to take notice



of the field[25]. As such, the future of neuromorphic circuit design in its current format is in application specific sensors or data processors, where energy efficiency is of paramount performance, such as the sensor described in this thesis.

## 1.5 Engineering Solution within Neuromorphic Framework

Many of the potential strengths of adopting a neuromorphic approach to the design of image processors map directly to the requirements of the research sponsored by QinetiQ. Both use analogue signal processing and allow integration of light sensing elements with signal processing circuitry. A neuromorphic approach satisfies the requirement for continuous time operation, with focal-plane processing reducing the information required by subsequent processing stages. Low power constraints are also met by using transistors biased in the subthreshold region of operation.

However, the major stumbling block regarding a fully neuromorphic approach to the solution of this particular problem is the lack of a biological equivalent to model. Neuromorphics can be split into two main camps; those who attempt to enhance knowledge of biology by creating models whose constituent parts are subject to the same physical limitations as nature[18], and those who wish to create subtle, efficient solutions to engineering problems. In effect, the field can be split into *Scientific* and *Engineering* approaches. However, there are no biological entities whose vision system is concerned with analysing temporal frequency at the expense of all other data, ruling out a direct neuromorphic approach. However, the correlation between the key advantages of such an approach and the requirements for the sensor to be designed lead to an amalgamation of neuromorphic and engineering approaches. The idea is to take the neuromorphic framework for vision sensors but apply an engineering solution, thus benefiting from the advantages of each. As such, a review of papers that use low power, continuous time, analogue focal plane techniques was undertaken. The emphasis is on some form of neuromorphic focal plane technique, be it 'scientific' or 'engineering' in conception, as this is what best suits the adopted approach. The results of the literature review can be found in chapter two.



## 1.6 Contributions

The development of a CMOS image-processor for the extraction of frequency signatures is itself a novel concept. The research described in this thesis began with software simulations of potential algorithms to ascertain which would both produce the most robust results and more importantly, be the simplest to implement in CMOS hardware. The algorithm developed and tested in software is novel. Based on results from those simulations, three test chips were developed, each building on any lessons learnt from previous implementations. Again, each of the test ICs contain novel circuits at the system level. The first test chip contains several individual pixel cells, capable of producing a pulse train whose frequency corresponds directly to the fundamental frequency of any incident light change. The second test chip contains similar elements, although an improved self-referencing scheme has been implemented to reduce manual input. The third and final test chip contains two versions of a novel algorithm for tuning a band pass filter's centre frequency to the fundamental frequency of the incident light. The algorithm uses a self-tuning system to automatically place the band pass filter onto the fundamental frequency of the input signal. Novelty in this research stems from the algorithm employed, principally its design, simulation and hardware implementation.

## 1.7 Structure

The structure of this thesis is as follows:

- Chapter Two provides an introduction to previous research in CMOS image sensors incorporating focal plane processing. Both neuromorphic and engineering approaches to focal plane processing are discussed.
- Chapter Three introduces the early algorithmic work, with MATLAB simulation results to assess the strengths and weaknesses of several potential candidates.
- Chapter Four introduces the first test chip design, with structures that produce pulses corresponding to the fundamental frequency of the incident light. Results from chip testing are included.
- Chapter Five introduces a second test chip, which once again creates pulses that encode the fundamental frequency, this time using an improved, self-referencing algorithm



- Chapter Six presents the *minipix* algorithm, included on the third and final test IC. It is essentially a miniaturised version of the self-referencing system described in chapter five.
- Chapter Seven introduces the phase-derived feedback algorithm, also included on the third IC. It is capable of automatically tuning a band pass filter to the incident light's fundamental frequency
- Chapter Eight summarises the work, presents critical evaluation and suggestions for future work as a conclusion to this thesis.



---

# Chapter 2

## Smart Sensors Incorporating Focal Plane Processing: Literature Review

---

### 2.1 Introduction

As the topic under investigation in this thesis is novel, there are few directly relevant papers aimed at CMOS imagers analysing scenes for temporal frequency information. Nevertheless, as an engineering approach to neuromorphic image processing is to be pursued, many relevant papers incorporating focal plane processing to perform other image processing tasks exist. The review of relevant literature will be split into three different sections. The first section deals with focal-plane approaches to the spatial processing of image data. The second section is concerned with focal-plane approaches to temporal processing, while the third deals with spatio-temporal processing. Each category is split into **Scientific** and **Engineering** approaches to Neuromorphic system implementation, with the core similarity being the type of processing implemented and the use of focal plane techniques.

### 2.2 Spatial Processors

The spatial processing of image data refers to techniques that can be performed on static, single frames, without the inclusion of any temporal aspects. For instance, de-blurring, edge enhancement and magnification are just some examples of spatial processing available with standard image processing software tools.

Dedicated CMOS implementations of spatial processing incorporating focal plane processing have been reported in the literature, with aims such as edge-enhancement, dynamic range reduction and object orientation calculation. Generally speaking, the aim of such dedicated spatial processing circuitry is to enhance certain elements from a scene while suppressing others, reducing the data provided to subsequent processing stages. In the following sections, the different techniques employed by both scientific and more standard engineering approaches to neuromorphic processing are explored.



### 2.2.1 Silicon Retina: 'Scientific' Implementations

Some of the original work in neuromorphic vision systems concerned modelling the retina's spatial processing characteristics. For instance, image smoothing, dynamic range reduction, contrast enhancement and feature extraction/orientation can all be considered examples of retinal spatial processing[6].

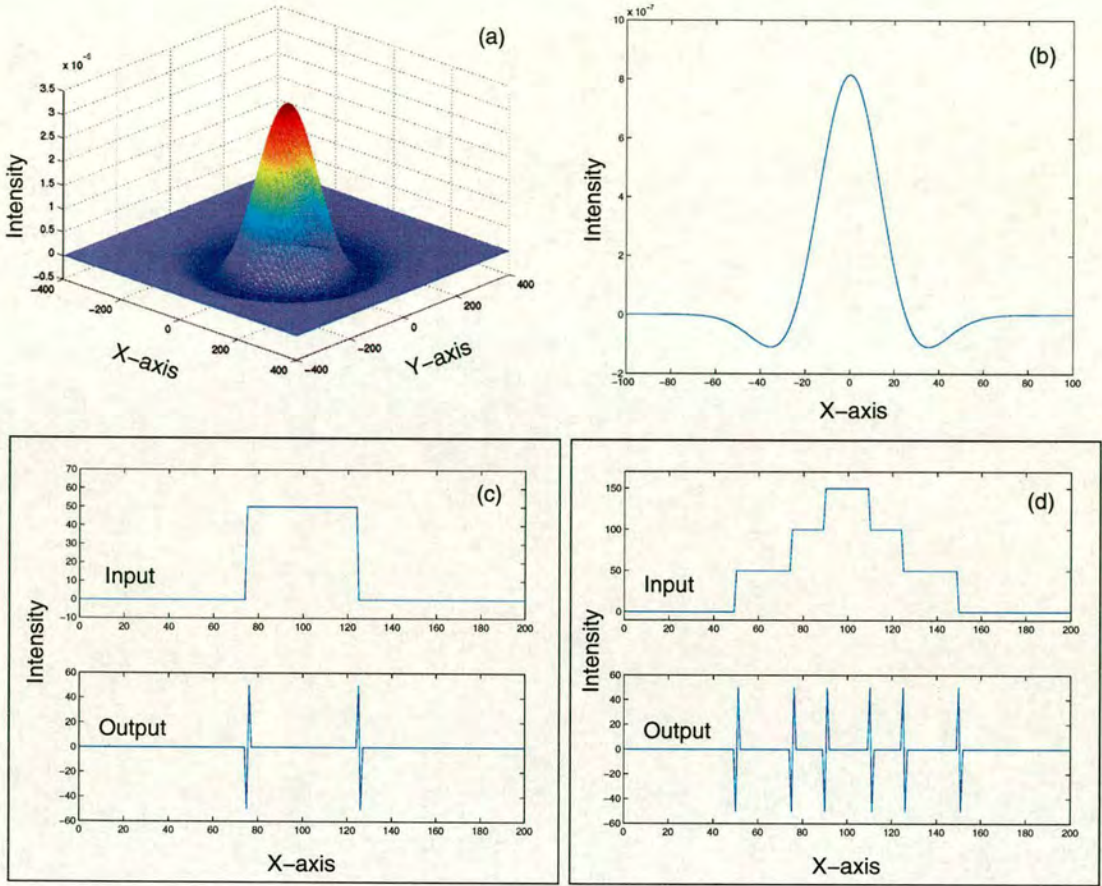
There are many different approaches to such neuromorphic spatial processing, some of which employ a *mexican hat* operator as the underlying convolution kernel. A two dimensional representation can be seen in figure 2.1(a), with the more familiar one dimensional version in figure 2.1(b). The kernel depicted here was constructed with the 'Laplacian of Gaussian' approximation[26], seen in equation 2.1.

$$LoG(x, y) = \nabla^2 G(x, y) = -\frac{1}{\pi\alpha^4} \left[ 1 - \frac{x^2 + y^2}{2\alpha^2} \right] e^{-\frac{x^2 + y^2}{2\alpha^2}} \quad (2.1)$$

The idea is that pixels are processed using information from neighbouring pixels. Those within the central region are given higher weighting, while those towards the outskirts are diminished. This form of excitation and inhibition can be used to enhance spatial gradients in the scene, as depicted in figure 2.1(c) and (d). Notice also that the output is independent of the actual intensity level and is only concerned with the edges or events in the scene. For example figure 2.1(d) shows that despite the input varying from 0 to 150 units of intensity, the output remains centred on zero except when a sudden intensity gradient is present. This has the effect of reducing the dynamic range of the input data, simplifying the task of further processing stages. It is clear that such a convolution kernel spatially high pass filters the image. The Laplacian of Gaussian is just one of the approximations to the 'mexican hat' kernel. Others include subtraction or division of incident illumination from a spatial average, difference of two Gaussians, Gabor functions and both linear and non-linear lateral inhibition. The last two can be considered plausible biological models for the retina, while the former are mathematical models used in software and certain hardware implementations.

With the useful characteristics mentioned above, many researchers have attempted to integrate the spatial processing of retinal systems onto vision chips. As previously described, there are two differing approaches to the implementation of so called *silicon retinas*, 'scientific', where the circuit blocks emulate biological processing steps and 'engineering', where conventional





**Figure 2.1:** Retinal Approach to Spatial Filtering: (a) 2D 'Mexican Hat' Kernel (Laplacian of Gaussian), (b) 1D version, (c) and (d) Contrast enhancement with edge detection.

circuit techniques are used yet the end result can be compared with biology.

Mead's model of early visual processing was one of the original implementations[14]. The system uses a resistive grid to create a smoothed version of the incident image. This is then subtracted from the actual intensity present at each pixel, re-creating the operation of the photoreceptors, horizontal and bipolar cells in retinal processing. The outputs from the silicon implementation correspond directly to those in the biological equivalent. The photoreceptors are implemented with logarithmic photodetectors, while amplifiers are used to produce an output proportional to the difference between the incident light and the averaged version. One of the problems with this implementation is its sensitivity to amplifier offset, causing many pixels to be either completely on or off. An improved version using floating gate MOSFETs to correct for mismatch has also been developed[27].



Another example of spatial processing is the silicon retina designed by Boahen and Andreou[28]. The design models the shunting inhibition found in the distal retina by using two smoothing networks, for the interaction between horizontal cells and cone cells in the photoreceptor. Both smoothing networks have different conductive properties, with the horizontal cells allowing signals to propagate over a much larger distance than the cone cells. A cone cell's activity increases when it is exposed to incident light, exciting the surrounding horizontal cells. These respond by trying to impede the cone cell, using inhibition. The diffusive networks are set up in such a way that the excitation received by cones close to the incident light is stronger than the inhibition from the horizontal cells. However, further away from the active cone cell the inhibition begins to dominate, producing the overall centre-surround 'mexican-hat' kernel seen before. The implementation uses subthreshold, current mode circuitry to produce a compact, low power sensor. However, a problem with the implementation is the fact that the size of the centre-surround receptive field varies with the incident light level. A second generation implementation with 48,000 pixels was designed[29]. Despite the large imager array, a power consumption of only 50 mW was reported from a 5 V supply.

A more recent silicon retina implementation uses two bipolar phototransistors to model the photoreceptors and spatial smoothing of the horizontal cells[30, 31]. Instead of implementing a relatively large resistive grid to achieve spatial smoothing, an array of phototransistors with common base region is realised. When light is incident on the imager, excess carriers are generated and diffuse out, producing a current that decays logarithmically with distance. This implementation allows for compact pixel size, with each being  $60\text{ }\mu\text{m}$  by  $60\text{ }\mu\text{m}$  when designed in a  $0.8\text{ }\mu\text{m}$  process. An improved version, where the size of the centre surround receptive field is tunable has also been implemented[32]. By placing MOSFETs between the phototransistor elements, it is possible to control the amount of spatial averaging. A 64 by 64 array was designed in a  $0.5\text{ }\mu\text{m}$  process, with each pixel taking  $45\text{ }\mu\text{m}^2$ . The power dissipation of the entire imager varies from 3 mW to 30 mW depending on the incident light level. The authors claim that such a system can be implemented in both BiCMOS and CMOS processes using parasitic phototransistors. In the latter case, mismatch in the parasitic elements may inhibit the performance.

A similar technique using photodiodes instead of phototransistors was implemented by Ikeda et al[33, 34]. The aim is to develop an edge extraction imager by modelling the three most important neuronal structures in the retina, the photoreceptor, horizontal and bipolar cells. A



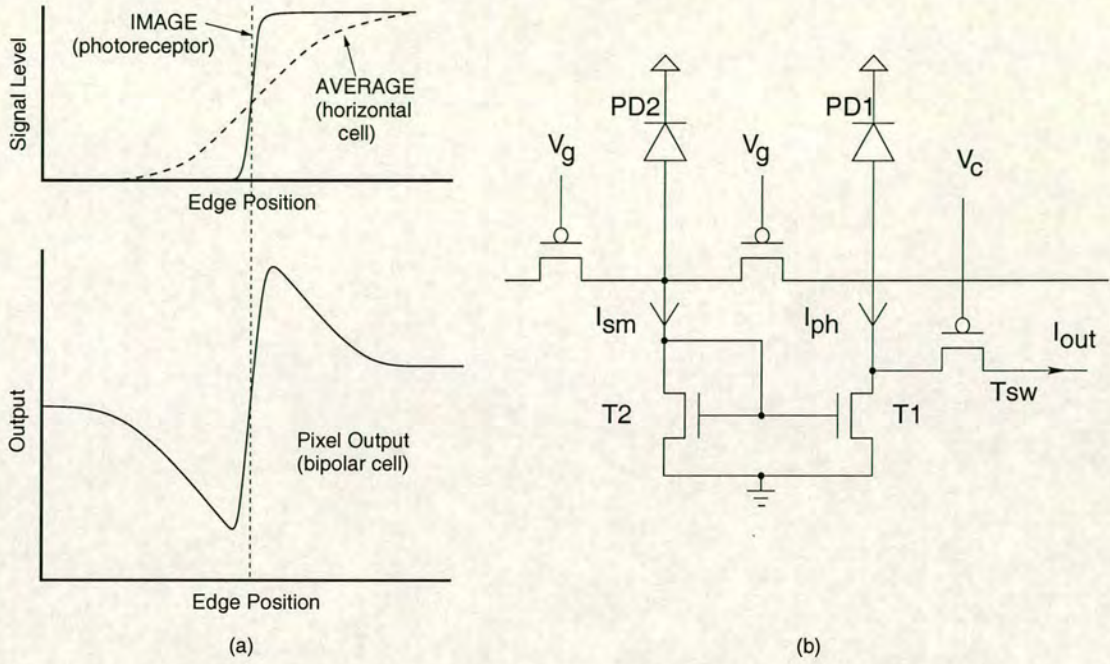
simplified version of the processing performed by these cells can be found in figure 2.2, together with the adopted circuit techniques. As a spatial gradient moves across the surface of the retina, the difference between the actual intensity change (photoreceptor cell) and a locally smoothed average (horizontal cell) is computed by the bipolar cell. The zero-crossings of the bipolar cell's output give the location of the edges. Ikeda et al use only two photodiodes and three MOS transistors in each pixel cell to model the edge enhancement capabilities of the retina. A projected pixel size when designed in a  $0.5\text{ }\mu\text{m}$  process is  $12\text{ }\mu\text{m}$  by  $14\text{ }\mu\text{m}$ , allowing for high density imagers. Each pixel contains an isolated photodiode (PD1 in figure 2.2) to model the photoreceptor. The second photoreceptor (PD2) is connected to the equivalent in the neighbouring pixels by the transistors controlled by  $V_g$ , producing a spatially smoothed version of the intensity change. The amount of spatial averaging can be controlled by varying the gate voltage of the linking transistors, effectively varying the size of the averaging area and therefore the sensitivity of the system. A current mirror incorporating T1 and T2 is then used to calculate the difference between the isolated and connected photodiodes, mimicking the bipolar cell.

Potential problems with such a system include mismatch between the two photodiodes, which may produce different photocurrents despite exposure to the same intensity. A current mode approach such as this may also suffer from inaccuracies in the subtraction calculation performed by the current mirror, particularly if the photocurrents bias transistors T1 and T2 in the subthreshold region of operation.

Other research on implementations of silicon retina includes that by Kameda et al. A model based on regularisation theory was implemented using analogue VLSI[35]. The model relies on the fact that the responses of retinal photoreceptors and horizontal cells are graded potentials with respect to light. Such responses can be mimicked with analogue network models. The VLSI implementation of the model uses two resistive grids of variable conductivity, similar to that used by Boahen and Andreou. However, the employed photocircuit samples the photocurrent at discrete time intervals, meaning it is not a continuous time implementation. Although only a one dimensional implementation, the circuit includes buffers and sample and hold circuitry in an effort to compensate for the mismatch and fixed pattern noise attributed to sampled-data photocircuits. Adding such circuitry will increase the size of the implementation, yet the achieved increase in accuracy means it could potentially be employed in applications such as retinal implants[36] and robot vision[37].

Another interesting implementation of a silicon retina for spatial processing was developed by





**Figure 2.2:** Retinal Processing for Edge Extraction: (a) Operation of the photoreceptors, horizontal and bipolar cells within the biological retina. As an edge moves across the surface, the photoreceptors produce an output corresponding to the actual intensity variation, while a locally smoothed average is computed by the horizontal cells. The difference between the two is calculated by the bipolar cells, enhancing the location of the edge. (b) Compact circuit for modelling photoreceptors, horizontal and bipolar cells, adopted by Ikeda et al (adapted from [33].)

Kobayashi et al [38]. A retina is implemented with two resistive networks to achieve the familiar Laplacian of Gaussian style response. However, the size of the receptive field of the spatial filter adapts depending on either global or local light intensity. This is achieved by implementing the conductances in the resistive grid with MOSFETs in the triode region of operation, thus tunable with voltage. Such a tunable receptive field allows the system to maximise its response depending on the signal to noise ratio of the particular image. For instance, if it is assumed that the intrinsic noise is constant, the relative noise to signal ratio during daylight will be small. In this situation, the size of the receptive field can be small, increasing the spatial resolution. However, when exposed to moonlight, the receptive fields need to increase to counter the reduced signal to noise ratio. More work by this author includes a 40 x 45 pixel silicon retina with user variable receptive field [39]. The implementation uses both negative and positive resistances to produce a better approximation to the Mexican hat convolution kernel. In a 2  $\mu\text{m}$  process, each



pixel measures  $170 \mu\text{m}$  by  $200 \mu\text{m}$  and the entire chip consumes 2 W.

Instead of using a resistive grid, Harris et al[40] used non-linear elements they termed resistive fuses. These have the properties of linear I-V relationships for small voltage drops, yet the fuse 'breaks' and current reaches zero for large voltage drops. When incorporated in a vision sensor, small discontinuities are smoothed as with standard resistive grids, but large changes turn off the fuse, allowing abrupt spatial gradients to be segmented[6]. Such a system was applied to robot vision in[41].

### 2.2.2 Silicon Retina: 'Engineering' Approaches

By producing smart sensors that incorporate focal plane processing, researchers hope to produce fast, low-power front ends for complex image processing tasks. While many take their inspiration from biology, it is possible to include both analogue and digital processing within each pixel in an effort to produce superior vision processing systems.

An interesting approach to extracting spatial gradients from a scene with a CMOS smart sensor was implemented by Barbaro et al[42]. The system uses steerable filters to extract both the magnitude and direction of spatial gradients in the imagers field of vision. The approach relies on the definition of the first order derivative of a two-dimensional function performed in the vector direction  $\vec{\xi}$ , seen in equation 2.2, where  $\alpha$  is the vector's angle.

$$\frac{\partial I(x,y)}{\partial \vec{\xi}} = \frac{\partial I(x,y)}{\partial x} \cos \alpha + \frac{\partial I(x,y)}{\partial y} \sin \alpha \quad (2.2)$$

When applied to the discrete pixel locations associated with a CMOS imager and with the angle  $\alpha$  swept in time, equation 2.2 becomes that in equation 2.3, where  $i$  and  $j$  represent the pixel coordinates.

$$\frac{\partial I_{i,j}}{\partial \vec{\xi}(t)} = \frac{I_{i+1,j} - I_{i-1,j}}{2} \cos \omega.t + \frac{I_{i,j+1} - I_{i,j-1}}{2} \sin \omega.t \quad (2.3)$$

It can be seen from equation 2.3 that the local spatial derivative in the vector's direction can be computed using only nearest neighbour connectivity and two sine and cosine interpolating functions. Note that the partial derivatives in the x and y directions from equation 2.2 have been



replaced with numerical approximations based on the central difference theory in equation 2.3. The second equation is applied to each pixel in the array, with the amplitude of the result representing the magnitude of the gradient, and the phase its direction. The chip uses current mode analogue circuitry to apply the equation, with a pixel size of  $80\text{ }\mu\text{m}$  by  $80\text{ }\mu\text{m}$  in a  $0.5\text{ }\mu\text{m}$  process. The entire imager of 10000 pixels consumes 50 mW of static power and is designed to operate at a frame rate of 1000 Hz.

### **2.2.3 Vision Sensors for Object Orientation/Selection**

Another interesting form of spatial processing is concerned with computing the position and orientation of objects within the field of vision.

An early attempt was developed by Standley[43], which determines the position and orientation of an object against a dark background. A resistive grid is used to calculate the first and second moments of the spatial intensity distribution. These moments communicate the centroid of the object and the axis of least inertia, which convey the position and orientation respectively. The sensor implements a  $29 \times 29$  array of pixel cells, with each cell measuring  $190\text{ }\mu\text{m}$  by  $190\text{ }\mu\text{m}$  and incorporating photoreceptors along with current mode brightness thresholding circuitry to remove a dim scene background. When tested, the angle of orientation was found to vary around the mean by  $\pm 2\%$  for moderately sized objects.

A more recent approach by Shi[44], applies Gabor-type filtering algorithms to a two-dimensional imager array. The applied filters can be tuned to respond strongly to particular orientations and ignore others. The design uses subthreshold transistors in a current mode system producing a power dissipation of only  $1.2\text{ }\mu\text{W}$  per pixel. The system relies on weighted summations of nearest-neighbour photocurrents to implement the algorithm, with each pixel containing a photo-sensor, two current amplifiers and some subthreshold transistors configured as conductance elements. Each pixel is  $132\text{ }\mu\text{m}$  by  $108\text{ }\mu\text{m}$  in a  $1.2\text{ }\mu\text{m}$  process, with a 20% fill factor. When tested, the system performed reasonably well although fixed pattern noise had to be digitally removed in a post-processing step.

A one dimensional imager with the ability to group pixels into objects based on their intensity was designed by Morris et al[45]. The idea is to produce a type of visual attention scheme, where processing is applied to objects rather than to individual pixels. This will allow processing to be concentrated on areas of saliency within the scene, improving the efficiency of



such smart sensors. The algorithm first normalises the photocurrents before spatially filtering them using a variation on Boahen's silicon retina[28]. The spatially filtered signal is then thresholded to produce a binary signal that essentially signals the presence or absence of an object. Those pixels that have currents larger than the global threshold value are assumed to belong to the same object. Within these pixels, the largest current is selected and copies used for every pixel in the object. This essentially forces all the selected pixels to act together as a single object. The circuitry is implemented with subthreshold current mode blocks in a  $2\text{ }\mu\text{m}$  process. However, there is no mention of the size of each pixel processing unit.

A two-dimensional version was also designed[46]. A single processing unit is shared between four photodetector cells to produce a multi-resolution architecture. The imager was designed in a  $1.2\text{ }\mu\text{m}$  process and each four photodetector element is  $159.6\text{ }\mu\text{m}^2$ . The entire array consumed approximately 5.6 mW during operation.

Other examples of spatial processing for visual attention include work by Brajovic et al[47], which describes a system that targets an intensity peak in the incident image and continuously reports its location and magnitude. The idea is to reduce data flow for subsequent processing stages.

#### 2.2.4 Other interesting Spatial Processing Performed on the Focal-Plane

There are many other possible applications for incorporating spatial processing with focal-plane architectures.

For instance, a novel application was developed by Delbruck[48] for digital camera auto-focus applications. The chip would be included in a feedback system for camera applications, measuring the sharpness of the image and correcting as necessary. A de-focused image corresponds to a circular 'cookie-cutter' kernel whose diameter encodes the distance the image is displaced from the plane of focus. The aim of the system is therefore to minimise the effect of this kernel, which it achieves by comparing the absolute difference between three neighbouring pixels in a hexagonal grid. An *anti-bump circuit*[49] computes an expansive measure of difference, with the sharper the image, the larger the output. The measure of sharpness for the image is the sum of all the anti-bump circuits in the array. Each processing element in the 25 by 26 array is  $60\text{ }\mu\text{m}^2$  when designed in a  $1.2\text{ }\mu\text{m}$  process.

Another interesting application involves an imager designed to convert the intensity present in



the scene into timing sensitive events[50]. The idea is that cells receiving more light generate events before those cells receiving less light, meaning the intensity of the incident image is encoded in the timing of the events. This allows for the sharing of communication resources but also produces extremely rapid results based on the intensity profile in the image. The more time allowed for the processor, the more inputs that are received so the system builds up a global decision, initially based on only a few pixel cells but becoming more detailed as other pixel's contribute. Such global processing allows extremely fast decisions to be made about the relative intensities in the scene, without the need to read out the entire image. A 21 by 26 cell sensor was developed in a 2  $\mu\text{m}$  process, with each cell measuring 152  $\mu\text{m}$  by 180  $\mu\text{m}$ .

### **2.2.5 Comments on Focal-Plane Approaches to Spatial Processing**

It is clear that CMOS image sensors performing dedicated spatial processing tasks have created much academic interest recently. In general, such sensors can be considered input stages for more complex image processing tasks, with edge or feature extraction combined with dynamic range reduction the key processing requirements. The 'scientific' approaches benefit from low power consumption, with transistors biased in subthreshold a recurring theme for such implementations. However, it appears that the motivation for much of the early work on neuromorphics was to model the biological retina as accurately as possible. While this was successful, the use of such techniques for real-world engineering problems remains to be seen.

The 'engineering' approaches to focal plane spatial processing highlight interesting solutions to niche problems, such as auto-focus and object orientation. Despite large pixel sizes and low fill factors, advantages in power consumption and speed of processing make such techniques attractive in certain applications. Of particular interest are sensors with visual attention systems, which concentrate on regions of interest in a scene.

## **2.3 Temporal Processors**

Systems developed with purely temporal aspects as the focus are rare, due to the nature of vision processing. Nevertheless, this section aims to review papers that are more concerned with the temporal aspects of vision processing than purely spatial or spatio-temporal phenomenon. The reason for the distinction is that the aim of the sensor described in this thesis is concerned only with the temporal variations of the image intensity, rather than the location of an object







ison between the prediction the model makes and the actual photocurrent comprises the output. The different gain responses to 'slow' and 'fast' temporal inputs are created by the dual feedback paths, through  $C_2$  and the adaptive element. During a low speed variation in intensity, the adaptive element dominates. This element is designed to have large resistance for small signals and a low resistance for large voltage variations, allowing rapid adaptation to huge changes in illumination while remaining sensitive to small contrast changes. If the change in illumination occurs quickly however, the capacitor in the feedback dominates and the gain of the capacitive divider is applied to the signal. The dual feedback path provides high gain for quickly varying signals and low gain for steady state signals.  $V_b$  can be used to set the cutoff frequency of the circuit, as it controls the bias current of the amplifier created by transistors  $Q_p, Q_{cas}$  and  $Q_n$ .  $Q_{fb}$  is the transistor used to create the prediction of the input photocurrent. The addition of the cascode transistor  $Q_{cas}$  reduces the miller effect in  $Q_n$  as well as increasing the gain of the inverting amplifier. Results from this photocircuit are promising, with successful operation over almost seven decades of input illumination.

Two variations on this circuit were designed by Liu[52, 53]. By replacing the adaptive element from Delbruck's photoreceptor with a non-linear resistor created from a single transistor, it was possible to vary the frequency response of the circuit. The temporal filtering properties of both circuits adapt with background intensity, similar to the operation of the vertebrate and invertebrate retina. Tests have shown that the retina behaves as a temporal band pass filter at high background intensity, yet modifies to a low pass filter at lower intensity levels, where the signal to noise ratio is lower. The photoreceptors designed here achieve this by the biasing conditions of the non-linear transistor. While the time constant of the adaptation in Delbruck's circuit is fixed by design decisions, both circuits presented here have tunable adaptation rates, producing better results throughout five orders of input illumination.

Earlier work on temporal light adaptation was attempted by Mann[54]. The circuit combines high pass and low pass filters along with a gain stage to produce a temporal band pass filter. The idea is once again to allow the circuit to adapt to a temporal average while still remaining sensitive to small changes in illumination, such as an object in a very bright background. While only documenting simulation results and integrating large amounts of circuitry into each pixel, the results seem to verify operation.



### **2.3.2 Temporal Processing: 'Engineering' Approaches**

A recent 48 by 48 CMOS imager array developed by Kramer[55] is concerned with temporally high pass filtering the incident image. The aim is to exploit redundancy in mostly static images by producing a sensor that responds only to fast temporal changes within a scene. Using an adapted version of Delbruck's photoreceptor, each pixel produces an output at either the ON or OFF channel, depending on the polarity of the temporal illumination variation. An output is also created encoding the address of the pixel producing the response. The sensor can be operated to respond only to positive or negative illumination gradients, as well as producing multiple or individual pulses depending in the chosen refractory period. Each pixel measures  $32.8 \mu\text{m}^2$  and has a fill factor of 9.2% in a  $0.35 \mu\text{m}$  process. Such a sensor could be used as a general purpose input stage for temporal processing algorithms. Specific details of the pixel level transient processing, together with characterisation and test measurements can be found in a later paper by Kramer[56].

Another example of temporal processing concerns computing the difference between consecutive frames of data[57]. Such a technique can be used to extract moving objects from a static background as well as in video compression. This particular implementation uses a focal plane technique incorporating memory to facilitate the differencing. Each pixel has two outputs: the current frame output and that of the previous frame. The chip operates in one of two possible modes, pipelined which allows continuous difference evaluation and snapshot which is a 'one-off' difference calculation. However, no details of pixel size are provided.

A temporal pre-processor designed by Gopalan et al[58] is aimed as an input stage for a motion detection algorithm. The idea is to enhance transients and reduce DC effects using analogue, continuous time filtering techniques. A low power, low area implementation initially low-pass filters the photoreceptor output to remove flickering from AC lighting. A high pass filter is used to remove low frequency effects and enhance the temporal aspects of the image data. When implemented in a  $0.5 \mu\text{m}$  process, each pixel measures  $59 \mu\text{m}$  by  $59 \mu\text{m}$ .

### **2.3.3 Comments on Focal-Plane Approaches to Temporal Processing**

In general, the 'scientific' approaches to temporal processing aim to mimic the retina's ability to adapt with different time delays to slow and fast changes in intensity. Of particular interest from the literature review of temporal processing regarding this research was the adaptive photore-



ceptor developed by Delbruck[12]. The ability to remain sensitive to sudden, transient changes in image irradiance over the eight decades of background intensity would make it an excellent input stage for the system described in this thesis. However, the aim of the work described here is to investigate system level architectures for the extraction of frequency signatures. As such, a decision was made to adopt simple circuit techniques to allow for rapid IC prototyping to test the validity of the algorithm. If individual circuits are discovered that out-perform those adopted, they could be 'slotted' into the final algorithm to improve the overall performance accordingly. Despite the power of Delbruck's photoreceptor, it was felt that time spent on it could be better used elsewhere.

Also of interest was the temporal pre-processor by Gopalan et al[58]. It seems the idea was to develop a general purpose temporal processor as the input stage for more advanced image processing algorithms. The circuit adopts a standard engineering approach to signal processing but uses analogue continuous time circuit techniques biased in weak inversion, similar to 'scientific' neuromorphic approaches. This coupling of an engineering approach at the system level with the advantages of so-called scientific neuromorphic circuit techniques is similar to the approach adopted in this research.

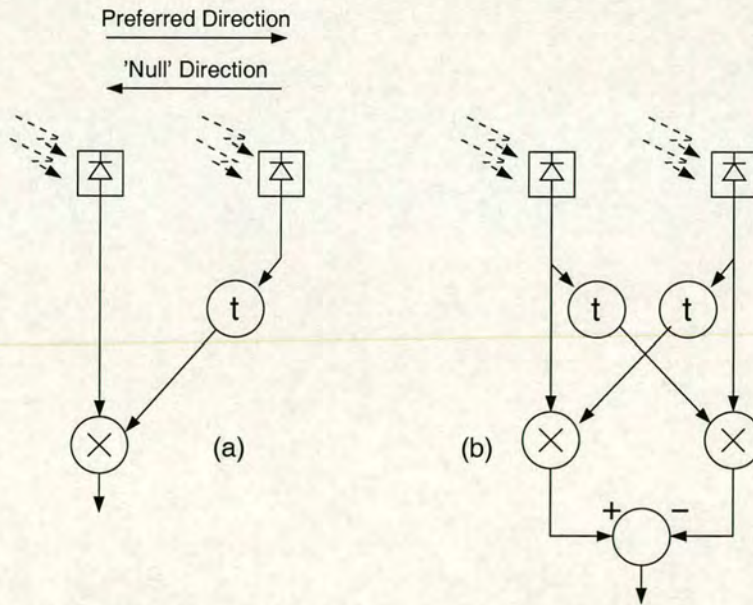
## 2.4 Spatio-Temporal Processors: Motion/Velocity Estimation

Spatio-temporal processing refers to algorithms that rely on spatial information, such as intensity gradients, combined with some temporal data, such as how these gradients move in time. The majority of research in this area has concentrated on motion detection, which involves computing the existence and direction of moving objects in the field of vision. Such work can be extended to include velocity estimation, which provides information on the speed of the motion in a particular scene. Other examples of spatio-temporal processing include time to collision detectors, as well as systems that track objects over space and time.

The ability to detect the direction and speed of motion with simple, low power vision systems has created much academic interest over the past 15 years. There is some disagreement within the literature over the means of categorising the different algorithms that have been developed. Moini[6], draws the distinction between computational implementations and those derived from biological inspiration. The computational versions can be further sub-divided into *intensity-based*, *feature-based* and *correlation-based*. Intensity based algorithms use calcula-



tions of spatial and temporal gradients to estimate the optical flow in the scene. Feature-based algorithms first extract *tokens* from the scene, such as edges or corners and then attempt to track them over time. Limitations with such an approach include the correspondence problem, which describes the ambiguity in ensuring a feature at one position, at a particular time instant corresponds to the same object at another place and time. Correlation based algorithms involve the direct correlation of the current frame with the the previous. By finding the maximum points of correlation between the two, estimates to the change in motion can be made. Biological algorithms according to Moini's classification mostly stem from the Hassenstein-Reichardt model of early visual processing in the fly, seen in figure 2.4(a). The output from one pixel is compared with that of its neighbour, delayed by a time constant. If the object is moving in the preferred direction, and the time it spends passing from one pixel to the next matches the delay, a strongly correlated output will result. If, however, the object is moving in the 'null' direction, no output will be produced. The output from such a system is maximised for a particular velocity in a particular direction. A bi-directional version is represented in figure 2.4(b).



**Figure 2.4:** Hassenstein-Reichardt Biological Motion Detection Algorithm (adapted from [6] and [15]): (a) Single direction, (b) Bi-directional version.

Moini states that a sensor based on the Reichardt model can be considered a correlation sensor where the correlation area has been reduced from the entire image to a single pixel.

A separate classification scheme was suggested by Sarpeshkar et al [15]. Motion detection



algorithms are split into *intensity-based* and *token-based*, the difference being the use of the incident illumination. An intensity based algorithm uses the image irradiance or a linearly filtered version **directly** to estimate the optical flow. A token based algorithm first searches for objects such as lines, edges and corners and then tracks them through time, making use of the redundancy present in many scenes. Sarpeshkar et al further sub-divide intensity based algorithms into *gradient* methods and *correlation* methods. Gradient based algorithms are similar to those described as intensity-based by Moini, in that spatial and temporal derivatives are calculated as a means of estimating optical flow. Sarpeshkar's definition of correlation algorithms corresponds with Moini's biologically inspired category, in that most are based on the Reichardt model. Sarpeshkar et al mention the difficulty in applying correlation models with direct incident illumination as the input, which led to the development of hybrid techniques, first extracting tokens and then applying correlation techniques to these. Also mentioned are *time-of-travel* algorithms, where a token is tracked between two fixed locations to estimate its speed.

While both classification schemes have their merits, for the purposes of this thesis an approach borrowing ideas from both will be applied. Algorithms are split, as with the spatial and temporal processors described earlier into *scientific or biologically plausible* versions and *computational or engineering* solutions. The distinction is based on the extent to which the particular algorithm uses a conventional technique to extract information, compared to a biologically inspired implementation. Both categories share an element of focal plane processing, and are further divided into sub-categories. Within computational approaches are *gradient or differential* algorithms and *token-based* algorithms, both described earlier. Scientific approaches to motion detection include both *pure correlation* algorithms and *correlation with token inputs*. While one may be simply a subset of the other, the difference in performance[15] warrants the distinction. While most of the scientific approaches rely on the Reichardt correlation model, there are other biologically inspired algorithms which will also be investigated.

#### 2.4.1 'Scientific' Motion Detection: Reichardt Correlation Algorithms

The most popular technique for implementing biologically plausible motion detection algorithms is the Hassenstein-Reichardt model, depicted in figure 2.4. Various implementations of this algorithm have been reported in the literature.

One of the earliest was by Andreou et al[59]. A one-dimensional implementation using sub-threshold, current-mode circuitry was constructed using a spatially filtered version of the image

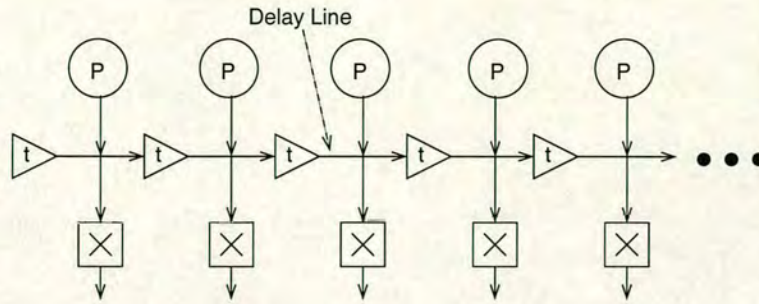


as an input. When designed in a  $2\text{ }\mu\text{m}$  process, each pixel row is approximately  $1000\text{ }\mu\text{m}$  by  $40\text{ }\mu\text{m}$ . The author admits the chip was not practically useful, due to limitations in the spatial filtering producing an illumination dependent input for the system. However, results do show a relatively linear output velocity measurement for input speeds of 0 to 16 pixel/msec. An improved, two dimensional version was also reported[60]. The spatial filtering is performed by Boahen's current mode silicon retina[28], reducing the illumination dependency of the first implementation. Further details of the system, along with a description of Boahen's silicon retina and an excellent introduction to the benefits of Neuromorphic approaches to certain image processing tasks can be found in[18].

The problems with outputs that are illumination dependent led to research in hybrid systems, where a feature is first detected and then used in subsequent correlation stages. An early example of such a system was developed by Horiuchi et al[61]. This one-dimensional sensor uses quick temporal rises in intensity as a token to provide input to a correlation stage. The underlying algorithm calculates the time such a feature takes to travel from one photoreceptor to another. The mechanism is two parallel delay lines, which propagate signals in opposite directions and a series of correlation units. Imagine a row of photoreceptors, with an edge passing from left to right. If the edge moves with infinite velocity, there will be no difference between when the two 'end' pixels experience the edge and fire accordingly. As such, the resultant signals will propagate down the two delay lines and meet in the centre, producing a strong correlation in this location. If the time difference is small (high velocity), correlations occur near the centre of the delay line. As the time difference increases, the correlation occurs further towards the edges as the velocity reduces. Motion direction can be calculated from which side of the centre the strongest correlation occurs. A winner takes all circuit is then used to choose the strongest correlation. Results seem promising, with a fairly linear output for an increase in stimulus speed, however spatial aliasing caused by the spacing of the photoreceptors can introduce errors.

Another correlation based system was designed by Delbruck[62]. The idea is once again based on a delay line, although the mechanism is different. The concept is illustrated in figure 2.5. The photoreceptors (P in figure 2.5) are coupled to a uni-directional delay line, with each element having a delay of  $t$ . If the velocity of an edge matches the delay in the line, the signal on the delay line is reinforced. However, an edge in the wrong direction or at the wrong speed has its signal diminished. By using some form of non-linear operation (multiplication in figure 2.5) it





**Figure 2.5:** One Dimensional Delay Line in Delbruck's Velocity Tuned Pixel (adapted from[62])

is possible to extract information on the amplitude of the delay line signal, which is sensitive to both direction and velocity. A two dimensional system using a hexagonal layout and three delay lines was implemented, with each pixel occupying  $224 \mu\text{m}$  by  $225 \mu\text{m}$  when using a  $2 \mu\text{m}$  process. Much of the area is consumed by the capacitance required to achieve the long time constants necessary for the delay elements. A possible limitation with such an implementation is the need to tune the delay elements to allow a range of object velocities to be detected. Other motion sensors can produce outputs for different stimulus speeds without external tuning.

A clever circuit that attempts to automatically tune the delay in a Reichardt sensor was developed by Liu[63]. The output from each photoreceptor is passed through both high and low pass filters with similar time constants. The output from each filter is compared with a peak amplitude detector, the output of which provides an error for tuning the time constants of the filters. If the input frequency does not match the time constant of the filters, the output of the two filters will be different, with the error encoding the direction the filter's frequency response needs to move. While not implementing an actual sensor, possibly due to the size of the circuitry required, chip results show the filter's time constant adapting accurately over four decades of input frequency.

A more recent correlation algorithm with token based inputs was developed by Jiang et al[64]. The circuit uses the silicon retina developed by Wu et al[65] to create pulses corresponding to the edges in the scene. These edges are used as the tokens for input into a correlation system. The system converts the zero-crossings from the silicon retina into binary pulses, allowing digital circuitry to implement the correlation. The authors suggest this produces more accurately correlated outputs, improving the performance of the motion detector. When designed in a  $0.6 \mu\text{m}$  process, each pixel of the  $32$  by  $32$  array is  $100 \mu\text{m}^2$  with a  $20\%$  fill factor. Test results show the sensor can detect motion at any angle for a fixed stimulus speed of  $0.5 \text{ m/s}$ .



Another recent implementation of a Reichardt based correlation motion sensor was developed by Harrison and Koch[66]. This claims to be one of the most accurate implementations of Reichardt's model, working on the hypotheses that more closely modelling nature will produce better results. The aim was to outperform other implementations, particularly with respect to image contrast dependence. The algorithm uses a Gm-C high pass filter to fix the DC level of the photoreceptors output. The delay is implemented with a tunable Gm-C low pass filter, while a Gilbert multiplier performs the correlation. Each cell occupies an area of  $61\text{ }\mu\text{m}$  by  $199\text{ }\mu\text{m}$  when designed in a  $1.2\text{ }\mu\text{m}$  process with much of the area consumed by the capacitors needed for the filtering. In the paper, the circuit was strenuously tested to gauge performance, particularly with low contrast images. The non-linear saturating characteristics of the Gilbert multiplier provide better operation than previous implementations. The authors also claim that one pixel cell consumes a mere  $50\text{ nW}$  when operated with a  $2.5\text{ V}$  power supply. The circuit seems to outperform many of the previous implementations. However, a two-dimensional version has not been implemented, which may increase pixel size. The authors estimate that an  $80$  by  $80$  pixel sensor could be implemented on a  $7\text{ mm}$  by  $7\text{ mm}$  chip with less than  $700\text{ }\mu\text{W}$  power consumption.

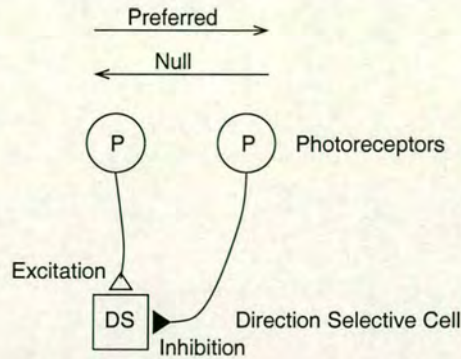
#### **2.4.2 'Scientific' Motion Detection: Alternative Algorithms**

While the Reichardt model for motion selectivity is the most prevalent in analogue VLSI implementations, other algorithms do exist.

An alternative was suggested by Barlow and Levick and is implemented in a sensor by Benson and Delbruck[67]. The model was originally developed to describe direction selectivity in rabbit retinas and utilises inhibition in the null direction. An illustration of the Barlow and Levick model can be seen in figure 2.6. As an object moves across the sensor in the preferred direction, the photoreceptor on the left excites the direction selective cell, causing it to fire. However, when the object reaches the neighbouring photoreceptor to the right, it inhibits the firing of the DS cell. An object moving in the null direction produces no outputs from any of the direction selective cells, as each is inhibited by the photoreceptor to its right. An estimate of velocity can be calculated by the speed with which the DS cell's output is inhibited. Slowly moving objects in the preferred direction will produce outputs for longer than quickly moving objects, as the inhibition occurs later. The circuit uses Delbruck's adaptive photoreceptor[12] as an input stage to enhance the transients in the incident illumination. An array of  $47 \times 41$  cells



was designed, with results verifying the direction selectivity of the system.

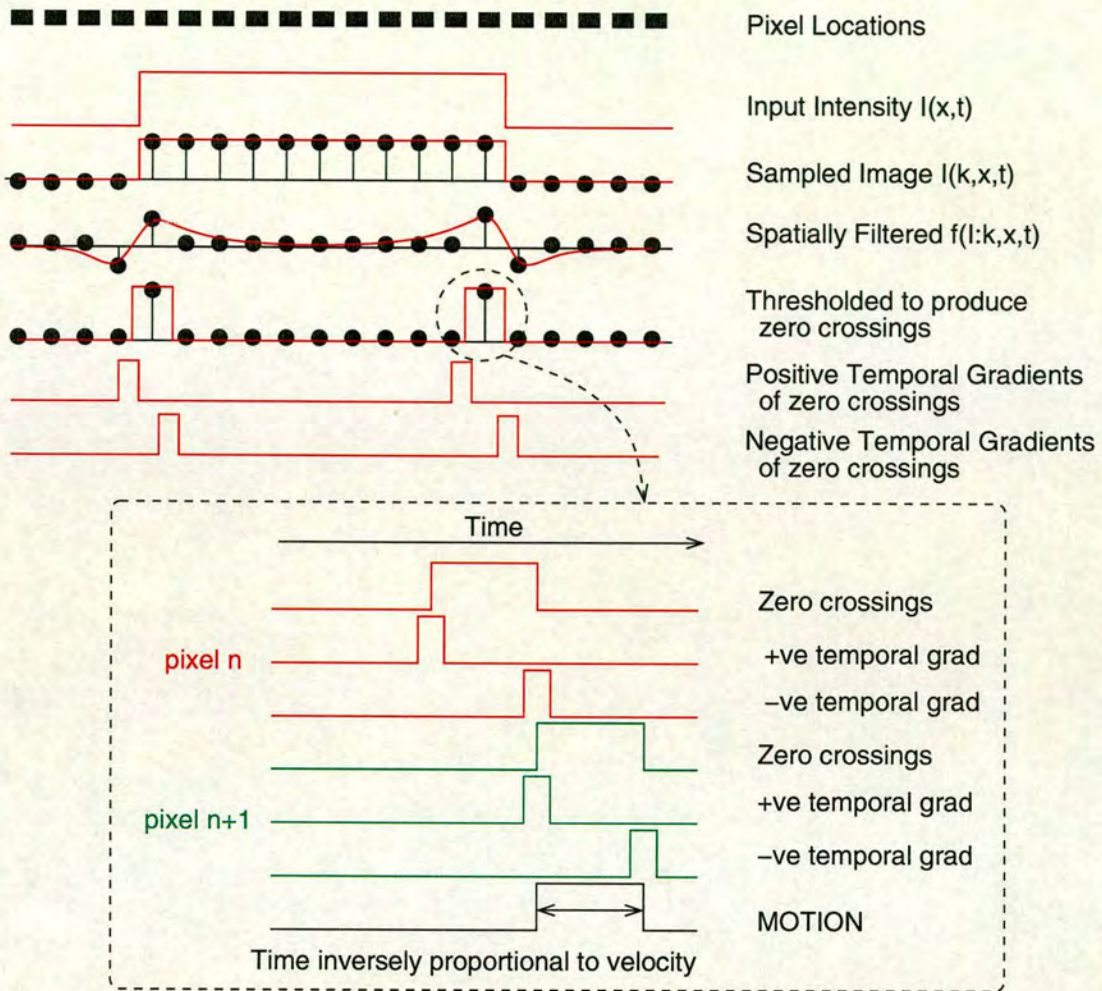


**Figure 2.6:** *Barlow and Levick's Model of Direction Selectivity in the Rabbit Retina (adapted from [67])*

An approach that combines the Reichardt sensor with an algorithm by Ullman and Marr was developed by Etienne-Cummings et al[68, 69]. The Ullman-Marr model uses the temporal derivatives of zero-crossings for velocity estimation. The idea is to measure the time between the disappearance of an edge (negative temporal derivative) at one pixel and its appearance (positive temporal derivative) at a neighbouring pixel. The authors state that by combining the direction selectivity of the Reichardt model with the velocity estimation of the Ullman-Marr algorithm, a more accurate and robust motion detection system is produced. The implemented algorithm can be considered a hybrid token-correlation system, based on the classification used in this thesis. A resistive grid is used to perform the spatial filtering and extract the spatial tokens to be tracked. However, contrary to many other silicon retina implementations[14, 29], the transistors are operated in strong inversion rather than subthreshold. The authors state that weak inversion resistive grids suffer from large offsets and small signal to noise ratios, problems which can be alleviated with a strong inversion implementation. Results from a test chip show that the algorithm responds to contrasts as low as 5 % in dim room lighting and bright sunlight. Plots of detected speed versus actual speed exhibit good linearity.

A more detailed description of the algorithm, together with improvements can be found in[70]. The operation of the algorithm can be seen in figure 2.7. The image irradiance is spatially filtered and then thresholded to produce the zero-crossings. These encode the position of any edges in the scene with a binary representation. The pulses of the zero crossings are then temporally differentiated, with a positive result representing the arrival of a zero crossing and a negative temporal gradient representing its disappearance. Motion is detected when the dis-





**Figure 2.7:** Etienne-Cummings et al Correlation/Token Motion Detector Hybrid (adapted from [70])

appearance of a zero crossing at one pixel corresponds to its appearance at the neighbouring pixel. This can be achieved by correlating the positive temporal gradient at one pixel with the negative from the neighbouring. As the inset in figure 2.7 depicts, a correlation occurs and the length of time that passes until the edge disappears from the pixel is inversely proportional to its velocity. In the improved version of the algorithm, each cell measures  $110 \times 220 \mu\text{m}^2$  when designed in a  $2 \mu\text{m}$  process, with a fill factor of 41 %.

Another interesting approach to motion detection is described by Liu and Mead[71]. The idea is to dynamically adapt the delay in a delay line system such as Delbruck's[62] to allow different velocities to be detected without external bias adjustments. The system described by Liu is



similar to a phase locked loop, in that a phase difference is calculated and used as an error signal to match the delay to the time taken for an object to pass between two pixels. The system is based on a model of the optomotor response of the bee. The circuitry is complex and no details of pixel size are included, although a detailed analysis of convergence properties of the feedback system is included.

A different neuromorphic approach that relies on an implementation of the template model of insect vision systems was developed by Moini et al[72]. The template model was proposed by Horridge and operates on the principle that the temporal contrast can be quantised to three values; increase, decrease and no change. The signals from two neighbouring detectors can have nine possible combinations of temporal contrasts, with 81 in total from two separate detectors at two separate sampling times. In the implementation, the temporal differentiation and thresholding circuits are implemented in analogue circuitry, with digital techniques performing the template matching. The system makes use of a current mode spatial filter, similar to that designed by Andreou[29]. Temporal differentiation is achieved with an operational transconductance amplifier based system, biased in subthreshold to create the large time constants required. Results show the successful operation of the spatial filter and overall motion detection system. However, bias currents need to be tuned to obtain reasonable outputs for different lighting conditions.

#### **2.4.3 'Engineering' or Computational Motion Detection: Token-Based Algorithms**

Despite the scientific interest in translating biologically inspired motion detection and velocity estimation algorithms into VLSI implementations, many are unpractical or limited in their utility. For instance, many of the algorithms based on the Reichardt model have fixed time delay elements, resulting in limited velocity estimation range. In addition, many implementations produce large pixel circuits, meaning high resolution sensors are unpractical. An engineering approach to motion detection and velocity estimation may provide a smaller, more robust circuit. However, as previously mentioned, the discretisation of image irradiance associated with pixel arrays introduces the correspondence problem. The difficulty stems from ensuring one token at a particular location corresponds to the same object as it moves in space and time[6].

One such technique involves first identifying features or *tokens* in the incident image. It is possible to extract either spatial tokens, such as edges, corners and lines, or temporal features,



such as fast changes in intensity[66].

#### **2.4.3.1 Token-Based Motion Detection: Spatial Tokens**

The sensor developed by Etienne-Cummings et al[70] could be included in this section as it employs a hybrid correlation/token based approach. The tokens that are tracked over time are the zero-crossings of the edges of the image when filtered with a difference of Gaussian kernel. Figure 2.7 highlights the operation of this algorithm in more detail.

Another interesting approach was developed by Yamada and Soga[73]. The algorithm calculates the time in which a spatial edge moves a constant distance, related to the spatial distance between pixels. A  $10 \times 2$  array of pixel cells was implemented, with a detectable velocity range of 0.2 to 100 mm/s. However, the accuracy is only  $\pm 20\%$ , which may be unacceptable for certain applications.

#### **2.4.3.2 Token-Based Motion Detection: Temporal Tokens**

Algorithms that search for abrupt temporal changes in image intensity and track them in time have proved popular with regards VLSI implementation.

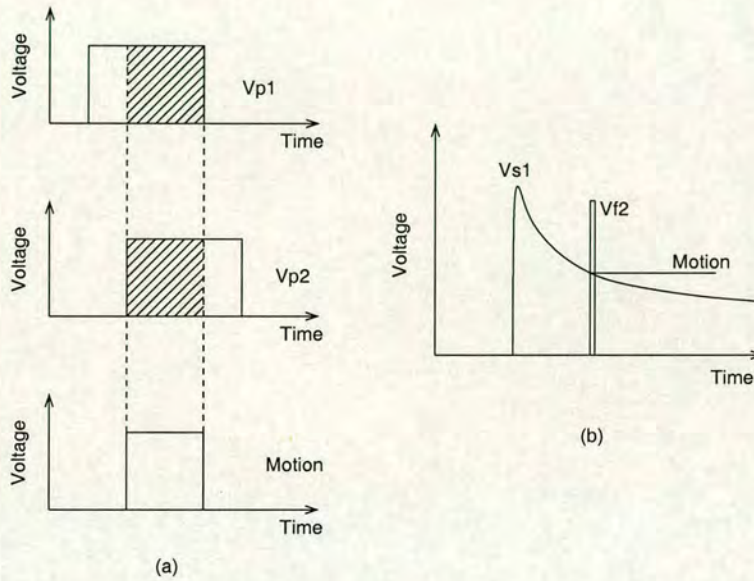
An algorithm developed by Kramer et al searches for abrupt temporal changes in image irradiance. An original version computed velocity in only one direction[74], while a subsequent update operated in two directions[75]. The algorithm works by converting temporal changes into thin current pulses, using a circuit based on Delbruck's adaptive photoreceptor[12]. The current pulses are then transformed into voltage pulses with pulse-shaping circuits, the outputs of which are fed to direction selective motion circuits. There are two separate algorithms implemented on the chip, both using the pulse shaping front end.

The first is termed facilitate and trigger (FT) and works by comparing the timing of pulses created by two adjacent pixels. The overlap between the two is directly proportional to velocity. Direction selection is created by circuitry that responds differently depending on the order of the pulses' arrival.

The second algorithm is called the facilitate and sample (FS) algorithm. Each pulse shaping circuit creates both a thin voltage spike ( $V_f$ ) and a slowly decaying signal ( $V_s$ ) at the onset of a current pulse. The voltage spike from one pixel is used to sample the slowly decaying



output from its neighbour. If  $V_s$  precedes  $V_f$ , the sampled value is a measure of the time delay between them, and is therefore inversely proportional to velocity. If however the sampling pulse arrives before  $V_s$ , it samples the decayed output from the previous edge, which should have diminished to a low value unless there is a high frequency of edges. This algorithm was conceived to produce outputs over a larger velocity range than the FT version. The operation of both can be seen in figure 2.8.



**Figure 2.8:** Kramer et al's Token Based Velocity Sensor(adapted from [75]): (a) FT algorithm; pulses from neighbouring pixels are compared, with overlap encoding motion and pulse-width inversely proportional to velocity. Direction selectivity maintained by circuits sensitive to the order the pulses appear in. (b) FS algorithm; Each pixel produces a thin sample pulse ( $V_f$ ) and a slowly decaying pulse ( $V_s$ ). If ( $V_f$ ) occurs after ( $V_s$ ), it samples its value which provides an estimate to the time delay between them and is therefore inversely proportional to velocity.

One dimensional versions of both algorithms were implemented to verify performance. Both produce a cell size of approximately  $0.05 \text{ mm}^2$  when implemented in a  $2 \mu\text{m}$  process. The authors predict that with the same technology, a two-dimensional array of 1250 pixels would create a chip measuring  $62.5 \text{ mm}^2$ . Despite the increase in pixel size to accommodate two dimensional performance, a  $128 \times 128$  pixel array could be integrated into a  $16 \text{ mm} \times 16 \text{ mm}$  chip using a  $0.7 \mu\text{m}$  process. The FS algorithm operates successfully down to low speeds but is limited at higher velocity due to the finite width of the sampling pulse. The FT algorithm operates better at high velocity but is limited at lower speeds due to the mechanism employed in the direction selective cells. The FT algorithm is better with respect to different input illumination



than its counterpart, although the authors state that both respond better to low contrast stimuli than many of the alternative algorithms reported in the literature. Further details of the system, along with a detailed review of previous motion detection algorithms can be found in[15]

A similar approach was developed by Higgins et al[76, 77]. Two separate token based algorithms are presented, Inhibit, Trigger and Inhibit (ITI) and Facilitate, Trigger Compare (FTC). Both once again use a circuit based on Delbruck's adaptive photoreceptor to enhance transients in the image. The ITI algorithm operates on the principle that an edge crossing any pixel triggers a direction voltage for both left and right directions. The same temporal token, when seen by a neighbouring pixel inhibits the original pixels left or right pulse, depending on orientation. The output current is the difference between the left and right channels. The FTC algorithm is similar to the FT algorithm described by Kramer et al[75], in that speed is calculated by timing the occurrence of an edge at one pixel and its reappearance at a neighbouring pixel. Two dimensional versions of both were created, with a 14 x 13 ITI array and a 12 x 13 FTC sensor. Pixel size is 110 x 120  $\mu\text{m}^2$  for the ITI and 128 x 119  $\mu\text{m}^2$  for the FTC, with 47  $\mu\text{W}$  and 29  $\mu\text{W}$  power consumption per pixel respectively.

The authors allude to the inspiration for these sensors stemming from work by Kramer[75] yet claim their implementation is more suitable to two dimensional implementation, with the two dimensional ITI pixel cell occupying 20% less area than the one dimensional FS alternative. There is little to choose between the two algorithms in terms of performance, with both computing motion over two orders of magnitude.

It is interesting that both Kramer and Higgins feel real improvements in the production of robust motion detection algorithms would be helped by the design of an improved temporal token detector. Although Delbruck's circuit[12] is the basic approach, both authors feel their circuits are not limited by the actual algorithm, but the ability to successfully detect rapid temporal changes in the image.

#### **2.4.4 'Engineering' or Computational Motion Detection: Gradient-Based Algorithms**

The gradient method for motion detection relies on the assumption that image brightness remains constant with respect to time[6, 15]. This assumption allows an estimate of velocity to be derived from first order spatial and temporal derivatives, as seen for a one dimensional case



in equation 2.4.

$$v = -\frac{\partial E / \partial t}{\partial E / \partial x} \quad (2.4)$$

However, the calculation of accurate temporal and spatial derivatives with analogue VLSI is difficult, particularly with such circuitry's inherent offset and noise problems.

One of the earliest implementations of a focal plane motion detection array was that by Tanner and Mead[13]. The system utilises analogue computation to solve a mathematical model of optical flow. By computing the first spatial derivative of intensity in the x and y directions, together with the first temporal derivative of the intensity variation, it is possible to approximate the global velocity of the image flow. An error signal is then computed to continuously refine this velocity estimation, improving the accuracy. This system was one of the first to demonstrate the potential of implementing motion processing algorithms with dedicated analogue computation, despite possible inaccuracies caused by the calculation of accurate derivatives with analogue circuitry.

Another implementation of a gradient based motion detection chip was developed by Chong et al[78]. When a time-invariant but spatially variant image is projected onto an array of photoreceptors, the output current from each is constant over time. If these output currents are temporally differentiated, any movement in the image will produce non-zero outputs, proportional to the first derivative of the output current. A current mirror differentiator is used and test results from a 25 by 25 pixel array prove the successful operation.

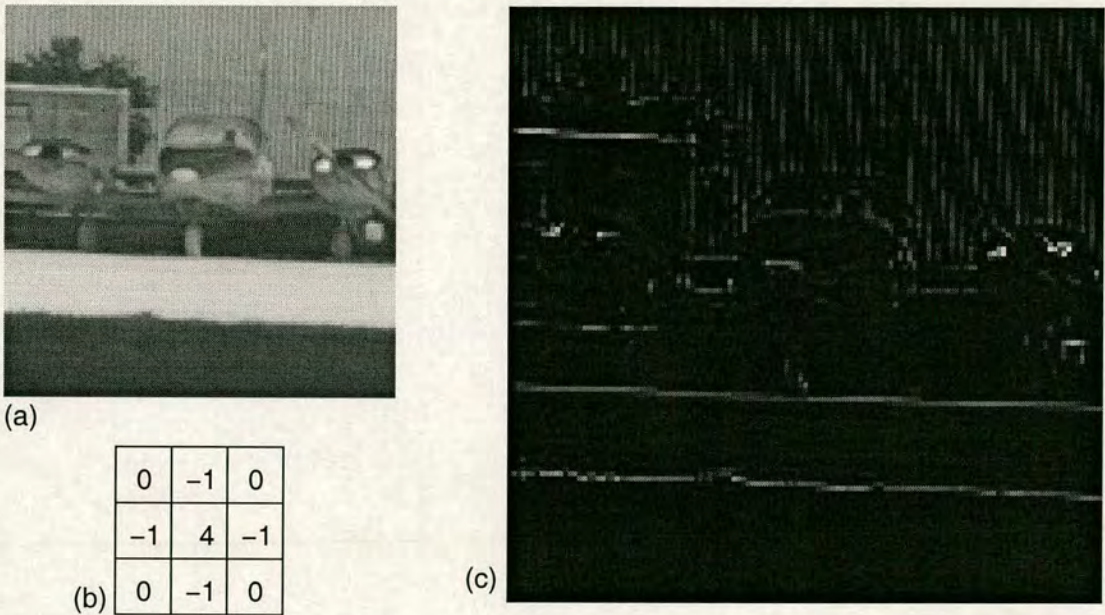
A more recent gradient-based algorithm was developed by Deutschmann and Koch[79]. This circuit translates equation 2.4 directly into analogue VLSI, using Delbruck's adaptive photoreceptor[12], spatial and temporal derivative circuits and a division circuit. Floating-gate techniques are used to increase the linear range of the amplifier used to implement the spatial derivative. Test results from a one directional velocity sensor exhibit a fairly linear output for increasing stimulus speed, over a range of approximately 150 mm/sec. Tests show successful operation down to approximately 4 % contrast. No details of pixel size are provided, yet the authors admit it has to be optimised if inclusion in a two-dimensional array is to be achieved.



### 2.4.5 Programmable Focal-Plane Image Processing

The CMOS image processors reviewed so far have been hard-wired for a specific task, be it motion detection, velocity estimation or spatial filtering. However, the complexity of image processing suggests some element of programmability may be useful in certain applications. Recently, research on programmable focal plane CMOS image processors has produced interesting systems.

Etienne-Cummings et al[80] produced a processor with programmable spatial kernels. As with digital signal processing techniques, a spatial filtering operation is achieved by convolving the incident image with a discrete two dimensional kernel. The process is highlighted in figure 2.9.



**Figure 2.9:** *Spatial Processing Using Convolution Kernels: (a) Original image, (b) 3 x 3 convolution kernel, approximating the Laplacian of Gaussian, (c) Spatially filtered result.*

The convolution kernel is applied to each pixel in the original image, the results for each pixel summed and then thresholded to produce a binary image representing the edges in the scene. The ability to vary the underlying convolution kernel allows for different spatial filtering effects, such as horizontal or vertical edge enhancement and positive or negative intensity gradient suppression. A more recent implementation[81] has included temporal processing in the form of frame differencing to produce a truly programmable spatio-temporal image processor. The



circuitry works by using current mirrors to create copies of the photocurrent, scaled to represent the particular convolution kernel being implemented. This produces a relatively small pixel size at only  $30\ \mu\text{m}^2$ , yet the mismatch involved with mirroring subthreshold current may cause spurious results. However, the reported results appear to verify the versatility of the system.

An earlier programmable approach to modelling the retina was implemented by Paillet et al[82]. This circuit incorporates a digital processing element within each pixel, effectively creating a digital, programmable retina. The ever decreasing minimum feature size of CMOS processes allows up to 100 minimum sized transistors to be included, with minimal effect on the imaging array dimensions. Each pixel is  $60\ \mu\text{m}$  by  $60\ \mu\text{m}$ , with a fill factor of 30 %. The processing element first converts the analogue signal to digital before performing boolean algebra operations on the data. Image processing tasks such as motion detection, segmentation and shape recognition are possible. The processing element is designed to be programmable, allowing more flexibility than previous hard-wired analogue approaches. However, the 128 by 128 pixel imager uses about 1 W when clocked at 80 MHz.

Another approach to implementing a programmable CMOS image sensor-processor was developed by Dudek et al[83–85]. The approach uses analogue sampled data techniques combined with digital control circuitry to produce an analogue microprocessor, capable of performing different image processing tasks. The system allows for the convolution of image data with programmable kernels, similar to the system developed by Etienne-Cummings. By using switched current techniques and simple weighted current mirrors, it is possible to achieve addition, subtraction and multiplication of individual pixel data with the corresponding convolution kernel weighting. However, such SI circuit techniques are prone to mismatch, particularly with the charge injection caused by analogue switches. As such, a technique involving correlated double sampling was adopted. The idea is to sample the pixel data twice per cycle, once to measure the actual signal and a second time to measure any underlying noise. The noise can then be subtracted from the signal, improving the accuracy at the cost of increased area of implementation. Each analogue microprocessor measures  $600\ \mu\text{m}$  by  $70\ \mu\text{m}$  and consumes 100  $\mu\text{W}$  when implemented in a  $0.8\ \mu\text{m}$  process. The authors aim to produce a general, low power analogue microprocessor and have aimed in particular at image processing applications. As such, the results seem promising, particularly with the ability to program the system to perform different tasks.



#### **2.4.6 Comments on Focal-Plane Approaches to Spatio-Temporal Processing**

The 'scientific' neuromorphic systems developed to perform spatio-temporal processing seem to concentrate on motion and velocity estimation algorithms. The reason for this probably stems from the simplicity of the biological models developed to explain such processing in the biological retina. Despite this relative simplicity, scientific motion sensors are still far more complex than the purely spatial and temporal processors described earlier, combining techniques from both alongside additional processing circuitry. The results highlight the potential for developing complex image processors with dedicated analogue circuitry, although are limited when compared with more powerful computer based algorithms.

### **2.5 Summary**

The CMOS image processors reviewed in this chapter, both biologically inspired or employing more traditional engineering techniques, all share some element of focal plane processing. Whether employed to perform spatial, temporal or spatio-temporal image processing tasks, the fact that signal processing is included with light sensitive elements in each pixel may provide advantages in certain key aspects of a system level implementation.

Most of the reviewed image processors employ analogue signal processing techniques which can be more economic regarding both power consumption and silicon area than digital circuitry of similar complexity[75]. In addition, the vast majority employ transistors biased in the subthreshold region of operation, where bias currents are in the nA or pA range, substantially reducing power consumption. Analogue processing also has the advantage of operating in continuous time, allowing precise timing of events and reducing the effects of aliasing, a potential problem with sampled data systems. The fact that pixel level processing is performed in parallel also allows for more rapid processing times, without the bottleneck associated with data being read out per column or row. While the pixel sizes are larger than standard CMOS imager arrays, the trade-off between spatial resolution and enhanced functionality produces smaller, more elegant system level designs.

The research documented in this thesis would be classified in the temporal processing section of the literature review. The aim is to produce a CMOS imaging system capable of analysing temporal frequencies present in any scene. Ultimately, the target is the extraction of a frequency signature, incorporating the fundamental frequency and the early harmonics. Such a system



could be used in object classification as well as diagnostic testing. An emphasis on low power, low area, continuous time processing ties in with the potential advantages of a focal-plane implementation, hence the adopted approach involves the development of an algorithm that employs parallel, pixel level analogue signal processing.

It seems that the key to successful neuromorphic image sensor design stems from the particular needs of the project. The complexity of real-time image processing coupled with the limitations of analogue circuits, particularly when biased in the subthreshold region, raises questions about the general application of such techniques. It is clear that for particular applications, successful neuromorphic systems can be developed. For example, motion detection systems based on the Hassenstein-Reichardt model seem particularly suited to such implementation techniques. Despite this, the results from such systems appear limited, either in detectable speed range, detectable illumination range or precision, when compared with computer vision implementations. Indeed, Sarpeshkar et al[15] recognise that the velocity sensors they produce need to be employed in applications where *qualitative* motion estimates, rather than precise values, are required. Knowing whether an object has increased or decreased in speed, or changed direction, rather than providing exact measurements to the magnitude of object velocities, can be useful in systems incorporating feedback control systems.

As previously mentioned, the study of neuromorphic systems with regard to image processing can be split into 'scientific' and 'engineering' categories. The former is concerned with furthering the understanding of biology by accurately modelling neural processing using a platform that faces similar constraints in terms of power consumption and implementation area. The latter approach aims to mimic biological signal processing to provide solutions to common visual processing problems. Such neuromorphic sensors aim to provide elegant solutions to problems where power consumption and area of implementation are more important than the overall precision of the system's output. As such, the development of neuromorphic image processors remains entirely dependent on the identification of relevant applications. In the authors opinion, a general purpose, low power neuromorphic image sensor to rival more traditional image processing techniques seems unlikely.

However, the requirements of the image-processor described in this thesis seem closely linked to the potential advantages of employing an engineering approach to neuromorphic processing. The aim is a dedicated, low power, continuous time sensor capable of extracting frequency signatures from any scene it is exposed to. There exists a trade-off between the accuracy or



processing power of the solution and its corresponding power consumption, with the sponsor company placing an emphasis on minimising the latter. For this reason, an approach combining the advantages of certain neuromorphic design principles in the form of parallel, low power focal-plane processing techniques along with more traditional *engineering* signal processing structures has been employed.



---

## Chapter 3

# Software Development of Temporal Frequency Analysis Algorithm

---

The requirements for low power, low area, high speed processing specified by the sponsor company led to the decision to concentrate on a focal plane processing solution, incorporating analogue, continuous time circuits biased in the subthreshold region of operation. After a review of image processors incorporating focal-plane processing, the potential advantages of such an approach seemed to closely match the requirements of the project.

With a design framework in place, development of algorithms that could perform the required processing while fulfilling the necessary system level requirements could begin. All potential candidates were developed with simple circuit level realisations for each processing step in mind, to ease the translation from software simulation to hardware realisation. The MATLAB programming tool was used to develop and simulate the different approaches, due to the large number of in-built, standardised signal processing routines. An emphasis was placed on analysing potential techniques for the application specified by the sponsor company.

### 3.1 Test Data

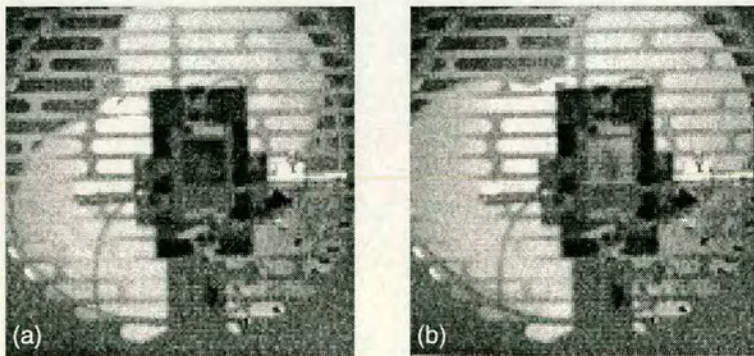
To test potential algorithms, QinetiQ provided a series of image sequences, each containing objects that exhibit temporal frequencies of interest. The data sequences were captured with an infra-red camera, the reasons for which were two-fold: the sponsor company has an interest in infra-red applications and the camera was selected for its high frame rate. The nature of the project requires high sampling rates for the test data to avoid aliasing of the temporal frequencies. The adopted camera sampled at 500 frames per second, meaning the highest detectable temporal frequency was 250 Hz. Each data sequence comprises 500 frames, with each individual frame having a resolution of 128 by 128 pixels, and each pixel having an 8 bit range of data values.



### 3.1.1 Fan Data Sequence

The fan data sequence contains two different objects, a fan and a negative luminescence device, each producing a different temporal frequency of interest. The fan can be clearly seen in the two selected frames in figure 3.1, while the negative luminescence device sits in the very centre of the image. Such a device can be thought of as a thermal torch and was used due to the infra-red capabilities of the video camera. By biasing certain IR LED devices in reverse bias, the carrier densities in the near intrinsic active region fall below the equilibrium value[86], allowing it to absorb infra-red radiation without emitting it, effectively contravening Kirchoff's law[87]. This means that the device emits less radiation than its surroundings and therefore appears as if instantaneously cooled. It was this quick transferral between 'hot' and 'cold' conditions that made the negative luminescence device useful for the development of the test data employed in this research.

In all there were seven different fan data sequences. The rotational frequency of the fan remains constant in each, acting as a control for the luminescence device, whose flashing frequency was changed for each of the eight sequences. The frequency of the thermal torch ramps from 10 Hz to 20 Hz, 30 Hz, 40 Hz, 50 Hz, 70 Hz and finally 90 Hz.

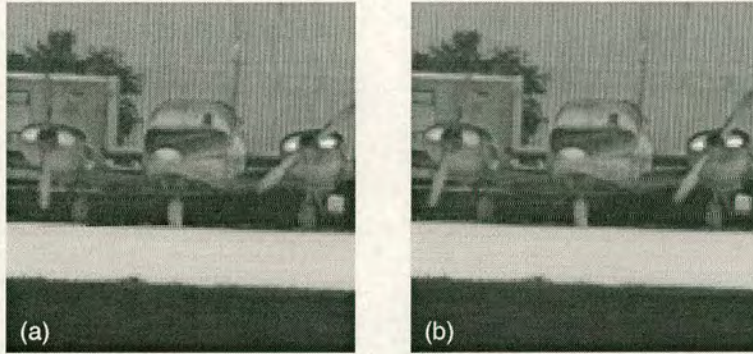


**Figure 3.1:** *Selected Consecutive Frames from Fan Data Sequence: In total there are seven different data sequences, each comprising 500 frames. There are two objects exhibiting temporal frequencies of interest, a fan and a negative luminescence device situated in the centre of image. Its intensity changes from dark in (a) to bright in (b). The fan rotates at a constant frequency in each data sequence, while the frequency of the negative luminescence device's flashing is ramped from 10 Hz through to 90 Hz*



### 3.1.2 Propeller Plane Data Sequence

The propeller plane data sequence depicts a twin engined plane, with both propeller blades rotating as it prepares to take-off. As such, the temporal frequencies of interest are the two rotors. The plane is static throughout the data sequence, with the only motion belonging to the propeller blades. Two consecutive frames from the sequence can be seen in figure 3.2.



**Figure 3.2:** *Selected Consecutive Frames from Propeller Plane Data Sequence: The temporal frequencies of interest are the twin engines of the aircraft*

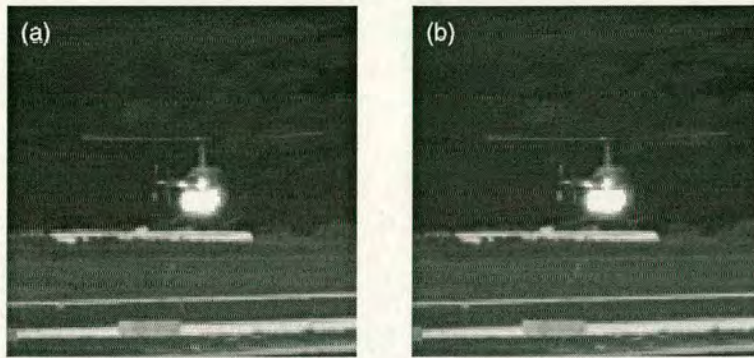
### 3.1.3 Helicopter Data Sequence

With the helicopter data sequence, the temporal frequencies of interest are the main rotor and the tail rotor. The helicopter itself is airborne in the data sequence, and moves relative to the camera from the centre to the bottom right of the image. Two consecutive frames can be seen in figure 3.3. The rotors in the images are very faint, due partly to their width and partly to the low contrast between them and the background.

## 3.2 Dyadic Tree Algorithm

The ability to extract the underlying frequency signature from transient changes in illumination is the underlying aim of the research described in this thesis. As previously mentioned in chapter one, the ideal approach would be the integration of a Fourier style processor within each pixel. Such an approach proved unrealistic given the design criterion imposed by the sponsor company. Nevertheless, the idea of converting the time domain variation in intensity into its corresponding frequency domain representation seemed the best approach to a successful implementation. An initial study of signal transform techniques led to an investigation of wavelet





**Figure 3.3:** *Selected Consecutive Frames from Helicopter Data Sequence: The temporal frequencies of interest are the main and tail rotors of the helicopter*

transforms.

Wavelet transforms differ from Fourier transforms in that the scale of the underlying basis function is variable, giving potentially more detail about the input signal's frequency content. The dyadic tree is a simple method of implementing a wavelet style decomposition of the input signal into *frequency bins* of differing size. Essentially, the approach uses a bank of low and high pass filters to split the signal's frequency content in two. The lower frequency band is then further sub-divided in two, and so on, producing a series of frequency bins covering different sections of the signal's frequency content. Specific details of the dyadic tree and its simulation with regards to the research documented here can be found in appendix A. It was discovered that the approach was unsuitable for this application, as it was unable to successfully discriminate between different temporal frequencies given the imposed size constraints. The technique also required the use of sampled data filter techniques, which is opposed to the sponsor companies initial design criterion.

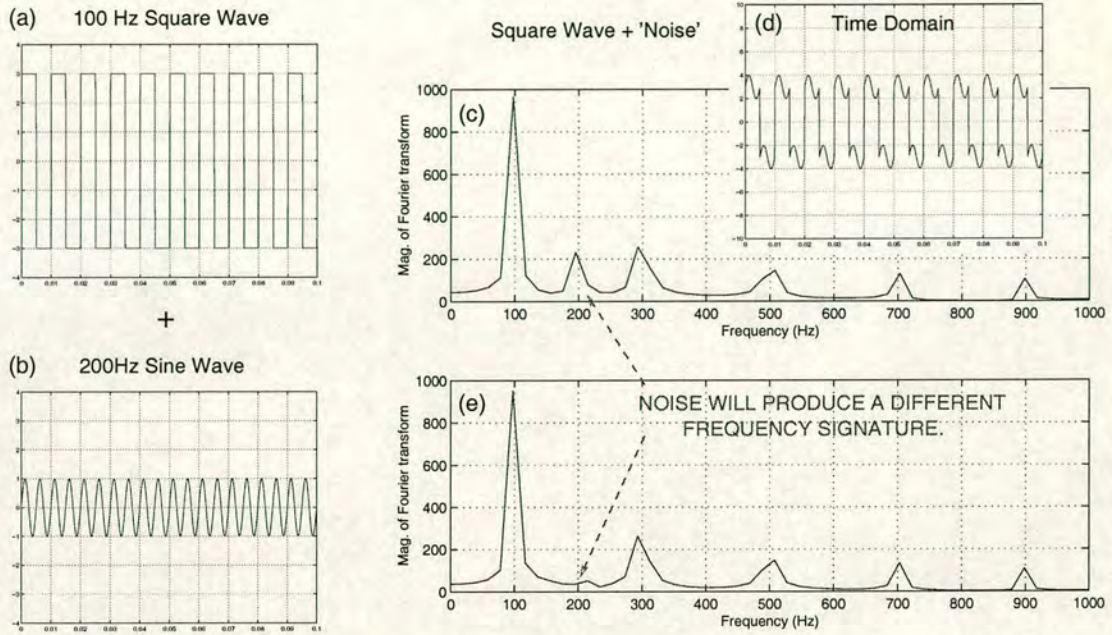
### 3.3 Focal Plane Extraction of Fundamental Temporal Frequency

With the failure of the dyadic tree algorithm, attention shifted to alternative approaches to the problem. Consider figure 3.4, which shows temporal and frequency domain representations of a 100 Hz square wave, with and without an additional 200 Hz sine wave. Imagine the sine wave is a noise signal, possibly caused by two objects contributing to the same pixel's intensity variation. The difference in the frequency domain signals can clearly be seen, suggesting a further problem with the dyadic tree algorithm. The tree filterbank technique is unable to differentiate





between signal and noise, producing a different frequency signature dependent on the relative strength of the signal and the noise.

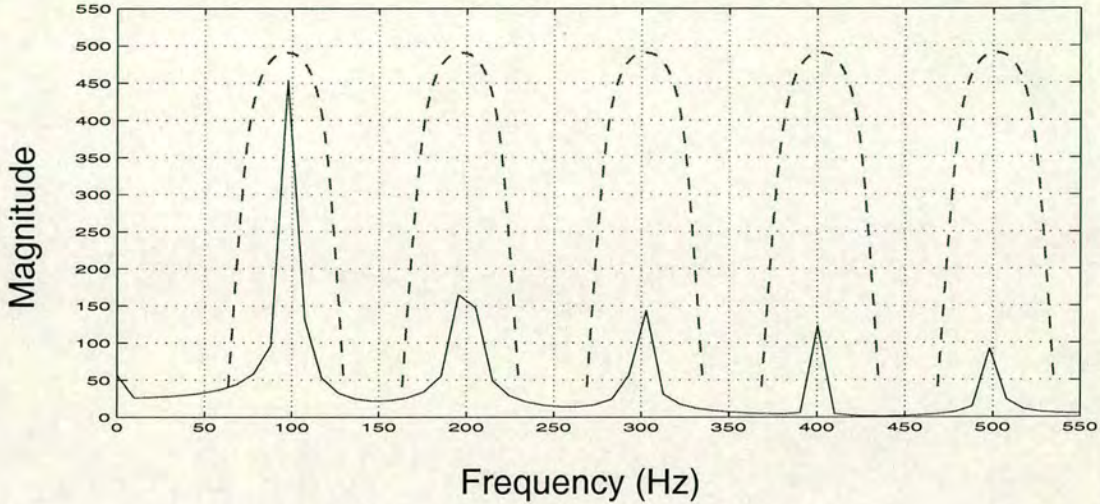


**Figure 3.4:** Temporal and Frequency Domain Representations of 100 Hz Square Wave, with and without 200 Hz Sine Wave Noise Source:(a) 100 Hz square wave, (b) 200 Hz Sine wave (noise), (c) frequency domain version of sine wave combined with square wave, (d) temporal domain version of sine wave compared with square wave, (e) frequency domain version of square wave only. The addition of the noise changes the frequency domain signal, highlighting a potential problem with the dyadic tree algorithm.

This potential problem became the inspiration for another approach to the extraction of frequency signatures from visual data. Once again, the idea relies on filterbanks being placed in the frequency domain, with the outputs from each giving an estimate of the energy in that band. However, the approach differs from the dyadic tree in that the filters are programmable and placed depending on a calculation of the fundamental frequency, as depicted in figure 3.5. The first step involves the calculation of the fundamental frequency as accurately as possible, possibly using focal plane processing techniques. A tunable band pass filter is then placed at this frequency, and at the first four integer multiples, ensuring that each is in a *sensitive* place. In effect, the algorithm constructs a *pseudo Fourier-processor*. Obviously, the success of the algorithm depends strongly on the accuracy with the the fundamental frequency is calculated. For this reason, at this stage of the research, the focus moved to finding accurate techniques



for finding the fundamental frequency of temporal variations. An emphasis was placed on algorithms which allowed simple circuit level realisations, as well as elements of focal plane processing.



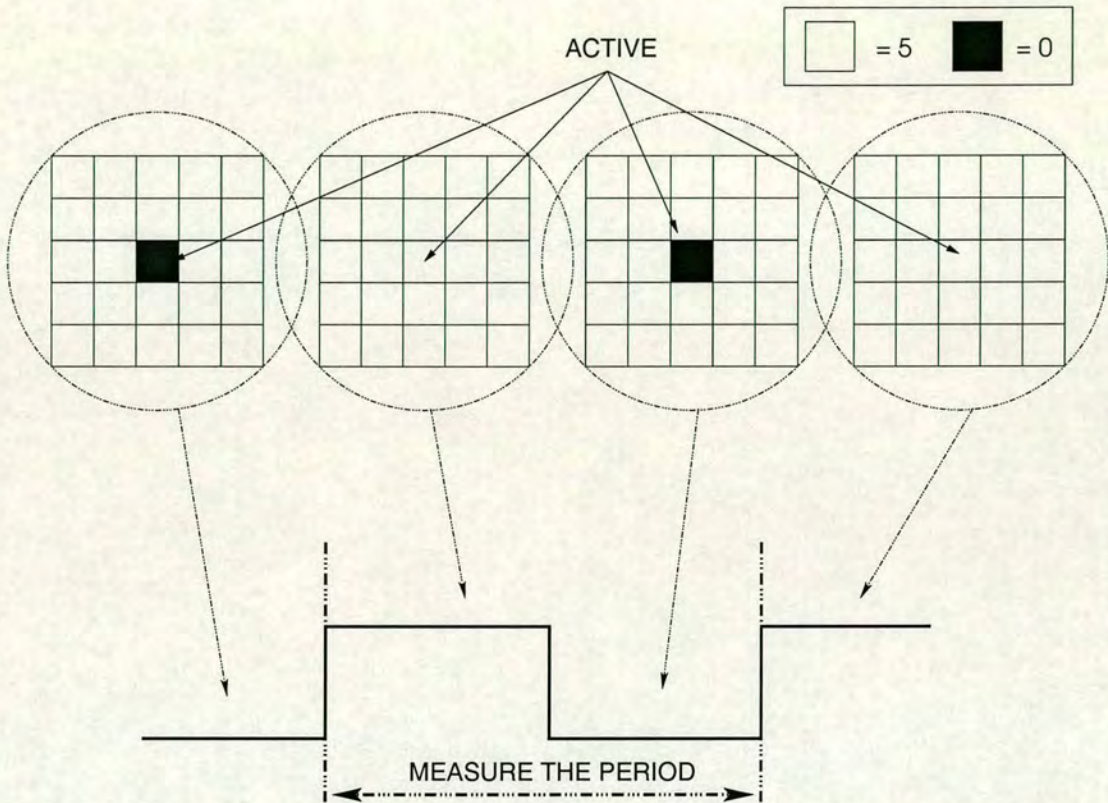
**Figure 3.5:** *Tuning Band Pass Filters to Integer Multiples of the Fundamental Frequency: First, the fundamental frequency is calculated as accurately as possible. A band pass filter is tuned to this frequency, and the first four integer multiples, effectively creating a pseudo Fourier processor.*

### 3.4 The Average vs Active Algorithm

The initial approach to finding the fundamental frequency stemmed from research into implementations of silicon retina circuitry. Many rely on creating an averaged or spatially smoothed version of the incident light intensity, using either resistive grids[14] or current mode techniques [28–30]. This is then compared with the incident photocurrent from a single photoreceptor, in an effort to re-create the centre-surround property of the retina and extract edges accordingly. By continually detecting the appearance and disappearance of edges in a scene with some form of comparator, a series of pulses will result. The frequency of these pulses will correspond directly to the fundamental frequency of the object producing them. This pulse train could then be used to place the band pass filters accordingly. The idea is that the frequency of the pulse train will be simpler to measure accurately than the small signal variations produced by a CMOS image sensor. The process is highlighted in figure 3.6. The 'flashing' of the active pixel corresponds to the intensity change caused by an object producing a temporal frequency of interest. If



the intensity of the active pixel is compared directly with the average of the surrounding pixels using a comparator, the resultant pulse train encodes the fundamental temporal frequency.



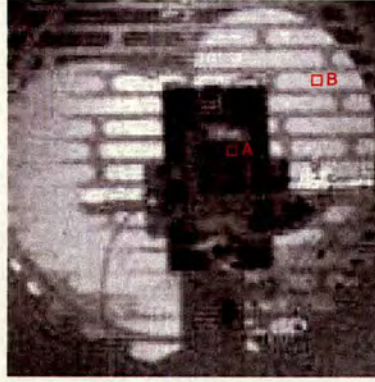
**Figure 3.6:** Operation of the Average vs Active Algorithm: The intensity of the active (central) pixel is compared to the average of the surrounding pixels with a comparator. The resultant pulse train encodes the fundamental frequency of the temporal variations in intensity.

### 3.4.1 Software Simulation of the Average vs Active Algorithm

To test the performance of the algorithm, a series of simulations using the MATLAB processing tool were developed. A simple routine to calculate the average intensity of a programmable area around the active pixel was implemented. Once again, the 'fan' data sequence was used to test the performance of the algorithm. The pixels highlighted in figure 3.7 (a) and (b) were chosen as the active pixels for the luminescence and fan respectively. The algorithm was applied to the test data sequence with the luminescence device flashing at 20 Hz. Figure 3.8 shows the results. The luminescence device produces an intensity variation that approximates a square



wave, hence the frequency domain representation in figure 3.8 (a). As such, the algorithm manages to successfully estimate the fundamental frequency as highlighted in the table in figure 3.8 (c), even with a relatively small averaging area of seven by seven pixels.



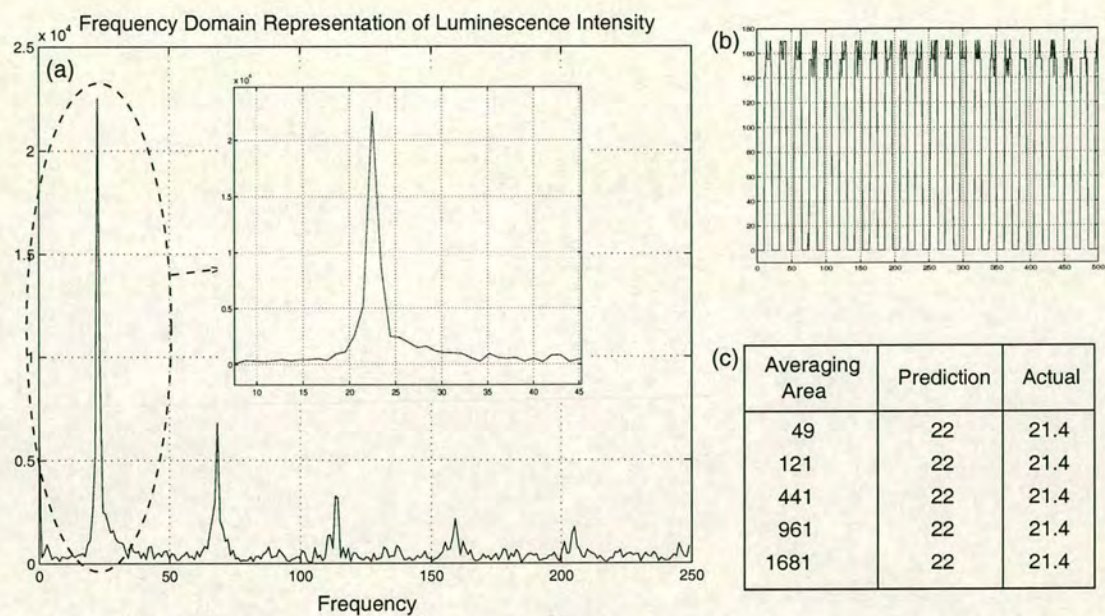
**Figure 3.7:** *Selected Pixels from the 'Fan' Data Sequence used to test the Average vs Active Algorithm: (A) corresponds to the negative luminescence device, while (B) represents the fan itself.*

The results from applying the algorithm to the pixel corresponding to the fan can be found in figure 3.9. It is clear that the algorithm struggles to accurately estimate the fundamental frequency of the fan's temporal intensity variations. The averaging area was increased in steps up to 41 by 41 pixels, yet the estimate is still not accurate. The poor performance of the algorithm regarding the fan, when compared with the luminescence device may be due a number of factors. The fan rotates at higher frequency and also produces a less 'well-defined' temporal variation, as can be seen by comparing figure 3.8(b) with figure 3.9(b). The variation in intensity from the luminescence device is far sharper than the fan, producing a better estimate to a square wave. Whatever the reason, it is clear that the *average vs active* algorithm struggles to resolve the fundamental frequency of the fan's rotation.

### 3.4.2 Comments on the Average vs Active Algorithm

The results from the *average vs active* algorithm suggest that it is unsuitable for the application described in this thesis. Although tested on limited data, it is clear that the results, particularly when tested on a pixel corresponding to the fan, are not accurate enough. Coupled with this are the complexities of a circuit level implementation. For the best estimate of the fan's rotational frequency, the algorithm required an averaging area of nearly 1000 pixels. While simple in software, the interconnect required for such an implementation in silicon would severely limit





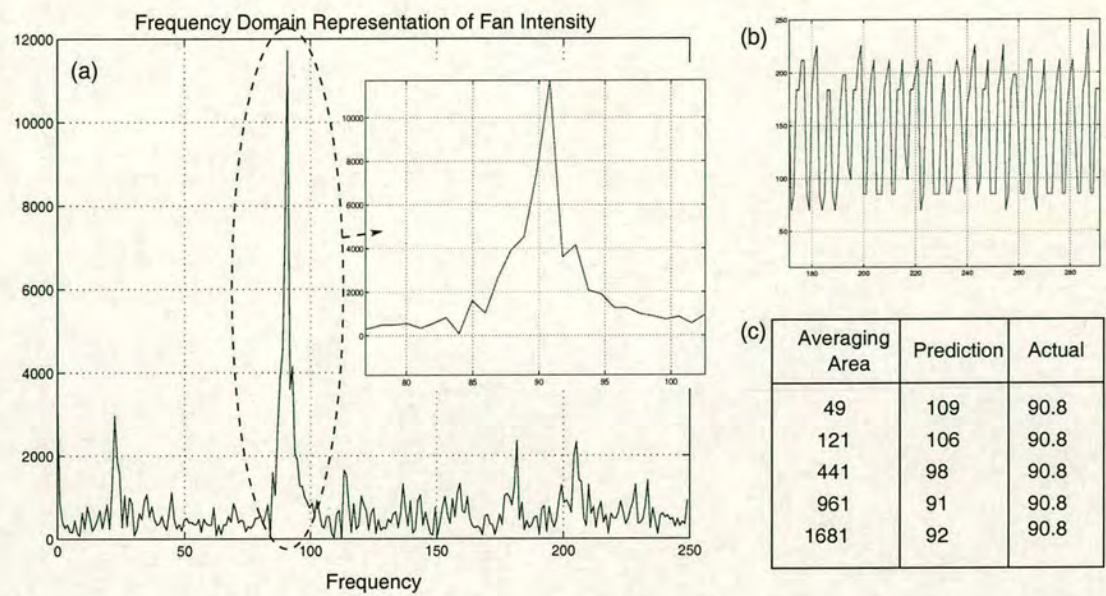
**Figure 3.8:** *Average vs Active Algorithm Applied to Luminescence Device Flashing at 20 Hz: (a) luminescence device active pixel - frequency domain, (b) luminescence device active pixel - temporal domain, (c) fundamental frequency estimate versus actual. No matter the averaging area, the prediction of the fundamental frequency is close to the actual value.*

the feasibility of the resultant image-processor. In addition, a direct comparison of the active pixel's intensity with the surrounding pixel's average with a comparator is not guaranteed to produce pulses. If an area of low or high intensity is included in the average calculation, the active pixels variation may not 'cross' the average value, meaning the comparator will be unable to switch. Any missed pulses will produce highly inaccurate estimates of the fundamental frequency. It may be this phenomenon that is responsible for the inaccuracy of the algorithm when applied to the fan. Given all these reasons, it was decided a circuit level implementation of the *average vs active* algorithm was too large and inaccurate to be a feasible solution to the problem.

### 3.5 The Flashing Pixel Algorithm

With the *average vs active* algorithm highlighting potential problems with grouping pixels together in processing steps, a decision was made to investigate methods of reducing the required interconnectivity. The results of this investigation developed into the *flashing pixel* algorithm.





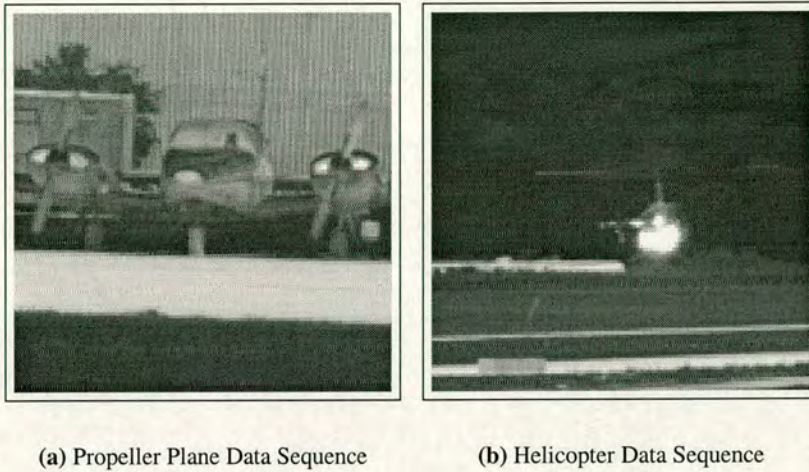
**Figure 3.9:** Average vs Active Algorithm Applied to Fan: (a) fan active pixel - frequency domain, (b) fan active pixel - temporal domain, (c) fundamental frequency estimate versus actual. The algorithm struggles to estimate the fundamental frequency accurately.

The approach is still based on the simple premise of producing pulses that chart the appearance and disappearance of edges in a scene. The frequency of the pulse train then directly encodes the fundamental frequency of the temporal intensity variation within that scene. However, a simpler approach to edge-enhancement was required, motivated by the short-comings of the *average vs active* algorithm. Attention shifted to work performed by Dudek et al[83–85, 88], which detailed the development of a general purpose analogue microprocessor. The targeted application was image processing, with the microprocessor allowing convolution of image data with edge enhancing kernels, such as the Laplacian seen in figure 2.9. The circuitry employed switched current techniques and, crucially, nearest neighbour connectivity to perform the necessary processing. Such techniques suggested a simple method of enhancing the edges in a scene, which might serve as a useful front-end for the *Flashing Pixel* algorithm.

### 3.5.1 Laplacian Mask vs Half-Laplacian Mask

The Laplacian mask highlighted in figure 2.9 is an attempt to model the mexican-hat response of the retina. As such, when applied to images such as those in figure 3.10, both positive and





**Figure 3.10:** *Selected Frames from the Propeller Plane and Helicopter Data Sequences*

negative intensity gradients are enhanced, as depicted in figure 3.11. A three by three Laplacian kernel (figure 3.11 (a)) was convolved with every pixel in figure 3.10 (a) and (b), producing the results in figure 3.11 (b) and (c).

It is clear that the Laplacian convolution kernel provides equal weighting to both positive and negative intensity gradients. While this is essential for standard edge-enhancing algorithms, in this application it may produce spurious pulses, reducing the accuracy of the fundamental frequency calculation. The aim is to edge enhance the scene, using a comparator to create pulses according to the appearance and disappearance of edges. If a pulse is produced for both positive and negative edges, then both rising and falling edges of an object may produce a pulse, effectively creating a spurious double pulse, where only one is required. For this reason, an emphasis was placed on the development of a new mask, which highlights only positive intensity gradients, ignoring the corresponding negative gradients.

The operation of the desired convolution mask is to enhance positive intensity gradients, while suppressing negative intensity gradients. The adopted approach was to adapt the weighting of the three by three Laplacian mask to effectively *skew* the output. After much experimentation in MATLAB, the convolution kernel seen in figure 3.12 (a) was chosen as the best solution. It is clear from figures 3.12 (b) and (c) that the desired effect has been produced. A direct comparison of the results can be found in figure 3.13.

From the propeller plane data sequence, the different effect of the Laplacian and half-Laplacian



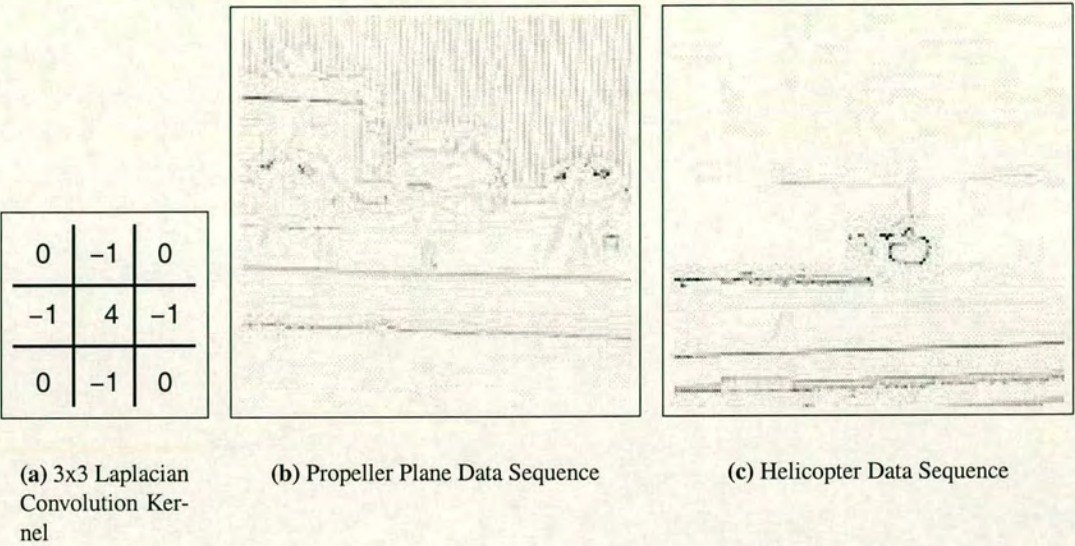


Figure 3.11: Edge-Enhancement with the Laplacian Convolution Kernel

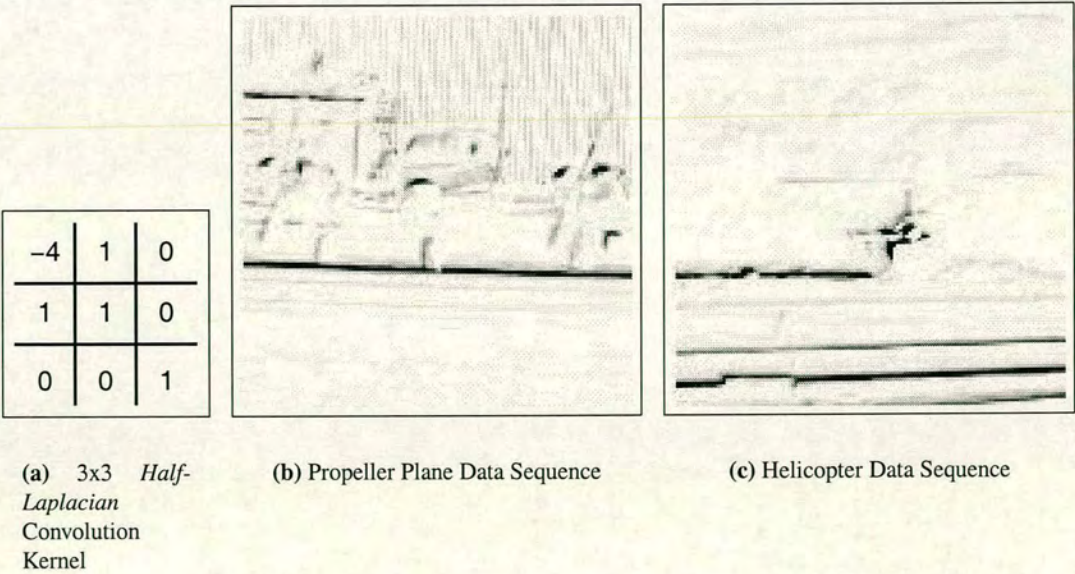


Figure 3.12: Edge-Enhancement with the Half-Laplacian Convolution Kernel



masks can be clearly seen in the enhancement of the runway underneath the plane in figure 3.13 (a). As expected, the Laplacian kernel highlights both positive and negative intensity gradients, as seen in figure 3.13 (c). However, the half-Laplacian mask, depicted in figure 3.13 (e), strongly enhances the positive intensity gradient, dark to light as we move both left to right and top to bottom, while completely ignoring the corresponding negative intensity gradient. A similar effect with the engine of the helicopter can be seen in figure 3.13 (d) and (f). Notice also that the edges enhanced by the half-Laplacian mask appear stronger than the equivalent by the standard Laplacian technique. This may make subsequent thresholding of the edges an easier task, producing a more robust input to the algorithm.

This hypothesis was tested by selecting an individual pixel from each of the propeller plane and helicopter data sets. That pixel was then processed with both the half-Laplacian and Laplacian masks, with the output from each thresholded to produce a series of pulses. The results can be seen in figure 3.14. As expected, the output from each edge-enhancing mask is centred at zero, only moving when an edge is present in the scene. It is clear that for both selected pixels, the half-Laplacian mask produces a 'stronger' output, making thresholding a simpler task. The resultant pulse trains are a better estimate of the fundamental frequency than those from the Laplacian mask algorithm.

### 3.5.2 Development of the *Flashing Pixel* Algorithm

The aim of the *flashing pixel* algorithm is to first enhance the edges in a scene, using a three by three half-Laplacian mask. These edges are then thresholded using a comparator to produce a series of pulses, the frequency of which corresponds to the fundamental frequency of the intensity variation. To this end, a series of MATLAB simulations were performed to test the validity of the approach, using the plane and helicopter data sequences as inputs. Initially, individual pixel cells from each data set were selected, with the algorithm applied to their intensity variation. The pixels chosen to test the algorithm can be seen in figure 3.15. Pixels were selected if they experienced a temporal frequency of interest.

Figure 3.16 highlights the processing steps performed by the algorithm, when applied to pixel (100,72) from the propeller plane data sequence. The half-Laplacian mask is applied to the selected pixel, taking weighted information from its nearest neighbours. The output from the mask is then thresholded, to produce a series of pulses. The frequency of the pulse train corresponds to that of the fundamental frequency of the original intensity variation. This information

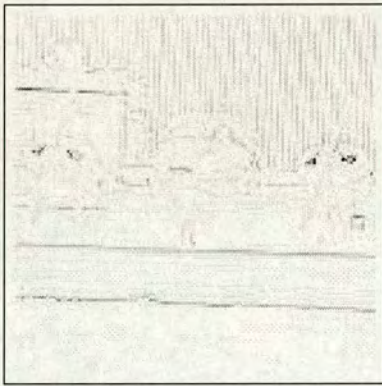




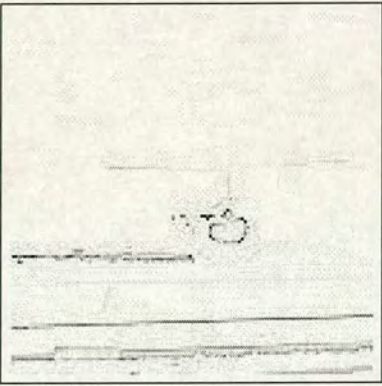
(a) Original Frame from Propeller Plane Data Sequence



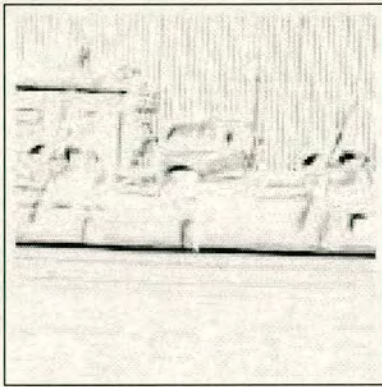
(b) Original Frame from Helicopter Data Sequence



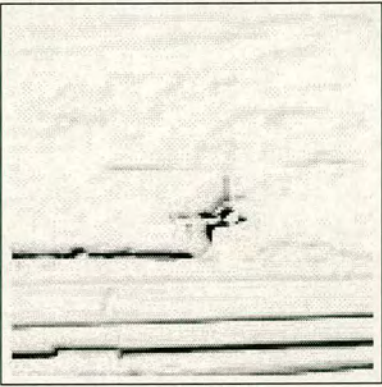
(c) Plane Edge-Enhanced with Laplacian Mask



(d) Helicopter Edge-Enhanced with Laplacian Mask



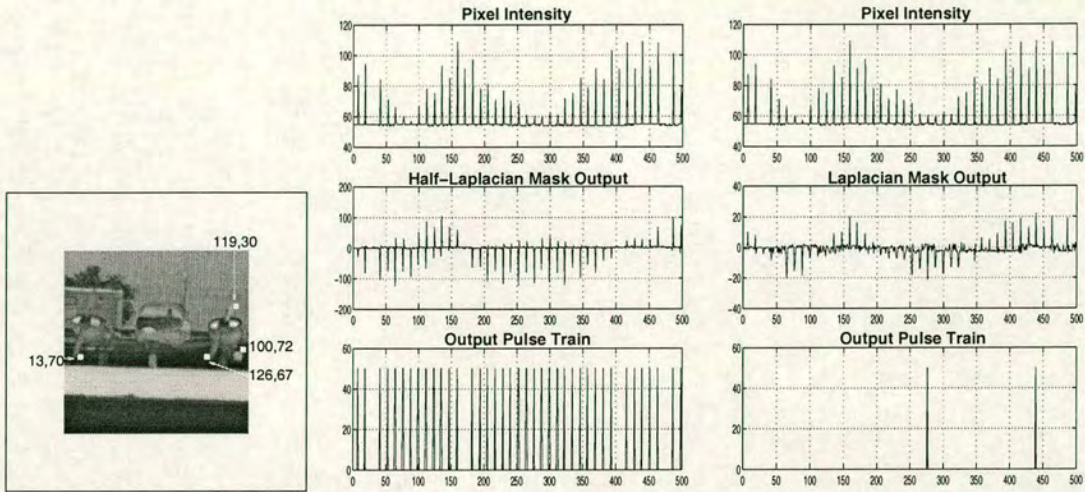
(e) Plane Edge-Enhanced with Half-Laplacian Mask



(f) Helicopter Edge-Enhanced with Half-Laplacian Mask

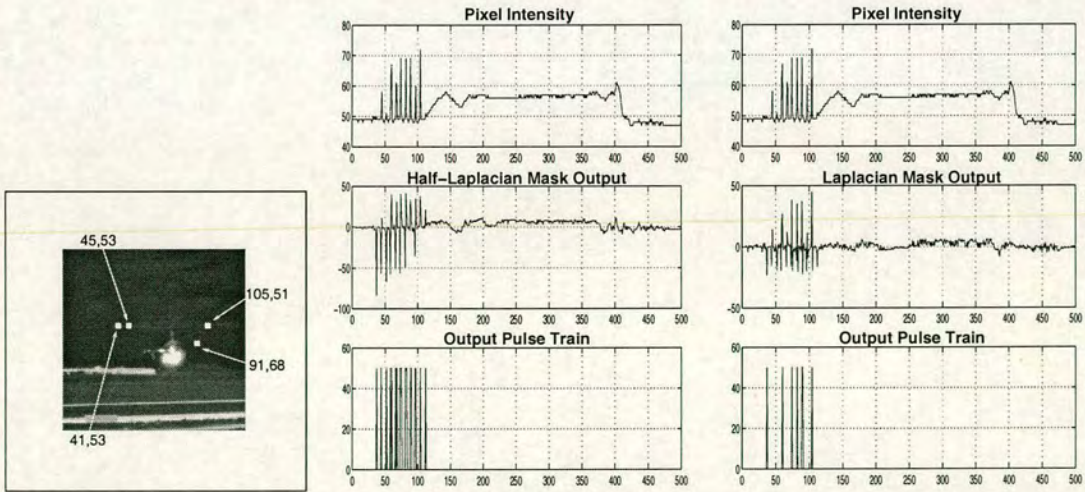
**Figure 3.13:** Comparison of Edge-Enhancement Convolution Kernels





(a) Original Frame from Propeller Plane Data Sequence

(b) Comparison of Convolution Kernels for Plane Data, Pixel 100,72

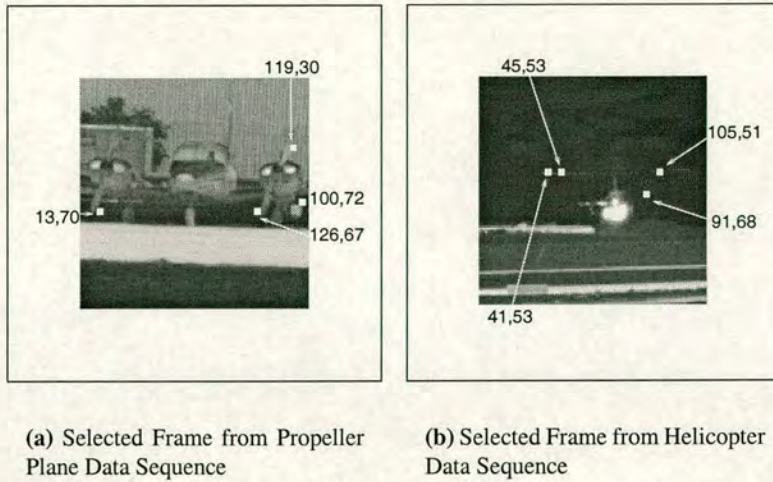


(c) Original Frame from Helicopter Data Sequence

(d) Comparison of Convolution Kernels for Helicopter Data, Pixel 45,53

**Figure 3.14:** Comparison of Half-Laplacian and Laplacian Convolution Kernels on Single Pixel Data



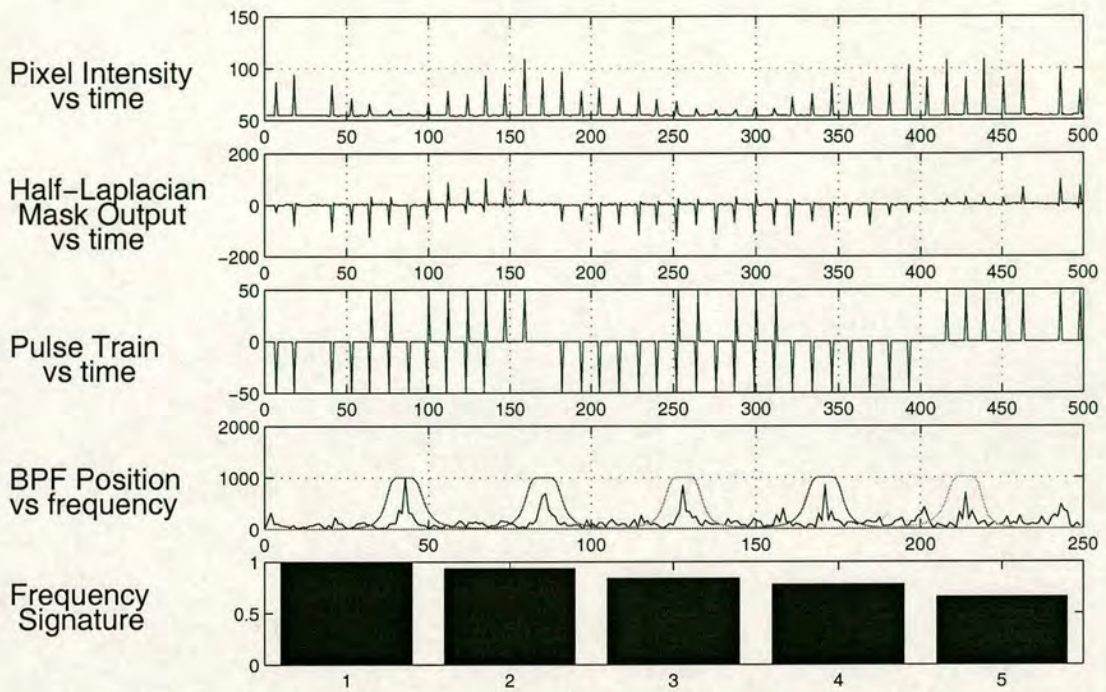


**Figure 3.15:** *Selected Frames from the Plane and Helicopter Data Sequences, Highlighting the Chosen Pixel Locations*

is then used to place a series of band-pass filters at integer multiples of the fundamental. The fourth row superimposes the position of the band-pass filters on the frequency domain representation of the pixels intensity variation. An estimate of the energy at the output of each filter produces a frequency signature of the object that produces the temporal intensity variations. It is clear from figure 3.16 that the algorithm successfully places the band pass filters in the relevant area in the frequency domain. For the purposes of this simulation, a frequency counting algorithm that ignores certain gaps in the pulse train was employed. The original pixel intensity displays a cyclical variation in the magnitude of the propeller blade, due to the sampled nature of the image data sequence. It is envisaged that a continuous time implementation of the algorithm would produce a more constant pulse train, allowing the use of a simpler frequency counting technique. For all subsequent simulations, a threshold level was selected to ensure that the required pulse train would be created. In a circuit-level implementation, the thresholding will be performed by a comparator circuit, meaning that the temporal variations will be compared to their quiescent DC level. As such, the thresholding stage in these simulations is an analogy to the process employed in the final version, and is chosen to ensure the greatest number of output pulses.

The same algorithm was applied to a pixel from the helicopter data sequence, the results of which can be found in figure 3.17. The fundamental frequency of the helicopter's main propeller is such that the harmonics are folded down to lower frequencies, hence the unorthodox



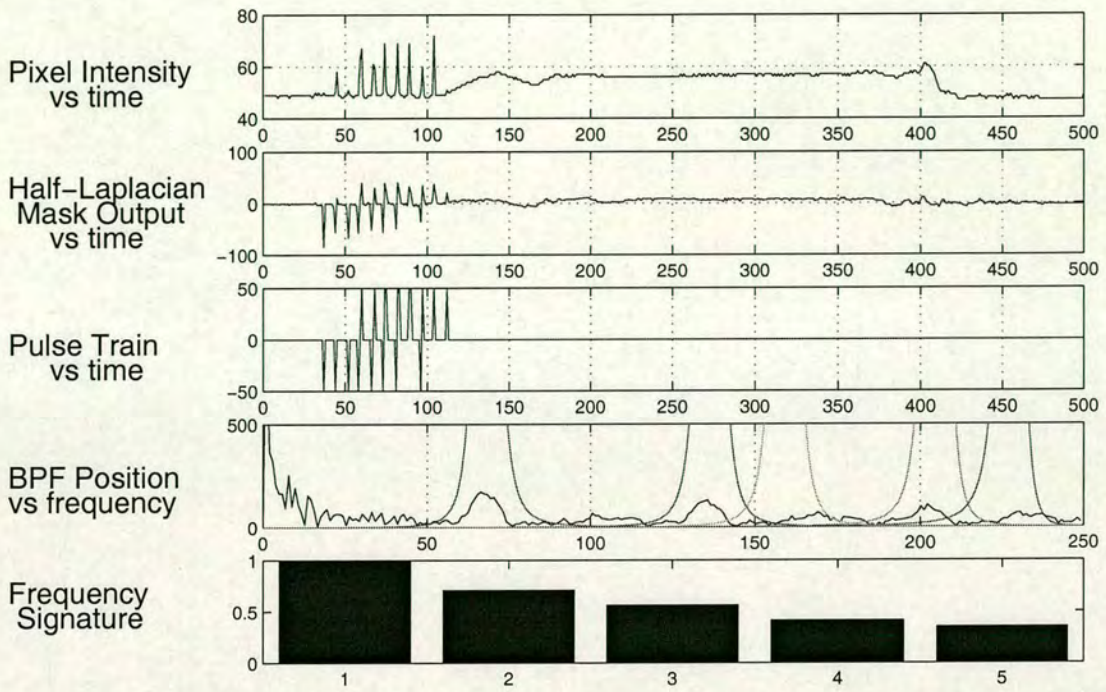


**Figure 3.16:** Operation of the Flashing Pixel Algorithm Applied to Pixel 100,72 in the Plane Data Sequence: The half-Laplacian mask is applied to the intensity variation before being thresholded. The frequency of the resultant pulses is used to place a series of five band pass filters, the energy content from each displayed in the frequency signature bar-chart.

appearance of the frequency content. In addition, the frequency domain representation exhibits a large DC offset, due probably to the helicopter's engine passing through the selected pixel. Nevertheless, the algorithm does place the band pass filters over the frequency content, ignoring the DC offset. To further test the technique, the algorithm was applied to all four highlighted pixels in figure 3.15. For the plane data set, the positioning of the band pass filters can be seen in figure 3.18. It is clear that for each of the four selected pixels, the algorithm successfully positions the first band pass filter on the fundamental frequency. The subsequent band pass filters are accurately placed on the integer multiples of the fundamental, building a Fourier-style decomposition of the original signal.

The same experiment with the selected pixels from the helicopter data set can be seen in figure 3.19. The algorithm copes well with (a),(c) and (d), but is slightly inaccurate with pixel (41,53). The slight difference between the calculated fundamental frequency and the actual value becomes more apparent at the higher frequency integer multiples. However, in general,





**Figure 3.17:** Operation of the Flashing Pixel Algorithm Applied to Pixel 45,53 in the Helicopter Data Sequence

the algorithm copes well with the plane and helicopter data sequences.

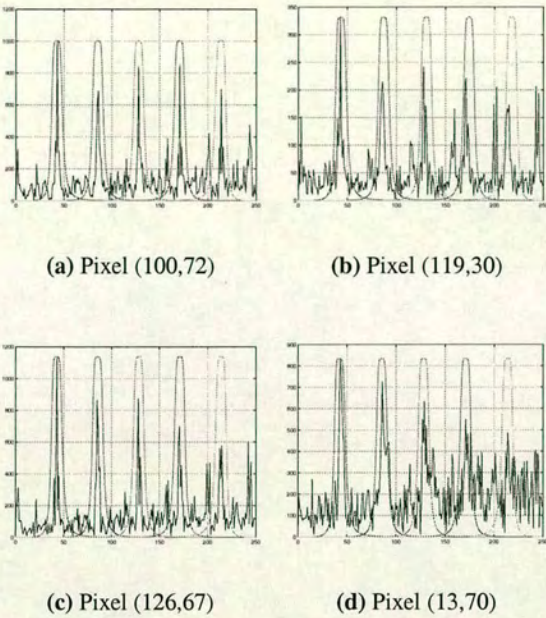
### 3.5.3 Noise Analysis of the Flashing Pixel Algorithm

With the results from the *flashing pixel* algorithm appearing promising, physical limitations from a CMOS implementation had to be considered and factored into the simulations. CMOS imagers traditionally suffer more from noise than their CCD equivalents[89]. Of particular interest regarding this project are fixed pattern noise and random transient noise.

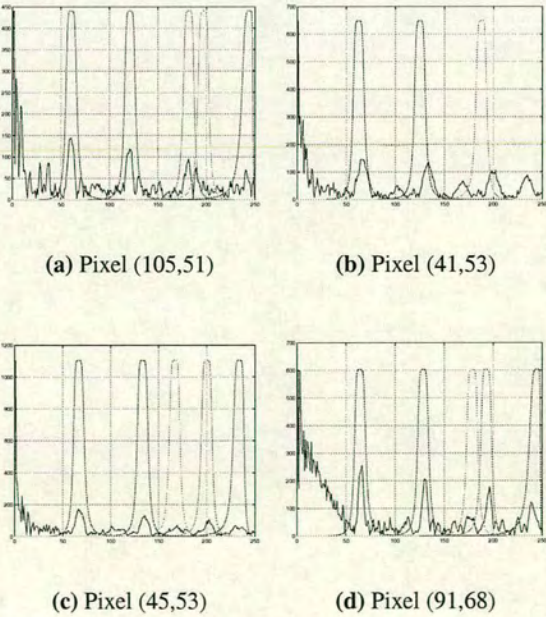
#### 3.5.3.1 Fixed Pattern Noise: Appearance and Causes

In a standard CMOS process, mismatch between process parameters across the surface of the die produce offsets which effect the performance of the implemented circuitry. Such mismatch effects passive devices such as resistors and capacitors as well as the active MOSFETs. In a CMOS image sensor, such mismatch manifests as fixed pattern noise, so-called because it is strictly a DC phenomenon. If every pixel in the imager is illuminated with the same light





**Figure 3.18:** *Band Pass Filter Positioning with the Plane Data-Set, using the half-Laplacian Mask algorithm*



**Figure 3.19:** *Band Pass Filter Positioning with the Helicopter Data-Set, using the half-Laplacian Mask algorithm*



source, the output image will display a pattern corresponding to the mismatch. From a circuit level point of view, techniques such as common centroid layout, dummy transistors and ensuring transistors are large and given the same orientation can be adopted[90]. When considering an image sensor, approaches such as 'remembering' the fixed pattern noise and subtracting it from each frame can help to reduce the effect. Such correlated double sampling techniques[6] involve taking two readings of the photocurrent to be used differentially in subsequent processing stages.

For the purposes of this research, fixed pattern noise can be modelled as a different DC offset applied to each pixel, which is then held constant for each frame in the data sequence. The amount of fixed pattern noise added to the simulation is variable, with a threshold setting the maximum value. For this simulation, the threshold was set such that the maximum possible value of fixed pattern noise is 7.8% of the intensity range. This manifests as a maximum possible noise value of  $\pm 10$  (20 units from the maximum intensity value of 255) added to the actual intensity value for that particular pixel. To allow direct comparison, the selected pixels for the propeller plane and helicopter data sets were those highlighted in figure 3.15.

3.5.3.2 Simulating the *Flashing Pixel Algorithm* with Fixed Pattern Noise

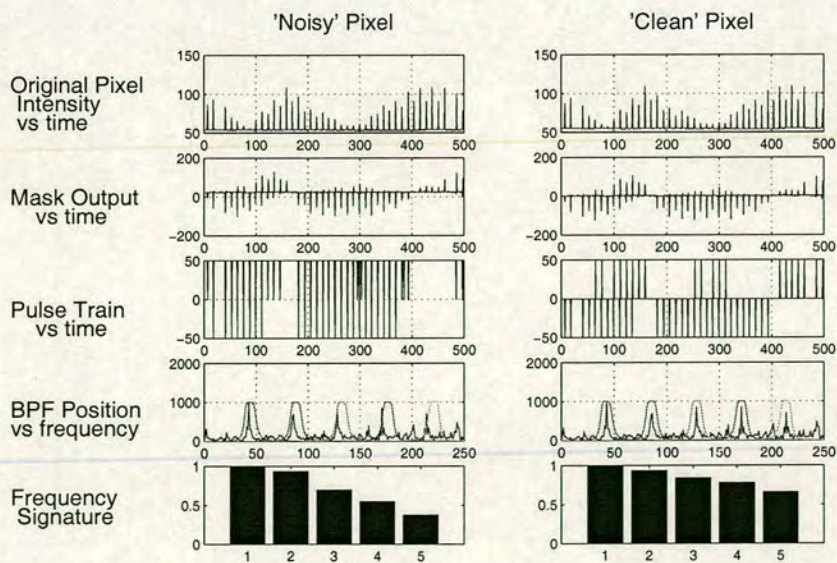


Figure 3.20: Effect of Fixed Pattern Noise on the Flashing Pixel Algorithm: Plane Pixel (100,72)

The effect of fixed pattern noise on the flashing pixel algorithm can be seen in figure 3.20,

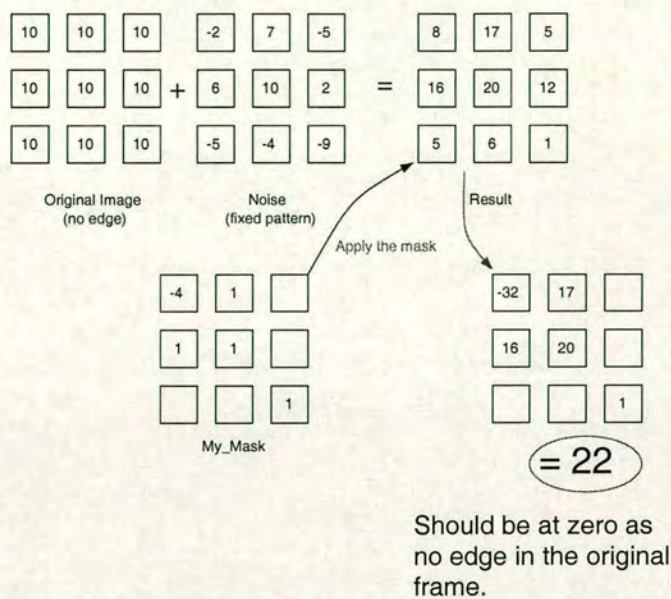


when applied to the plane data set. The algorithm was applied to the selected pixel, both with and without the added fixed pattern noise, such that a direct comparison of the effects can be made. The left hand side depicts the algorithmic flow for the noisy pixel, while the right is the same pixel without noise added. At first glance, the difference between the *noisy* pixel and its *clean* equivalent appear minimal, with both appearing to place the band pass filters relatively accurately. However, looking at the output from the half-Laplacian mask for both shows the presence of a DC offset in the noisy pixel. This is a potential problem for the algorithm, as the next phase relies on thresholding the output from the mask to produce a pulse train. In this case, the pulses for the noisy pixel appear relatively close to those for the clean version, but a few spurious pulses are produced, compromising the accuracy with which the band pass filters are positioned. The presence of this DC offset can be explained by the nature of fixed pattern noise and the operation of the half-Laplacian mask, as highlighted in figure 3.21. The original image contains no edges, yet the fixed pattern noise produces an artificial edge which is subsequently amplified by the half-Laplacian convolution kernel. In effect, the discontinuities in the fixed pattern noise are amplified, producing spurious edges where none in fact exist. As the sum of the weights in the half-Laplacian mask equal zero, the output should sit at zero when no edge is present. When the result from the application of the mask is computed, an intensity of 22 units is produced, simply by the presence of fixed pattern noise.

It is clear that fixed pattern noise creates potential problems for the algorithm in its current state, in the form of a DC offset. The same experiment applied to the helicopter data set produced the results in figure 3.22. Once again, the DC offset caused by the presence of fixed pattern noise effects the accuracy of the technique. with spurious pulses produced as a result.

However, a simple adaptation could be made to solve the problem. If the mask is applied as usual, but the output is passed through a high pass filter before thresholding, any DC offset caused by fixed pattern noise will be removed. This led to two new versions of the algorithm. The first is termed the *half-Laplacian HPF algorithm*, and is exactly as described above with the DC offset from the mask's output removed with a high pass filter. The second is called the *no mask algorithm* and simply involves high pass filtering the original pixel intensity before thresholding, without applying an edge-enhancing mask. It was felt that any degradation in performance of such a technique may be offset by the simplified circuit-level implementation. Not applying a convolution kernel to the image means that no nearest-neighbour connectivity is required, effectively allowing each pixel to operate independently. The two techniques are





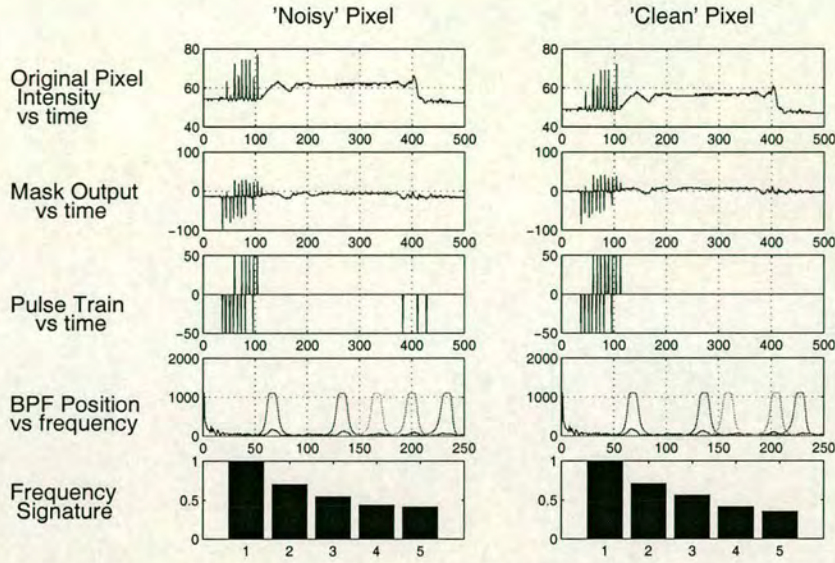
**Figure 3.21:** DC Offset caused by Fixed Pattern Noise with the Flashing Pixel Algorithm: Despite the absence of an edge in the original frame, the fixed pattern noise mimics the appearance of an edge, and is thus amplified by the half-Laplacian convolution kernel

depicted pictorially in figure 3.23, together with the original version.

**3.5.3.3 Simulating the Half-Laplacian HPF Algorithm with Fixed Pattern Noise**

The first variation on the original flashing pixel algorithm uses a high pass filter to remove any DC offset caused by the presence of fixed pattern noise. To test the algorithm, the pixels from the plane and helicopter data sequence highlighted in figure 3.15 were selected. As before, a randomly generated 128 by 128 array was added to each of the 500 frames in the sequence, to model the DC nature of fixed pattern noise. The processing steps when applied to the pixel corresponding to the plane data sequence can be seen in figure 3.24 (a). As before, the left hand side represents the pixel with added fixed pattern noise, while the right hand side is the same pixel without any additional noise. The DC offset caused by fixed pattern noise is clear to see at the output from the half-Laplacian mask in the noisy pixel. However, after high pass filtering this signal, it resembles exactly the same processing stage in the clean pixel. Removing the DC offset allows simple thresholding, with the result that the filterbank is accurately positioned and the underlying frequency signatures for noisy and clean pixels are similar. It is clear that removing the DC level caused by fixed pattern noise with a high pass filtering stage makes the





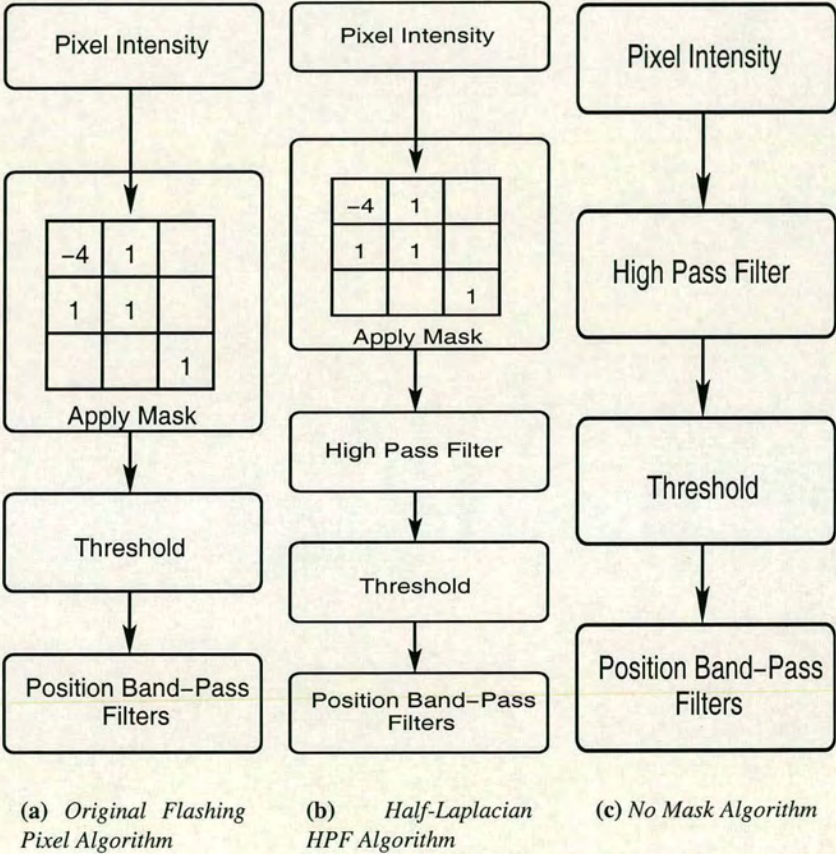
**Figure 3.22:** Effect of Fixed Pattern Noise on the Flashing Pixel Algorithm: Helicopter Pixel (45,53)

noisy pixel behave exactly like its clean counterpart. The same algorithm was applied to the helicopter data set, with the results in figure 3.24 (b). Once again, the DC offset is removed with the filtering stage, resulting in both noisy and clean pixels producing the same frequency signature.

### 3.5.3.4 Simulating the No-Mask Algorithm with Fixed Pattern Noise

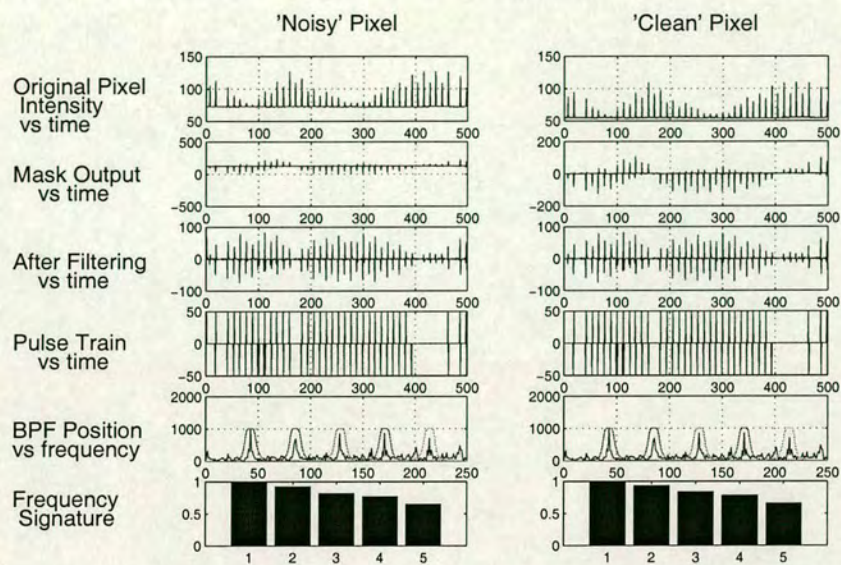
The No-Mask Algorithm was conceived as a simpler alternative to the Half-Laplacian HPF algorithm. By not applying an edge-enhancing convolution kernel to the image, the savings in terms of inter-connect and processing time may prove advantageous for the application concerned. However, the original aim of applying a mask was to enhance edges in the scene, making thresholding a far simpler task. Without the mask, such a thresholding step may be more complex, with the possibility that certain objects may be missed. Nevertheless, given the potential savings from a circuit-level perspective, it was decided to simulate such a system to see how it compares with the other techniques. The algorithm was tested on the same pixels that were selected for the flashing-pixel and half-laplacian HPF versions, to allow a direct comparison of results. The results when applied to the plane data sequence can be seen in figure 3.25 (a). The pixel's intensity is passed directly through a high pass filter, removing any DC offset and centring it on zero. This is then thresholded to produce pulses, the frequency of which is



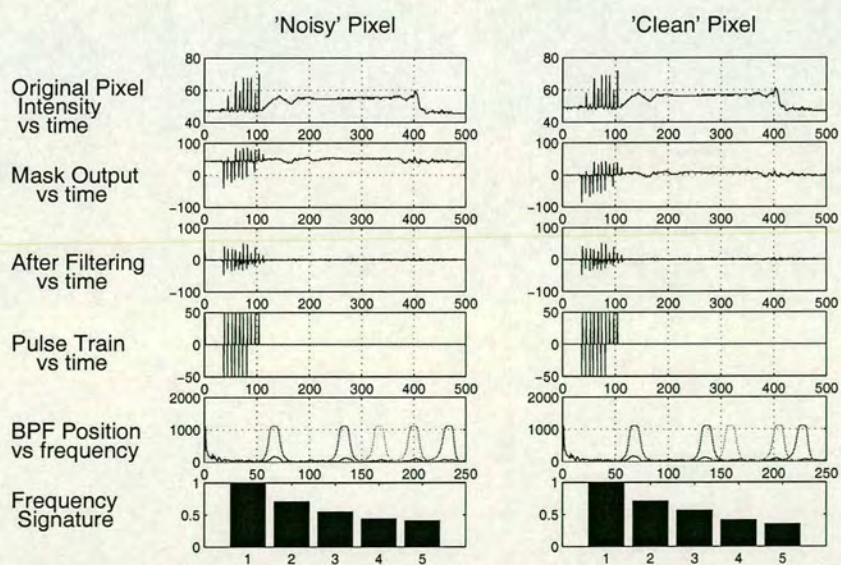


**Figure 3.23:** Differences Between the Flashing Pixel Algorithm, the Half-Laplacian HPF Algorithm and the No-Mask Algorithm.





(a) Propeller Plane Pixel (100,72)



(b) Helicopter Pixel (45,53)

Figure 3.24: Effect of Fixed Pattern Noise on the Half-Laplacian HPF Algorithm



used to position band pass filters. In this case, the combined effect of the actual DC bias of the pixel intensity and the fixed pattern noise is completely removed with the high pass filter. In effect, only the transient properties of the signal are passed to the thresholding stage. As such, both clean and noisy pixels produce the same pulse train when thresholded, ensuring similar underlying frequency signatures. The results when the algorithm is applied to the helicopter data set can be found in figure 3.25 (b). Once again, the DC is removed with the high pass filter and the transient signal is thresholded. In both cases, the results are comparable with the half-Laplacian HPF algorithm, suggesting that this technique may be a viable alternative.

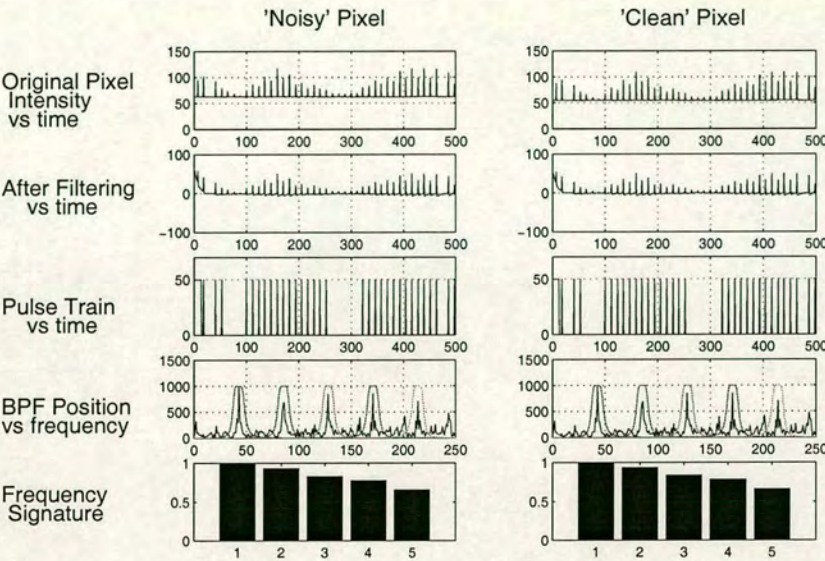
### 3.5.3.5 Comments on Fixed Pattern Noise

The aim of the simulations within the previous section was to ascertain the robustness of the flashing pixel algorithm to a common source of noise with CMOS imagers. It was discovered that fixed pattern noise manifests as a fixed DC offset, which can vary from pixel to pixel. As such, the original approach of applying an edge-enhancing mask to the image data was proved to be unsuitable, as edges caused by the fixed pattern noise were enhanced together with the actual edges in the scene, producing a DC pedestal onto which the mask's output is superimposed. However, a solution in the form of high pass filtering the signal to remove this DC offset was discovered. This led to the development of two alternative algorithms, the *half-Laplacian HPF* algorithm and the *no-mask* algorithm, both of which were found to be robust to fixed pattern noise. The filter simply removes any DC level created by fixed pattern noise, ensuring that both techniques operate well despite the presence of fixed pattern noise. It is clear that the robustness to fixed pattern noise of both the half-Laplacian HPF algorithm and the no-mask algorithm is *data-independent*, in that no matter the magnitude of the noise, the systems will function correctly.

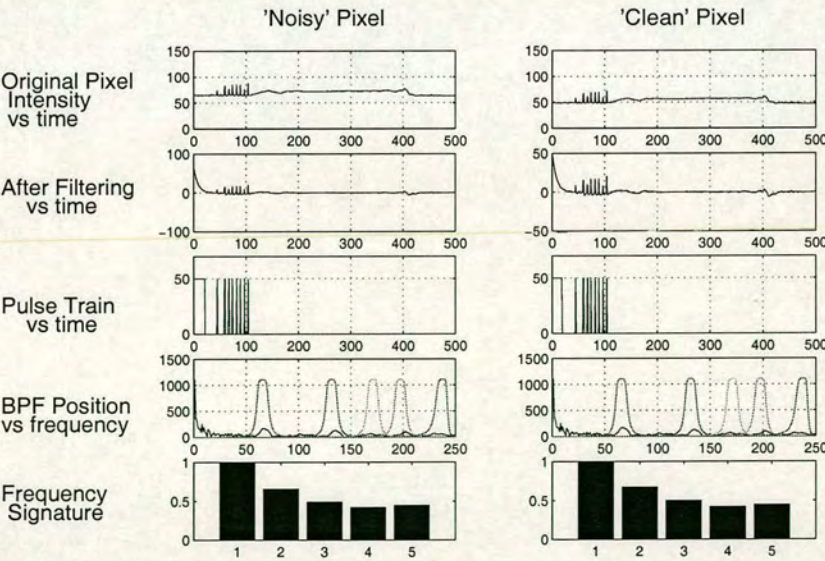
### 3.5.3.6 Random Transient Noise: Appearance and Causes

Each component in a CMOS circuit, either passive or active, introduces some element of random transient noise. Such noise exhibits random amplitude versus time and has an average of zero when measured over extended time periods. Thermal noise, shot noise and flicker noise are all examples of random transient noise[91], which can combine to produce the total noise for the circuit. The nature of such noise means it is unavoidable, yet its effects can be diminished with clever design at both algorithm and circuit levels. Random transient noise is termed





(a) Propeller Plane Pixel (100,72)



(b) Helicopter Pixel (45,53)

Figure 3.25: Effect of Fixed Pattern Noise on the No-Mask Algorithm



*white-noise* in that it appears at all frequencies and therefore cannot be removed with a filtering step.

Transient noise was modelled by creating a random temporal signal to add to each pixel's temporal intensity variation. The magnitude of this noise signal was set at roughly 3 % of the maximum signal swing, modelling what might be expected from a circuit-level implementation. The amplitude of random transient noise is less than that from fixed pattern noise. Once again, the pixels highlighted in figure 3.15 were used to perform the simulations.

#### **3.5.3.7 Simulating the *Flashing Pixel Algorithm* with Random Transient Noise**

The results from applying transient noise to the flashing pixel algorithm can be seen in figure 3.26. The added noise is clearly visible in the left hand column, adding a 'fuzziness' to the pixel intensity when compared with the clean signal in the right hand side. This in turn produces a noisy output from the mask, which introduces problems for the thresholding step. Despite the spurious pulses, the frequency counting algorithm manages to place the band pass filters fairly accurately in both plane and helicopter data sets, with the underlying frequency signatures similar to the clean counterparts.

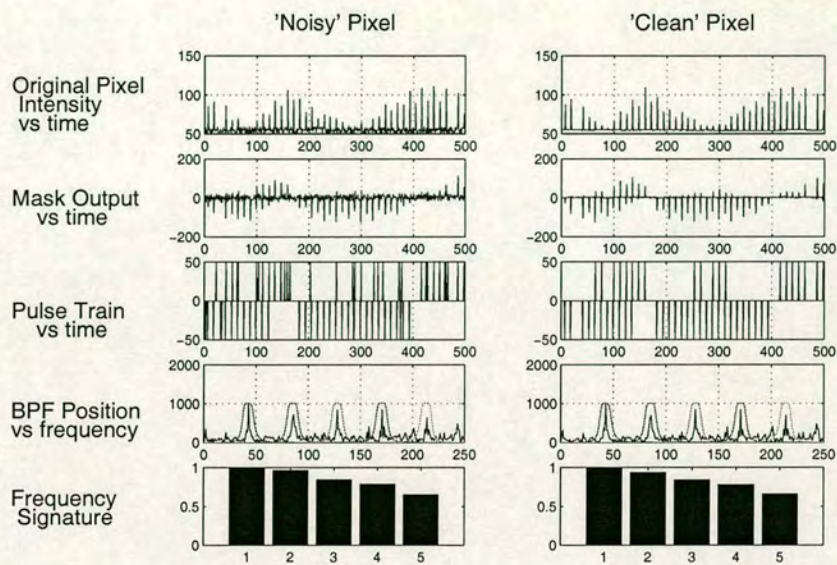
#### **3.5.3.8 Simulating the *Half-Laplacian HPF Algorithm* with Random Transient Noise**

The same simulation setup was applied to the half-Laplacian HPF algorithm, with the results in figure 3.27. The presence of temporal noise once again serves to produce spurious pulses, but the algorithm manages to place the filterbank fairly accurately for the plane data sequence. However, the algorithm struggles with the helicopter sequence, producing different frequency signatures. Changing the thresholding level may improve the performance, but it is clear that random transient noise poses difficult questions of the technique.

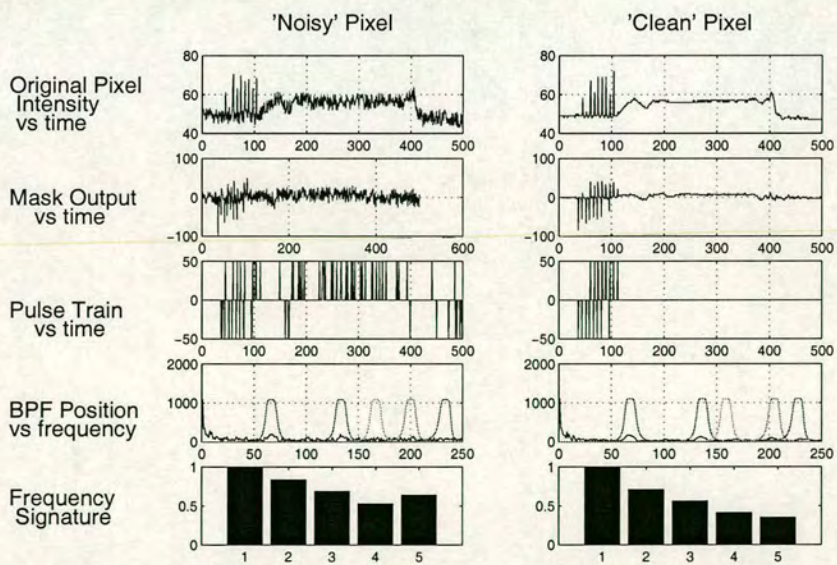
#### **3.5.3.9 Simulating the *No-Mask Algorithm* with Random Transient Noise**

Finally, transient noise was added to the no-mask version of the algorithm. The results for both plane and helicopter data sets can be seen in figure 3.28. As before, the presence of noise creates spurious pulses at the thresholding stage, particularly with the helicopter data sequence. However, the system seems to cope well with the plane data set, producing fewer spurious pulses than both the flashing pixel and half-Laplacian HPF versions for the same pixel.





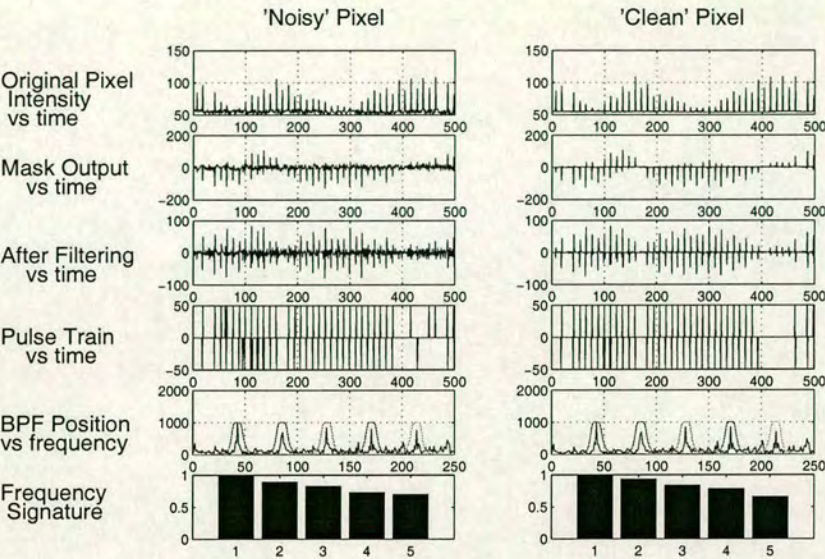
(a) Propeller Plane Pixel (100,72)



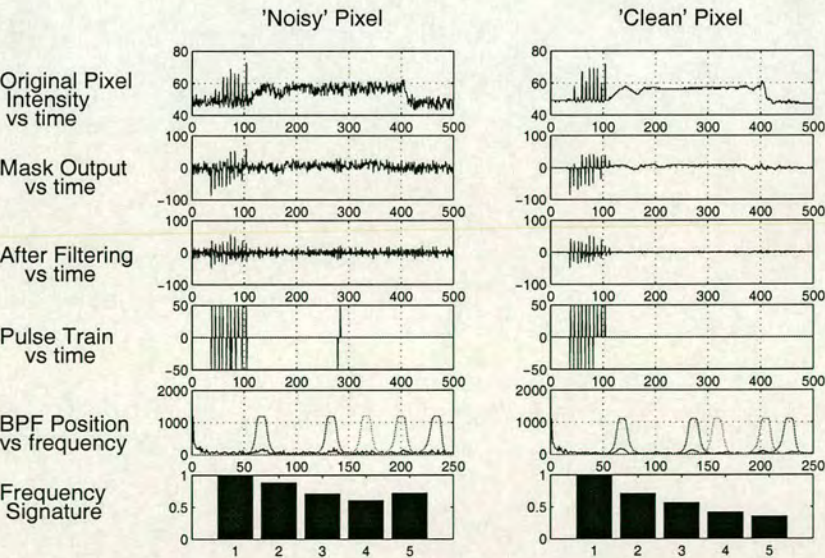
(b) Helicopter Pixel (45,53)

**Figure 3.26:** Effect of Random Transient Noise on the Flashing Pixel Algorithm





(a) Propeller Plane Pixel (100,72)



(b) Helicopter Pixel (45,53)

Figure 3.27: Effect of Random Transient Noise on the Half-Laplacian HPF Algorithm



As such, the frequency signatures for the plane sequence for both noisy and clean pixels are extremely close. It may be that the absence of an edge-enhancing mask in some way helps the no-mask algorithm out-perform its counterparts.

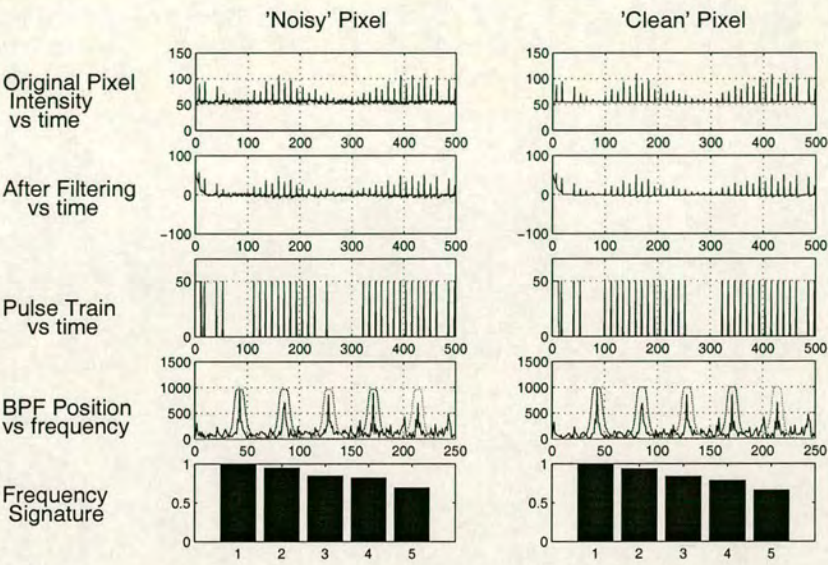
#### 3.5.3.10 Comments on Random Transient Noise

It is clear that transient noise creates more problems for the three versions of the algorithm than fixed pattern noise. In effect, a random signal in terms of both amplitude and frequency content is superimposed onto the existing intensity variation for each pixel, making the job of extracting the fundamental more difficult. If the magnitude of the noise is comparable to the size of the signal of interest, no amount of clever processing will allow the two to be separated. However, the size of the noise in the simulations from figures 3.26, 3.27 and 3.28 may be slightly larger than could be expected from a circuit level implementation. A value of 3% of the maximum possible signal swing was selected for the analysis. This manifests as approximately 8 intensity units out of the maximum 255 produced by the camera. As the maximum signal swing for the plane data is about 70 units, with that of the helicopter even less at about 30 units, the adopted noise percentage can be considered a harsh test of the algorithms robustness, with a signal to noise ratio of 18.84 dB and 11.48 dB respectively. Despite this, on the whole the three systems cope fairly well, with the thresholding process providing some degree of protection. Unlike with fixed pattern noise, the robustness of the three algorithms to random temporal noise is *data-dependent*, in that the magnitude of the noise effects the performance.

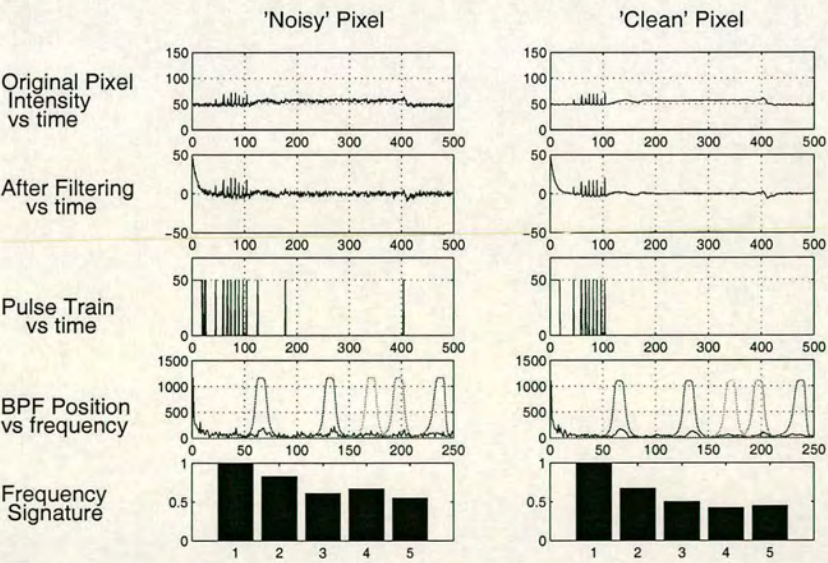
### 3.5.4 Whole Image Analysis of the *Flashing Pixel* Algorithm

The noise analysis of the *flashing pixel algorithm* led to the development of two new versions, both relying on a high pass filter to remove any DC offset caused by fixed pattern noise. A choice of whether to implement the *half-Laplacian HPF algorithm* or the *no-mask algorithm* on a CMOS test IC had to be made. To this end, a series of simulations analysing every pixel in the test data sequences were undertaken. Previous simulations had applied the algorithm to single pixels in the data sequence, selected as they experienced temporal frequencies of interest. It was decided that a truer picture of the power of the candidate algorithms could be made by analysing each pixel in the image sequence. As the success of each algorithm depends on its ability to extract the fundamental frequency as accurately as possible, it was decided to create a series of fundamental frequency maps of the test data sequences





(a) Propeller Plane Pixel (100,72)



(b) Helicopter Pixel (45,53)

Figure 3.28: Effect of Random Transient Noise on the No-Mask Algorithm



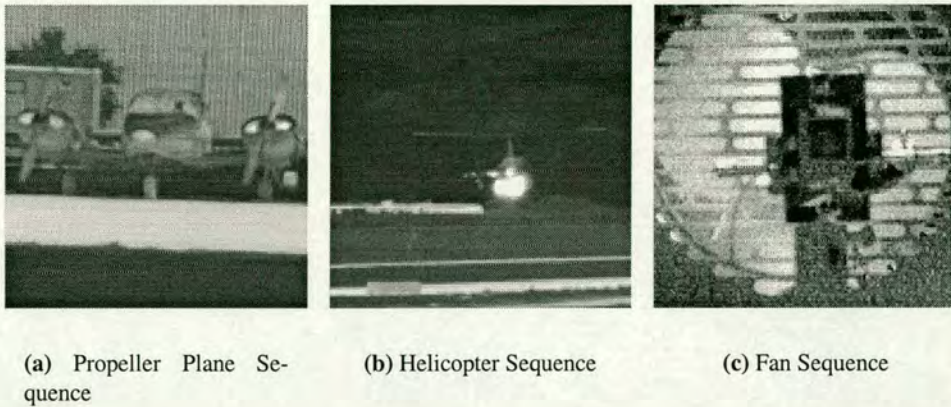
### 3.5.4.1 Fundamental Temporal Frequency Maps

The critical point of each candidate algorithm is the accuracy with which the fundamental frequency is extracted from the temporal intensity variation. A band pass filter array is positioned at the calculated fundamental frequency and its first four integer multiples, making the accuracy of this initial calculation of paramount importance. To test each algorithms' ability to successfully extract the fundamental frequency of any temporal variation in the scene, a series of fundamental temporal frequency maps were created. The idea is to first apply the algorithm to each pixel in the data sequence. The frequency of the generated pulse train is calculated for each pixel, and encoded as a normalised intensity between 0 and 255. The intensity value is then placed in the fundamental frequency map at the same location as the pixel whose fundamental frequency has been calculated. In this way, each of the 500 frame data sequences are represented with a single frame, whose intensity at each pixel maps to the temporal fundamental frequency at that pixel location. To make the frequency maps more easily readable, the intensities were normalised to the highest detected frequency in the sequence, in an effort to maximise the contrast. Each fundamental frequency map provides a simple, visual method of ascertaining the accuracy of each candidate algorithm. In all of the subsequent simulations, fixed pattern noise of approximately 30 % of the maximum pixel intensity was added, to prove the robustness of the half-Laplacian HPF and no-mask algorithms. Random transient noise was not included, as previous simulations had shown that there is very little that can be done about this phenomenon, apart from varying the threshold level of the comparator. Examples of individual frames from the propeller plane, helicopter and fan data sets can be found in figure 3.29. For the fan data sequence, frequency maps of the negative luminescence device flashing at both 10 Hz and 90 Hz were produced.

### 3.5.4.2 Frequency Maps: *Flashing Pixel* Algorithm

Although the problems with the original *flashing pixel* algorithm regarding fixed pattern noise are well documented, it was decided to use it to create frequency maps for comparison. The generated fundamental frequency maps can be see in figure 3.30. As expected, the addition of fixed pattern noise to the data renders the algorithm completely unable to differentiate between pixels that experience temporal frequencies and those whose intensity remains static. Both propeller plane and helicopter data sequences show no discernible output, while the guard mesh from the fan data sets is almost visible. Nevertheless, this simulation proves the unsuitability of





**Figure 3.29:** *Selected Frames from the Test Data Sequences*

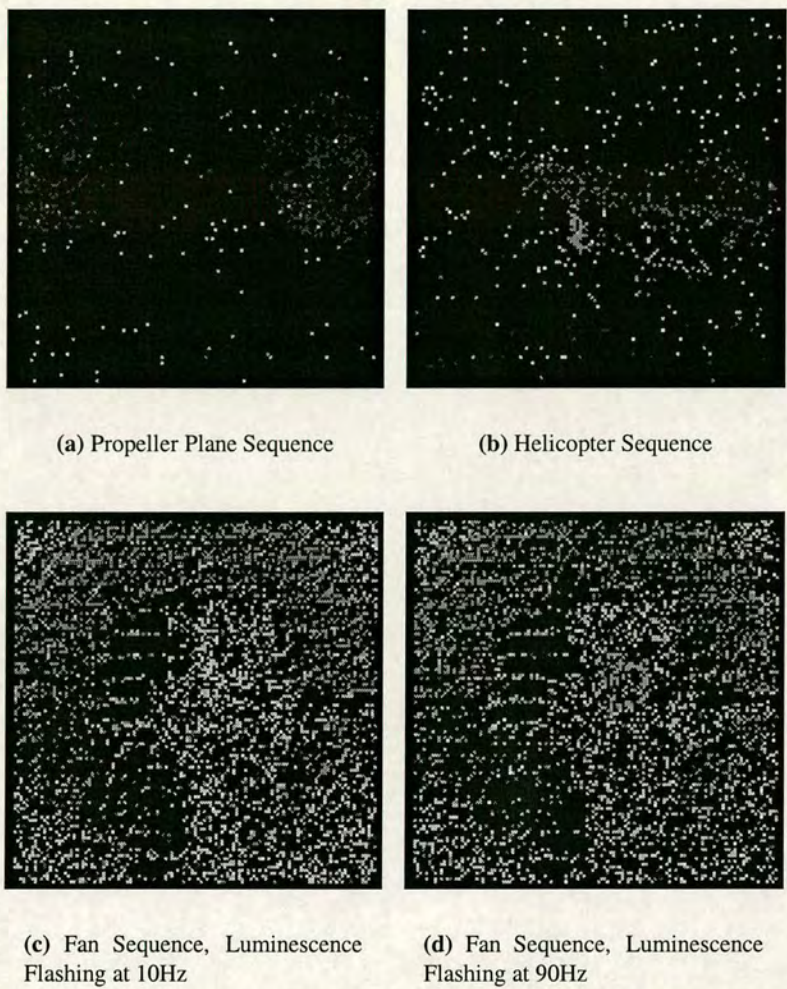
the *flashing pixel* algorithm, when noise simulating a CMOS implementation is introduced. As such, there is little point in producing a CMOS implementation of this version of the algorithm.

### 3.5.4.3 Frequency Maps: *Half-Laplacian HPF* Algorithm

The frequency maps generated with the *half-Laplacian HPF* algorithm can be seen in figure 3.31. The inclusion of a high pass filter ensures the algorithm is completely robust to fixed pattern noise. The map for the propeller plane clearly highlights the twin engines, while ignoring all the other stationary pixels. Ideally, the frequency seen by each pixel, and therefore the intensity in the frequency map would be constant. It is clear that this is not the case, with several patches of differing intensity. The reasons for this stem from the infra-red camera used to capture the image sequence. The difference in temperature between the propeller and its background is so little that the algorithm cannot differentiate between the two. In short, there is simply too little contrast for the algorithm to operate successfully in certain pixel locations. Despite this, it is clear that the algorithm successfully differentiates between static pixels and those that experience a temporal frequency of interest.

The frequency map for the helicopter data sequence shows two different 'blocks' of intensity, corresponding to the main and tail rotor blades of the helicopter. Note also that the helicopter moves in the data sequences with respect to the camera, hence the intensity blocks seeming to gravitate to the bottom right hand corner. Once again, the algorithm produces very few spurious frequencies, with only pixels of interest producing an output.





**Figure 3.30:** *Fundamental Frequency Maps Generated with the Flashing Pixel Algorithm*

The fan data sequence contains both a fan and a negative luminescence device, flashing at 10 Hz in figure 3.31 (c) and 90 Hz in figure 3.31 (d). In both cases, the fan produces a near uniform intensity, with the protective mesh clearly visible. The luminescence device is partly visible in the centre of (d) although many pixels are missing. In general, the two frequency maps from the fan data sequence contain many spurious high frequency readings. Such noise suggests the algorithm is 'seeing' frequencies where in fact none exist.



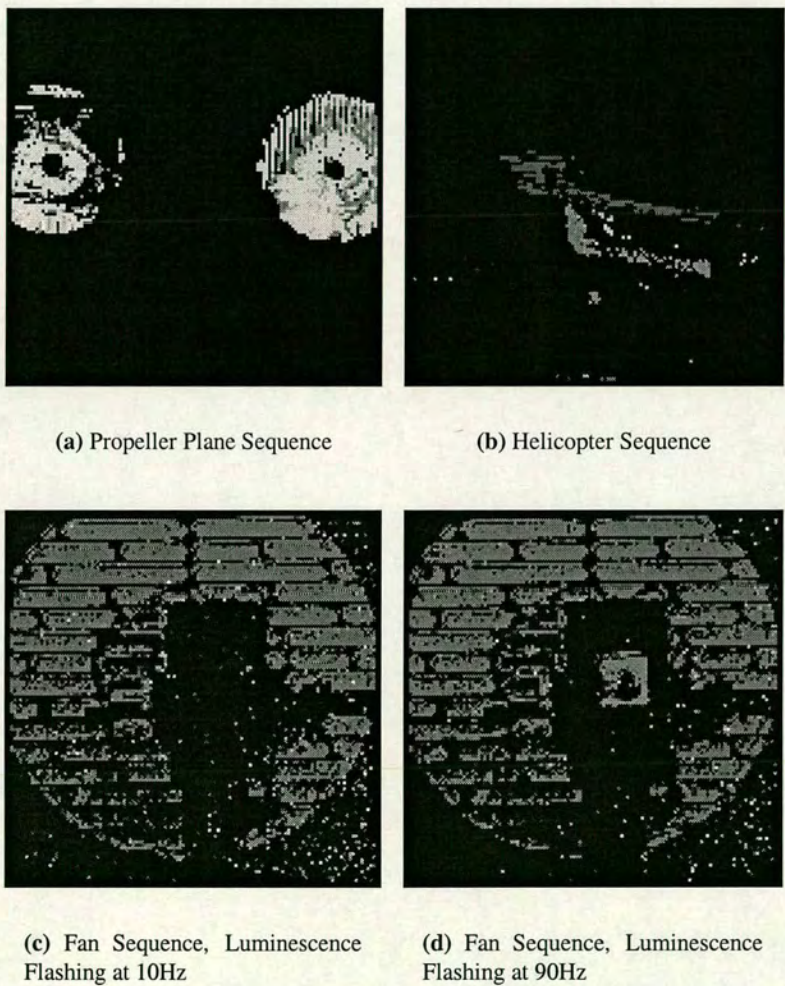
#### 3.5.4.4 Frequency Maps: *No-Mask* Algorithm

The no-mask algorithm simply high pass filters the intensity variation from each pixel before thresholding to produce a series of pulses. The frequency maps generated with this technique can be seen in figure 3.32. There is little discernible difference between the plane map and the equivalent generated with the half-Laplacian HPF algorithm. The two tone effect caused by lack of contrast is still present. The results for the helicopter data set also appear similar, with the main rotor and tail blades clearly visible. There are a few spurious frequencies at the bottom of the frequency map, probably due to camera-wobble on the concrete runway present in figure 3.29(b). The results from the fan data set are the most striking, with very few spurious frequencies created. The luminescence devices are clear to see in the very centre of the frequency map, with the 10 Hz and 90 Hz alternatives producing low and high intensity blocks respectively. The fans themselves produce very uniform blocks of intensity, meaning the algorithm computes the same fundamental frequency for the vast majority of the pixels it passes through. There are a few spurious frequencies in the bottom right corner, but very little compared to the corresponding frequency maps created with the half-Laplacian HPF algorithm.

#### 3.5.4.5 Comments on Fundamental Temporal Frequency Maps

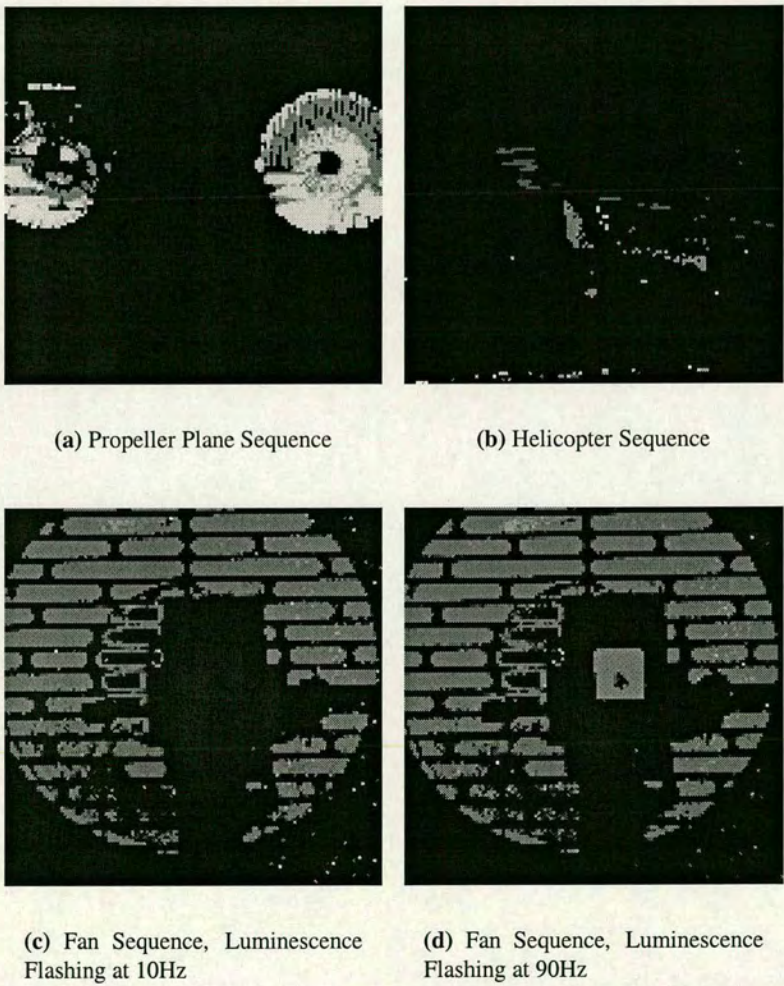
Comparing the frequency maps generated with the three algorithms suggests that the *no-mask algorithm* may be the best in terms of a CMOS implementation. All the frequency maps were generated with the addition of fixed pattern noise, which, as expected, rendered the *flashing pixel algorithm* completely unable to distinguish between actual frequencies and those caused by noise. The results from the *half-Laplacian HPF algorithm* were more promising, with the rotational elements of the data sequences clearly visible against the stationary background. However, particularly with the fan sequence, an alarming number of spurious frequencies were detected, suggesting the approach may attempt to 'lock' onto frequencies that do not exist. Although much simpler, the results from the *no-mask algorithm* appear the best, with very few spurious frequencies and relatively uniform blocks of intensity representing the same object. The better results, coupled with the far simpler circuit level implementation led to the *no-mask algorithm* being selected as the approach for implementation in a CMOS test IC.





**Figure 3.31:** *Fundamental Frequency Maps Generated with the Half-Laplacian HPF Algorithm*





**Figure 3.32:** *Fundamental Frequency Maps Generated with the No-Mask Algorithm*



### 3.5.5 Circuit-Level Implementation of the *Flashing Pixel* Algorithm

With the *no-mask algorithm* selected as the best approach for determining the fundamental frequency of any temporal intensity variations, focus switched to the circuit-level implementation. A decision to concentrate on analogue, continuous time circuitry incorporating focal-plane techniques was made at the inception of the research. All of the candidate algorithms were developed with such a frame-work in mind, with the selected technique involving very simple processing steps.

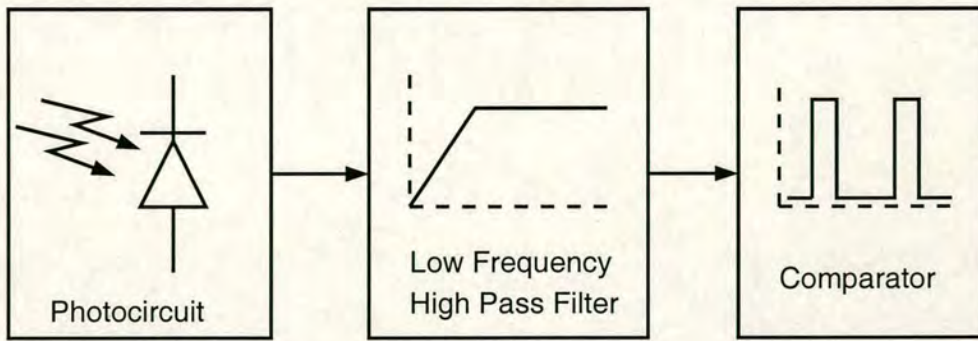
The *no-mask* algorithm can be effectively split into two main tasks: first extracting the fundamental frequency as accurately as possible, before using this information to place a programmable band pass filter array. The frequency map simulations confirm the ability of the *no-mask* algorithm to successfully extract fundamental frequencies while ignoring pixels that experience no temporal intensity variation. It was decided to concentrate first on this fundamental frequency extraction, with the development of a focal-plane technique. With this in place, an algorithm to tune the programmable band pass filter array could be developed.

The adopted algorithm uses two very simple processing steps to find the fundamental frequency of any 'flashing' objects in the field of view. From a single pixel's perspective, the light intensity is high pass filtered, to remove any DC offset and centre the AC signal on a pre-determined bias level. This signal is then thresholded with a comparator to produce a series of pulses, the frequency of which corresponds to the fundamental frequency of the original temporal intensity variation. The process is depicted pictorially in figure 3.33. A photocircuit converts the light into an electrical signal, which is then high pass filtered to superimpose it on a known DC level. The resultant signal is then thresholded with a comparator to produce the pulse train. The next phase of the research involves finding suitable circuit techniques to realise a focal-plane implementation of the algorithm as efficiently as possible. The aim is to include such circuitry within each pixel of the final CMOS image-processor, placing an emphasis on low power and low area circuit implementations.

## 3.6 Conclusions

This chapter has introduced the initial software simulations that were performed to identify possible algorithms for the extraction of frequency signatures from visual data. Two underlying approaches were considered. The first involved wavelet processing in the form of a dyadic tree





**Figure 3.33:** *Circuit-Level Implementation of the No-Mask Algorithm*

filter structure to split the signal into frequency bands of differing resolution. The results from software simulations of the technique suggested that it was not powerful enough to distinguish accurately between objects with similar fundamental frequencies. Other concerns over the size and feasibility of a circuit-level implementation led to other techniques being investigated. Focus instead shifted to a two-phase system, with the fundamental frequency first being detected using focal-plane techniques, before being used to position a pseudo-Fourier filterbank. A technique based on early-processing within the retina was briefly analysed, but deemed unrealistic for circuit implementation. Research into analogue circuit techniques for applying simple convolution masks to the image data led to the flashing pixel algorithm, which was then superseded by the no-mask algorithm, owing to both superior performance and simpler implementation.

Of the different algorithms considered for the extraction of frequency signatures from the temporal intensity variations subject to a CMOS imager, the *no-mask algorithm* was selected. The adopted approach seemed the best compromise in terms of processing power and ease of circuit level implementation, while still adhering to the sponsor companies requirements for a compact, continuous time approach incorporating focal-plane processing techniques. The noise analysis proved the technique is completely robust to fixed pattern noise, a common problem with CMOS imager arrays. Random transient noise is more problematic, but the thresholding process provides some protection. Finally, the whole image analysis proved that the no-mask algorithm clearly distinguishes between pixels that experience frequencies of interest, and those whose intensity remains static. Combined with the simple circuit level implementation, it is clear that the adopted approach is well-suited to a CMOS implementation. The strength of the approach lies in the fact that each pixel is treated as an independent frequency sensitive unit, providing massively parallel processing across the surface of the array. With the algorithm in



place, the next phase of the research involved translating each processing step into a circuit level equivalent.



---

## Chapter 4

# Test IC One: Fundamental Frequency Extraction

---

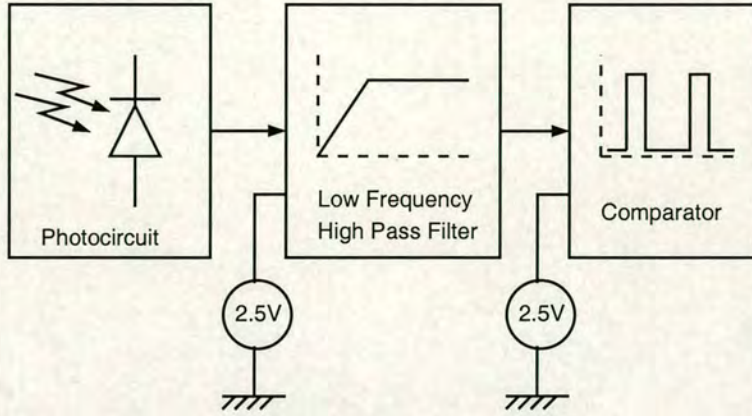
The software simulations detailed in chapter three identified the *no-mask* algorithm as the best approach for extracting the fundamental frequency of temporal intensity variations, regarding a CMOS implementation. This chapter details the development of a test chip incorporating the no-mask algorithm, coupled with test results from the fabricated ICs. All of the test chips were developed using the Cadence suite of tools, incorporating the analog artist simulation environment and the virtuoso layout package. The chips were developed with an AMS 0.6  $\mu\text{m}$  process available through Europractice, and were financed by QinetiQ. A third metal layer was employed on this chip and all subsequent IC's developed during this research to shield those sections of circuitry not directly exposed to the incident light intensity, thus reducing the potential effects of unwanted photo-induced currents. This is particularly important when subthreshold bias conditions are utilised, as such photocurrents can be similar in magnitude to the bias currents, seriously effecting the subsequent performance.

### 4.1 System-Level Design

The three processing steps of the fundamental frequency extraction algorithm are depicted in figure 4.1. A photocircuit converts the incident light intensity to an electrical signal. The DC level is then removed with a high pass filter, designed with a low cutoff frequency due to the low frequency nature of the transient changes in intensity. In order to 'pass' a signal at 10 Hz, a time-constant of 0.1 s or longer is required, which poses difficult design constraints on the filter implementation. Finally, a comparator is used to threshold the signal, producing the required pulse train. A reference of 2.5 V is selected in figure 4.1 as it occupies the midrange of the 5 V power supply, ensuring maximum possible signal swing.

At the system level, a choice of continuous time circuit techniques had already been made, based on the sponsor companies requirements. This led to an exploration of continuous time





**Figure 4.1:** *Circuit Level Implementation for the No-Mask Algorithm: The incident light is converted to an electrical signal which is then filtered to remove the DC level, before thresholding with a comparator*

circuit techniques that could be employed in the development of the system. A choice between current or voltage mode processing techniques had to be made. The difference stems from the electrical signal used to perform the processing steps. In general, current mode circuit techniques are thought to be faster than their voltage mode counterparts[92, 93]. This is due to the ability of transistors used as current amplifiers to operate right up to the maximum realisable frequency,  $f_T$ . The frequency range of interest is relatively low, meaning very high speed processing is not essential. In addition, the emphasis on transistors biased in subthreshold to reduce power consumption may make current mode techniques impractical. Mismatch between subthreshold currents can be as much as 5 % - 20 % depending on transistor dimensions[94], creating potential problems for such an approach. However, a recent paper by Schmid[95] casts doubt on the relevance of defining circuitry as either current or voltage mode, arguing that both share the same underlying properties. An emphasis was placed on simple circuit techniques to allow quick and easy simulation and testing.

## 4.2 Circuit-Level Design

With the decision to pursue an analogue, continuous time approach incorporating focal-plane techniques, the next phase of the research was the identification of circuit structures that could perform the three processing steps depicted in figure 4.1. The emphasis on the test chips designed in this research was proof of concept. As such, if superior circuit-level topologies are discovered, they could be 'slotted' into the algorithm at a later date.



## 4.3 Photoelement

The job of converting the incident light intensity into a useful electrical quantity is performed by the photoelement. In a standard CMOS process, there are a number of alternative structures, all with certain strengths and weaknesses. The response of silicon to incident light photons of sufficient energy is the liberation of a valence electron from the bonds holding the silicon atoms together. This process leaves behind a positively charged 'hole', thus creating an 'electron-hole pair' and allows the structure to conduct a certain amount of electrical current. As the light intensity increases, the number of liberated electrons also increases, producing a higher current. By exposing certain structures to light intensity, photocurrents related to the strength of that light energy are created.

### 4.3.1 Photodiode

The photodiode tends to be the most widely used photoelement in commercial CMOS imagers due to its relatively good matching properties[96]. In general, such devices are used in reverse-bias, due to the relatively linear relationship between light intensity and photocurrent. As a consequence of this, the photocurrents produced by a photodiode tend to be in the  $pA$  to  $nA$  range. There are different ways of realising photodiodes in a CMOS process, including well-substrate diode, diffusion-well diode, diffusion-substrate diode and lateral diode.

### 4.3.2 Phototransistor

Parasitic bipolar transistors exist in CMOS processes due to the layers of differently doped n-type and p-type material. It is possible to create a phototransistor by exposing such a structure to the incident light. A standard bipolar transistor operates by multiplying the base current by a gain factor  $\beta$ , which then appears at the collector terminal. If the base current is generated by light energy, the effective photocurrent will be multiplied by the  $\beta$  factor, producing approximately 100 times the value from a photodiode. However, this also proves to be the phototransistors downfall, due to the poor matching characteristics of this gain factor. For CMOS imagers, it is imperative that each pixel produces the same photocurrent for the same incident light intensity. Such a situation is unlikely with a phototransistor, without post-processing circuit or software techniques. Nevertheless, the phototransistor is a useful structure for converting light into photocurrent in certain applications.



## 4.4 Photocircuit: Logarithmic Compression Photocircuit

The task of the photocircuit is to condition the current from the photoelement into a useful format for further processing stages. The choice of a continuous time approach at the system level reduced the number of potential candidates considerably. Many industrial CMOS imagers make use of active pixel sensors, where the charge is integrated on a node and then read-out when required[6]. Such an approach requires clock signals to control the charge collection and read-out periods, with the output appearing as a sampled version of the incident light intensity. The rationale behind a continuous time approach to the system stemmed from the underlying requirement to extract a meaningful frequency signature from each pixel. With a sampled data approach, the sampling time would have to be sufficient to pass the requirements for Nyquist's criterion regarding aliasing. Standard approaches to CMOS imagers take a sample from each pixel in a row or column in turn, meaning the clocking frequency in this application would need to take into account the overall dimensions of the imager array, placing a potential upper-limit on the maximum resolution of the system. A continuous time approach stops potential aliasing problems, allowing each pixel to be accurately analysed for frequency content.

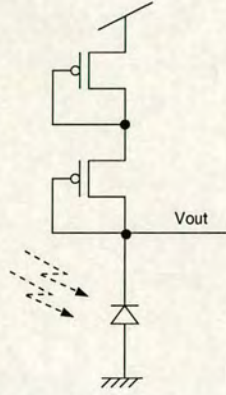
A potential problem with continuous time photocircuits is the huge range of illumination levels incident to the imager. There are almost eight decades of incident illumination, producing a huge range of possible inputs for a photocircuit to cope with. Active pixel circuits adapt the integration period depending on the background illumination, allowing long periods for dark conditions and short periods for bright conditions. While a useful approach, if the same image incorporates regions of dark and bright light, certain details will be lost. A potential solution to this dynamic range problem is the logarithmic compression photocircuit, depicted in figure 4.2. The circuit works on the basis that the photocurrent produced by the photoelement (in figure 4.2, a photodiode) is small enough to bias the transistor loads in the subthreshold region of operation.

### 4.4.1 Large-Signal Characteristics

A simplified version of the  $I_{ds}$ - $V_{gs}$  relationship for a subthreshold transistor[97] can be found in equation 4.1, along with the corresponding strong-inversion version in equation 4.2. In equation 4.1,  $W$  and  $L$  represent the physical dimensions of the transistor,  $I_{D0}$  is a process dependent reverse voltage saturation current,  $V_{gs}$  is the gate-source voltage of the transistor,  $n$



is the subthreshold slope factor,  $k$  is Boltzmann's constant,  $T$  is the ambient temperature and  $q$  is the charge on an electron. The parameters for equation 4.2 once again include the transistor dimensions  $W$  and  $L$ , the mobility factor  $\mu_0$ , oxide capacitance  $C_{ox}$ , and the threshold voltage of the device  $V_t$  subtracted from the gate source voltage. The exponential nature of the current of a subthreshold transistor with respect to the gate-source voltage results in a logarithmic compression of the output voltage for the photocircuit, as seen in the simulation results in figure 4.3. When the photocurrent starts to reach strong inversion magnitudes, the square-law relationship seen in equation 4.2 begins to dominate and the output voltage drops significantly.



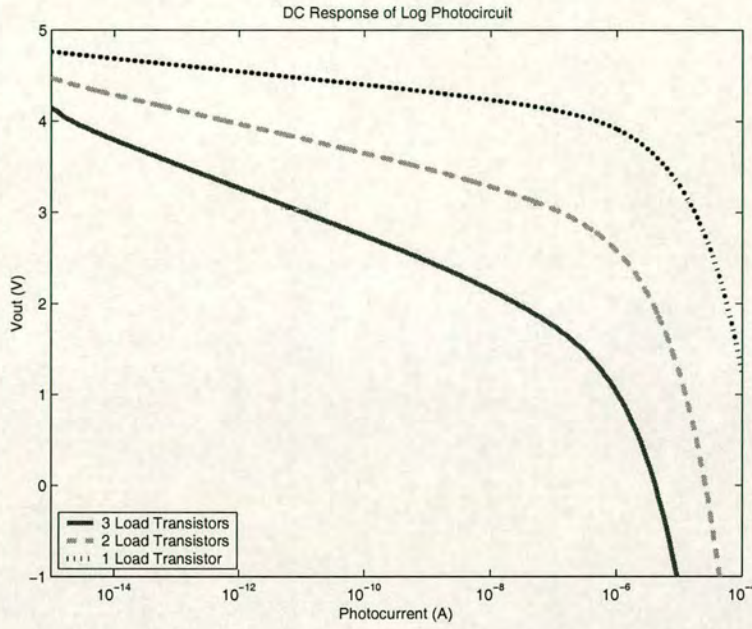
**Figure 4.2:** *Logarithmic Compression Photocircuit: The current produced by the photoelement bias the transistors in the subthreshold regime, resulting in a logarithmic compression in the output voltage*

$$I_{ds} = \frac{W}{L} I_{D0} \exp \left( \frac{V_{gs}}{n \frac{kT}{q}} \right) \quad (4.1)$$

$$I_{ds} = \frac{\mu_0 C_{ox}}{2} \frac{W}{L} (V_{gs} - V_t)^2 \quad (4.2)$$

The slope of the DC response for the logarithmic photocircuit can be modified by varying the number of load transistors. By increasing the number of load transistors, the slope of the logarithmic photoreceptor's DC characteristic can be increased, effectively increasing its sensitivity to transient signals[98]. This property was simulated using the Spectre simulation tool, for photocircuits with one, two and three load transistors. The results in figure 4.3 highlight the change in slope as the load is varied.





**Figure 4.3:** Simulation of the Logarithmic Photoreceptor's DC Response: As the number of load transistors increase, the slope of the DC characteristic steepens, increasing the sensitivity

#### 4.4.2 Small-Signal Characteristics

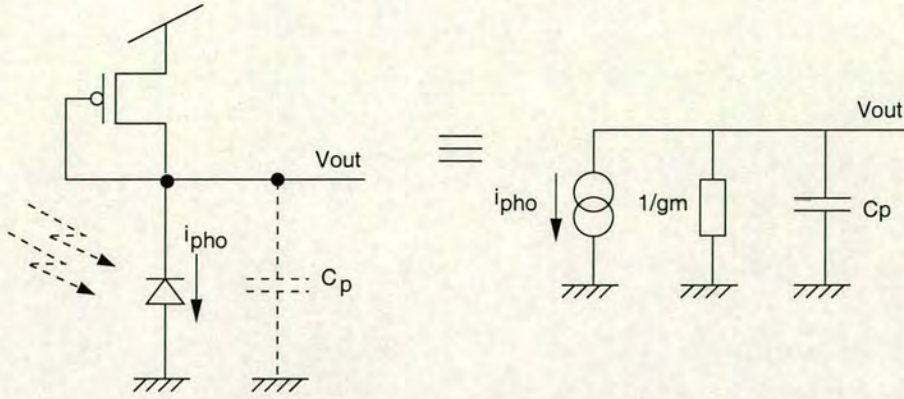
The frequency response of the logarithmic compression photocircuit is of paramount importance to its use in this application. The photocircuit has to be able to operate at the frequencies of interest, in this case from about 1 Hz to 10 kHz. For simplicity, the small signal response of the photocircuit was derived with a single load transistor. Based on figure 4.4, with  $C_p$  representing the parasitic capacitance of the photoelement, the small signal transfer function can be calculated as that in equation 4.4, using Kirchoff's current law at the output node.

$$g_m V_{out} + i_{pho} + V_{out} s C_p = 0 \quad (4.3)$$

$$\frac{V_{out}(s)}{i_{pho}(s)} = \frac{-1}{g_m + s C_p} \quad (4.4)$$

It is clear that the logarithmic compression photocircuit exhibits a first order low pass response, with dominant pole  $p_1 = \frac{-g_m}{C_p}$ . Depending on the ambient light conditions, the photocurrent will bias the transistor load in the subthreshold regime, producing a small value of transcon-





**Figure 4.4:** *Logarithmic Compression Photocircuit: Small Signal Model*

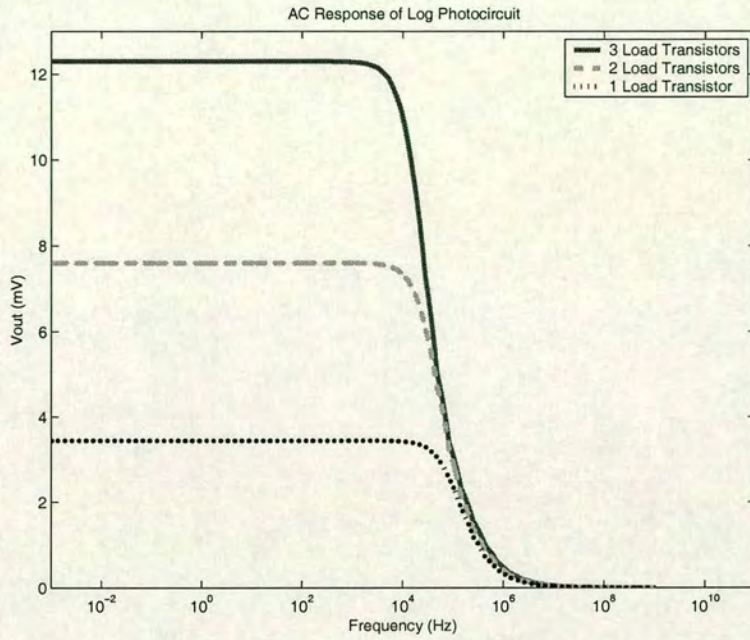
ductance. The main limiting factor in the frequency response is the parasitic capacitance of the photoelement, which is directly proportional to its size. This suggests a design trade-off between the size of the photoelement and therefore the magnitude of the photocurrent, with the maximum input frequency that the photocircuit can 'pass'. Methods of decreasing the dependence of the bandwidth on the parasitic capacitance include using an amplifier with feedback to shift the dominant pole to higher frequencies[6].

Once again, the photocircuit was simulated with the Spectre simulation tool, in an effort to ascertain the effect of the number of load transistors on the AC response. From the DC characteristics in figure 4.3, the slope increased as the load increased, suggesting higher gain. It is clear from the simulated AC response in figure 4.5, that more load transistors does provide higher gain. However, the increased sensitivity to transient signals comes at the cost of a reduction in bandwidth.

#### 4.4.3 Implementation of the Logarithmic Compression Photocircuit

The actual circuit implemented on test chip one can be seen in figure 4.6, together with the transistor dimensions. Based on the simulation results, three load transistors were used in an effort to increase the slope of the logarithmic compression, providing larger output voltage swings for the same transient photocurrent[6, 98]. However, as figure 4.5 highlights, this approach has the disadvantage of reducing the bandwidth of the device. Both are important considerations for the photocircuit's application in this research, but it was decided that the advantage of increased gain from three load transistors out-weighed the reduction in bandwidth. Two different





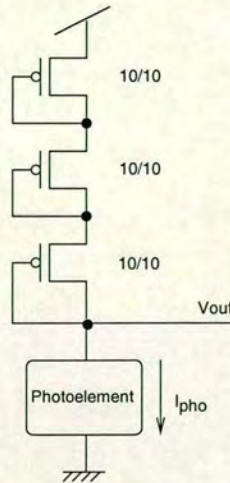
**Figure 4.5:** *Simulation of the Logarithmic Photoreceptor's AC Response: As expected, the gain increases as the number of load transistors increases. However, the cost is a reduction in bandwidth*

photoelement structures were included, both with dimensions  $50\ \mu\text{m}$  by  $50\ \mu\text{m}$  to allow a direct comparison of results. The first photoelement was a diffusion-substrate photodiode, with the other being a parasitic vertical bipolar phototransistor.

#### 4.4.4 IC Test Results: Logarithmic Compression Photocircuit

The logarithmic photoreceptor implemented on the test IC was tested with the aid of an LED controlled by a signal generator, to allow input frequencies to be changed. In order to test the frequency response of all the implemented circuits, a simple buffer circuit was employed to prevent the relatively high capacitance of the analogue pads from loading the sensitive nodes. The circuitry within each pad, particularly the protection diodes, produce a large capacitance which can effect the frequency response of the circuits under test. The details of the buffer circuitry can be found in appendix B.





**Figure 4.6:** *Logarithmic Compression Photocircuit: Implementation on Test Chip One*

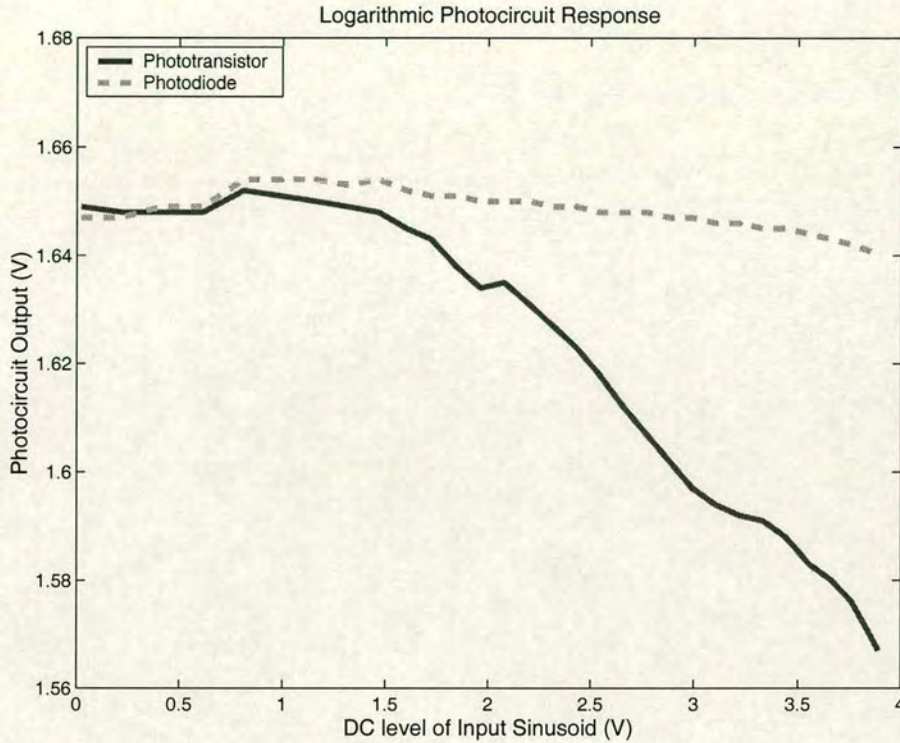
#### 4.4.4.1 Comparison of Output Voltage Between Diffusion Substrate Photodiode and Vertical Parasitic Phototransistor

The first experiment involved proving the fact that the vertical parasitic bipolar phototransistor produced more photocurrent than a photodiode for the same input intensity. To test this, the LED was illuminated with a DC supply source which was slowly varied to increase the level of illumination. As the light level increases, the corresponding photocurrent should also increase, with that of the phototransistor producing a more pronounced change than the photodiode. As such, the output voltage from the phototransistor should vary over a wider range than the photodiode. The test results in figure 4.7 confirm this, with the output from the phototransistor driven photocircuit varying from 1.65 to 1.56 Volts, while the photodiode version moves down only to approximately 1.64V. In this application, any increased signal swing from the photocircuit will aid the thresholding stage of the algorithm.

#### 4.4.4.2 Frequency Response of the Photocircuit

The frequency response of the log compression photocircuit is dominated by the size of the parasitic capacitance caused by the photoelement, as seen by the relationship in equation 4.4. An effort to measure the photocircuits frequency response was made by controlling the LED with a signal generator, which allowed the input frequency to be ramped. The ratio of the output voltage to the input was taken at each frequency step, with the results plotted in the bode plot of figure 4.8. The experiment was repeated three times, with the DC level of the LED's light





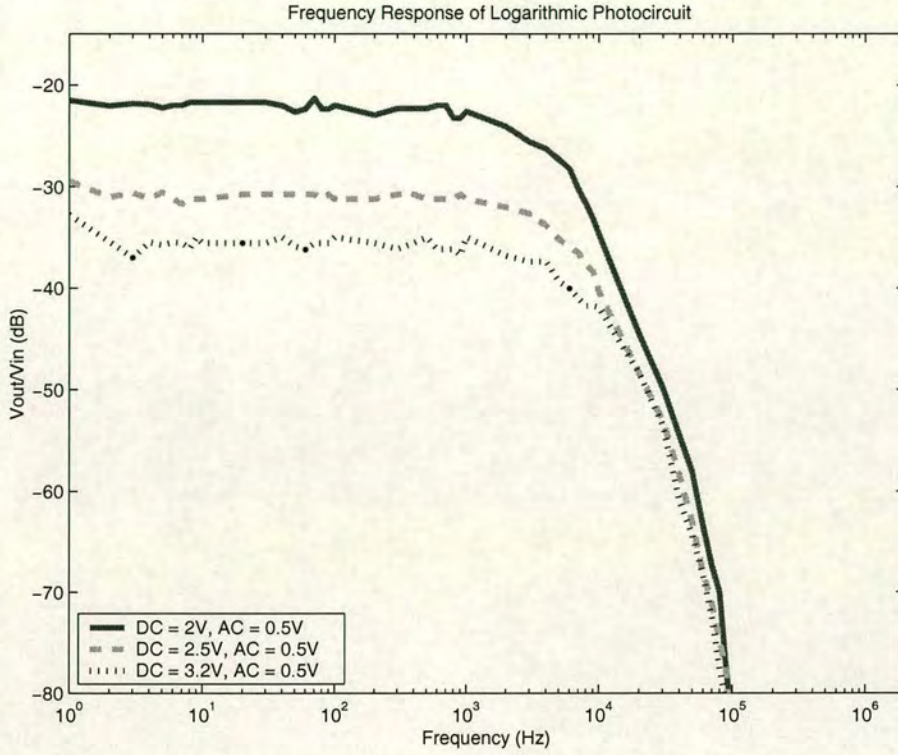
**Figure 4.7:** *Measured Test IC Results- Comparison of Logarithmic Photoreceptor with Photodiode and Phototransistor Loads: As expected, the phototransistor produces a larger output swing due to the inherent amplification of the photocurrent by its  $\beta$ . For all subsequent implementations, the phototransistor was the employed photoelement*

intensity varied but the AC kept constant throughout at 500 mV. As expected, the differing DC level manifests as a different level of attenuation in the low frequency pass band. However, the cutoff frequency for all three remains fairly constant at around 1kHz. This could be increased by reducing the number of load transistors or the size of the phototransistor.

#### 4.4.5 Comments on the Logarithmic Compression Photocircuit

The logarithmic compression photocircuit exploits the physics of a transistor when biased in the weak inversion region of operation to reduce the huge potential input photocurrent range to a much smaller output voltage. Potential disadvantages of this approach regarding the chosen application include the fact that the high compression may result in small, transient intensity changes being effectively 'missed' as the voltage change they produce are so small. Also, the frequency response is low pass in nature, with a dominant pole caused by the size of the pho-





**Figure 4.8:** *Measured Test IC Results- Frequency Response of the Logarithmic Compression Photocircuit with Phototransistor: Despite the variation in pass band attenuation, the cutoff frequency tends to be constant at around 1kHz*

toelement. Nevertheless, the circuit does fit the design criterion in that it operates in continuous time, with low power consumption and produces an output voltage signal that is related to the input photocurrent.

The use of a phototransistor as the photoelement is acceptable in this particular application, as each pixel in the imager operates independently from the others. As such, any mismatch in the gain from each photocircuit will have no effect on the system's performance, with the larger signal swings afforded by the  $\beta$  multiplication factor an advantage for subsequent thresholding.

As previously mentioned in chapter two, Delbruck's adaptive photoreceptor[12] would be an excellent photocircuit for the chosen application. However, it was felt that the simplicity of the logarithmic compression approach made it a more feasible input stage. Although its performance is poorer, the ease with which it could be implemented was the reason for using it as the input stage for the prototype CMOS implementation of the fundamental frequency extraction algorithm.



## 4.5 High Pass Filter: Gm-C First Order Filter

The second stage of the fundamental frequency extraction algorithm uses a high pass filter to remove the DC level from the photocircuit's output and bias the transient information at a well-defined reference level. Due to the relatively low frequency range of the input, the cutoff frequency of the filter had to be low, such that signals close to 1 Hz could be passed without considerable attenuation. Cutoff frequencies of this order are generally difficult to achieve in CMOS VLSI, particularly given the constraints in this particular application. The adopted approach requires that the filter consumes as little area as possible, as the three processing elements in figure 4.1 are to be included within each pixel of the image-processor array. This, coupled with the low power constraints imposed by the sponsor company, led to a decision to implement a simple first order high pass filter.

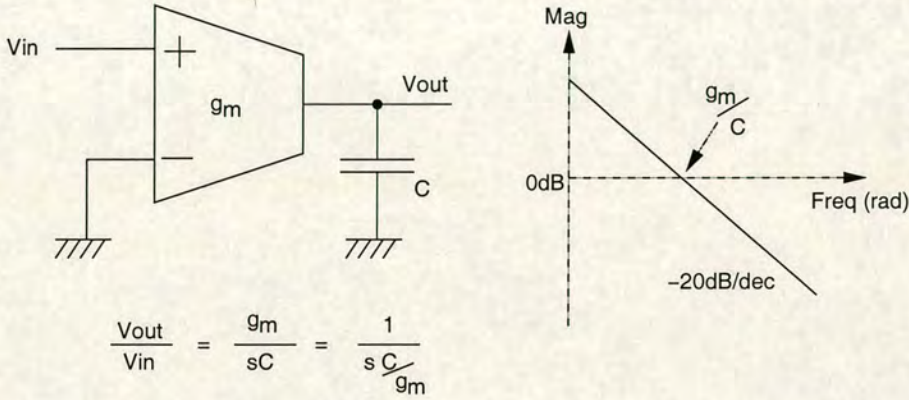
### 4.5.1 Low Frequency Filtering with Gm-C Circuit Structures

Various continuous time methods for realising filter structures exist in CMOS VLSI. Active and passive RC filters, current mode filters and log-domain filters are all examples of design methodologies for producing frequency selective circuits. Based on work published in the literature, a decision to implement a Gm-C/OTA-C filter was made. Such circuits use transconductance elements combined with capacitors to produce integrators, which act as the basis of Gm-C filtering[99]. These can then be combined to implement higher order structures, either by simply cascading low order stages or converting an LC ladder prototype.

The reason for the choice of Gm-C filter stems from the need for large time constants to realise the low cutoff frequency required. Work into the realisation of very low frequency Gm-C filtering has recently been published [100–103], together with more general work on such filter techniques[99, 104–106]. The ability to bias the transconductance element in the weak inversion region of operation allows for the creation of very large time constants, due to the resultant small values in transconductance. As an example, consider the simple first order integrator illustrated in figure 4.9. As previously mentioned, such integrators act as the basic elements in the realisation of higher order Gm-C filter structures. The figure illustrates the fact that the unity gain frequency depends directly on the ratio of the transconductance to the capacitance. If the designer wishes to realise this unity gain at low frequency, as is the case with the implementation of low frequency filters, one approach may be to increase the size of the capacitance.



However, capacitors consume large amounts of space on a CMOS IC. The only other possibility is to reduce the magnitude of the transconductance, which is readily achievable by biasing the element with a subthreshold current. A device biased with a weak-inversion current has the added benefit of consuming very little power, which was one of the driving factors behind the implementation of this system. It seems that low frequency filters implemented with this technique are ideal for the application presented in this thesis.



**Figure 4.9:** First Order Integrator- The Basis for many Gm-C Filter Structures: The unity gain point depends on the ratio of the transconductance to the capacitance.

However, there are disadvantages regarding implementing filters with subthreshold currents. Mismatch between weak inversion currents[19, 94] can manifest as similar filter structures with wildly different cutoff frequencies. Gm-C filters biased in the strong inversion region of operation require tuning circuitry to accurately control the frequency response, due to the inherent mismatch in CMOS elements[106]. The error only increases when using subthreshold currents, suggesting that accurate filter cutoff frequencies are difficult to achieve without expensive tuning techniques. Another disadvantage with subthreshold circuitry is its inherent temperature sensitivity, as highlighted in the dependence on  $T$  in equation 4.1. Changes in the ambient temperature of the IC will produce different filter time constants. It is possible to develop systems which are more tolerant to temperature variation, but at the cost of increased area and complexity.

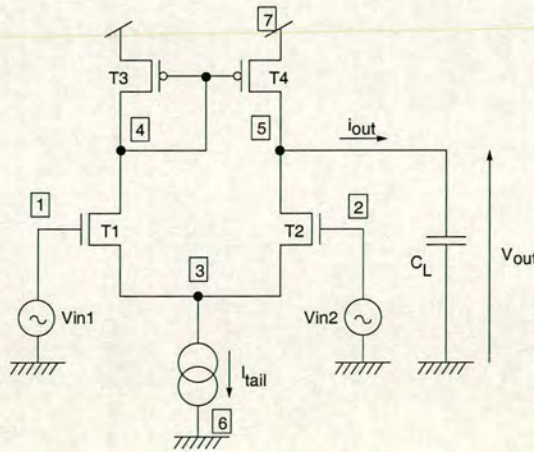
For the purposes of the research documented in this thesis, the fact that the time constant of the high pass filter will vary is of little concern, given its task in the fundamental frequency extraction algorithm. The sole aim of the filter is to separate the transient information from the photocircuit's output from its bias level. To achieve this, the cutoff frequency simply has



to be lower than the input fundamental frequency, but exactly how much lower is of little consequence. At worst, the pixel may produce no output, however the chances of every pixel doing so are small. Crucially, the system won't produce a spurious output frequency if the filter's time constant varies, meaning the system can be relied upon at all times.

#### 4.5.2 Realising Transconductance Elements with Operational Transconductance Amplifiers

There are many methods for implementing the transconductance element in figure 4.9. Operational transconductance amplifiers (OTAs) are a simple way to realise a tunable transconductance element. An OTA is essentially an operational amplifier with a low impedance output. The gain of an OTA is normally characterised by the transconductance of the input differential pair, which serve to convert the input voltage to a current. Consider the simple OTA in figure 4.10, which is essentially a standard differential stage with current mirror load. When biased in strong inversion, the transconductance of this element is given by equation 4.5[91], where  $W$  and  $L$  are the dimensions of the differential pair transistors,  $\mu_0$  is the mobility factor,  $C_{ox}$  is the oxide capacitance and  $I_{tail}$  is the tail current. It is clear that the transconductance, and therefore the time constant of the circuit, can be controlled by varying the tail current. However, there is a square-root relationship, reducing the potential tuning range of any subsequent filter implemented with this approach.



**Figure 4.10:** Operational Transconductance Amplifier: NMOS Differential Pair with Current Mirror Load



$$g_m = \sqrt{2\mu_0 C_{ox} I_{tail} \frac{W}{L}} \quad (4.5)$$

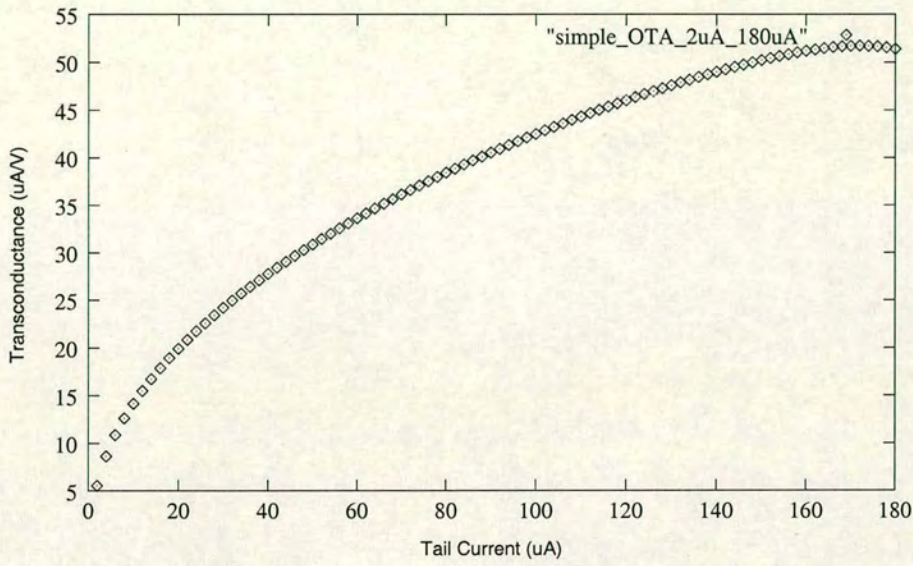
The relationship for the transconductance of the same circuit when biased in subthreshold becomes that in equation 4.6[100], where  $K$  is a process dependent constant and  $V_{thermal}$  is the thermal voltage. In this case, there is a linear relationship between transconductance and tail current, allowing a wider tuning range for filters implemented using this technique.

$$g_m = K \frac{I_{tail}}{V_{thermal}} \quad (4.6)$$

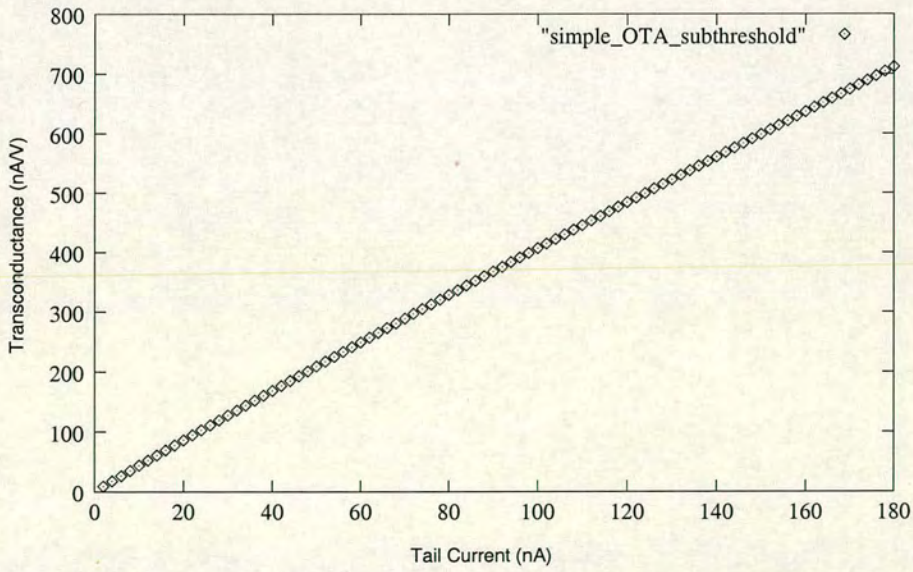
The relationships in equations 4.5 and 4.6, for the simple OTA in figure 4.10 were simulated using the Spectre simulation tool. The transconductance versus tail current for both strong and weak inversion bias can be seen in figure 4.11. Figure 4.11 (a) clearly highlights the square-law relationship between tail current and transconductance, while the linear relationship for the same circuit biased in subthreshold is depicted in figure 4.11 (b). The ability to realise large time constants using subthreshold currents is the prime motivation for adopting the technique in this research. The fact that the device can be tuned over a wide range of transconductance values may have no explicit benefit at this stage, but any implementation of programmable band pass filter banks at a later stage would benefit greatly.

For the purposes of this research, two simple OTA structures were implemented in an effort to compare the relative strengths and weaknesses. The simple differential stage with current mirror load was the first structure, with the mirrored OTA acting as the second. Both OTA topologies are highlighted in figure 4.12. The mirrored OTA uses three current mirrors to convert the differential input to a single-ended output. The idea is that it is a more balanced circuit than that in figure 4.12 (a), with a resultant improvement in output offset voltage[105]. The transistor sizes for the two OTA structures can be seen in tables 4.1 and 4.2. The elements were sized by using the simulator to achieve reasonable values of transconductance. Given that the elements are biased in the subthreshold region of operation, large transistor aspect ratios were adopted to improve the effects of current mismatch.





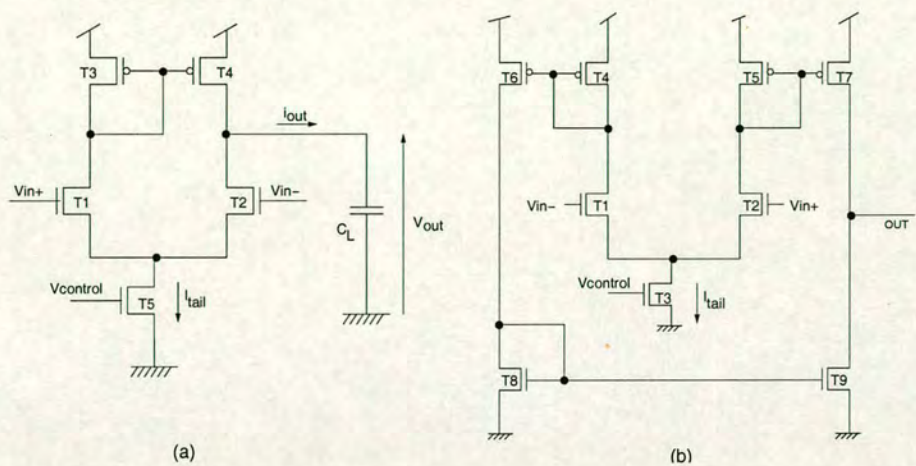
(a) OTA biased in strong inversion



(b) OTA biased in weak inversion

**Figure 4.11:** Simulated Comparison of Transconductance versus Tail Current for Simple OTA: The OTA biased in strong inversion exhibits a square root relationship, while that for the same circuit biased in subthreshold is linear.





**Figure 4.12:** OTA Structures Implemented on Test Chip One: (a) Differential Stage with Current Mirror Load, (b) Mirrored OTA

Transistor	$\frac{W}{L}$ ( $\mu\text{m}$ )
T1	40/10
T2	40/10
T3	30/10
T4	30/10
T5	10/6

**Table 4.1:** Transistor Dimensions for Differential Pair OTA

### 4.5.3 OTA-C First Order High Pass Filter

The generic form for a first order high pass filter based on Gm-C/OTA-C principles is highlighted in figure 4.13 (a), together with its transfer function. The circuit consists of a capacitor and a transconductance element connected with negative feedback. The non-inverting terminal of the OTA is connected to a reference voltage, normally midway between the power supply rails. It is this reference voltage onto which the filter's passband is superimposed.

The idealised frequency response depicted in figure 4.13 (b) shows the high pass response of the circuit, coupled with the assertion that the cutoff frequency is equal to  $f_{-3dB} = \frac{g_m}{2\pi C}$ . It is clear that the position of the cutoff frequency can be reduced by either increasing the capacitance or reducing the transconductance of the OTA.

The high pass filter in figure 4.13 (a) was implemented on board test chip one, using both OTA structures highlighted in figure 4.12. The capacitor was realised with a 1 pF polysilicon



Transistor	$\frac{W}{L} (\mu\text{m})$
T1	5/6
T2	5/6
T3	10/6
T4	4/4
T5	4/4
T6	4/4
T7	4/4
T8	4/4
T9	4/4

Table 4.2: Transistor Dimensions for Mirrored OTA

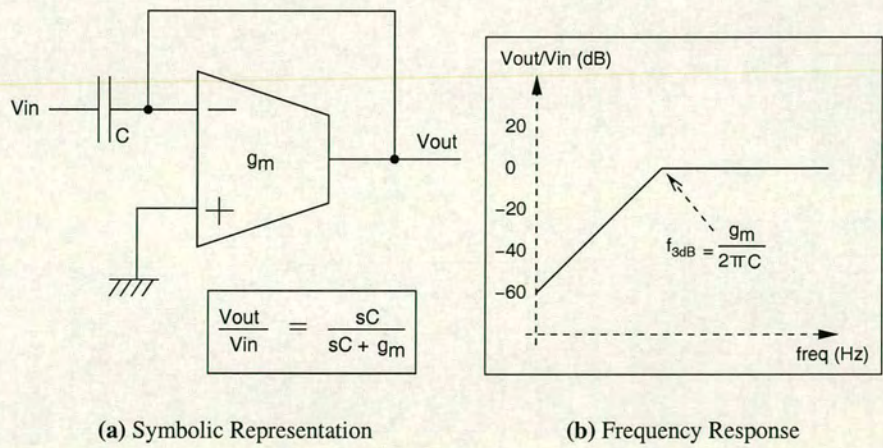


Figure 4.13: 1st Order Gm-C High Pass Filter



structure, due to the good matching properties of such devices. The layout of both filters can be seen in figure 4.14. High pass filter one is approximately  $125\ \mu\text{m}$  by  $75\ \mu\text{m}$ , while filter two consumes almost  $110\ \mu\text{m}$  by  $70\ \mu\text{m}$ . Although both fairly large, it is clear from figure 4.14 that there is a large amount of redundancy in both layouts, with guard rings and dummy transistors employed. Little effort was made at this stage to minimise the size of the filter structures, with the emphasis on a working prototype of the system. However, it is clear that smaller implementations of the filters could be realised.

#### 4.5.4 OTA-C High Pass Filter: Power Consumption

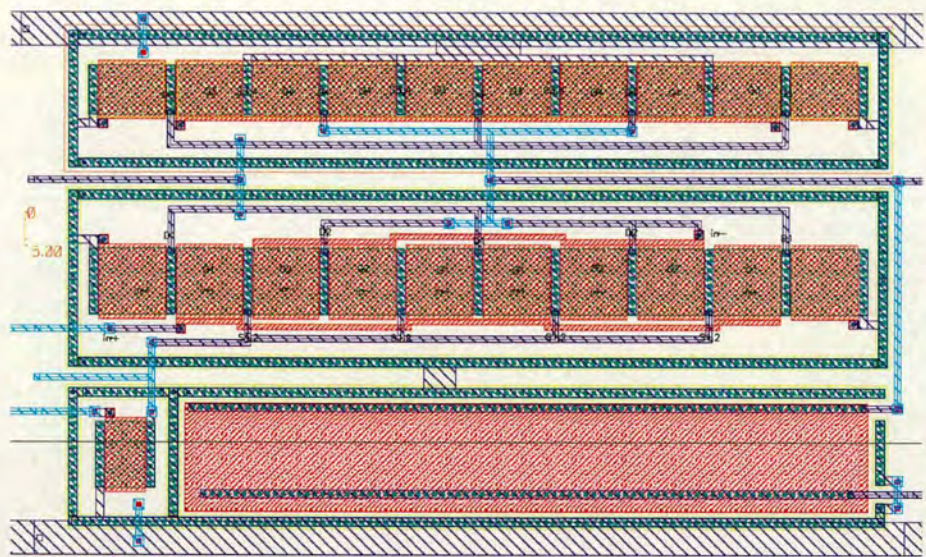
The need for very low filter cutoff frequencies led to the use of subthreshold bias currents. This in turn ties in well with the sponsor's requirement for low power processing. This rare mutually beneficial scenario allows for the realisation of relatively small, low frequency filter structures that consume extremely low power levels.

An estimate of the power consumption of both filter structures was made using the Spectre simulation tool. Measurements from test IC's were not made, due both to the extremely small nature of the bias currents and the difficulty in isolating individual processing elements. Table 4.3 highlights the current consumption for both filter structures for a bias of 0.51 V and 0.56 V, typical values that may be used in fundamental frequency extraction system. If a power supply of 5 V is adopted, the first high pass filter consumes between 237 pW and 830 pW, while the second version uses between 549 pW and 1.76 nW. The higher values for the second filter are expected, as the mirrored OTA structure has more current consuming paths than the simple differential version. Although these values are likely to be inaccurate regarding the actual IC power consumption, it is clear that the filter structures consume extremely low power. Even if the values are multiplied by a factor of 10 or 100, the filters still consume power in the nano-Watt range, allowing for potential battery-powered operation.

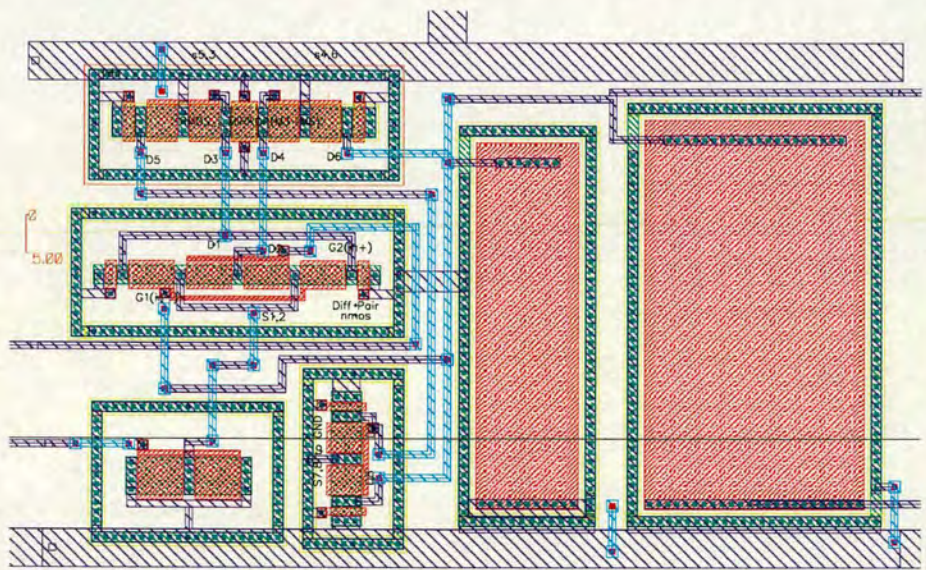
Control Voltage	HPF1	HPF2
0.51 V	47.4 pA	109.8 pA
0.56 V	166 pA	352.7 pA

**Table 4.3:** Simulated Bias Current Consumption of OTA-C High Pass Filter Structures





(a) HPF1: Simple Differential OTA



(b) HPF2: Mirrored OTA

**Figure 4.14:** *Physical Layout of the High Pass Filters on Test IC One*



#### **4.5.5 IC Test Results: OTA-C First Order High Pass Filter**

The two different versions of the first order OTA-C high pass filter implemented on IC one were tested for their frequency response, in an effort to verify the low frequency nature of biasing them in the subthreshold regime. Another test that was performed was to ascertain the DC offset between the output signal and the reference voltage. The aim of the high pass filter is to separate the transient (AC) information from the ambient light conditions (DC), before biasing the former at the reference level. As such, any discrepancy between the reference level and the filter's output is of particular interest.

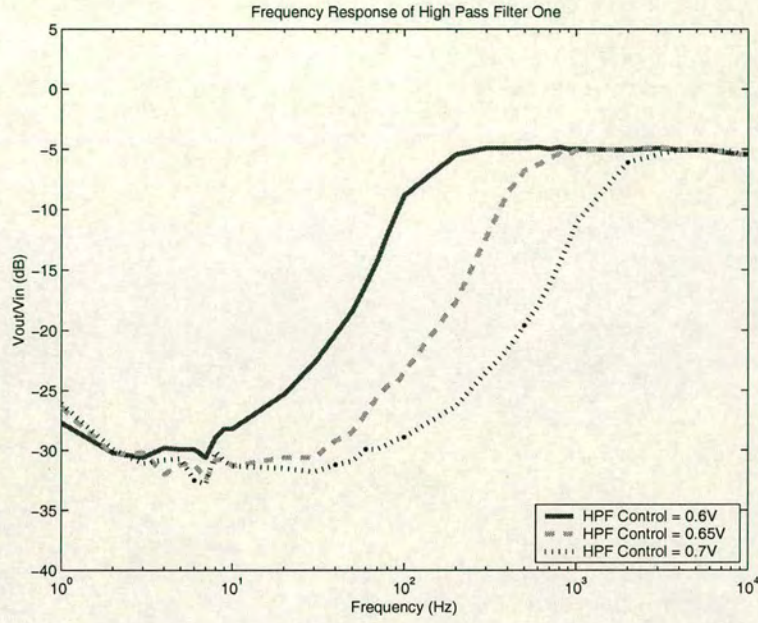
##### **4.5.5.1 Frequency Response of the OTA-C First Order High Pass Filters**

The frequency response of the filter was estimated by connecting a signal generator to the input and measuring the ratio between the input and output voltage. Once again, the buffer circuit highlighted in figure B.1 was used to prevent any loading effect. The results for the high pass filter realised with the simple differential OTA (HPF1) can be found in figure 4.15 (a), with the corresponding results from the mirrored OTA implementation (HPF2) in figure 4.15 (b). The tunability of both circuits is clear, as the control voltage is varied from 0.7 V to 0.6 V. As expected, the deeper into subthreshold the filter is biased, the lower the cutoff frequency. With a control voltage of 0.6 V, the cutoff frequency of both high pass filters is approximately 100 Hz, suggesting transconductances in the region of 60 nA/V. The pass-bands of both filters exhibit almost five decibels of attenuation, which is a potential problem for the approach. Any reduction in signal swing makes the next step of thresholding the signal more complex. Both filters exhibit the expected 20 dB per decade attenuation associated with a first order filter. At very low frequency, the bode plots appear to level off, suggesting the presence of an unwanted zero. In fact, at such low frequency, with such high levels of attenuation, the difference between signal and background noise is hard to distinguish, resulting in the 'levelling-off' of the bode plot. It is however clear from the test results in figure 4.15 that both high pass filters perform as expected.

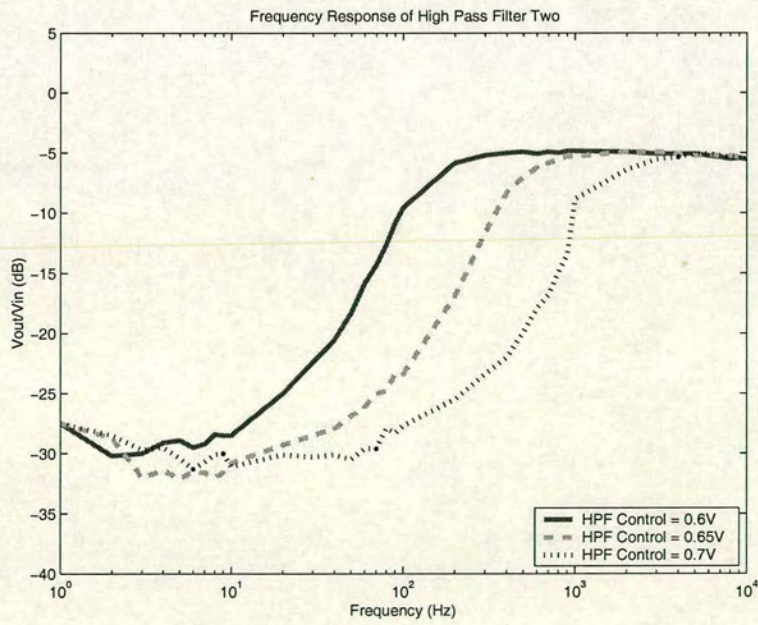
##### **4.5.5.2 DC Offset of the OTA-C First-Order High Pass Filters**

Another property of the high pass filter of interest to its application in this research is the DC offset between the output voltage and the reference voltage. All differential stages introduce





(a) HPF with Simple Differential OTA (HPF1)



(b) HPF with Mirrored OTA (HPF2)

**Figure 4.15:** Measured Test IC Results-Frequency Response of the Two OTA-C High Pass Filter Structures



some element of offset, due to the inherent mismatch present in CMOS processes. Layout techniques can be adopted to reduce this offset, including placing similar structures with the same geometry and common centroid layout[90]. In an effort to reduce the offset of the high pass filters, common centroid techniques were employed, in the form of constructing large transistors from smaller, inter-digitated versions. In addition, dummy transistors were added at the ends of the common-centroid implementation, in an effort to ensure each small transistor 'sees' the same structures to the left and right. Guard ring structures were also included to protect the transistors from noise, as highlighted previously in figure 4.14.

When measuring the DC offset from the filter, it is important to remember that the buffer circuit may also exhibit some offset, which will effect the measured results. A measure of the buffer's DC offset when biased with a control voltage of 1V can be seen in figure B.2. It seems that the buffer's offset increases with the input DC level, from 32 mV at 1 V to 91 mV at 4.5 V.

The DC offset for each filter was tested by fixing the reference point of the filter at 2.5 V. The input was then generated with a signal generator, with the input DC level ramped from 0 V to 5 V. In theory, the filter's output should remain biased at the 2.5 V reference level, completely independent of the rising input. The results for HPF1 and HPF2 can be seen in figure 4.16. It is clear from (a) and (c) that the output from both high pass filters does remain close to 2.5 V, confirming the correct operation. This also suggests that the low frequency data in the bode plots of figure 4.15 is misleading due to the extremely low signal to noise ratio. The detail in figure 4.16 (b) shows the variation in offset for HPF1 as the control voltage of the filter is varied from 0.7 V to 0.6 V. From first inspection, all three bias conditions produce relatively constant levels of offset, with 0.6 V exhibiting roughly 50 mV, 0.65 V having approximately 40 mV and 0.7 V producing about 45mV between the filter output and the reference level.

The same experiment was applied to the second high pass filter, with the results detailed in figure 4.16 (d). Once again, the offset seems fairly constant, with all three bias conditions producing very similar offsets. In this case, a bias of 0.6 V provides approximately 55 mV of offset, 0.65 V gives around 54 mV, and 0.7 V exhibits 55 mV. The initial reason for implementing a mirrored OTA structure was its superior offset performance, although these results suggest this may not be the case when biased in the subthreshold regime. The mismatch between weak inversion currents may go some way to explaining this, with the simple differential OTA having only one current mirror, while the mirrored version incorporates three. However, the mismatch appears more constant for the second HPF, which may be an advantage for this particular ap-



plication. Despite all these points, it is clear there is very little difference between the two filter structures in terms of DC offset, with both exhibiting expected amounts.

#### **4.5.6 Comments on the OTA-C First Order High Pass Filter**

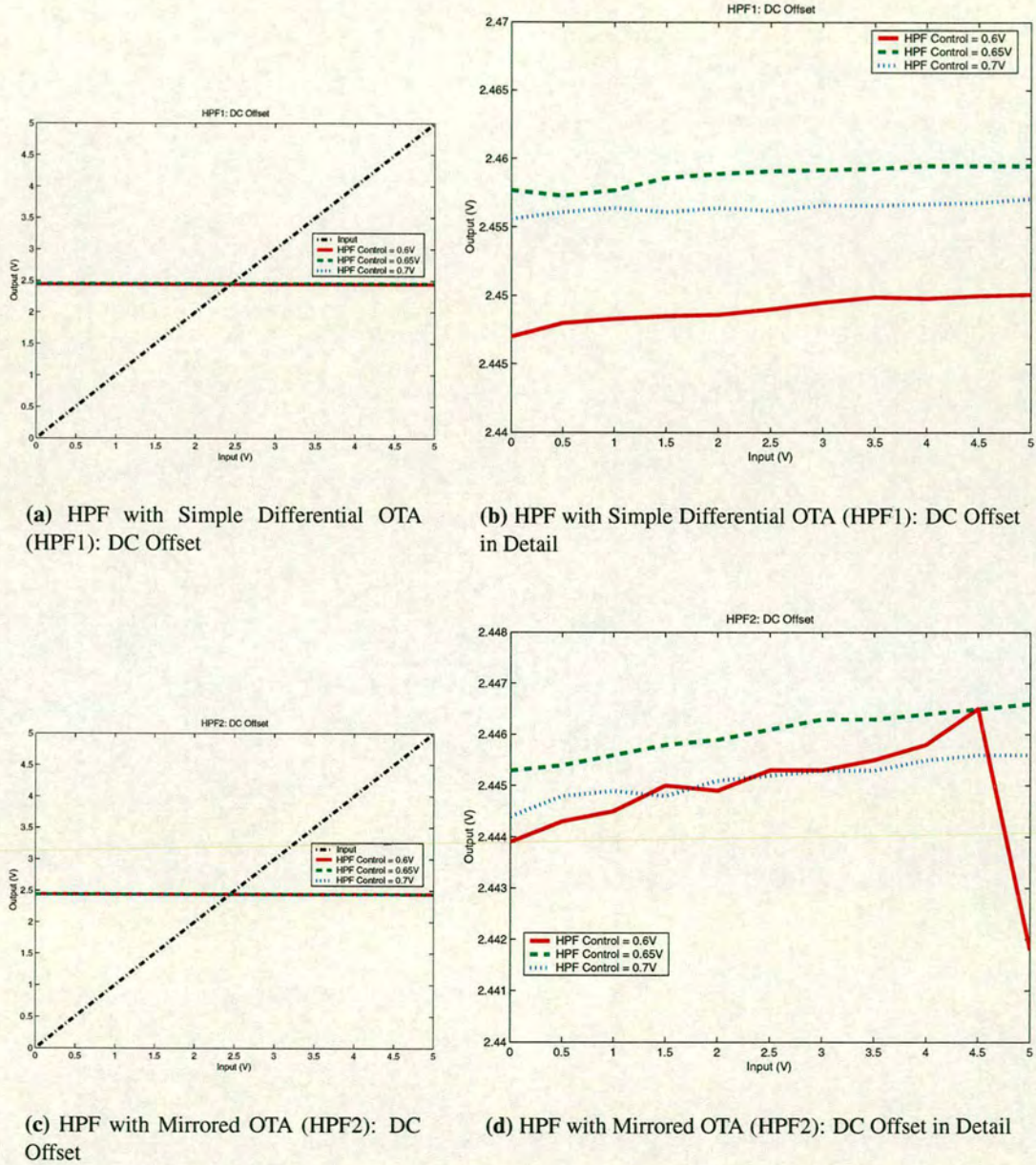
The aim of the high pass filter in the algorithm depicted in figure 4.1 is to remove the DC level from the output of the photocircuit and superimpose the transient information on a well-defined reference level. The need for continuous-time circuitry led to an investigation of Gm-C filter techniques, while area constraints led to the implementation of a simple, first order structure. The need for very low cutoff frequencies suggested the use of subthreshold bias currents for the transconductance elements, allowing large time constants to be achieved. This approach is also relevant from a system viewpoint, given the emphasis on low power signal processing specified by the sponsor company. Other potential advantages of OTA-C filter structures biased in weak inversion include tunability with a single control voltage or current, as well as a much wider potential tuning range than the same technique biased in strong inversion. However, the disadvantages of the approach include high mismatch between subthreshold currents and temperature sensitivity, all which serve to produce similar filter structures on the same IC having different cutoff frequencies. Nevertheless, it was felt that for this particular application, the fact that the cutoff frequency may vary from pixel to pixel will not have too drastic an effect on system-level operation, and the benefits of simple, low power and low area filtering outweighed the potential disadvantages.

The test IC results for both the filter structures behave as expected, with both exhibiting readily tunable low frequency time constants. However, the attenuation level in each passband may be a disadvantage, as may the level of DC offset between the filter's output and its reference level. The structures behaviour in this respect is similar, with perhaps the reduced levels of DC offset for the simple differential OTA giving it a slight advantage over its mirrored OTA alternative. Despite the inherent DC offset, it is clear both filters superimpose the transient information onto the reference voltage, and as such can be considered successful.

## **4.6 Comparator**

The final element of the fundamental frequency extraction algorithm involves thresholding the filter's output with a comparator. In keeping with the previous circuit blocks, an emphasis was





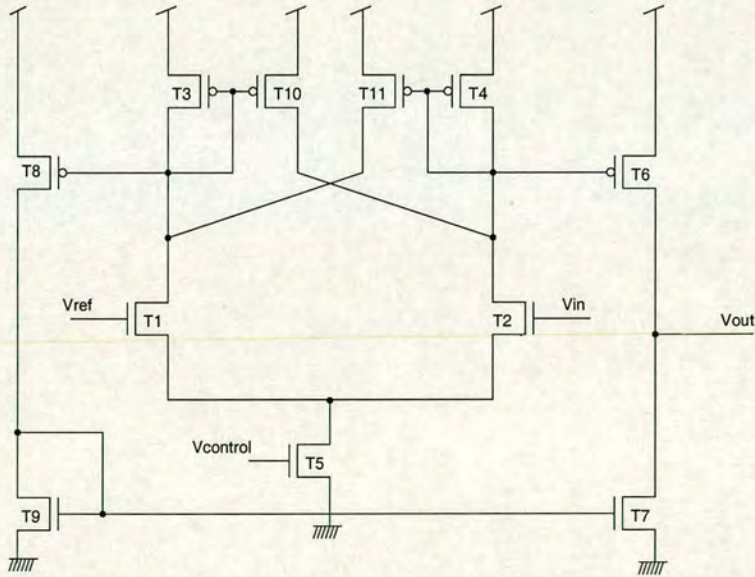
**Figure 4.16:** Measured Test IC Results-DC Offset of the Two OTA-C High-Pass Filter Structures: The output should remain biased at 2.5V, despite the increasing DC level of the input



placed on low power, continuous time and simple circuit techniques. The adopted approach involves the use of positive feedback to force an amplifiers output to either the positive or negative power supply rail, depending on the input conditions.

#### 4.6.1 Comparator with Positive Feedback and Optional Hysteresis

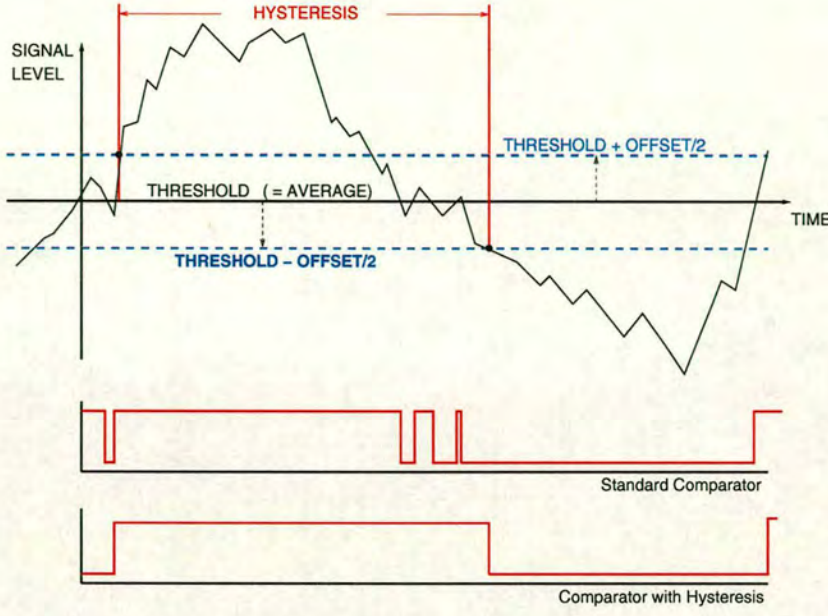
The chosen circuit[107] can be seen in figure 4.17. The reason for its choice stems from the ability to include optional hysteresis at the circuit design stage. Hysteresis allows a comparator to operate successfully in a noisy environment, by varying the *trip-point* of positive and negative excursions. It was felt that the output from the filter may exhibit some temporal noise, which could conceivably cause spurious pulses in the comparators output. The effect of hysteresis can be seen in figure 4.18. The comparator switches at the threshold plus or minus the offset, for positive and negative excursions respectively, reducing the likelihood of noise producing spurious output pulses.



**Figure 4.17:** Comparator with Optional Hysteresis

The circuit in figure 4.17 works by exploiting positive feedback to force the comparators output to either the positive or negative rail. Consider the case when the circuit in figure 4.17 is powered by a zero to five volt supply rail, with the reference point set in the centre at 2.5 V. If the input is at ground, T1 will be fully on, while T2 will be completely off. This means that the tail current from T5 will all flow down the left hand side of the T1-T2 differential pair, turning





**Figure 4.18:** *Effect of a Comparator with Hysteresis on a Noisy Signal: One a positive excursion, the comparator switches at the threshold plus the offset, while for the negative case it switches at the threshold minus the offset. Adapted from [107]*

T3 and T10 on and ensuring T4 and T11 are off. The gate of T8 is subsequently pulled low, meaning transistors T9 and T7 are turned on, creating a path from the output to ground. At the same time, the gate of T6 is pulled high, ensuring that the output is held low.

If the input now starts to increase, some of the tail current will begin to flow through T2. At some point, the current sunk by T2 will equal the current sourced by T10, and the comparator will begin to switch state. By equating these currents and using them to calculate the gate source voltages of T1 and T2, it is possible to calculate the positive trip voltage, based on the more extensive analysis in[107]:

$$i_1 = i_3 = \frac{i_5}{1 + [(W/L)_{10} / (W/L)_3]} \quad (4.7)$$

$$i_2 = i_5 - i_1 \quad (4.8)$$

The positive trip voltage can be found by reversing equation 4.2 to make  $V_{gs}$  the focus, and solving equation 4.9, where  $\beta = \mu_0 C_{OX} \frac{W}{L}$ :



$$V_{trip+} = V_{GS2} - V_{GS1} = \left[ \sqrt{\left(\frac{2i_2}{\beta_2}\right)} + V_{t2} \right] - \left[ \sqrt{\left(\frac{2i_1}{\beta_1}\right)} + V_{t1} \right] \quad (4.9)$$

Similar analysis yields a relationship between the currents in T4 and T11 for the negative trip point:

$$i_2 = i_4 = \frac{i_5}{1 + [(W/L)_{11} / (W/L)_4]} \quad (4.10)$$

$$i_1 = i_5 - i_2 \quad (4.11)$$

$$V_{trip-} = V_{GS2} - V_{GS1} = \left[ \sqrt{\left(\frac{2i_2}{\beta_2}\right)} + V_{t2} \right] - \left[ \sqrt{\left(\frac{2i_1}{\beta_1}\right)} + V_{t1} \right] \quad (4.12)$$

The important thing to notice is the dependence on the aspect ratio of T3 and T10 for the positive trip voltage, and the corresponding negative trip voltages relationship to the size of transistors T4 and T11. It is clear that if T3, T4, T10 and T11 are all made the same size, the circuit behaves as a standard comparator without hysteresis, with the positive and negative switch points coinciding with the reference voltage. However, if the ratio of T3 to T10 and T4 to T11 is varied, the positive and negative trip voltages are changed, introducing hysteresis into the system. It is this property that made this particular circuit topology interesting for this application.

#### 4.6.2 Comparators Implemented on Test IC One

Two versions of the comparator circuit were implemented on the test IC, with and without hysteresis. For the purposes of this thesis, the comparator without hysteresis is termed the 'standard' comparator. In keeping with previous circuit elements, an effort to bias the comparator in the subthreshold region of operation was made to reduce the switching current of the device. It was felt that the low frequency range of the input signals may allow the tail current  $i_5$  to be limited, based on the low slew rate requirements. The idea was to allow control over the tail current on test IC one and to vary it until an acceptable compromise between power consumption and speed of operation was found. However, the amount of hysteresis in the com-



parator is directly related to the value of the tail current as equations 4.7 and 4.10 highlight. If the tail current is to be variable, there seemed little point in designing for a precise value of hysteresis. For this reason, a decision was made to minimise the value of hysteresis by selecting the smallest possible difference between T10 and T3, with the same difference between T4 and T11. This lead to the choice of transistor aspect ratios found in table 4.4. It was felt that the small signal variations from the output of the photocircuit, coupled with the 5 decibels of attenuation from the high pass filter would warrant small values of hysteresis.

Transistor	Standard Comparator $\frac{W}{L}$ ( $\mu\text{m}$ )	Comparator with Hysteresis $\frac{W}{L}$ ( $\mu\text{m}$ )
T1	6/1.5	6/1.5
T2	6/1.5	6/1.5
T3	3/1.5	3/1.5
T4	3/1.5	3/1.5
T5	5/2	5/2
T6	3/3	3/3
T7	1.5/3	1.5/3
T8	3/3	3/3
T9	1.5/3	1.5/3
T10	3/1.5	4/1.5
T11	3/1.5	4/1.5

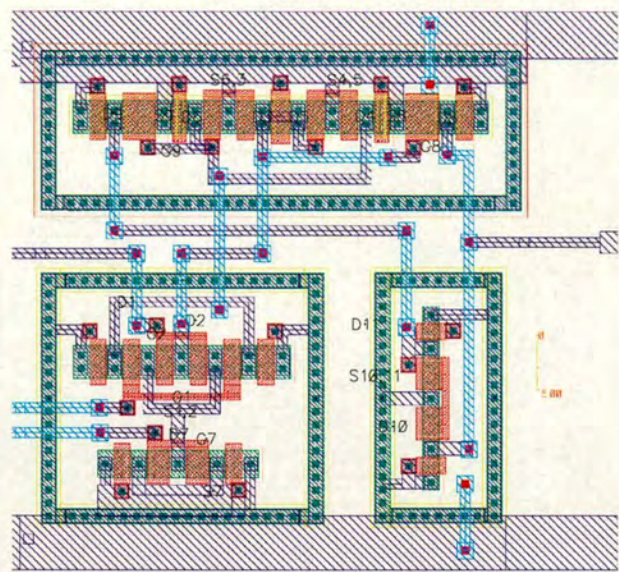
**Table 4.4:** Transistor Dimensions for the Comparator With and Without Hysteresis

The layout of the standard comparator can be seen in figure 4.19. As before, common centroid techniques, dummy transistors and guard structures were adopted in an effort to improve performance. Both comparator's dimensions are approximately  $45 \mu\text{m}$  by  $50 \mu\text{m}$ , although once again this could be reduced considerably.

### 4.6.3 Comparator Power Consumption

The approximate power consumption for the comparator for different control voltages was calculated using the Spectre simulation tool. The comparator exhibits both static power consumption, when the output is stable, and dynamic power consumption when the output is switching state. In an effort to combine both into a general power consumption value for the comparator, a time-domain plot of the current consumption was created. The reference was fixed at 2.5 V and the input varied around this level, to ensure the comparator output switches. The transient current waveform therefore contains information about both static and dynamic current consumption. This waveform was integrated using the waveform calculator's *average* function,





**Figure 4.19:** Physical Layout of the Standard Comparator (Without Hysteresis)

which performs the integral in equation 4.13. If  $y = f(x)$ :

$$\text{average}(y) = \frac{\int_{\text{from}}^{\text{to}} f(x)dx}{\text{to} - \text{from}} \tag{4.13}$$

Equation 4.13 produces a single output value which is effectively a time referred average of the input signal. Using this command, the results in table 4.5 were created. Both Sine waves were 10 mV in amplitude, biased at the reference of 2.5 V, while the square wave swept the full range from 0 V to 5 V.

Control Voltage	100 Hz Sine	1 kHz Sine	100 Hz Square
0.6 V	677 pA	-	787 pA
0.75 V	32.15 nA	32.2 nA	37.7 nA
1 V	3.065 $\mu$ A	3.065 $\mu$ A	3.41 $\mu$ A

**Table 4.5:** Simulated Bias Current Consumption of Standard Comparator

As expected, the current consumption reduces as the control voltage is reduced, biasing the comparator further into the subthreshold regime. As the frequency of the input increases, the power consumption remains relatively constant as the averaging function takes into account the reduced time interval. However, the gradient with which the input passes through the com-



parator's reference point is increased as the waveform changes from a sine wave to a square wave, producing a corresponding increase in current consumption. Large switching transients are seen at the actual switch points, although these last for extremely brief time periods.

#### **4.6.4 IC Test Results: Comparator**

Both versions of the comparator were tested to ensure correct operation, as well as to ascertain certain performance relevant criterion. Measurement of the effect of control voltage on speed of operation was made, together with the magnitude of the hysteresis.

##### **4.6.4.1 Comparator Switching Frequency**

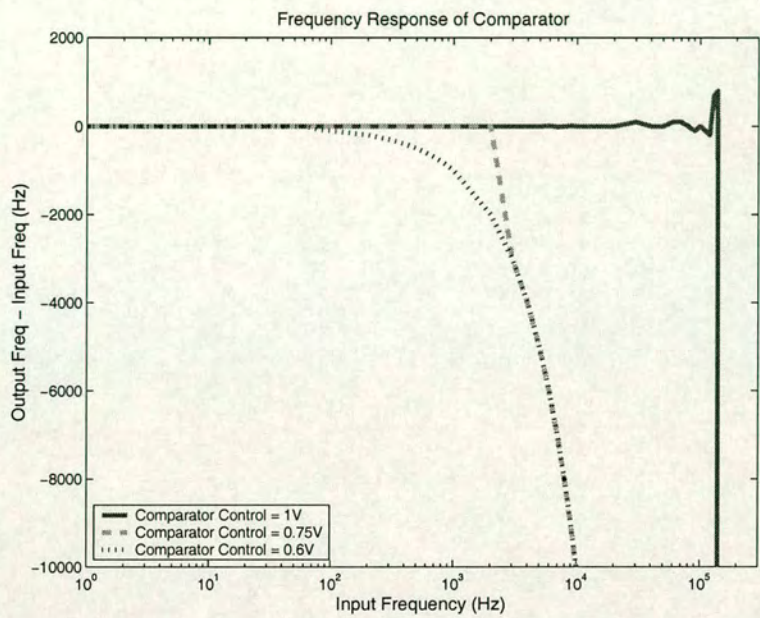
One of the fundamental aims for the system developed in this research is low power processing. As such, the control voltage for the comparator was made variable in an effort to limit the bias current, therefore reducing power consumption. However, reducing the current in the system has the effect of reducing the maximum speed of operation that the comparator can switch. By fixing the reference at 2.5 V and applying a sinusoidally varying input around this reference, the comparator is guaranteed to switch. The results in figure 4.20 were made by calculating the difference between the input and output frequencies. The results for the standard comparator in figure 4.20 (a) are very similar to those for the version with hysteresis, with a strong inversion bias of 1 V able to switch up to almost 100 kHz. As the control voltage is reduced to 0.6 V, both comparators maximum switching frequency moves down to nearer 100 Hz.

##### **4.6.4.2 Hysteresis Measurement**

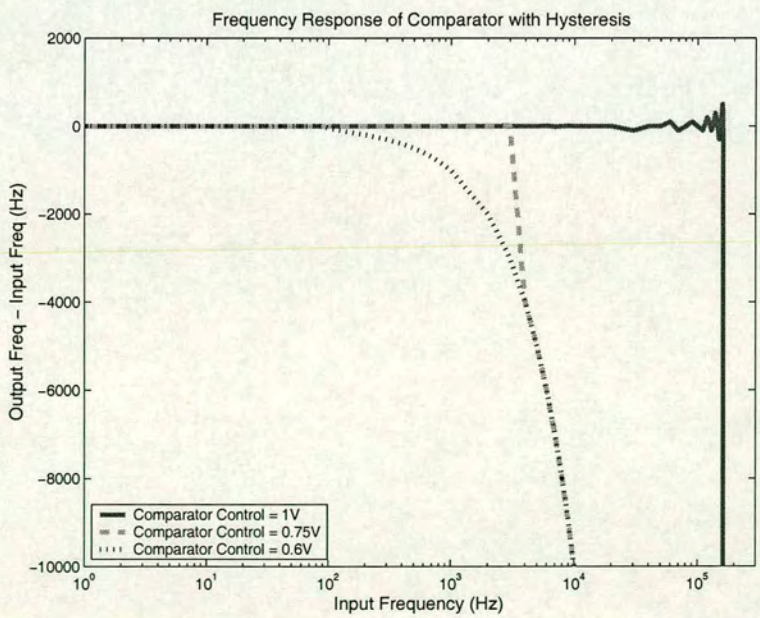
To confirm the difference between the two comparators, an estimate of the amount of hysteresis present in the second comparator was made, for a selection of different bias conditions. By once again fixing the reference point at 2.5 V and applying a sinusoid around this DC level as the input, the comparator produced a series of output pulses. By applying such an input to both comparators, the difference between the positive and negative switch points for the standard and hysteresis comparators could be calculated.

The screen-shots in figure 4.21 were made with an Agilent A54624A oscilloscope, and depict the difference in switching between the two comparators implemented on test IC one. Figure 4.21 (a) and (b) depict the positive and negative transitions of both comparators respectively,





(a) Standard Comparator



(b) Comparator with Hysteresis

**Figure 4.20:** Measured Test IC Results-Comparator Switching Frequency: The maximum switching frequency decreases as the bias current is reduced



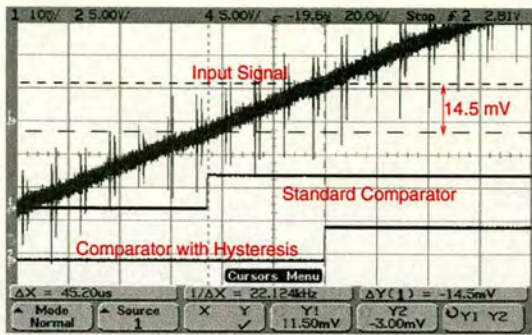
with the control voltage at 1V and a 100 Hz input signal. On the positive transition there is a difference of 14.5 mV between the standard comparator and the version with hysteresis, while the downward switch exhibits 22.5 mV hysteresis. For the same control conditions but with an increased input frequency of 1 kHz, the positive hysteresis is 12 mV while the difference on the negative transition is 20.5 mV, as highlighted in figure 4.21 (c) and (d). As expected, the hysteresis remains roughly the same for similar bias conditions. However, the difference between the positive and negative hysteresis is unaccounted for, particularly considering the ratio between transistors T3 and T10 is the same as that between T4 and T11. Despite this, the actual size of the hysteresis is not so important in this application, as it will have little effect on the accuracy of the frequency pulses. The same test was applied with a reduction in bias current, with the results for a control voltage of 0.75V and a 10 Hz input depicted in figure 4.21 (e) and (f). As expected, the positive hysteresis is reduced, down to almost 5 mV. However, the negative hysteresis remains relatively high at 22.5 mV, which was unexpected. It may be that by biasing the comparator with a subthreshold current, the resultant increase in current mismatch effects the hysteresis. The circuit relies heavily on current mirrors to produce ratioed versions of the tail current. As the mismatch is a DC phenomenon, the fact that the hysteresis is different for positive and negative excursions is unimportant in this application. If however the variation in hysteresis changes with time, this may effect the accuracy of the pulse timing, ultimately reducing the accuracy of the extracted fundamental frequency.

#### **4.6.5 Comments on the Comparator**

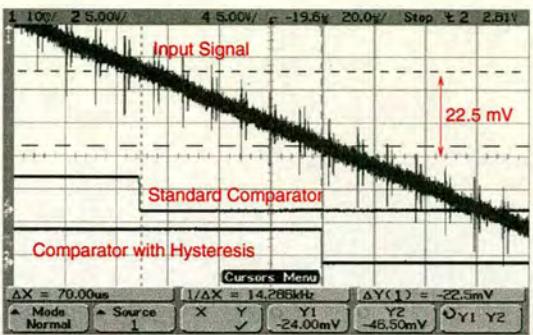
The IC test results confirm the fact that both comparators operate as expected, and are capable of operating with subthreshold currents as the results in figure 4.20 highlight. With a control voltage of 0.75 V, the comparator is still able to switch up to an input frequency of almost 2 kHz, while a 0.6 V control voltage reduces this maximum switching frequency to nearer 100 Hz. However, it is clear that for true subthreshold operation, the switching current from the comparators output stage would need to be limited. As the circuit diagram in figure 4.17 shows, the output is formed by transistors T6 and T7 sourcing or sinking sufficient current respectively to create the output voltage swing. It is this switching current that will govern the dynamic power consumption of the comparator circuit. Nevertheless, by using reduced bias currents elsewhere in the system, a significant power saving will be made.

The specific value of hysteresis in the comparator is directly dependent on the value of the tail

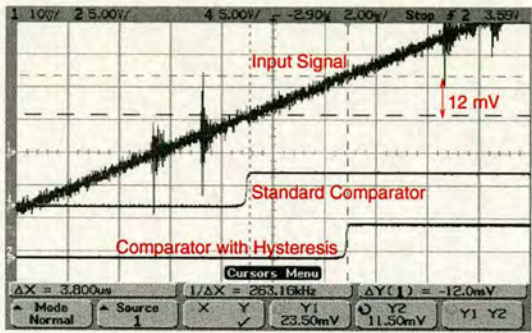




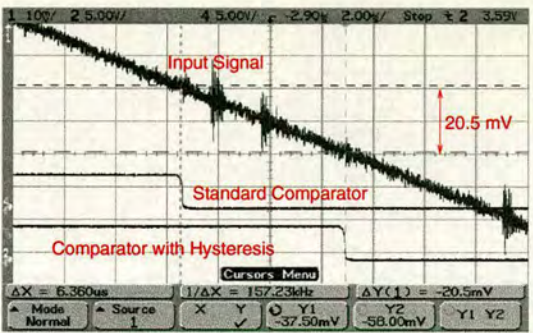
(a) Positive Transition: Control = 1V, Freq = 100 Hz



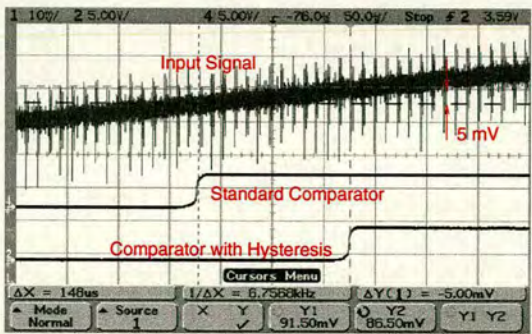
(b) Negative Transition: Control = 1V, Freq = 100 Hz



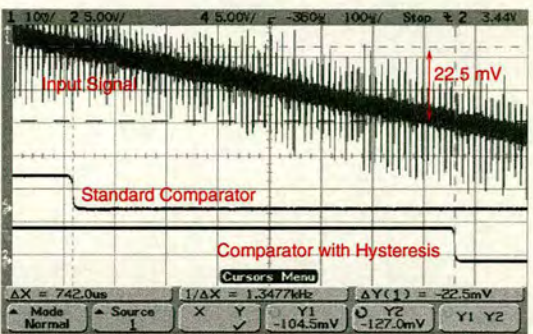
(c) Positive Transition: Control = 1V, Freq = 1 kHz



(d) Negative Transition: Control = 1V, Freq = 1 kHz



(e) Positive Transition: Control = 0.75V, Freq = 10 Hz



(f) Negative Transition: Control = 0.75V, Freq = 10 Hz

**Figure 4.21:** Measured Test IC Results-Direct Comparison of Switching for Comparator with and without Hysteresis: The amount of hysteresis on positive and negative transitions



current. However, the screen-shots in figure 4.21 suggest that there is a difference in the magnitude of the positive and negative hysteresis. The aim of the comparator is simply to threshold the high pass filter's output, producing a pulse train whose frequency corresponds directly to the fundamental frequency of the filter's output. As such, providing the relative size of the hysteresis does not vary greatly, the noted difference between positive and negative value is irrelevant. More specifically, providing the positive and negative hysteresis for a specific control voltage remains relatively constant, the difference between the two will not effect the accuracy of the output pulse frequency. From the screen-shots in figure 4.21 (a) and (c), the positive hysteresis for a 1 V control voltage appears similar, despite the change in input frequency. The same is also true for the negative hysteresis, highlighted in figure 4.21 (b) and (d). This seems to suggest that the hysteresis will not effect the accuracy of the output pulses.

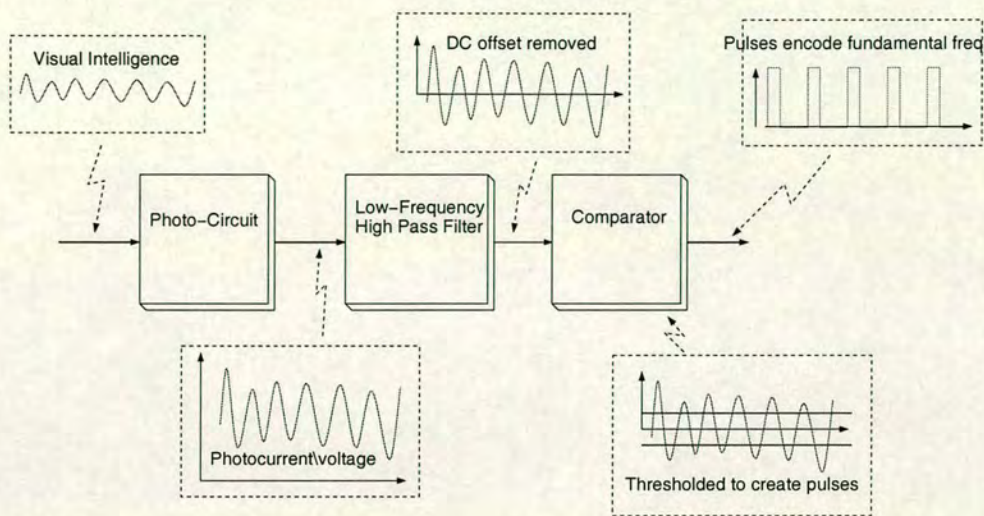
## 4.7 System Level Test IC Results

From a system viewpoint, the three circuit structures described in this chapter are combined to implement the *No-Mask* Algorithm described in detail in chapter three. The **photocircuit** takes the photocurrent from the **photoelement** to make a voltage signal dependent on the incident light intensity. This voltage is then passed through a low frequency, first-order **OTA-C high pass filter**, biased in subthreshold, to remove the DC level and place the AC information on a stable reference level. This reference is then used as the thresholding level for a **comparator**, the idea being that the frequency of the output pulses directly encodes the fundamental frequency of the incident light intensity. The filter essentially ensures that the comparator is biased in a *sensitive place* to ensure that it switches with the photocircuits output. The various processing steps can be seen in figure 4.22.

The layout of such a fundamental frequency extraction algorithm can be seen in figure 4.23. The dimensions of the pixel processing unit are approximately  $180\ \mu\text{m}$  by  $170\ \mu\text{m}$ , with a photoelement of  $50\ \mu\text{m}^2$ , producing a fill factor of just over 8 %. However, as the layout highlights, little effort was made to minimise the processing unit. It should be possible to drastically reduce the size of the implementation once correct operation has been confirmed.

The ultimate test of the success of the approach stems from the accuracy of the output pulse with respect to that of the incident light intensity. In order to test this property, a series of combinations of the different processing elements described in this chapter were included on





**Figure 4.22:** Signal Flow for Fundamental Frequency Extraction Algorithm One: The No-Mask Algorithm

the IC, each described as a fundamental frequency finding *pixel cell*. A summary of the different pixel cells and their constituent parts can be found in table 4.6. The different combinations of the the two high pass filter structures and the two comparator structures allow an assessment of the relative strengths and weaknesses of each.

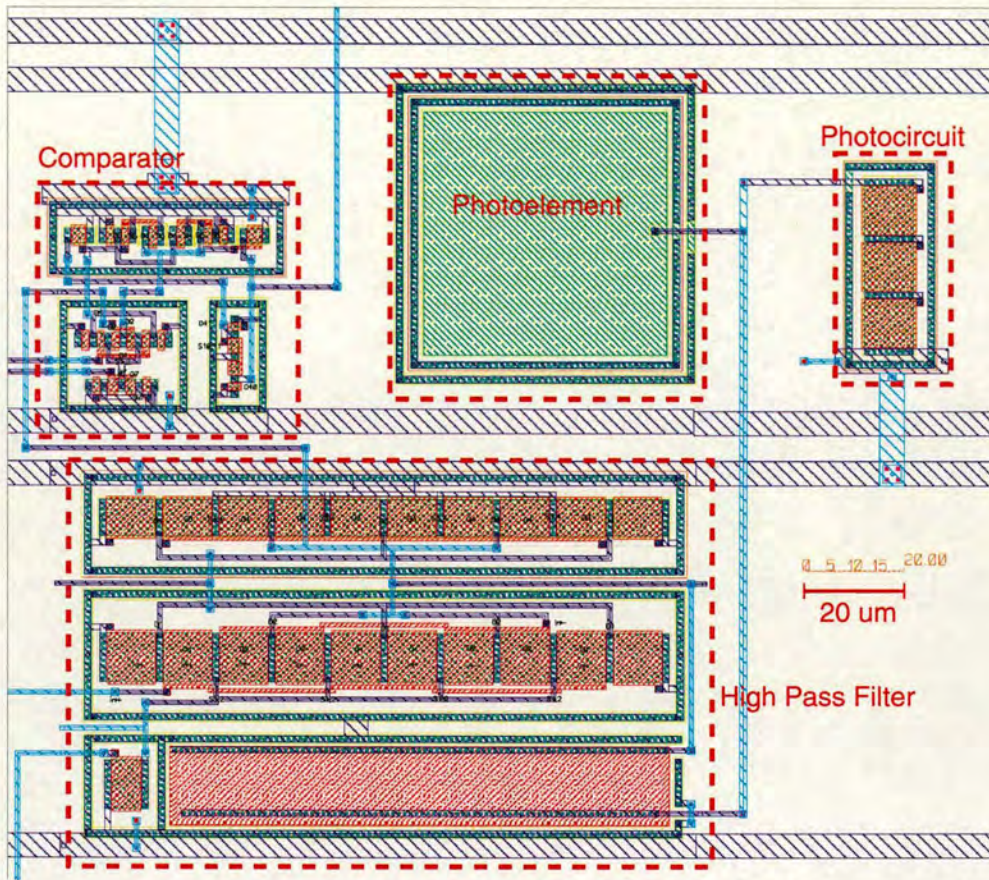
Pixel Cell	Photoelement	Photocircuit	HPF	Comparator
One	Photodiode	Logarithmic	HPF1	Standard
Two	Phototransistor	Logarithmic	HPF1	Hysteresis
Five	Photodiode	Logarithmic	HPF2	Standard
Six	Phototransistor	Logarithmic	HPF2	Hysteresis

**Table 4.6:** Pixel Processing Cells On Test IC One

**4.7.1 Pixels One and Two: Frequency Accuracy**

The differences between pixels one and two are the inclusion of hysteresis in the comparator of the latter, as well as the use of a photodiode in pixel one and a phototransistor in pixel two. Both pixels were tested for frequency accuracy by illuminating an LED with a signal generator. The transient intensity is therefore directly controllable, allowing the input frequency and therefore the illumination’s temporal frequency to be varied. By setting both the reference point of the filter and the comparator’s reference midway between the power rails at 2.5 V, the filter’s output should cause the comparator to pulse. The frequency of these pulses should then correspond to





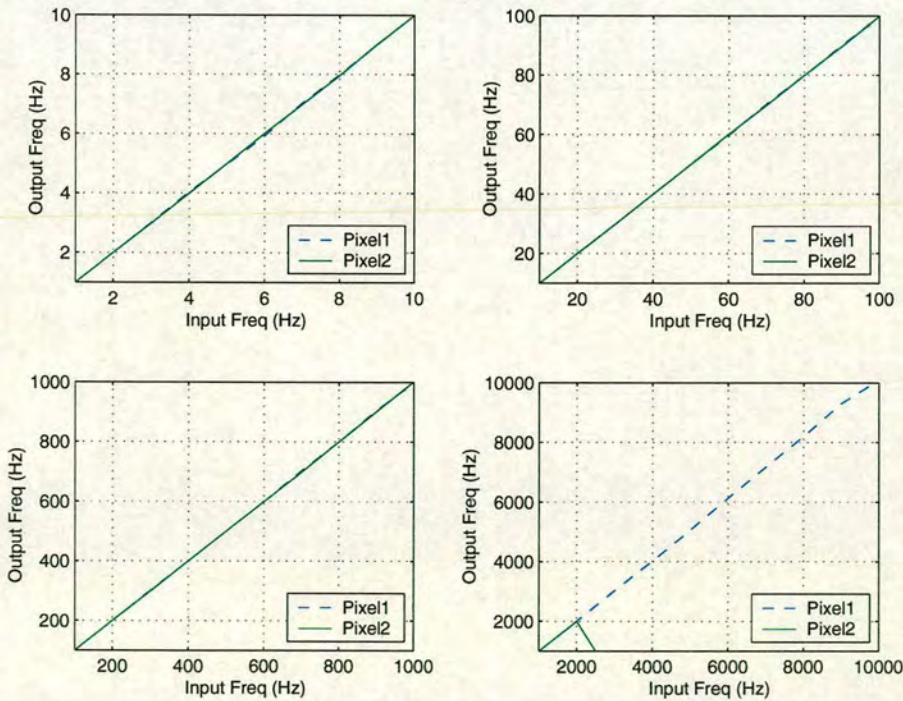
**Figure 4.23:** Physical Layout of Fundamental Frequency Extraction Algorithm: Pixel Two

the current LED modulation frequency.

This experiment was carried out for pixel's one and two, with the filter control voltage fixed at 0.51 V. The input frequency was ramped from 1 Hz to 10 kHz in logarithmic steps, with the results in figure 4.24 showing the accuracy of the output pulse train. Between 1 Hz and 10 Hz, both pixels' output pulse train frequency accurately reflects the input. The same is true for 10 Hz to 100 Hz and 100 Hz up to 1 kHz. However, in the 1 kHz to 10 kHz frequency band, pixel two ceases to operate at approximately 2 kHz, while pixel one is still relatively accurate up to 10 kHz. The reasons for this stem from the inclusion of hysteresis in pixel two. Figure 4.25 shows the combined frequency response from all three circuit blocks in pixels one and two. These were created by once again by modulating an LED with a signal generator. The magnitude of the output voltage from the filter was measured with the comparator by varying the threshold voltage and noting its value when the comparator output has just fallen low and stopped producing pulses. If the threshold is now varied until the comparator's output

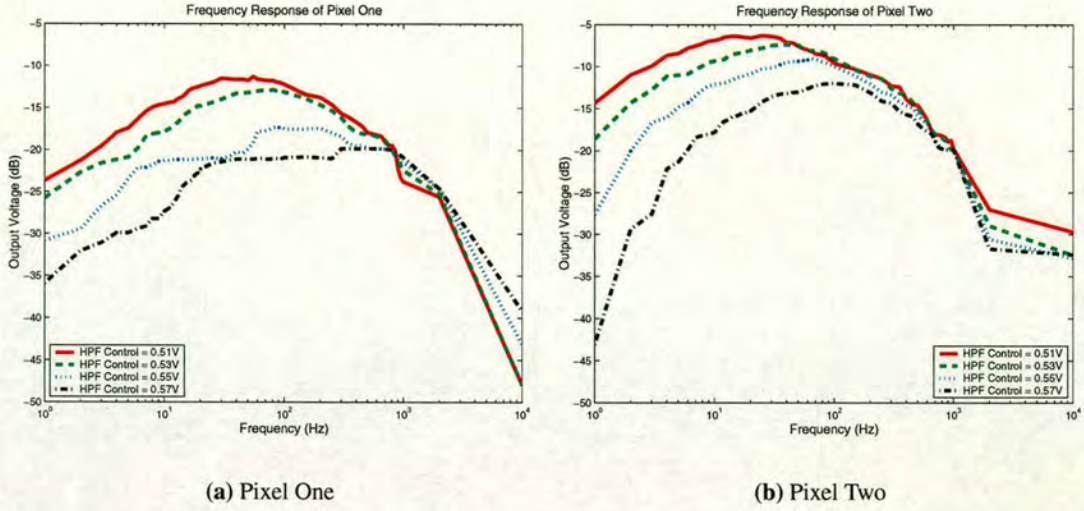


is completely high, the difference between the two reference points conveys the magnitude of the comparators input. As such, the bode plots for pixels one and two in figure 4.25 (a) and (b) respectively combine the frequency performance of the photocircuit, high pass filter and the comparator. The increased photocurrent produced by pixel two's phototransistor when compared with the photodiode from pixel one is clearly evident, resulting in an increased signal swing of over 5 decibels. The pixel's frequency response appear band pass in shape, due to the combination of the high pass filter and the low pass response of the photocircuit. At higher frequencies, the attenuation of the photocircuit is such that the comparator with hysteresis is simply unable to switch, due probably to the fact that the signal swing is smaller than the hysteresis itself. This is highlighted in the bode plot in figure 4.25 (b), where above about 2 kHz the bode plots remain flat as no magnitude information could be extracted. The comparator without hysteresis is still able to switch, as the results in figure 4.25 (a) for pixel one highlight. This hypothesis is confirmed by the frequency accuracy results for pixels one and two in figure 4.24. It would appear that the hysteresis may intuitively offer advantages in terms of accuracy, as the smoothness of the bode plot in figure 4.25 (b) seems to suggest, but at the price of a reduction in sensitivity.



**Figure 4.24:** Measured Test IC Results-Frequency Accuracy of Pixel Cells One and Two





**Figure 4.25:** *Measured Test IC Results-Combined Frequency Response of the Circuit Elements Combined to Produce Pixel Processing Cells One and Two: The Bode plots convey the combined frequency response of the photocircuit, HPF and comparator, which combine to produce a fundamental frequency extraction pixel cell*

#### 4.7.2 Pixels Five and Six: Frequency Accuracy

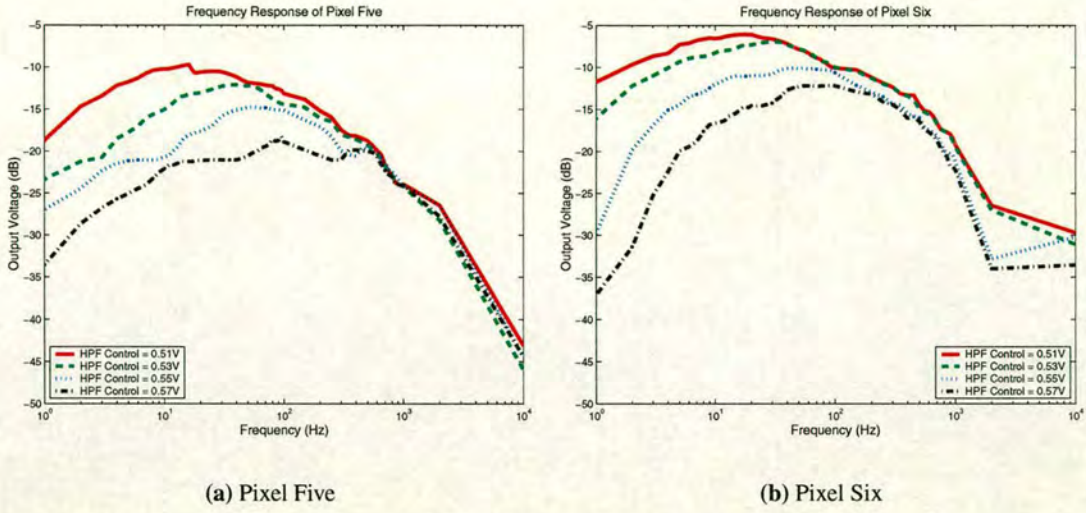
A similar experiment was performed on pixels five and six. These can be considered analogous to pixels one and two respectively, but the simple differential OTA filter has been replaced by the mirrored OTA structure. Once again, the system bode plots were developed as before, with the resultant bode plots in figure 4.26. A similar trend to that exhibited by pixels one and two is depicted, with the phototransistor from pixel six providing a larger signal swing. As previously discussed, this appears to be an advantage in this particular application, as each pixel essentially behaves as an independent frequency sensitive device. The hysteresis in pixel six also serves to reduce the sensitivity compared to pixel five.

The frequency accuracy for pixels five and six is depicted in figure 4.27. Both circuits operate over the full frequency range, from 1 Hz to 10 kHz.

### 4.8 Comments on Test IC One

The aim for test IC one was to investigate different methods of implementing the *no-mask* algorithm. To this end, a photocircuit, two different low frequency high pass filter implement-





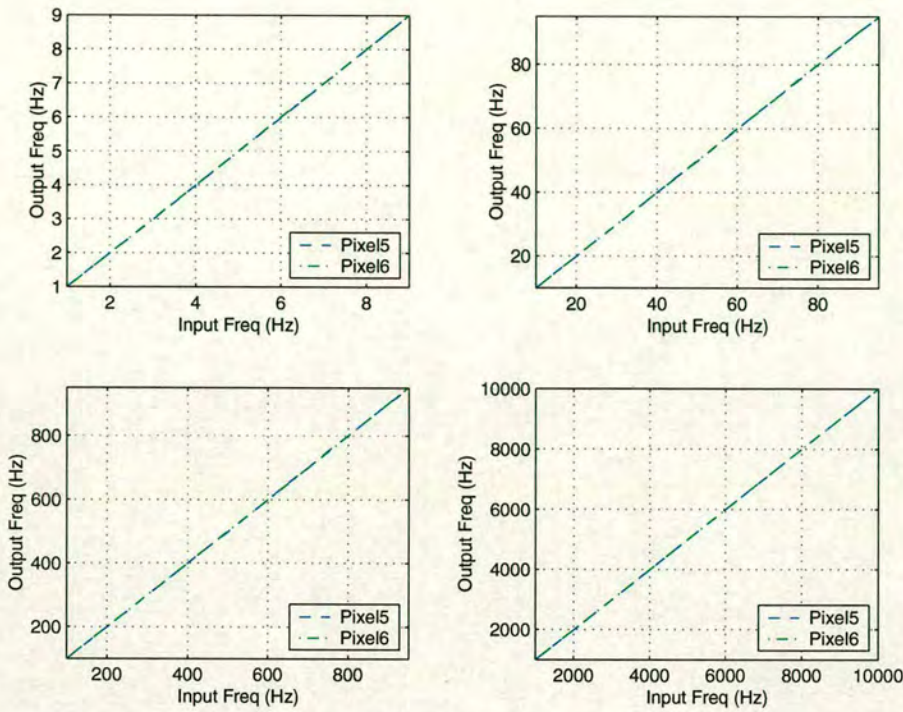
**Figure 4.26:** *Measured Test IC Results-Combined Frequency Response of the Circuit Elements Combined to Produce Pixel Processing Cells Five and Six: The Bode plots convey the combined frequency response of the photocircuit, HPF and comparator, which combine to produce a fundamental frequency extraction pixel cell*

ations and a comparator with and without hysteresis were designed. These elements were then combined to produce **pixel processing cells**, whose task was to extract the fundamental frequency from the incident light intensity. From the frequency accuracy results in figures 4.24 and 4.27, it is clear that the system is capable of accurately extracting the fundamental frequency, over a range of 1 Hz to 10 kHz, depending on the combination of circuit elements.

#### 4.8.1 Conclusions on Photoelement

It is clear from both theory[6] and IC results such as figure 4.3, that the phototransistor produces larger signal swings for the same incident light intensity, based on its  $\beta$  multiplication factor. However, this factor is subject to process variation, ruling out its inclusion in standard image sensors, where many pixel cells are combined to produce an accurate representation of the scene. The application described in this thesis does not suffer from the same limitations, as the absolute value of the incident light intensity from one pixel to the next is unimportant. The ultimate aim is to find the relative strength of the fundamental and the first four harmonics for each pixel location. As such, if two neighbouring pixels produce different amplitude signal swings for the same light intensity, the underlying relative amplitude between frequency signatures will





**Figure 4.27:** Measured Test IC Results-Frequency Accuracy of Pixel Cells Five and Six

remain the same. Each pixel can be considered as an independent processing element, allowing the use of a phototransistor as the photoelement and gaining from the increased photocurrent.

#### 4.8.2 Conclusions on Photocircuit

The logarithmic photocircuit employed in this research suffers from a series of potential limitations. The logarithmic compression that reduces the eight orders of potential light intensity into an acceptable voltage swing ensures that it can operate in a variety of ambient light conditions. However, the aim of the project is to analyse transient changes in intensity, which in real-world applications may be too small to register a significant output voltage. As demonstrated with simulations in figure 4.3, the sensitivity of the photocircuit can be increased by increasing the number of load transistors. However, this comes at the cost of a reduction in bandwidth, as the simulation results in figure 4.5 highlight. Another area of possible concern is the use of this photocircuit with the increased photocurrent from the phototransistor. As figure 4.3 shows, if the photocurrent nears strong-inversion magnitudes, the logarithmic compression gives way to a square root relationship, resulting in the output voltage collapsing. However, at low light levels, the increased photocurrent will be an advantage, suggesting this may be an application



specific design trade-off. As previously mentioned, the adaptive photoreceptor developed by Delbruck [12] may be a more sophisticated solution, which could be adopted at a later date.

The advantages of the logarithmic compression photocircuit include its simplicity and resultant low cost in terms of silicon area. The circuit also fits the design criterion in that it produces a continuous time output voltage that is a function of the incident light intensity. The circuit also benefits from very low power operation, as the photocurrent is used to bias the circuit. Despite its shortcomings, the circuit is a simple method for converting the light intensity into a suitable voltage signal. As such, it was deemed a good compromise between size, power consumption and processing power at this stage of the research.

### **4.8.3 Conclusions on High Pass Filter**

The high pass filter was implemented with OTA-C circuit techniques, due to the requirements for continuous-time operation. By biasing the circuit in the subthreshold regime, the constraints of both low frequency time constants and low power operation were achieved, meeting the system-level requirements. From this perspective, such an approach is highly suited to the application in this research. It is estimated that a first order high pass filter with a cutoff freq of 100 Hz consumes a mere 200 pA, allowing many to be realised on chip. Low cutoff frequencies are realised with a reasonable sized capacitance of 1 pF, producing small, low power filtering devices.

However, the approach does suffer from some potential limitations. The statistical mismatch between subthreshold currents[94, 97] means that no two filters will have the same cutoff frequency. In this application, the filter's task is to remove the DC level from the photocircuit's output, and superimpose the transient (AC) information onto a well defined reference level. It is clear from the IC test results in figure 4.16 that the filters successfully remove the input DC level, biasing the output at the reference. The frequencies of interest are as low as 1 Hz, meaning the filter's cutoff frequency should conceivably be lower than this value. However, as the filters are first order structures, the attenuation around this level is relatively small. As such, any variation in the filters cutoff due to poorly matched subthreshold currents will merely serve to reduce the magnitude of the filter output. At the very worst, the comparator for that particular pixel may not be able to switch, producing no output pulses as the filter's output is simply too small to threshold. No output pulse train is preferable to a spurious output frequency. If the filter's cutoff frequency is moved lower due to subthreshold current mismatch, the output mag-



nitude will be unaffected as the filter will simply try to pass even smaller input frequencies. It is theoretically impossible to pass the input DC level to the output, which is the filter's sole task to prevent. If the cutoff is moved higher in frequency, the output magnitude will decrease until the comparator can no longer switch. It should be possible to select a value of cutoff frequency for every pixel based on statistical measurements of subthreshold mismatch, to ensure that most pixel's comparators are capable of switching.

Another potential problem with the filter is the DC offset between the filter's output and the reference level. The idea was to use the same reference level for both the filter and the comparator for all the pixels in the imager system. However, the DC offset measurements in figure 4.16 suggest that this may not be feasible. The size of the offset is between 40 mV to 50 mV, which may be larger than the actual magnitude of the filter's AC output, meaning the reference point of the comparator would have to be varied to make it switch. The five decibels of attenuation in the passband of both OTA-C implementations is another potential problem. The AC magnitude of the photocircuit's output may be small as it is, without the extra attenuation through the filter. There appears to be very little difference between the two OTA-C filter structures implemented on test IC one, with both the simple differential stage and the mirrored version producing similar results. Both OTA structures are essentially behaving as nothing more than tunable resistors, based on the inverse of their transconductance.

Despite the potential disadvantages, there are few other circuit techniques that allow such low frequency, low area and low power filter structures to be realised. Coupled with the simple tunability properties, the OTA-C filters biased in subthreshold are the best approach for the given application.

#### **4.8.4 Conclusions on Comparator**

The two comparator structures implemented on test IC one were found to operate as expected. The difference between the standard version and that with hysteresis is highlighted in the oscilloscope screen-shots in figure 4.21. It was discovered that there seems to be some variation in the size of the positive and negative hysteresis. However, providing the size of both remains relatively constant, the fact that they are different with respect to each other will not effect the accuracy of the output pulse frequency. Based on the test results, this would seem to be the case.



The test IC results in figure 4.20 show that both comparators are able to switch with control voltages biasing the circuit in the subthreshold region of operation. With a control of 0.75 V, the current consumption is roughly 37 nA, based on circuit simulation measurements. It is clear that the static power consumption of the device is low. However, the dynamic power consumption is higher as the output stage will create momentarily large currents as the output switches. This could be limited in future implementations by controlling this switching current, possibly by including a transistor in the output stage to act as a current limiter.

#### **4.8.5 Conclusions on Pixel Processing Cells**

The circuit elements that combine to produce the fundamental frequency extraction elements have been shown throughout this chapter to operate as expected, although with certain performance limitations. When combined, the resultant pixel processing cells accurately extract the fundamental frequency of the incident light intensity[108], as the results in figure 4.24 and figure 4.27 confirm.

From a system viewpoint, it is clear from the pixel bode plots in figure 4.25 and figure 4.26 that the increased photocurrent from the phototransistor is an advantage in this application. As previously discussed, the increase in signal swing makes the thresholding task of the comparator simpler.

Including hysteresis in the comparator reduces the sensitivity of the system. Both pixels using comparators with hysteresis stop operating before the version without due to the increased signal attenuation at higher frequencies. However, the aim of hysteresis is to increase robustness to noise, meaning it may have advantages regarding accuracy. Both of these observations make intuitive sense, suggesting a trade-off between accuracy and sensitivity. The small size of the output signal swing from the photocircuit suggests that sensitivity may be of higher importance with this combination of components.

In general, the systems based on this approach worked well, with four pixel elements producing accurate output pulse trains. However, the DC offset from the output of the high pass filter may produce problems regarding the reference level of the system. As previously mentioned, if the magnitude of the filter's output is less than the offset, the reference point of the comparator will have to be changed. This effects the practicality of an array of such pixel processing structures operating independently. It is infeasible to have to set the reference for each pixel differently



from its neighbours. The output from the photocircuit is likely to be small in magnitude, due to the nature of its logarithmic compression. Coupled with this is the attenuation through the high pass filter, which reduces it further. The general feeling with the approach implemented on test IC one was that despite the promising results, it is too sensitive to be a realistic implementation. As feared, it was found that the reference points of the filter and the comparator had to be manipulated independently to create the output pulse train. The design meets the specifications for a low power, continuous time system capable of extracting the fundamental frequency of the incident illumination. However, it was decided to modify the system to test a slightly different system level approach. To this end, a second test IC was manufactured containing an improved fundamental frequency extraction algorithm.



---

## Chapter 5

# Test IC Two: Self-Referencing Fundamental Extraction

---

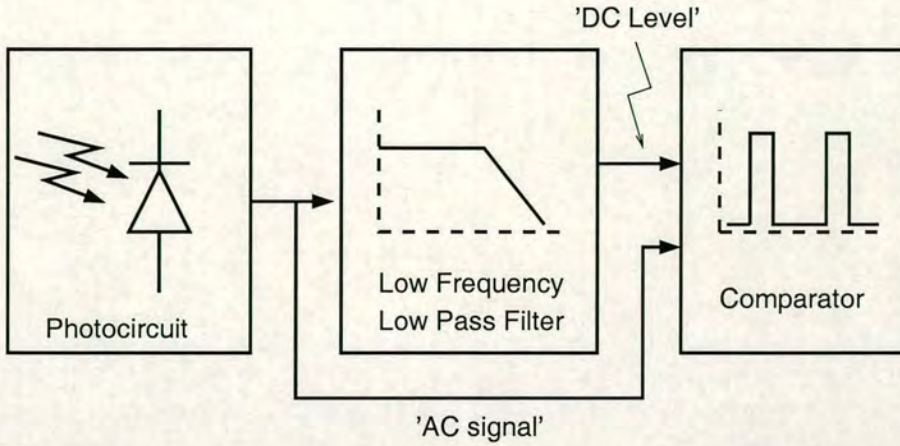
The results from the first test IC proved the feasibility of combining the photocircuit, low frequency OTA-C filtering and comparator structures into a fundamental frequency extracting processing unit. Test results showed the accuracy with which the fundamental frequency could be extracted, over the range of 1 Hz to 10 kHz. However, the ultimate aim is to create an imager consisting of an array of such pixel processing units. The initial approach relies on a fixed reference level of 2.5 V, onto which the filter's output is superimposed, and is then used as a reference level for the comparator. It was discovered that the system was extremely sensitive to this reference level, making a large array infeasible as each processing unit would have to be individually manipulated. The aim of the second test IC is to test an improved, self-referencing pixel processing unit.

As before, a 0.6  $\mu\text{m}$  AMS process available through Europractice was adopted, coupled with the Cadence design tools. Once again, a third metal layer was employed to provide shielding from unwanted photocurrents.

### 5.1 System Level Approach

The improved approach is similar to the original algorithm, but the high pass filter is replaced by a low pass version. Instead of removing the DC level from the photocircuit's output and biasing the transient information at fixed reference level, the DC level is used as the reference for the comparator. The approach is depicted in figure 5.1. Once again, the filter has to have a low cutoff frequency, to allow the low frequency transients to be separated from the underlying DC level. The new approach effectively creates the comparator's reference point internally, without the need for an external bias. This should allow an array of such pixel processing units to operate independently.





**Figure 5.1:** *Circuit Level Implementation of the Self-Referencing Algorithm: The low pass filter takes the DC level from the photocircuit and presents it as the reference for the comparator, ensuring it is in a 'sensitive' place*

## 5.2 Low Frequency OTA-C Low Pass Filter

The low frequency filter was implemented using the same OTA-C techniques described in chapter four. Once again, the benefits of low frequency, low power and wide tunability when biasing OTA-C filters in the subthreshold regime were exploited. The generic form for a first order gm-C low pass filter can be seen in figure 5.2. As before, the filter has a cutoff frequency of  $f_{3dB} = \frac{g_m}{2\pi C}$ .

### 5.2.1 Operational Transconductance Amplifiers on IC Two

The first test IC proved there was little in the way of performance difference between the simple differential stage and the mirrored OTA alternative. In fact, due to the increased number of current consuming paths, the mirrored version consumed more power. For this reason, a decision was made to concentrate on the implementation of simple differential stage operational transconductance amplifiers for the low pass filter structures.

Three OTA structures were implemented, an NMOS differential stage and PMOS differential stages with and without cascode transistors. The adopted process is an N-well technology, meaning PMOS transistors are fabricated in a separate well from the substrate. It is therefore possible to connect the bulk terminal directly to the source terminal for the PMOS differential OTA input transistors, reducing the bulk effect and the subsequent danger of variable threshold



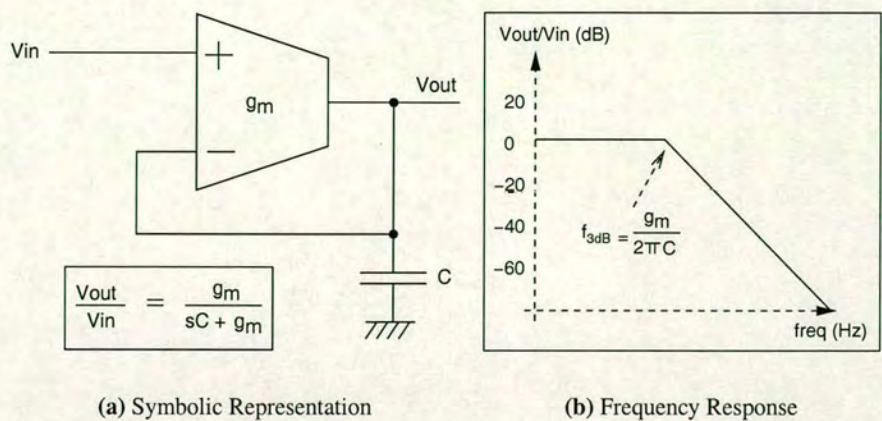


Figure 5.2: 1st Order Gm-C Low Pass Filter

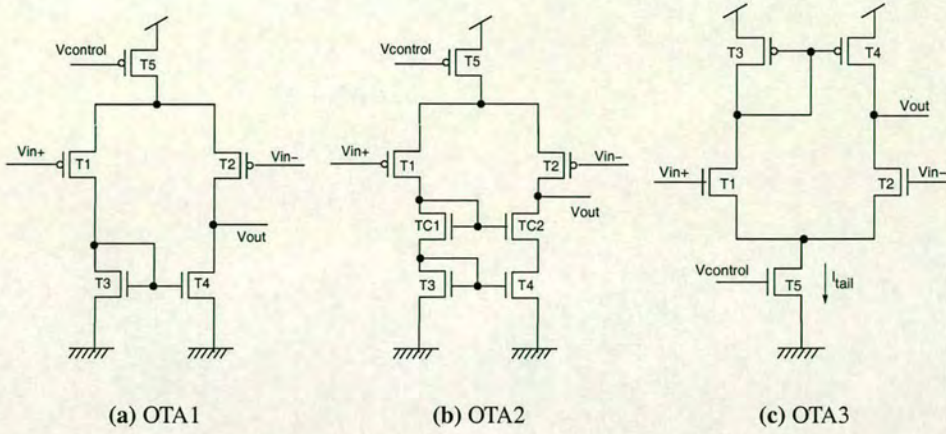
voltages. The NMOS and PMOS differential stage structures also have different *input common mode range* properties, which will have a direct effect on the system-level operation. The aim of cascode transistors is to increase the small signal gain of a differential stage. A detailed analysis of cascode operation is presented in [109] and as such will not be covered here. The cascode transistors serve to increase the output resistance of the differential stage. It was felt that the increased gain may provide some advantage due to the feedback properties of the low pass filter topology.

The three OTA structures included on test IC two can be seen in figure 5.3, with the transistor dimensions in table 5.1.

Transistor	OTA1 $\frac{W}{L}$ ( $\mu\text{m}$ )	OTA2 $\frac{W}{L}$ ( $\mu\text{m}$ )	OTA3 $\frac{W}{L}$ ( $\mu\text{m}$ )
T1	15/15	15/15	15/15
T2	15/15	15/15	15/15
T3	12/12	12/12	12/12
T4	12/12	12/12	12/12
T5	10/10	10/10	10/10
TC1	-	12/12	-
TC2	-	12/12	-

Table 5.1: Transistor Dimensions for the OTA Structures on IC Two





**Figure 5.3:** Operational Transconductance Amplifiers on Test IC Two

### 5.2.2 OTA structures: Input Common Mode Range

The range of common mode inputs over which the transistors in the OTA structures in figure 5.3 remain in the saturation region is termed the *input common mode range*. In this application, the photocircuit's output is applied directly to the OTA-C filter, meaning the ICMR of each OTA structure is important. The ICMR is calculated for the both the highest and lowest DC input for which the transistors remain in the saturation region.

For OTA1, the maximum possible input voltage can be calculated by starting at the positive supply voltage and subtracting the transistor voltage drops until the non-inverting input terminal (gate of transistor T1) is reached. The positive ICMR is therefore calculated as highlighted in equation 5.1:

$$ICMR+ = V_{DD} - V_{SD5(sat)} - V_{SG1} \quad (5.1)$$

The negative CMR can be calculated in the same manner, but beginning at the ground terminal as in equation 5.2:

$$ICMR- = AGnd + V_{GS3} + V_{SD1} - V_{SG1} \quad (5.2)$$

The same process for OTA2 yields the results in equations 5.3 and 5.4.



$$ICMR+ = V_{DD} - V_{SD5(sat)} - V_{SG1} \quad (5.3)$$

$$ICMR- = AGnd + V_{GS3} + V_{GSC1} + V_{SD1} - V_{SG1} \quad (5.4)$$

The results for the third operational transconductance amplifier can be seen in equations 5.5 and 5.6:

$$ICMR+ = V_{DD} - V_{SG3} - V_{DS1} + V_{GS1} \quad (5.5)$$

$$ICMR- = AGnd + V_{DS5(sat)} + V_{GS1} \quad (5.6)$$

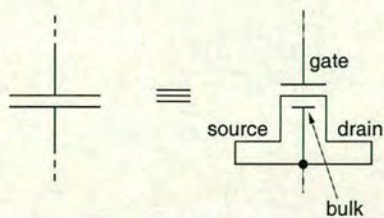
It is clear that the input common mode range for OTA one and three is very similar, with the diode connected transistor T3 limiting the lower and upper ICMR respectively. However, the cascode OTA structure has two diode connected transistors, resulting in two  $V_{GS}$  drops in its negative ICMR. The cost of increased gain is a severely reduced common mode input range, which may limit OTA2's usefulness in this application.

### 5.2.3 Low Pass Filter's Implemented on IC Two

The three OTA structures described previously were used in the implementation of first order OTA-C low pass filters, based on the topology in figure 5.2. The capacitor was realised with an MOS transistor, connected as shown in figure 5.4. The idea was to create a larger capacitor for the same silicon area than would be possible with a poly-poly capacitive structure. By shorting together the drain, source and bulk terminals, a relatively large capacitance is achieved for a small silicon area. A  $30 \mu m^2$  MOS structure was used to realise the 1 pF capacitance required by the low pass filter.

As with the first test IC, layout techniques were adopted to improve the matching properties of the filters. Common-centroid, inter-digitated small transistors were combined to realise large transistors, while dummy transistors and guard rings were also included. The physical layout of the three filter structures can be seen in figure 5.5. Low pass filter one is approximately  $160 \mu m$  by  $95 \mu m$ , LPF2 consumes  $170 \mu m$  by  $100 \mu m$ , while LPF3 uses  $155 \mu m$  by  $100 \mu m$ .





**Figure 5.4:** Capacitor created with MOS Transistor: By shorting the bulk, drain and source terminals, the parasitic capacitances combine to produce a larger capacitor

**5.2.4 Low Pass Filter: Power Consumption**

As with the previous test IC, the low pass filter structures will be biased with subthreshold currents to enable the very low cutoff frequencies required. The power consumption of each filter was estimated with the Spectre simulation tool. The estimated current consumption was calculated for three different values of control voltage which might be typically used in the fundamental frequency extraction system. The results can be seen in tables 5.2 and 5.3. As expected, both PMOS filters (LPF1 and LPF2) consume the same current, despite the presence of cascode transistors. The NMOS filter is biased further into the subthreshold regime, hence the reduced current consumption. All three filters consume power in the nano-Watt range, confirming the low power nature of the circuitry.

Filter	control = 4.3V	control = 4.35V	control = 4.4V
LPF1	3.166 nA	850 pA	190 pA
LPF2	3.15 nA	850 pA	190 pA

**Table 5.2:** Simulated Current Consumption for the PMOS OTA-C Low Pass Filter Structures

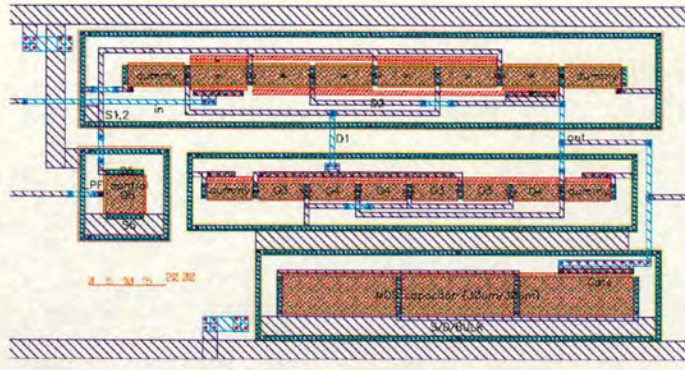
Filter	control = 0.55V	control = 0.5V	control = 0.45V
LPF3	80 pA	30 pA	15 pA

**Table 5.3:** Simulated Current Consumption for the NMOS OTA-C Low Pass Filter Structures

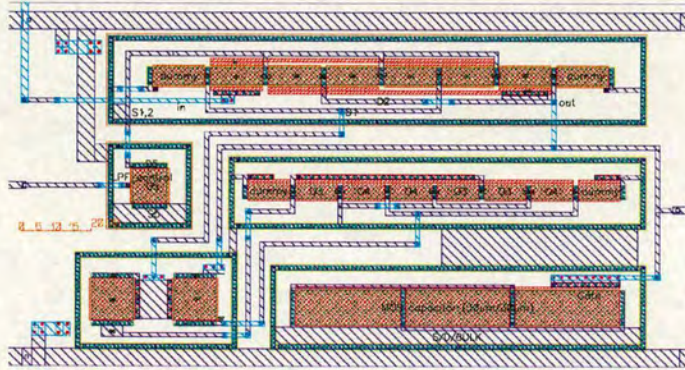
**5.2.5 IC Test Results: OTA-C First Order Low Pass Filter**

The three OTA-C low pass filter structures on IC two were tested for frequency response, input common mode range and DC offset. The same buffer circuit employed to measure the high pass filters was employed to prevent the high capacitance of the pad-ring from effecting the

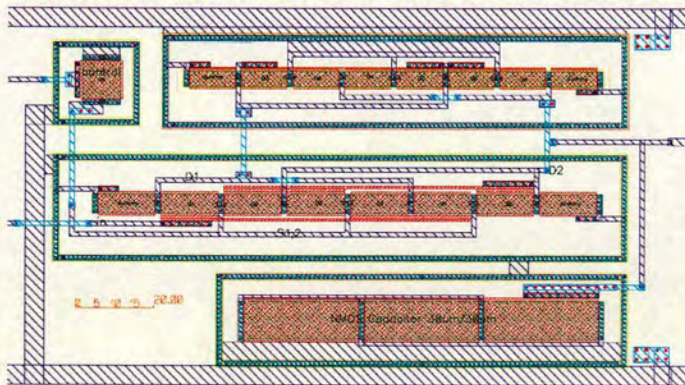




(a) LPF1 Layout



(b) LPF2 Layout



(c) LPF3 Layout

**Figure 5.5:** Physical Layout of the Three OTA-C Low Pass Filter Structures

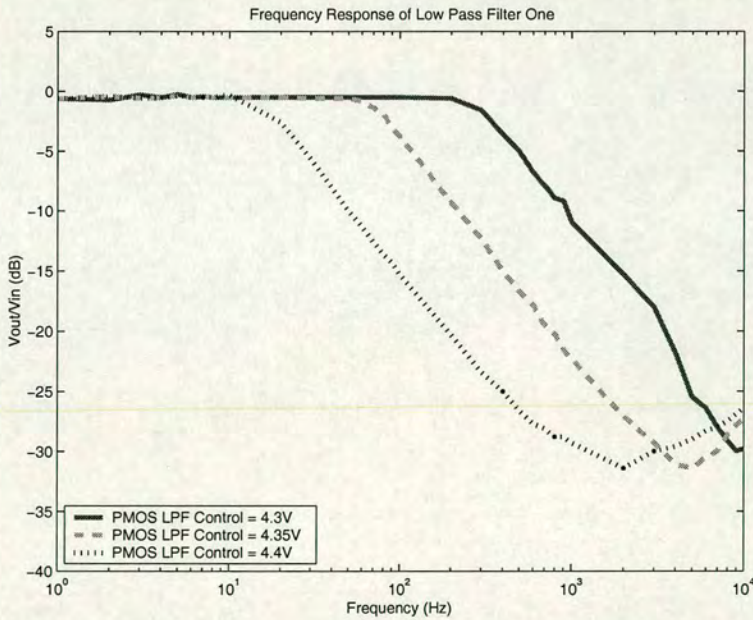


circuitry.

### 5.2.5.1 Frequency Response of the OTA-C Low Pass Filters

The frequency response of the low pass filter structures was measured as before, by increasing the input frequency with a signal generator and measuring the ratio between output and input voltages.

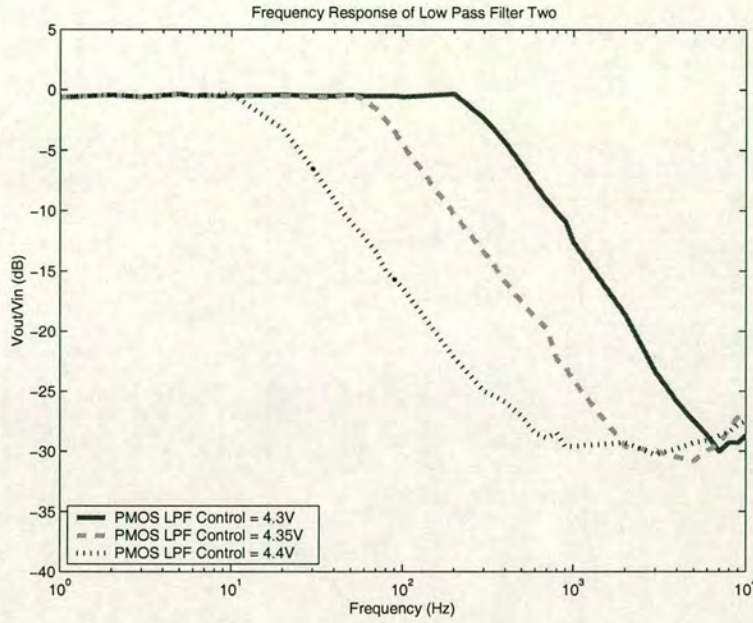
The frequency response of LPF1 is highlighted in figure 5.6. With a control voltage of 4.3 V, the cutoff frequency is approximately 300 Hz, reducing to below 10 Hz for 4.4 V. At high frequencies, the high attenuation makes discriminating between signal and noise extremely difficult, hence the inaccuracy of the results at frequencies above 1 kHz. Despite this, it is clear that the filter structure allows very low, tunable cutoff frequencies.



**Figure 5.6:** Measured Test IC Results: Frequency Response of OTA-C Low Pass Filter One

A similar test for the filter employing an OTA structure with cascode transistors was performed, with the results depicted in figure 5.7. The enhanced output resistance of the cascode configuration has little effect on the frequency response, with the results appearing very similar to LPF1. The task of the OTA in the filter is simply to convert the input voltage into a proportional current due to its transconductance. As such, the output resistance of the OTA has very little effect on the filter's frequency response.





**Figure 5.7:** Measured Test IC Results: Frequency Response of OTA-C Low Pass Filter Two

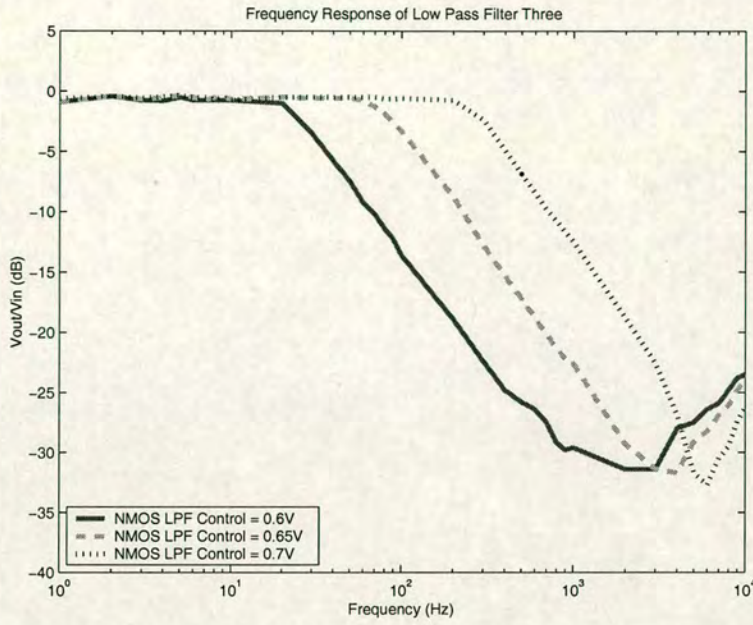
The filter employing the NMOS differential pair OTA was also tested for its frequency response in the same manner. The bode plots in figure 5.8 show the familiar low pass response, with the cutoff varying from about 200 Hz for 0.7 V control voltage to 20 Hz for 0.6 V.

### 5.2.5.2 Input Common Mode Range of the OTA-C Low Pass Filters

The range of input DC levels over which the low pass filter will operate is important in this application. The photocircuit's output is applied directly to the filter, and as its DC level varies with the background light intensity, it is important that the filter can operate over a range of input levels. The ICMR calculations in equations 5.1 to 5.6 suggest that the cascode transistors in LPF2 will severely limit its input common mode range. The ICMR was tested by applying a DC input to the filter, increasing from 0 V to 5 V. The output voltage was measured as the input increased.

The input common mode range for the three low pass filters can be seen in figure 5.9. The control voltage for filters one and two was fixed at 4.35 V, while that for the NMOS differential OTA in LPF3 was set at 0.6 V, ensuring all three are biased in the subthreshold region of operation. The results for LPF1 can be seen in figure 5.9(a) and show that the filter operates as expected from about 1 V to 4.5 V. The same experiment for LPF2 shows a similar range,





**Figure 5.8:** Measured Test IC Results: Frequency Response of OTA-C Low Pass Filter Three

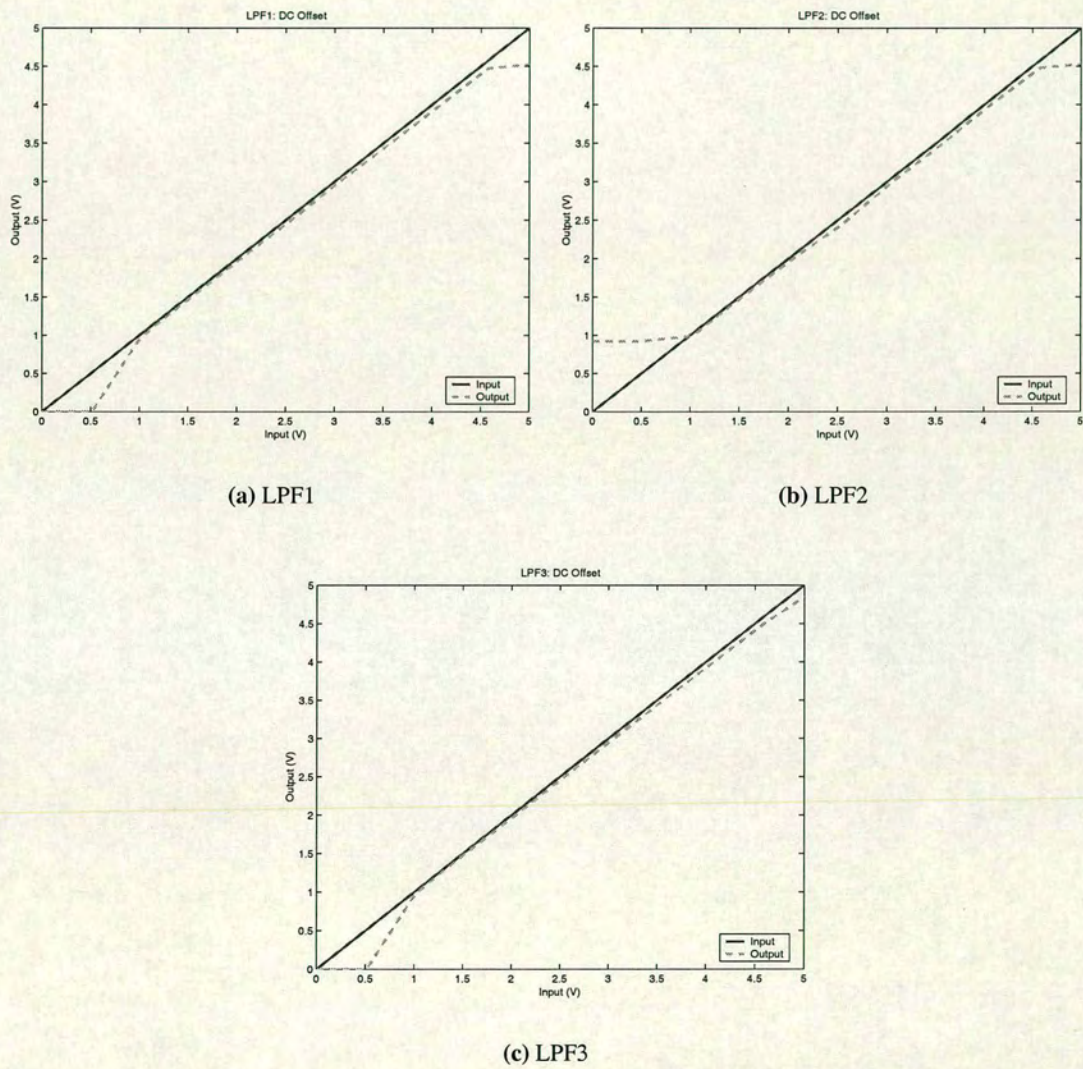
contrary to the earlier calculated expression. The third low pass filter appears to have the best input common mode range, operating from approximately 1 V up to nearly 5 V.

In the strong inversion region of operation, a  $V_{GS}$  term incorporates a threshold voltage, producing a large voltage drop. However, the expression for  $V_{GS}$  for a transistor biased in the subthreshold region does not include such a threshold voltage term, meaning the corresponding voltage drop is smaller. It is clear from the results in figure 5.9 that biasing the OTA structures in the weak inversion regime serves to maximise the potential input common mode range, yet another advantage of this approach. This may explain why the ICMR for the cascode OTA structures on LPF2 is not as bad as expected. All three filters allow operation over an acceptably wide range of input DC values, with the NMOS differential pair OTA employed in LPF3 producing the widest linear range.

### 5.2.5.3 DC Offset of the OTA-C Low Pass Filters

The aim of the low pass filter in the self-referencing fundamental frequency extraction algorithm is to separate the DC level from the transient AC signals of interest. The filter's output is then used as the reference for the thresholding stage, ensuring the comparator will switch. As such, the offset through the filter is of interest. If the offset is large, the filter's output may





**Figure 5.9:** Measured Test IC Results: Input Common Mode Range of OTA-C Low Pass Filter Three



not *cut* through the photocircuit's output, meaning the comparator will be unable to switch.

The results in table 5.4 highlight the DC offset between the input and output signals, in the range from 0 V to 5 V. All three filters exhibit similar levels of offset, which appears to increase as the input DC level increases. Of the three filters, LPF3 with the NMOS differential pair OTA appears to produce marginally the lowest offset, which becomes more pronounced at higher input voltages.

Input	LPF1	LPF2	LPF3
0 V	-	-	-
0.5 V	-	-	-
1.0 V	42.5 mV	30.7 mV	30.5 mV
1.5 V	32.6 mV	28 mV	23.5 mV
2.0 V	43.5 mV	43 mV	38.3 mV
2.5 V	55.5 mV	99.6 mV	50.2 mV
3.0 V	51.7 mV	57.1 mV	54.4 mV
3.5 V	73.9 mV	70.7 mV	66.1 mV
4.0 V	88.8 mV	66.2 mV	80.7 mV
4.5 V	104 mV	100 mV	92.4 mV
5.0 V	547 mV	543 mV	163 mV

**Table 5.4:** *Measured Test IC Results: DC Offset for OTA-C Low Pass Filters*

### 5.2.6 Comments on the OTA-C First Order Low Pass Filter

The benefits of employing OTA-C filter structures biased in the subthreshold region of operation have been explored previously, with low cutoff frequencies consuming nano-watts of power. The three low pass filter structures implemented on test IC two all perform as expected, with cutoff frequencies lower than 10 Hz easily achievable. In addition to low power processing, biasing the filters in the subthreshold regime would appear to maximise the input common mode range, based on the measured test results in figure 5.9. This is an advantage in this application, given the variable DC level of the photocircuit's output voltage.

The attenuation that was present in the pass band of the high pass filters implemented on test IC one is not present in the corresponding pass band of the low pass filters. This is an obvious advantage, as the signal swing will be maximised to allow easier thresholding by the comparator.

The cascode transistors in LPF2 seem to have little effect on performance, when compared with



the standard version in LPF1. As previously mentioned, the cascode transistors serve to increase the output impedance of the OTA, which is of little benefit in this application. The NMOS differential pair OTA in LPF3 seems to give the best results in terms of both input common mode range and DC offset, suggesting it might produce the best results when incorporated in the final algorithm.

### **5.3 System Level Test IC Results: Self-Referencing Pixel Processing Units**

The algorithm implemented on test IC two is highlighted in figure 5.1. The photocircuit creates a voltage signal that is dependent on the incident light. The low pass filter then separates the photocircuit's DC level from the transient signal, with the former supplied to the comparator as a reference voltage. The photocircuit's output is then applied directly to the comparator's input terminal. As with the first test IC, the aim is a pulse train whose frequency directly corresponds to the fundamental frequency of the incident light intensity. The difference stems from the internal creation of the comparator's reference voltage, as opposed to the use of an externally created reference signal.

The photocircuit and comparators with and without hysteresis from the initial test IC were once again adopted in IC two. The employed photoelement was a  $50 \mu\text{m}^2$  phototransistor, due to its high photocurrent capability. The load transistor dimensions of the photocircuit were reduced to  $5 \mu\text{m}$  by  $5 \mu\text{m}$ , in an effort to reduce the area of implementation. A parameter of interest in the system is the input common mode range of the comparator, as it will be expected to operate over a range of inputs. The comparator is essentially an NMOS differential stage, so the expressions calculated for OTA3 in equations 5.5 and 5.6 will be valid. The comparator will also be biased with moderate inversion bias currents, meaning the  $V_{GS}$  drops will not include a threshold voltage. As such, the comparator's ICMR should be adequate for the chosen application.

In total, six different pixel processing units were included on the second test IC, to allow separate testing of the different processing elements. The pixel cell's contents are summarised in table 5.5. There are two pixel cells for each of the low pass filter structures, using comparators with and without hysteresis. The layout of one of the pixels can be seen in figure 5.10. Little effort was made to minimise the size of the pixel cell, with the algorithm consuming approximately  $430 \mu\text{m}$  by  $100 \mu\text{m}$ .



Pixel Cell	Photoelement	Photocircuit	LPF	Comparator
Three	Phototransistor	Logarithmic	LPF1	Standard
Four	Phototransistor	Logarithmic	LPF1	Hysteresis
Five	Phototransistor	Logarithmic	LPF2	Standard
Six	Phototransistor	Logarithmic	LPF2	Hysteresis
Seven	Phototransistor	Logarithmic	LPF3	Standard
Eight	Phototransistor	Logarithmic	LPF3	Hysteresis

Table 5.5: Pixel Processing Cells On Test IC One

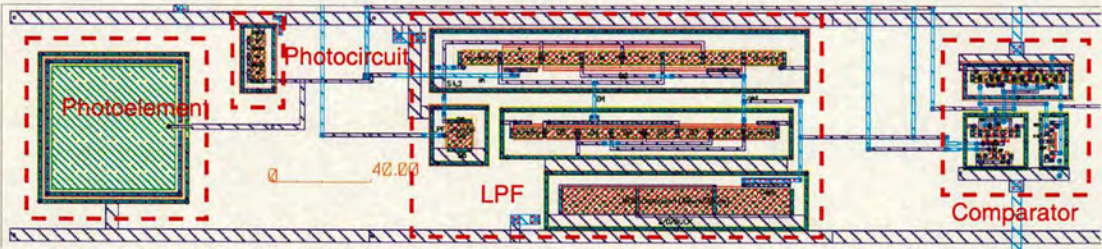


Figure 5.10: Physical Layout of Pixel Processing Cell Three

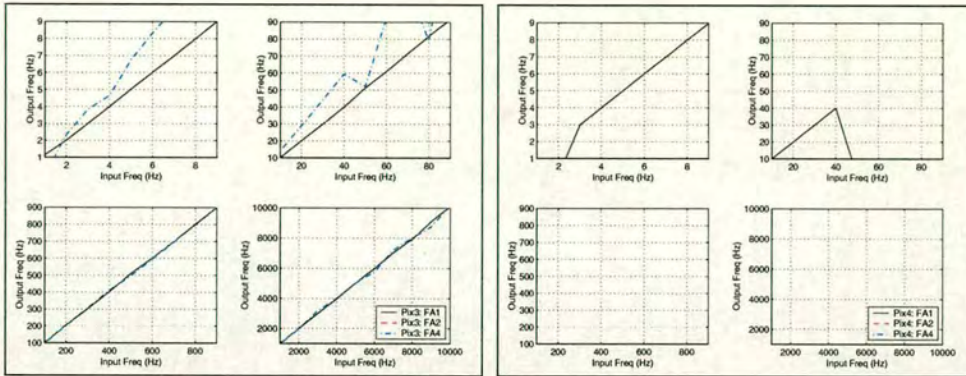
As before, the best test for measuring the performance of the system is the accuracy of the output pulse train’s frequency. To this end, a series of four different **frequency accuracy tests** were performed.

5.3.1 Frequency Accuracy Tests

The frequency accuracy tests were conceived to allow direct comparison of the different processing elements that combine to make fundamental frequency extraction pixel processing units. Parameters that were examined include response to different illumination levels, the effect of including hysteresis in the comparator, varying the control voltage of the filter structures and the potential benefits of including cascode transistors in the low pass filter. The different experiments are summarised in table 5.6. For all experiments, the comparator’s control voltage was fixed at 1 V. The frequency accuracy of each pixel’s output pulse train was once again measured with an Agilent A56424 oscilloscope. As with the first test IC, the intensity of an LED was modulated by a signal generator, to allow complete control over the incident illumination.

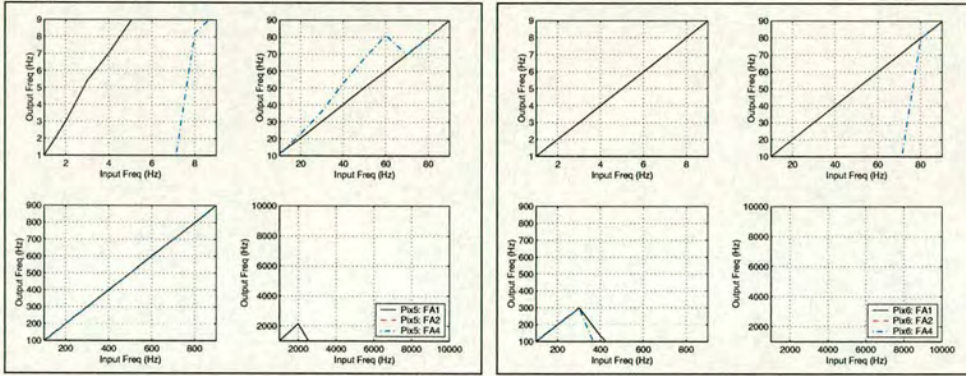
The frequency accuracy results for all pixel processing units can be seen in figure 5.11. Pixels





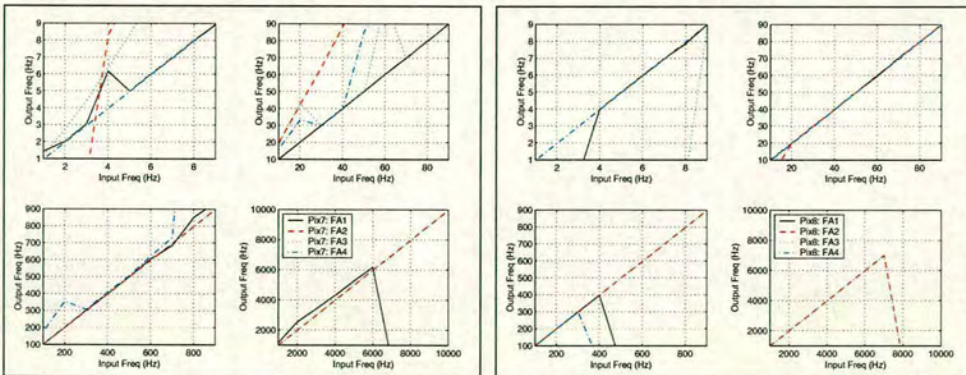
(a) Pixel Three

(b) Pixel Four



(c) Pixel Five

(d) Pixel Six



(e) Pixel Seven

(f) Pixel Eight

Figure 5.11: Measured Test IC Results: Frequency Accuracy of the Pixel Processing Elements



Test	LED: DC	LED: AC (sine)	NMOS LPF Control	PMOS LPF Control
FA1	1.2 V	100 mV	0.55 V	4.4 V
FA2	2 V	500 mV	0.55 V	4.4 V
FA3	2 V	500 mV	0.45 V	4.4 V
FA4	1.2 V	100 mV	0.45 V	4.35 V

**Table 5.6:** *Frequency Accuracy Testing: Parameter Values*

three to six produced no output for frequency accuracy test two, so FA3 was not applied to these pixels.

### 5.3.2 Comparison of Pixels with and without Hysteresis in the Comparator

The processing elements that combine to produce pixel four are similar to those for pixel three, except the comparator has a small amount of hysteresis to provide protection against noisy input signals. The same relationship exists between pixels six and five, and pixels eight and seven, with the former in each case using hysteresis and the latter using a standard comparator circuit. This allows a comparison of the potential advantages and disadvantages of hysteresis regarding the self-referencing fundamental frequency extraction algorithm.

#### 5.3.2.1 Pixel Three vs Pixel Four: Hysteresis Comparison

Pixel three operates well for the entire frequency range of 1 Hz to 10 kHz when tested with FA1. However, pixel four under the same test setup only begins to operate at 3 Hz, before failing to produce an output pulse train at frequencies over 40 Hz, as seen in figure 5.11(b). This would suggest that at very low frequency, the difference between the photocircuit's output and that from the low pass filter is too small for the comparator with hysteresis to discern. As the frequency increases, the size of the photocircuit's output reduces, and the comparator with hysteresis in pixel four is unable to threshold the signal.

For frequency accuracy test four (FA4), the low pass filter's control voltage has been reduced to 4.35 V, meaning the low pass filter's cutoff frequency has increased. The effect of this can be seen in figure 5.11(a), with the low frequency performance appearing less accurate than that for FA1. The reason for this is the lack of a large enough difference between the photocircuit and low pass filter's outputs, meaning the comparator is unable to threshold the signal. Pixel four produces no output for FA4, suggesting the signals are too small for the comparator with



hysteresis.

### **5.3.2.2 Pixel Five vs Pixel Six: Hysteresis Comparison**

The results for pixel five when tested with FA1 are highlighted in figure 5.11(c). The output pulse train is highly inaccurate at low frequency, but works well from 10 Hz up to almost 2 kHz. When compared with the results from the same test for pixel six in figure 5.11(d), it appears the inclusion of hysteresis improves the low frequency accuracy but at the cost of a reduction in range to 300 Hz. This is exactly what would be expected from a comparator with hysteresis, as the safety margin increases accuracy at the cost of reduced sensitivity.

The same pixels were tested with frequency accuracy test four (FA4). In this case, pixel five's low frequency operation is even less precise, as the filter's cutoff has increased in frequency. However, it does operate well from 70 Hz to about 1 kHz. As expected, the same test for pixel six produces a reduced operating range of approximately 80 Hz to 300 Hz. The presence of hysteresis once again restricts the operating range of the device.

### **5.3.2.3 Pixel Seven vs Pixel Eight: Hysteresis Comparison**

Pixels seven and eight use the NMOS OTA as the basis of the low pass filter structure. All four frequency accuracy tests were applied to these pixels.

For FA1 in figure 5.11(e), pixel seven is relatively accurate from 1 Hz to 6 kHz, with a small glitch at 6 Hz. The same test applied to pixel eight produces a more accurate output pulse train, but over a restricted range of 4 Hz to 400 Hz, as highlighted in figure 5.11(f).

The results for the second frequency accuracy test (FA2) when applied to pixel seven produce a very poor low frequency performance, with the pixel only beginning to produce an accurate output pulse frequency at about 100 Hz, continuing up to 10 kHz. When applied to pixel eight, FA2 produces a much more accurate pulse train over the range 20 Hz to 7 kHz. At very low frequency, pixel eight fails to produce an output pulse train.

Both pixels produce similar results to those for FA2 when tested with FA3. Pixel seven exhibits spurious output pulse train frequencies until approximately 70 Hz, when it becomes more accurate. Pixel eight works well from 10 Hz up to 7 kHz. The effect of reducing the cutoff frequency of the filter is to improve the low frequency performance very slightly.



Finally, frequency accuracy test four (FA4) was applied to both pixels seven and eight. Pixel seven operates well at low frequency under this test condition, but begins to produce occasional spurious output frequencies above 10 Hz. Pixel eight produces highly accurate output pulses from 1 Hz up to 300 Hz, above which it is unable to differentiate between its two inputs.

#### **5.3.2.4 Comments on the use of Hysteresis in the Comparator**

In general, hysteresis appears to increase the accuracy of the output pulse train, while reducing the sensitivity of the pixel processing units. The aim of hysteresis is to provide some degree of robustness to noise, by only switching when the input exceeds the threshold plus or minus the added hysteresis. The improved accuracy therefore makes sense, as spurious pulse are less likely to be created. The reduction in range is also expected as the large attenuation at higher frequencies means the signals may be too small for the comparator to switch.

### **5.3.3 Comparison of Different Illumination Levels**

Two different LED control voltages were used to test the response of the pixel processing units to a variation in illumination intensity. For FA1 and FA4, the LED was controlled with a DC level of 1.2 V and a transient signal of 100 mV peak to peak, while FA2 and FA3 use a DC level of 2 V and a larger AC signal swing of 500 mV peak to peak. Based on this, it is possible to directly compare the response of all six pixels to FA1 and FA2, as well pixels seven and eight to FA3 and FA4.

#### **5.3.3.1 Frequency Accuracy Test One vs Frequency Accuracy Test Two**

All six pixels were tested for their operation under FA1 when compared with the increased illumination levels of FA2. It is clear from figure 5.11(a) that while the response of pixel three to FA1 is very accurate, there is no output for FA2. A similar trend is highlighted for pixel four in figure 5.11(b), with FA1 producing a limited but highly accurate range of output frequencies, while FA2 produces nothing. The same is true for pixels five and six in figures 5.11(c) and (d), with FA1 producing a limited output range for each. However, when applied to pixels seven and eight in figures 5.11(e) and (f), both FA1 and FA2 produce outputs. For pixel seven, FA1 produces relatively accurate results over the entire range, while FA2 gives spurious output frequencies until approximately 100 Hz, when it begins to accurately represent



the input frequency. FA2 does however operate to a higher input frequency than FA1, which stops producing pulses at approximately 6 kHz. This trend is repeated for pixel eight, which works well at low frequency for FA1 before FA2 begins to produce better high frequency results.

### **5.3.3.2 Frequency Accuracy Test Three vs Frequency Accuracy Test Four**

It is also possible to directly compare the response of pixels seven and eight to differing illumination levels when tested with FA3 and FA4. As is the case for the comparison of FA1 and FA2, the lower intensity FA4 seems to work best at low input frequencies, while the higher illumination levels of FA3 produces superior high frequency results, as highlighted in figure 5.11(e). The same is true for pixel eight, as depicted in figure 5.11(f).

### **5.3.3.3 Comments on Different Illumination Levels**

It appears that for pixels seven and eight, low intensity illumination levels operate best at low frequency, while higher intensity produces better results at higher input frequencies. Pixels three to six produce no output for the higher illumination level. The difference between pixels three to six and seven to eight is the use of PMOS and NMOS OTA structures in the low pass filters respectively, suggesting this may be a factor in the differing performance. However, when tested individually, the three low pass filter structures produced similar results, with the NMOS OTA in LPF3 having slightly superior ICMR and DC offset performance. It may be that the increased illumination produces smaller signal swings from the photocircuit, as highlighted in figure 4.8, which in turn prevents the pixels using LPF1 and LPF2 from operating successfully. There may also be discrepancies between the test setups used, with the filters for pixels seven and eight biased further into the subthreshold regime than their counterparts in pixels three to six. However, the exact reasons for the failure of pixels three to six when tested with FA2 remain unclear.

### **5.3.4 Comparison of Pixels with Varying Filter Control Values**

Another variable that may have an effect on the frequency accuracy of the self-referencing pixel processing unit is the cutoff frequency of the low pass filter. Pixels three to six use PMOS differential pair OTAs as the basis of the filter, and therefore use a different control voltage to the NMOS differential pair OTAs of pixels seven and eight. For this reason, it is possible to



directly compare frequency accuracy tests one and four for all six pixels, while pixels seven and eight can also make a comparison between FA2 and FA3.

#### **5.3.4.1 Effect of Filter Control Voltage: Pixels Three to Six**

The sole difference between FA1 and FA4 is the reduction in PMOS filter control voltage from 4.4 V to 4.35 V. This will have the effect of increasing the bias current for the filter structures under frequency accuracy test four, therefore increasing the cutoff frequency, as highlighted in figures 5.6 and 5.7. Comparing the results for pixel three when tested with FA1 and FA4, it is clear from figure 5.11(a) that increasing the filter's cutoff frequency reduces the accuracy at low frequencies. This makes intuitive sense, as the system works by using a comparator to sense the difference between the output from the photocircuit and the low pass filter. From the measured test results for LPF1 in figure 5.6, a control voltage of 4.4 V gives a cutoff frequency of approximately 10 Hz, while 4.35 increases this to nearer 100 Hz. At the low frequency range of 1 Hz to 10 Hz for pixel three in figure 5.11(a), there is simply no discernible difference between the comparators' inputs when tested with FA4. However, reducing the filter's cutoff frequency allows the comparator to 'see' the difference between the two.

A similar argument seems to exist for pixel four, with FA1 producing a limited range of outputs while no output is produced for FA4, as seen in figure 5.11(b). The hysteresis in pixel four stops the comparator from thresholding the small signal variations between the photocircuit's and low pass filter's outputs.

Pixel five's operation in the 1 Hz to 10 Hz range is inaccurate for both FA1 and FA4. However, as highlighted in figure 5.11(c), the accuracy under FA1 is superior to that for FA4 in the 10 Hz to 100 Hz range, continuing the trend from pixels three and four. The same is true of pixel six in figure figure 5.11(d), whose low frequency performance under FA1 are superior to those for FA4.

#### **5.3.4.2 Effect of Filter Control Voltage: Pixels Seven and Eight**

Pixels seven and eight use LPF3, which is based on the NMOS differential pair OTA structure. It is possible to measure the effect of NMOS filter control on the frequency accuracy of these pixels by comparing FA2 and FA3, as well as FA1 and FA4.



The results for pixel seven are depicted in figure 5.11(e). It is clear that the accuracy for FA4 in the range of 1Hz to 10 Hz is superior to those for FA1 in the same region. FA4 uses a control voltage of 0.45 V, which means a lower cutoff frequency for the filter than the 0.55V of FA1. This follows the general results seen for the PMOS filters of pixels three to six, where reducing the cutoff frequency improves the low frequency results. A similar relationship is seen between FA3 and FA2 for pixel seven, although it is clear that the pixel processing unit is generally not as accurate as is the case for the lower intensity levels of FA1 and FA4.

Pixel eight also sees improved low frequency performance for FA4 when compared with FA1, as highlighted in figure 5.11(f). The difference between FA2 and FA3 is negligible, which is due to the presence of hysteresis in the comparator

#### 5.3.4.3 Comments on Varying the Filter Control Voltages

The comparison of varying the control voltage of the low pass filters suggests that the lower the cutoff frequency, the better the low frequency performance. The aim of the low pass filter is to extract the DC level from the photocircuit's output and present it as the reference point for the comparator. However, due to the extremely low frequency nature of the illumination variation, the low pass filter's output is not a stable DC level, but a slightly attenuated and phase shifted version of the photocircuit's output. It follows that if the cutoff frequency is too high, there is no difference between the photocircuit's and LPF's outputs, as the filters output is in the pass-band. In most cases, reducing the cutoff frequency improves the low frequency performance as it makes sure there is a *difference* between the comparator's input signals. Therefore, in general, reducing the cutoff frequency of the filter improves the low frequency performance of the system, while having little effect on the higher frequency operation.

#### 5.3.5 Comparison of Pixels with and without Cascode OTAs

As the IC test measurements for low pass filter's one and two highlighted, there appears to be very little difference between a PMOS differential pair with or without additional cascode transistors. The frequency response in the bode plots of figures 5.6 and 5.7 are very similar, as are the DC offsets in table 5.4 and the input common mode range in figure 5.9. However, from a system perspective, it seems worthwhile to observe if there is any difference between the pixels that include cascode transistors and those that do not.



Pixels five and six are similar to three and four respectively, except for the inclusion of cascode transistors in the OTA filter structure. It is therefore possible to compare pixel three(four) with pixel five(six) for both FA1 and FA4. There was no output from these pixels for frequency accuracy test two and three, hence the lack of meaningful comparison between them.

It would appear that the accuracy of pixel three for both FA1 and FA4 is superior to that of pixel five as highlighted in figures 5.11(a) and (c). Pixel three is able to operate up to the maximum input frequency of 10 kHz, for both FA1 and FA4, while pixel five stops at approximately 2 kHz. The low frequency performance of pixel three is also superior. However, the results for pixel six appear superior to those for pixel four. FA4 produces no output for pixel four, while there is a limited output for the same test setup for pixel six. The results for pixel six when tested with frequency accuracy test one are also superior to those for pixel four.

#### **5.3.5.1 Comments on the use of Cascode OTAs**

The discrepancy between the performance of pixel three and pixel five to pixel four and pixel six may be due to some inconsistency in the value of hysteresis in the comparator, or some other test parameter. However, it is hard to draw a conclusion on the use of cascode OTA structures based on these test results. Although the filters when tested individually show little in the way of a performance difference, the results when included in the pixel processing units seem to conflict with each other. There is no difference in power consumption between the two filter structures, but the use of cascode transistors does increase the area of implementation. In this application, the silicon area consumed by each pixel processing unit is important if a useful array is to be realised. For this reason, cascode transistors will not be included in future filter implementations.

#### **5.3.6 Comments on Self-Referencing Pixel Processing Units**

In general, the self-referencing pixel processing units work well. At first glance, system level results in figure 5.11 appear less impressive than those for the pixel processors implemented on test IC one. However, the great advantage of the approach on IC two is the self referencing system, which both removes the need for an external reference voltage and allows for much more realistic integration into a CMOS imager-processor array. Pixels seven and eight appear to operate over a wider range of test conditions than the others, producing outputs despite



varying illumination levels and filter control values. Adding hysteresis to the pixels improves the accuracy but reduces the sensitivity, while filter's should be biased deep into subthreshold, with lower cutoff frequencies giving the best performance.

## **5.4 Comments on Test IC Two**

Test IC two built on the experience gained from the first test integrated circuit, but introduced a small yet fundamental difference to the frequency extraction algorithm. Instead of trying to accurately superimpose the transient information from the photocircuit's output onto a pre-defined reference voltage, the actual DC level of the photocircuit's output voltage is extracted and used as the comparator's reference. This creates a **self-referencing** algorithm, that greatly increases the feasibility of an array of such pixel processing units being realised. The problems of sensitivity regarding the first approaches dependency on this external reference voltage are removed, allowing each pixel to operate without external reference manipulation.

### **5.4.1 Conclusions on OTA-C Low Pass Filters**

The first test IC highlighted the advantages of creating low cutoff frequency filters with OTA-C structures biased in the subthreshold region of operation. As well as realising large time constants with reasonably small silicon areas, the power consumption of such filters is in the nW region, agreeing with the requirements of the sponsor company. Another advantage is the wide tunability range of such filters when biased with weak inversion currents.

The low pass filters implemented on the second test IC performed well. The frequency response Bode plots in figures 5.6, 5.7 and 5.8 highlight the low cutoff frequencies that are easily achievable with the approach. In total, three different low pass filters were implemented, two with PMOS differential pair OTA structures (with and without cascode transistors) and one with an NMOS differential pair. The area of each implementation is similar, with LPF1 consuming  $160\text{ }\mu\text{m}$  by  $95\text{ }\mu\text{m}$ , LPF2 using  $170\text{ }\mu\text{m}$  by  $100\text{ }\mu\text{m}$ , while LPF3 is approximately  $155\text{ }\mu\text{m}$  by  $100\text{ }\mu\text{m}$ . Little effort was made to minimise the area of implementation as the physical layout in figure 5.5 highlights, with guard structures, common-centroid layout and dummy transistors incorporated to improve matching performance. The power consumption of all three filters is in the nW region, as the simulation estimates summarised in tables 5.2 and 5.3 confirm. Although likely to be inaccurate, it is clear that the filters consume very small bias currents, which



manifest as low power consumption.

The three filters were tested for input common mode range, as highlighted in figure 5.9. The mathematically derived expressions in equations 5.1 to 5.6 predict poor performance for LPF2, due to the cascode transistors. However, biasing the OTA structures with subthreshold currents appears to maximise the ICMR, as the  $V_{GS}$  drops do not include a threshold voltage. Of the three filters, the NMOS differential pair OTA in LPF3 seems to produce the widest input common mode range.

The DC offset between input and output was also measured for all three filters, with the results included in table 5.4. Although very similar, the results for the third filter appear slightly superior to those for the two PMOS differential pair devices.

All three filters perform as expected, with similar results in general. LPF3 is probably the best of the three, due to the superior ICMR and DC offset performance. However, any of the three could be included in a pixel processing unit incorporating the self-referencing fundamental frequency extraction algorithm.

#### **5.4.2 Conclusions on Self-Referencing Pixel Processing Cells**

The low pass filters described earlier were combined with phototransistors, logarithmic compression photocircuits and positive feedback comparators developed on the first test IC, to create self-referencing pixel processing units. The aim of these units is as before, to create an output pulse train whose frequency directly encodes the fundamental frequency of the variation in incident illumination. In total, six different pixel processing units were created, two with each low pass filter, to allow a comparison of the effects of including comparators both with and without hysteresis. The contents of the pixels can be seen in table 5.5.

The best measure of the success of the self-referencing pixels is the accuracy of the frequency of the output pulse trains. For this reason, a series of four different test procedures were performed, as summarised in table 5.6. The frequency accuracy tests allow direct comparisons of the effects of including hysteresis in the comparator, changing the illumination level, varying the filter control voltage and the use of cascode transistors in the OTA circuit.

It is clear from the frequency accuracy results in figure 5.11, that in general all six pixels operated in certain test conditions. Regarding the use of hysteresis in the comparator, it would



seem that hysteresis can improve the accuracy of the output pulse train, but at the cost of a reduced operating range. This is highlighted in the comparison of pixels seven and eight for frequency accuracy test 4 in figures 5.11(e) and (f), as well as that between pixel five and pixel six for FA1, as highlighted in figures 5.11(c) and (d). This property of hysteresis makes intuitive sense, and agrees with similar tests applied to the pixels on test IC one. It appears that hysteresis may be more important in this algorithm than the original implementation on test IC one. In the original technique, the high pass filter's output is compared with an external reference, which should be relatively clean. However, the self-referencing technique compares the photocircuit's output with that of the low pass filter, both of which may exhibit noise, suggesting hysteresis may offer advantages in this technique. It is clear from pixels four, six and eight compared to three, five and seven in figure 5.11, that there are no spurious output pulses from the pixels which incorporate hysteresis. The frequency accuracy tests prove the benefits of making the cutoff frequencies as low as possible, by biasing the OTA structures deep into the subthreshold regime. This has further advantages regarding the power consumption of the system. As previously mentioned, the performance benefits of including cascode transistors do not appear to merit the increased area of implementation. The two different illumination levels applied to the pixels suggests that the pixels incorporating NMOS differential pair OTA-C filters are superior to their PMOS counterparts.

As with the first pixel processing unit algorithm, the self referencing system benefits from low power operation, with the low pass filters consuming subthreshold bias currents. All six pixels perform relatively well, and importantly, do not require an external reference signal as was the case with the first approach. This, coupled with the positive test results led to this approach being adopted and improved for a third test IC. The aim of this new chip is to adapt the system to create a tunable band pass filter, whose centre frequency tunes automatically to the fundamental frequency of the incident illumination. Also, included on the third test IC is a miniaturised version of the self-referencing pixel processing unit, which combines, photoelement, photocircuit, low pass filter and comparator into an area of only  $60 \mu\text{m}^2$ .



---

## Chapter 6

# Test IC Three: *Minipix*

## Self-Referencing Pixel Processing Unit

---

As mentioned in the conclusions of chapter five, the self-referencing pixel processing unit was well suited to inclusion in a image processor pixel array. However, the size of the pixel processing units, at approximately  $430\text{ }\mu\text{m}$  by  $100\text{ }\mu\text{m}$ , was such that the spatial resolution of any resultant image processor would be too low to be of any practical use. An effort to reduce the size of the self-referencing pixel processing unit was made on test IC three, with the development of the **minipix** algorithm. The circuitry is the same as was implemented on the second test IC, but the area has been reduced considerably to  $60\text{ }\mu\text{m}^2$ , potentially allowing for a more realistic image processor.

### 6.1 Minipix: Physical Layout

The algorithm implemented in minipix is repeated in figure 6.1 (a), together with the transistor-level schematic and physical layout in (b) and (c) respectively. The circuitry consists of a logarithmic photocircuit with three transistor load, a first order PMOS OTA-C low pass filter and a comparator without hysteresis, all described in detail in chapters four and five. The transistor dimensions have been significantly reduced from previous implementations, as highlighted in table 6.1, along with the size of the capacitor and the phototransistor. The algorithm consumes approximately  $60\text{ }\mu\text{m}^2$ , with a 250 fF capacitor using approximately  $40\text{ }\mu\text{m}$  by  $11.5\text{ }\mu\text{m}$ , and a phototransistor taking  $48.8\text{ }\mu\text{m}$  by  $10.4\text{ }\mu\text{m}$ . The fill factor for the pixel is defined as the ratio of the area of the photoelement to the complete pixel, which in this instance equals 14.1%.

### 6.2 Minipix: Simulated Current Consumption

The current consumed by the minipix algorithm was estimated from simulation results. As before, the average current from a transient simulation was calculated, combining static and



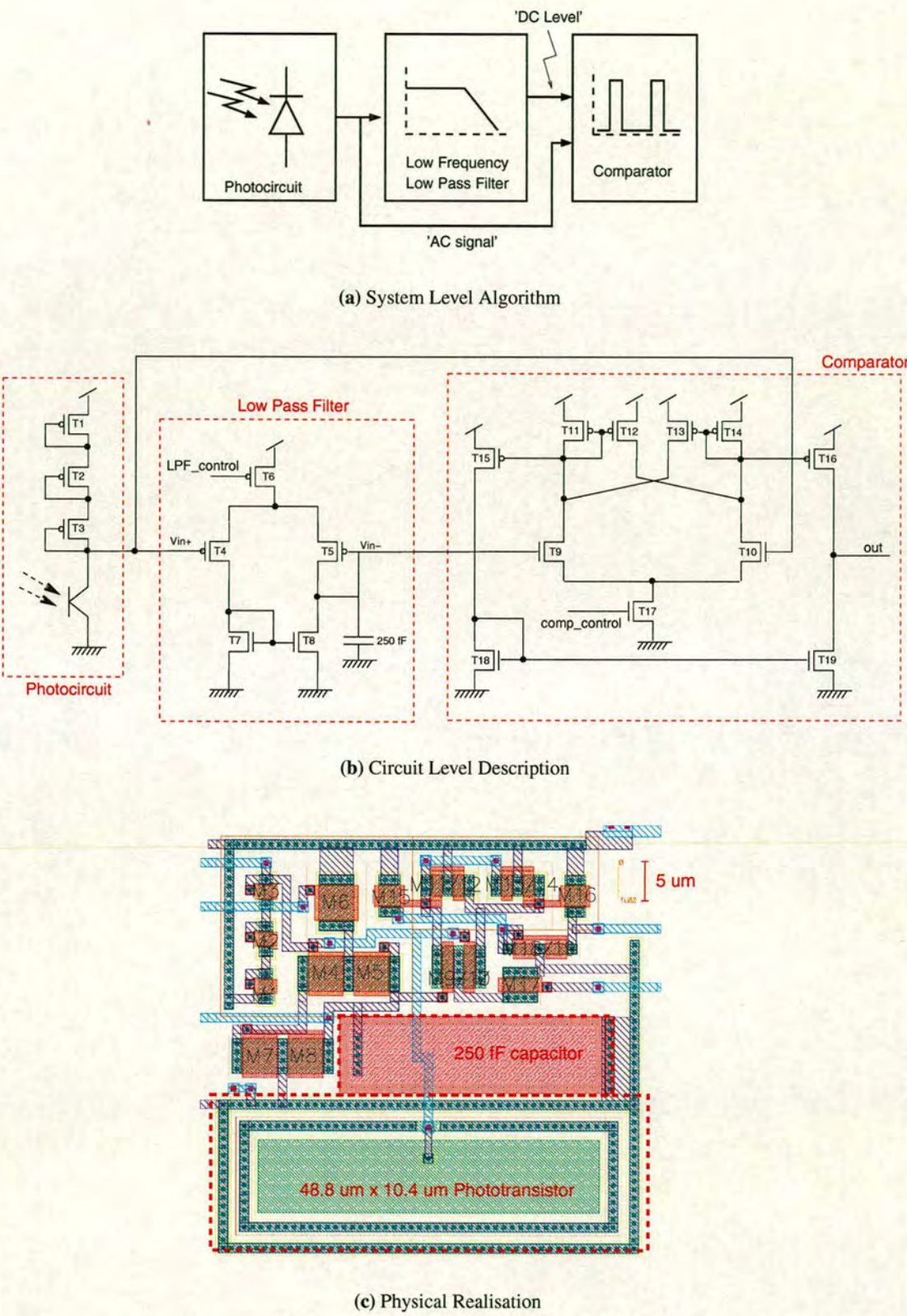


Figure 6.1: Minipix Self Referencing Pixel Processing Unit



Transistor	$\frac{W}{L}$ ( $\mu\text{m}$ )
T1	2/2
T2	2/2
T3	2/2
T4	5/5
T5	5/5
T6	5/5
T7	5/5
T8	5/5
T9	6/1.5
T10	6/1.5
T11	3/1.5
T12	3/1.5
T13	3/1.5
T14	3/1.5
T15	3/3
T16	3/3
T17	5/2
T18	1.5/3
T19	1.5/3

**Table 6.1:** Transistor Dimensions for the Minipix Algorithm

dynamic current consumption. The results in table 6.2 show current consumption for typical parameter values when the system is stimulated with a sinusoidal input. It is clear that the comparator consumes the majority of the current. By reducing its bias such that it operates in subthreshold, a significant reduction is achievable, but at the cost of a reduction in maximum input frequency. However, with 0.7 V bias the system can operate successfully up to 1 kHz.

	Parameter Values		
Input (Hz)	LPF Ctrl	Comp Ctrl	Current Consumption
100 Hz	4.5 V	1 V	3.577 $\mu\text{A}$
1 kHz	4.5 V	1 V	3.578 $\mu\text{A}$
100 Hz	4.5 V	0.7 V	13.98 nA
1 kHz	4.5 V	0.7 V	13.99 nA
100 Hz	4.59 V	0.7 V	13.76 nA
1 kHz	4.59 V	0.7 V	13.83 nA

**Table 6.2:** Simulated Current Consumption of the Minipix Algorithm



### 6.3 Minipix: Effect of Increased Variation in Filter Cutoff Due to Subthreshold Current Mismatch

A consequence of reducing the size of the transistors in the minipix algorithm is an increase in subthreshold current mismatch. Research suggests that there are three main causes of mismatch in subthreshold currents[94], edge effects, striation effects and random variations. Edge effects depend on the orientation of each transistor with its neighbouring structures, while striation effects manifest as a spatial variation in transistor current which can be as large as 30 % of the average. Random variations can be described by a Gaussian distribution, and are inversely proportional to the length of each transistor. The standard deviation of the random current mismatch can be as large as 20 %, causing a potential design trade-off between accuracy and area.

Current mismatch in the minipix algorithm will manifest as a variation in filter cutoff frequency across the surface of the image-processor. Although each minipix's low pass filter will receive the same bias voltage, current mismatch will result in each filter having different bias currents, therefore different cutoff frequencies. To test the robustness of the minipix to such current mismatch, a series of simulations were performed with the Spectre simulation tool. By varying the bias of the filter around the default value of 4.5 V, it is possible to simulate the resultant effect on the accuracy of the output pulse train. The variation in filter amplitude response can be seen in figure 6.2 (a), with the corresponding cutoff frequencies and simulated bias currents in Table 6.3. It is clear that the chosen values produce bias currents which far exceed the maximum expected variation in bias current, with well over a 2000 % increase from 4.6 V to 4.5 V and again from 4.5 V to 4.4 V. Despite this, the pulse trains in figure 6.2 (b) show that the system produced accurate outputs when attempting to measure a 100 Hz input signal. As the filter's cutoff varies, the phase response is also effected which explains the difference in mark to space ratio of the output pulse trains. However, this is of no consequence, with only the timing of subsequent rising or falling edges being of interest. It is clear that the minipix algorithm can cope with an extremely large variation in LPF cutoff frequency.

To further analyse the system's robustness, the results in table 6.4 were obtained from simulation. In this test, the low pass filter's cutoff was once again varied from 4.6 V to 4.4 V, with the output accuracy measured as the input frequency varies from 1 Hz to 10 kHz. It is clear from the results that the system operates extremely accurately over a large range of input



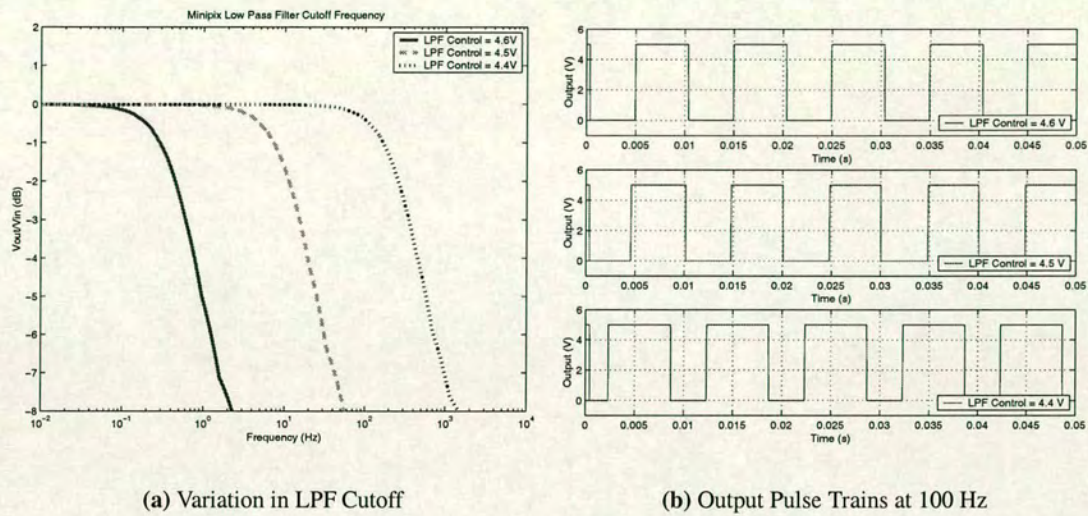


Figure 6.2: Effect of Subthreshold Current Mismatch on the Minipix Algorithm

LPF Control	Cutoff Freq	Bias Current
4.6 V	0.6 Hz	173.5 fA
4.5 V	15 Hz	4.5 pA
4.4 V	350 Hz	105.8 pA

Table 6.3: Minipix: Simulated Variation in LPF Cutoff Frequency

Input (Hz)	LPF Control		
	4.6 V	4.5 V	4.4 V
1 Hz	1 Hz	-	-
10 Hz	9.981 Hz	10 Hz	-
100 Hz	100.047 Hz	99.98 Hz	100 Hz
1 kHz	1000.036 Hz	999.6 Hz	1000.017 Hz
10 kHz	-	-	-

Table 6.4: Simulated Minipix Output Frequency Accuracy with Variable LPF Cutoff



values. If 4.5 V is taken as the default LPF bias voltage, the minipix can accurately encode the input fundamental at 100 Hz and 1 kHz despite the presence of massive variations in filter cutoff frequency. However, at extremely low or high input frequencies, the system cannot cope with these bias values. It can be concluded that an imager constructed with the minipix pixel processing units may require several frequency sensitivity settings, depending on the range of frequencies of interest. This would be a simple matter of varying the bias of the low pass filter. Despite this, it is clear that the minipix system is extremely robust to potential variations in LPF cutoff frequency caused by subthreshold current mismatch.

6.4 Minipix: Measured IC Test Results

The fabricated version of the minipix algorithm was tested in a similar fashion to the self-referencing pixel’s of test IC two. Once again, the most important parameter is the accuracy with which the output pulse train’s frequency maps to that of the input. Measurements of the frequency accuracy for the lighting conditions detailed in table 6.5 were taken. These correspond directly to those employed in the testing of the self-referencing pixel, allowing a direct comparison of the results. In addition, the frequency response of the photocircuit and LPF under FA2 were measured.

Test	LED: DC	LED: AC (sine)	PMOS LPF Control	Comp. Control
FA1	1.2 V	100 mV	4.59 V	1 V
FA2	2 V	500 mV	4.59 V	1 V

Table 6.5: Minipix Frequency Accuracy Testing: Parameter Values

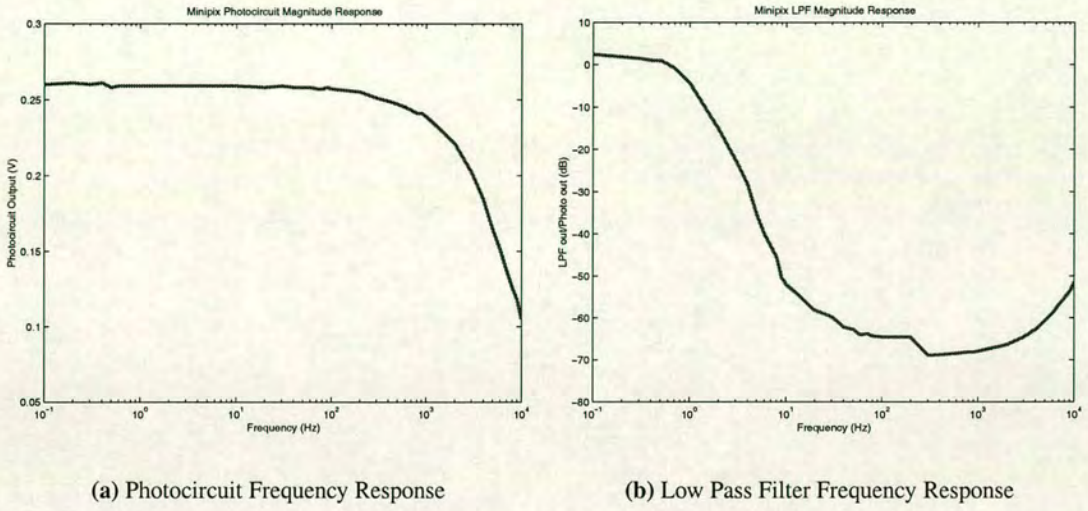
6.4.1 Minipix: Frequency Response

The frequency response for the minipix photocircuit and low pass filter are included in figure 6.3 (a) and (b) respectively. The cutoff frequency for the photocircuit is approximately 1 kHz, while the LPF is biased deep into subthreshold with a cutoff below 10 Hz.

6.4.2 Minipix: Frequency Accuracy

The accuracy of the minipix algorithm is included in figure 6.4. It is clear that the algorithm struggles at frequencies below 5 Hz, but operates with high accuracy to 10 kHz depending on





**Figure 6.3:** *Minipix Sub-Circuit Frequency Response*

the lighting conditions. This compares well with pixel three on test IC two, which is the closest to minipix in terms of circuit structure. Despite the significant reduction in implementation area, the minipix algorithm performs well.

## 6.5 Conclusions on the *Minipix* Algorithm

The performance of the minipix algorithm is comparable to the larger version implemented in test IC two. The system contains a photoelement, logarithmic photocircuit, low pass filter and comparator in an area of  $60 \mu m^2$ , with a fill factor of 14.1%. Simulations highlighting the robustness of the algorithm to variations in filter cutoff frequency caused by subthreshold current mismatch are introduced, together with an estimate of current consumption at approximately 14 nA when operating at 1 kHz. Such a pixel could be potentially integrated into a fully functional CMOS imager, producing a fundamental temporal frequency image-processor. By simply integrating the output from each pixel in a fixed time frame, an analogue level corresponding to the input frequency could be created, producing a hardware version of the fundamental frequency maps discussed in chapter three.



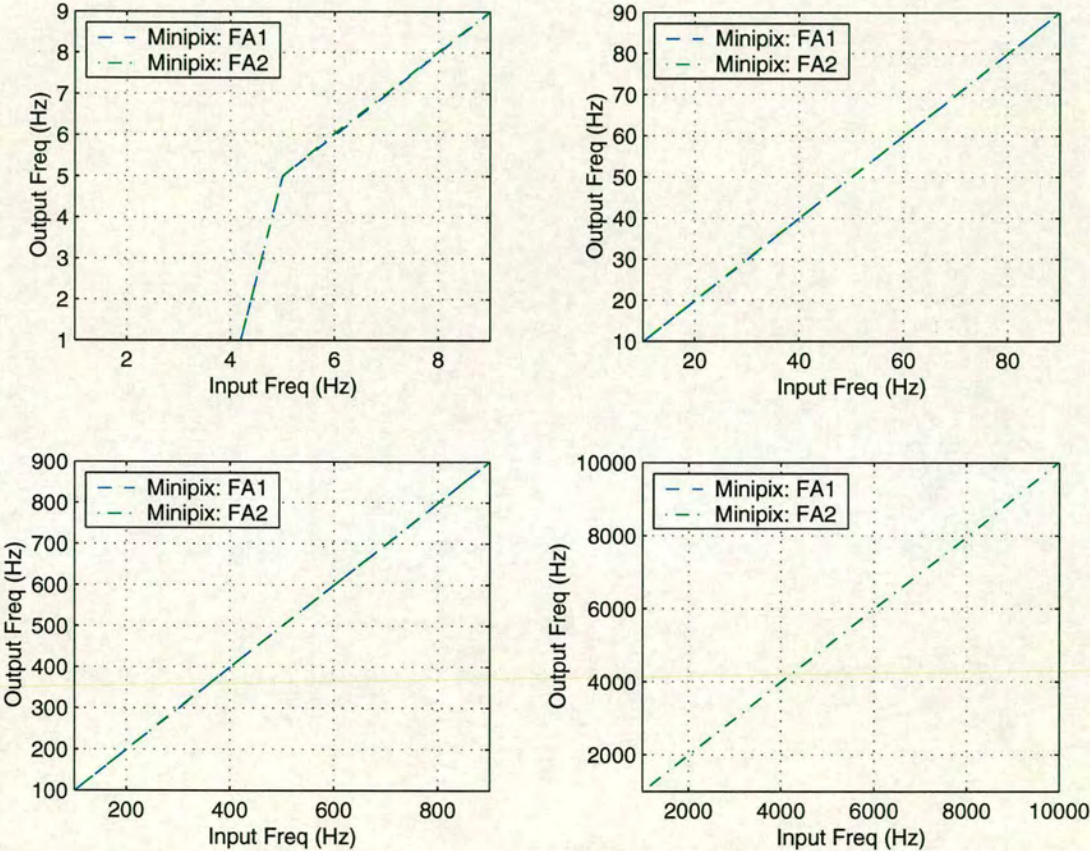


Figure 6.4: Minipix Frequency Accuracy



---

## Chapter 7

# Test IC Three: Automatically Tuned Band Pass Filter with Phase Derived Feedback

---

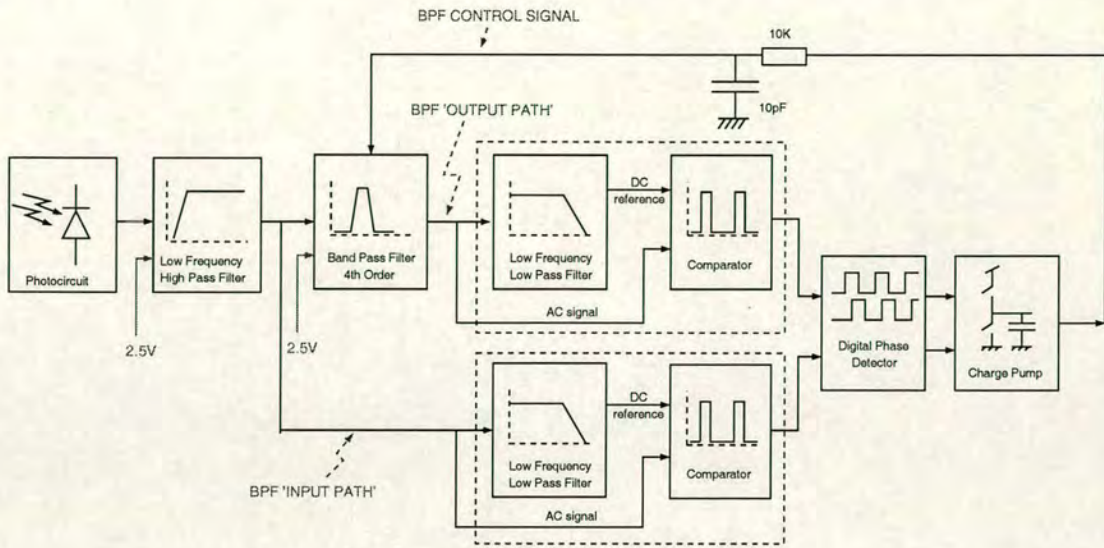
The test ICs detailed in chapters four, five and six were concerned with analysing pixel-level algorithms for the extraction of the fundamental frequency of any illumination variation. As such, various different processing structures were combined to produce pixel processing elements, highlighting the relative advantages and disadvantages of each approach. However, the ultimate aim of the research described in this thesis is not simply the extraction of the fundamental frequency, but also the relative strength of the first four harmonics. The approach, as detailed in chapter three, relies on using the fundamental frequency to place a series of band pass filters at relevant points in the frequency domain, thus building a *pseudo-Fourier processor*.

The circuits implemented on test IC three can be considered the first step in the development of such a pseudo-Fourier processor, while conforming to the low power constraints imposed by the sponsor company. As with the previous ICs developed in this research, a  $0.6\ \mu m$  AMS process available through Europractice was used, along with the Cadence design tools. As before, a third metal layer was employed to protect the signal processing circuitry from unwanted photo-induced currents.

### 7.1 Automatically Tuning BPF: System Level Approach

The success of the approach relies on the ability to accurately position band pass filters, based on the fundamental frequency extracted using the techniques described in previous chapters. Test ICs one and two concentrated on creating pulse trains, whose frequency directly corresponds to the fundamental frequency of the variation in illumination. The algorithm used in test IC three builds upon both techniques, to automatically tune an OTA-C band pass filter to the fundamental frequency of the incident illumination. The system-level algorithm can be seen in figure 7.1.





**Figure 7.1:** System Level Algorithm Implemented on Test IC Three: the BPF is automatically tuned to the incident illumination's fundamental frequency using phase-derived feedback

The algorithm uses the phase difference between the band pass filter's input and output to tune it to the relevant frequency. A filter was designed with a  $0^\circ$  phase difference between input and output at the peak of its magnitude response or centre frequency. If a negative feedback loop is created to force this phase difference to be zero, the BPF will automatically be tuned to the fundamental frequency of the incident illumination. As the input frequency changes, the feedback loop will force the filter's centre frequency to vary as it tries to ensure the phase difference between its input and output remains zero.

The phase-derived negative feedback system in figure 7.1 operates by first converting the incident illumination to a voltage signal with a logarithmic compression photocircuit. Due to the requirements of the employed OTA-C BPF, the DC level from the photocircuit is removed with a low frequency high pass filter, with the transient information superimposed onto a 2.5 V reference level. The BPF requires this well defined reference voltage for correct operation. At this stage, the algorithm splits into two paths, with both the BPF's input and the output signals being passed through self-referencing pulse creation units, as described in chapter five. These units create pulse trains that directly correspond to the frequency information seen before and after the band pass filter, meaning any phase difference between the two will be evident in the time delay between the two pulse trains. A digital phase detector is then used to find the difference



between these signals, raising one of two outputs depending on whether there is a negative or positive phase difference. These outputs are supplied to a charge pump circuit, which either charges or discharges a capacitor depending on the direction and magnitude of the phase difference. The charge pump's output voltage is then fed-back through a low pass filter to act as the control voltage of the band pass filter, completing the loop and ensuring the phase difference between the BPF's input and output is  $0^\circ$ .

As before, the sponsor companies requirements for low power circuit techniques and focal plane processing shaped the choice of circuit elements that were employed. The algorithm is too large to fit into a single pixel processing element if a realistic resolution is to be achieved. However, a system that employs such a BPF tuning unit for a group of pixels may be feasible, with each pixel acting as the input in turn. Wherever possible, circuits biased with subthreshold currents were used to minimise power consumption. Many of the circuit elements are similar to those previously implemented on test ICs one and two, with the logarithmic photoreceptor, low and high pass OTA-C filters and comparators documented in chapters four and five.

The use of circuits biased in the weak inversion region of operation may result in an increase in bias current mismatch. An advantage of the proposed system is that the filter is continually tuned by the feedback loop, increasing the accuracy accordingly. The phase of the band pass filter's output is used directly to tune its centre frequency, meaning that the potential mismatch between each transconductance element will be minimised. Such a technique is an example of direct filter tuning[110].

## **7.2 OTA-C 4th-Order Band Pass Filter**

As previously mentioned, the band pass filter is required to have a phase difference of  $0^\circ$  at its centre frequency. Coupled with this is the low frequency nature of the signals of interest in this research, along with the low power requirements imposed by the sponsor company. Previous work on OTA-C filters biased in the subthreshold region of operation has proved the success of the approach, with very low cutoff frequencies consuming nW power levels. Another potential advantage of the approach for this application is the ease with which the cutoff frequencies can be varied, using a single control voltage to vary the bias of OTA element. Combined with this is the wide tuning range available to filters when biased in the subthreshold region of operation when compared with strong inversion. This stems from the wider range of transconductance



values that can be achieved in the subthreshold regime due to the linear relationship between  $g_m$  and tail current, compared with the square root relationship for strong inversion. This principle was highlighted in chapter four, with equations 4.5 and 4.6 and the simulation results in figure 4.11. For all these reasons, coupled with the experience gained through test ICs one and two, a decision to implement the band pass filter using OTA-C techniques was made.

### 7.2.1 OTA-C 4th-Order BPF: Theory

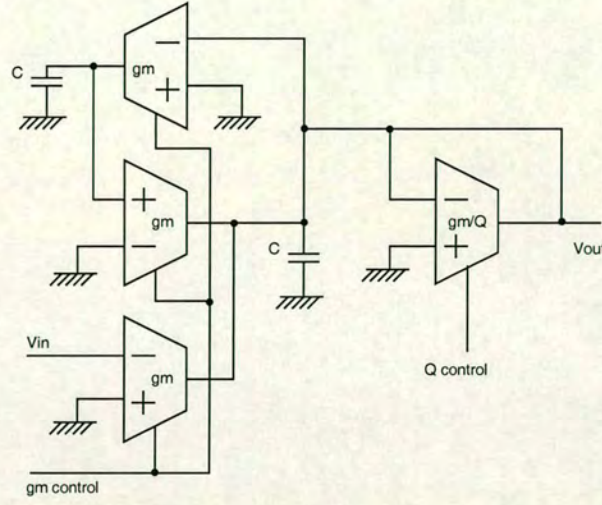
A simple approach to realising high order filter structures involves cascading lower order biquad filter sections. A biquad is essentially a hybrid second-order filter section that can be used create different filtering effects depending on which output is selected. The circuit in figure 7.2 is an OTA-C band pass biquad filter stage, which has the transfer function highlighted in equation 7.1. Comparing the transfer function to the standard form for a second order BPF as highlighted in equation 7.2, it is clear that the centre frequency  $\omega_0$  is equal to  $\frac{g_m}{C}$ , which is similar to the relationship of the cutoff frequencies for the first order low and high pass filters used in test ICs one and two. This property allows the filter to be tuned by varying the transconductance of the OTA elements. The factor  $Q$  refers to the *pole quality factor* and governs the distance of the poles from the  $j\omega$  axis when viewed on the pole-zero map. A high  $Q$  factor means the poles are close to the  $j\omega$  axis, resulting in a highly selective filter. To ease the design process, a pole quality factor of one was selected for the BPFs implemented on test IC three, allowing all four OTA elements to have the same transconductance values and the  $g_m$  and  $Q$  control lines to be shorted together. However, the option remains available to vary the filter's  $Q$  factor if required, at the cost of tuning two control parameters independently instead of one.

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{g_m}{C}s}{s^2 + \frac{g_m}{QC}s + \left(\frac{g_m}{C}\right)^2} \quad (7.1)$$

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{-K_1 s}{s^2 + \frac{\omega_0}{Q}s + (\omega_0)^2} \quad (7.2)$$

Of particular interest regarding the biquad BPF is its phase response corresponding to the centre frequency. Using the standard form for the biquad's frequency response, the phase response can be calculated using the relationship in figure 7.3. By replacing  $s$  with  $j\omega$  and applying equation 7.3 to equation 7.2, the expression in equation 7.4 is achieved.





**Figure 7.2:** Second Order OTA-C Band Pass Biquad Filter

$$\arg \left( \frac{V_{out}(s)}{V_{in}(s)} \right) = \tan^{-1} \left( \frac{Im}{Re} \right)_{\text{numerator}} - \tan^{-1} \left( \frac{Im}{Re} \right)_{\text{denominator}} \quad (7.3)$$

$$\arg \left( \frac{V_{out}(s)}{V_{in}(s)} \right) = \tan^{-1} \left( \frac{-K_1 \omega}{0} \right) - \tan^{-1} \left( \frac{\frac{\omega_0 \omega}{Q}}{\omega_0^2 - \omega^2} \right) \quad (7.4)$$

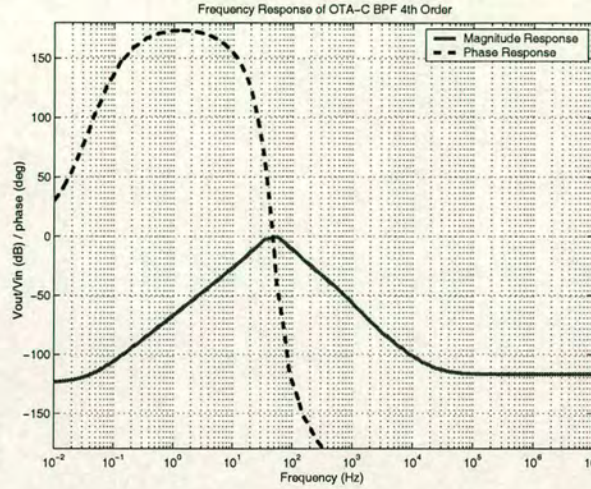
Now,  $\tan^{-1} \left( \frac{-K_1 \omega}{0} \right) = -90^\circ$ , while at the centre frequency of the filter,  $\omega_0 = \omega$ . Therefore, the relationship becomes that in equation 7.5:

$$\arg \left( \frac{V_{out}(s)}{V_{in}(s)} \right) = -90^\circ - \tan^{-1} \left( \frac{\frac{\omega_0 \omega}{Q}}{0} \right) \quad (7.5)$$

Once again,  $\tan^{-1} \left( \frac{\frac{\omega_0 \omega}{Q}}{0} \right) = -90^\circ$ , so the overall phase response for this OTA-C band pass filter biquad at its centre frequency is  $-180^\circ$ . If two such biquad stages were to be cascaded, to produce a fourth order BPF, the combined phase response at the centre frequency would be  $-180^\circ + -180^\circ$ , which equals  $0^\circ$  in total. Such a filter was implemented and simulated using the Spectre design tool, with the simulation results depicted in figure 7.3. It is clear that the phase response at the filter's centre frequency is  $0^\circ$ , allowing it to be used as the band pass filter in the phase-derived feedback system. In effect, any band pass filter built using cascaded biquad sections could be used in the automatic tuning algorithm, provided the order is an integer



multiple of four.



**Figure 7.3:** *Simulated Frequency Response for the 4th-Order OTA-C Biquad Band Pass Filter: The phase response at the centre frequency is  $0^\circ$ , meaning it meets the criterion for the phase-derived feedback system*

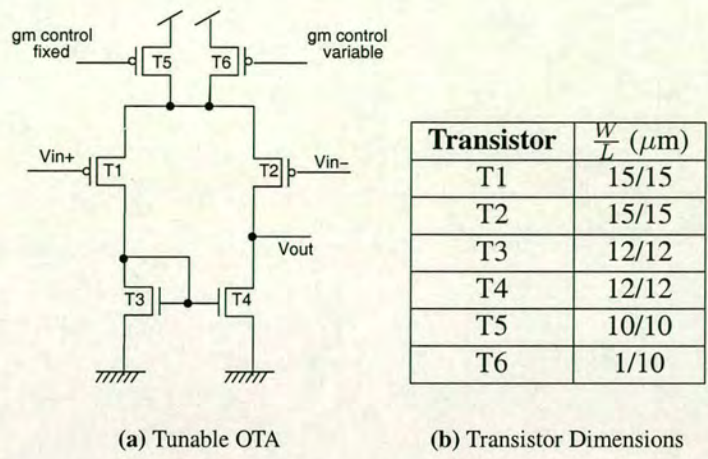
## 7.2.2 Operational Transconductance Amplifiers on Test IC Three

The operational transconductance amplifiers used in the implementation of the OTA-C band pass filter employ simple differential stages, similar to those on test IC two. The OTA can be seen in figure 7.4, together with the transistor dimensions. Two control transistors are included in the OTA, one which remains fixed to provide a certain minimum value of  $g_m$ . The variable control voltage can then be used to fine tune the transconductance value. From the filter's perspective, the fixed control sets a minimum possible centre frequency for the band pass filter, while the variable control receives the fed-back control signal, allowing the filter to tune itself to different centre frequencies.

## 7.2.3 Physical realisation of the 4th-Order OTA-C BPF

The OTA-C biquad band pass filter section comprises four OTA structures and two 10 pF poly-poly capacitors, connected as shown in figure 7.2. The fourth order BPF is realised by simply cascading two such biquad sections, meaning it requires four capacitors and eight OTA structures. The physical layout of the filter can be seen in figure 7.5. The structure consumes an area of approximately  $475 \mu m$  by  $550 \mu m$ . However, by reducing the size of the capacitors and





**Figure 7.4:** Tunable PMOS Differential Pair OTA Circuit Implemented on Test IC Three: The circuit has two control voltages, one which remains fixed and the other which is variable, to fine tune the transconductance

employing fewer guard structures, this could be reduced considerably.

### 7.2.4 OTA-C 4th-Order BPF: Power Consumption

An attempt to estimate the power consumption of the filter was made using the Spectre simulation tool. As with previous filter implementations, the BPF is biased with subthreshold currents, principally to allow the low centre frequencies required. The current consumption values in table 7.1 were taken by measuring the current drawn from the supply voltage, for different combinations of control voltages. It is clear that the current consumption increases as the bias voltages reduce, meaning that higher centre frequencies consume more power. This makes intuitive sense, with the filter biased in weak inversion for low centre frequencies but moving into moderate or even strong inversion for higher frequencies.

### 7.2.5 OTA-C 4th-Order BPF: Measured Test IC Results

The band pass filter implemented on IC three was tested for both its phase and magnitude frequency response. The success of the approach relies on the filter's centre frequency corresponding to a  $0^0$  phase difference between its input and output. The filter also has to be able to tune to the low frequencies required by the system. Another BPF feature of interest is the variation in magnitude response caused by the Q control parameter. As previously mentioned,



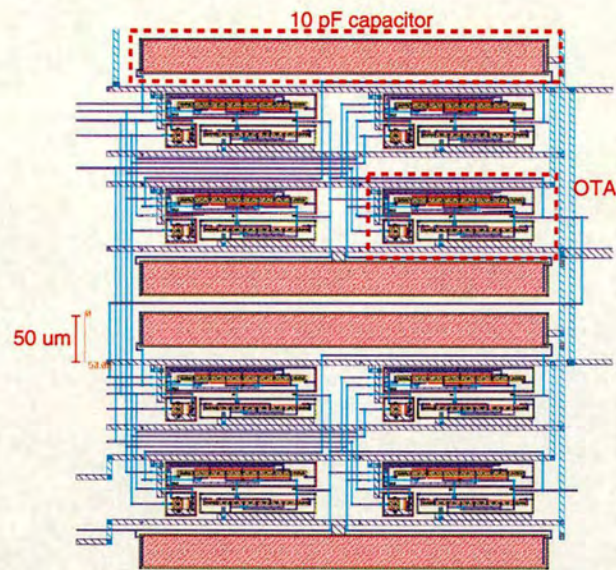


Figure 7.5: Physical Layout of the OTA-C 4th-Order Band Pass Filter

Fixed Control (V)	Variable Control (V)	Current Consumption
4.4	4.1	37.75 nA
4.4	4.0	178.67 nA
4.4	3.9	489.8 nA
4.3	4.1	61.39 nA
4.3	4.0	202.30 nA
4.3	3.9	512.4 nA
4.2	4.1	298.6 nA
4.2	4.0	439.48 nA
4.2	3.9	750.5 nA

Table 7.1: Simulated Current Consumption for the 4th-Order OTA-C Band Pass Filter

the Q factor of the filter governs its selectivity for a particular centre frequency.

7.2.5.1 OTA-C 4th-Order BPF: Frequency Response

The Bode plots in figure 7.6 were created by applying a sinusoidal input from a signal generator and measuring the output amplitude, with the phase response measured from the output to the input. The results were obtained with both the fixed control and Q control parameters set at 4.33 V, while the variable control parameter was varied from 4.25 V to 4.10 V.

Figure 7.6 (a) clearly depicts the tunability of the filter, with the four separate values of the



variable control parameter producing four different centre frequencies, ranging from approximately 45 Hz for 4.25 V to about 250 Hz for 4.10 V. The Bode plots in figures 7.6 (b) - (e) are detailed plots of the phase and magnitude response for each of the four different variable control values. It is clear from each that the phase difference of  $0^0$  corresponds directly to the peak in the magnitude response, meaning the filter will work well in the proposed system.

#### **7.2.5.2 OTA-C 4th-Order BPF: Response to Variation in $Q$ Control**

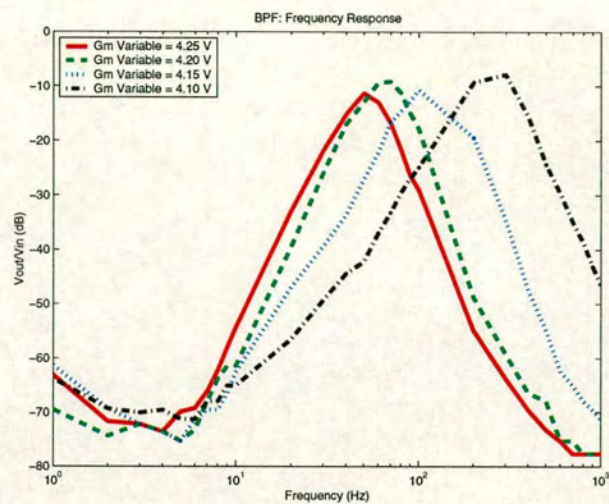
The ability to tune the BPF to different  $Q$  values allows for its selectivity to be altered, which may prove useful in this application. The magnitude responses in figure 7.7 were obtained in a similar manner to those in figure 7.6, with the fixed control set at 4.33 V, the variable control held constant at 4.20 V and the  $Q$  control varied from 4.35 V to 4.30 V. The results show that there is a clear difference in the magnitude response dependent on the value of  $Q$  control, with the peak more pronounced for a  $Q$  factor greater than one and becoming shallower as  $Q$  control is reduced. From figure 7.2, it is clear that the value of  $Q$  depends on the ratio of the transconductance of the three OTA's controlled by ' $g_m$  control' to that of the OTA governed by ' $Q$  control'. If the transconductance of the  $Q$  control OTA is greater (smaller) than the other three, the resultant  $Q$  factor will be less than (greater than) unity. As previously mentioned, when the BPF is employed in the phase-derived feedback network, the  $g_m$  control and  $Q$  control parameters are effectively shorted together, ensuring that all four OTA's have the same value of transconductance and that  $Q = 1$ .

#### **7.2.6 Comments on on the OTA-C 4th-Order BPF**

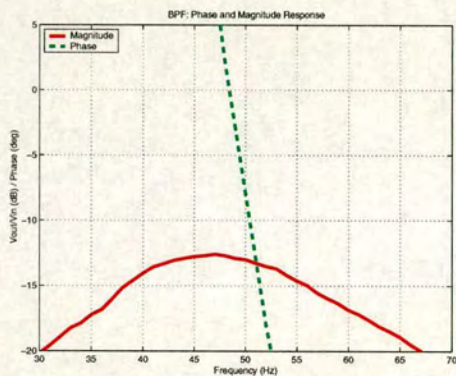
It is clear from the test results that the band pass filter behaves as expected. Crucially, the phase difference at the centre frequency is  $0^0$ , as proven mathematically, meaning the filter meets the requirements for inclusion in the phase derived feedback system. The results in figure 7.7 highlight the change in amplitude response achievable by varying the filter's  $Q$  factor. However, to make the system as simple as possible, the fixed control and  $Q$  control parameters of the BPF included in the on-chip feedback algorithm were shorted together, effectively guaranteeing a unity quality factor.

The size of the band pass filter implementation is a potential problem for inclusion in a system such as this. At  $475\ \mu\text{m}$  by  $550\ \mu\text{m}$ , the filter is too large to be repeated many times on the

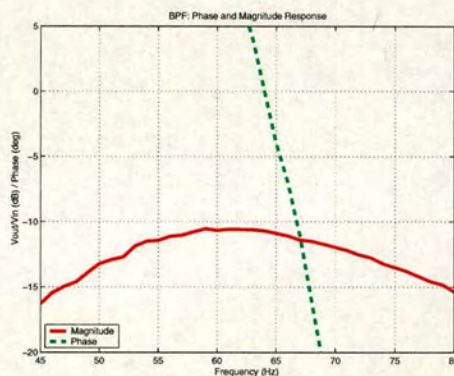




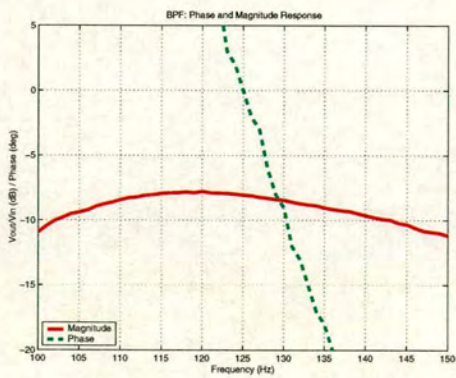
(a) BPF Frequency Response



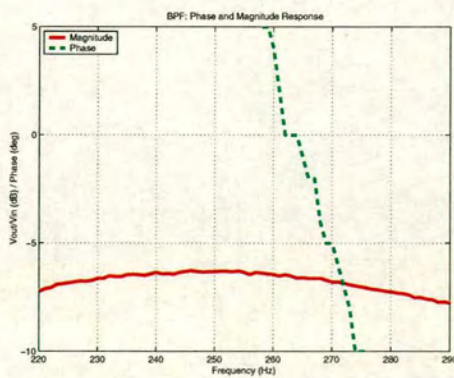
(b) Phase and Magnitude for 4.25 V



(c) Phase and Magnitude for 4.20 V



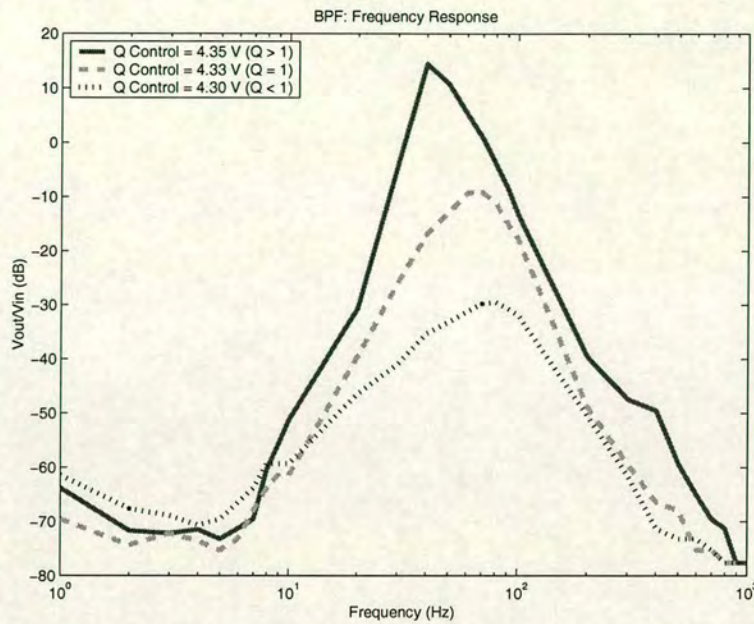
(d) Phase and Magnitude for 4.15 V



(e) Phase and Magnitude for 4.10 V

**Figure 7.6:** Measured Test IC Results: Band Pass Filter Frequency Response and the Effect of Varying the Control Voltage





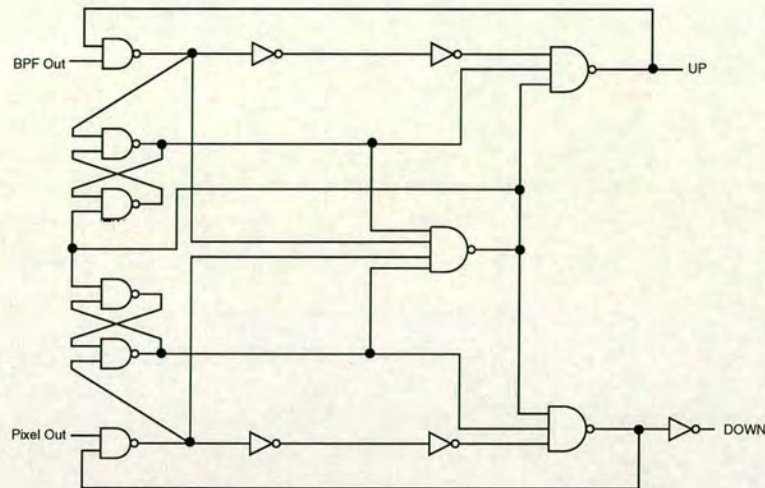
**Figure 7.7:** Measured Test IC Results: Response of the BPF to a Variation in  $Q$  Control

same IC. As this was a proof of concept, little effort was made to minimise the silicon area, with guard rings, dummy transistors and other safety structures included to ensure correct operation. In future versions of the system, the area of the band pass filter will be a critical parameter to reduce, with potential solutions such as using smaller transistors or capacitors. It should be possible to drastically reduce the area of the BPF, while still maintaining its correct operation.

### 7.3 Digital Phase Detector

The phase derived feedback system requires a phase detector circuit that can sense not only the magnitude of the difference in phase, but also the direction. More specifically, it requires a circuit that has two outputs, once which is asserted when there is a positive phase difference and the other which does the same for a negative difference. The reason for this stems from the phase response of the 4th order OTA-C band pass filter, which is positive at frequencies below the filter centre frequency and becomes negative at larger frequencies. The circuit[111] highlighted in figure 7.8 achieves this.





**Figure 7.8:** *Asynchronous Digital Phase Detector Circuit: The circuit has two outputs, which are asserted depending on the direction of the phase difference between the two inputs[111]*

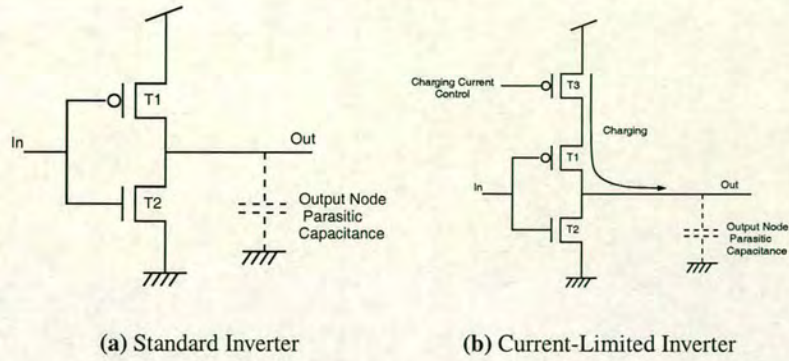
### 7.3.1 Digital Gates with Current Limiting Transistors

The original aim of the project was to develop an entirely analogue circuit level solution to the problem. The principle reason for this approach was to bias transistors in the subthreshold region of operation, thus producing a very low power system. However, given the nature of the pulse trains generated by the self-referencing pixel units, a digital phase detector seemed the best choice in terms of functionality. The requirement for the direction of any phase difference to be ascertained by the circuit added a level of complexity that could best be solved with digital techniques.

Despite meeting the functional requirements of the system, digital techniques in general suffer from higher dynamic current consumption than subthreshold analogue counterparts. The reason for this stems from the use of transistors to switch the output between high and low logic states, thus consuming high switching currents. An effort to reduce the dynamic current consumption of the phase detector was made by including an extra transistor in each pull up chain of the digital gates, effectively acting as a current limiter. As an example, consider the two inverters in figure 7.9. The inverter on the left is the standard technique, while that on the right includes an extra current limiting transistor in the pull-up path.

Limiting the current in this manner is possible due to the relatively low input frequencies of interest in this research. In high speed applications, the current required to charge and discharge





**Figure 7.9:** Comparison of Inverter Circuits with and without Current Limiting Transistors

the node capacitances is much higher, due to the relationship  $I = C \frac{dV}{dT}$ . In the inverter on the left of figure 7.9, the current used to charge the output node is set by the slew rate requirements, coupled with the rise and fall times of the input signal. In contrast, the inverter in figure 7.9 (b) has a reduced current with which to charge the output node, thus reducing the switching current, but at the price of a reduction in slew rate. This is a trade-off between power consumption and functionality which may not be useful in other applications, but was included here due to the low frequency operation and the sponsor companies requirements for low power consumption.

Estimates of the difference in current consumption for the two inverter circuits were made using the Spectre simulation tool. A series of transient simulations were performed for differing input rise times and control voltages, with both average and peak current consumption calculated. The simulations were performed at an input frequency of 100 Hz, with a load capacitance of 1 pF connected to each output. The results in table 7.2, highlight the power reduction caused by the inclusion of the current limiting transistor, with the final two columns reporting a power saving of at least an order of magnitude in both peak and average current consumption.

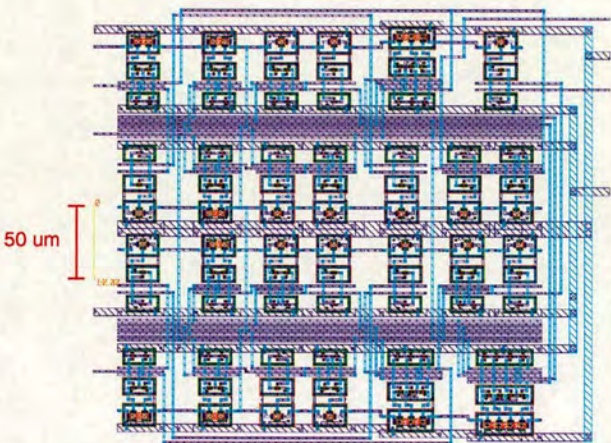
### 7.3.2 Digital Phase Detector: Physical Realisation

The physical layout of the phase detector circuit can be seen in figure 7.10. The circuit consumes approximately  $310 \mu m$  by  $285 \mu m$ . All the digital gates are surrounded by guard ring structures and emphasis was placed on functionality rather than minimising the implementation area.



Input Setup		Standard Inverter		Limited Inverter		Power Saving	
Rise time	control	Average	Peak	Average	Peak	Average	Peak
10 ns	4.1 V	1.845 $\mu$ A	36.72 $\mu$ A	105.8 nA	2.528 $\mu$ A	17.43 $\times$	14.53 $\times$
20 ns	4.1 V	5.821 $\mu$ A	56.43 $\mu$ A	165.8 nA	2.217 $\mu$ A	35.1 $\times$	25.45 $\times$
30 ns	4.1 V	10.74 $\mu$ A	68.71 $\mu$ A	209.3 nA	1.987 $\mu$ A	51.31 $\times$	34.58 $\times$
40 ns	4.1 V	16.23 $\mu$ A	77.27 $\mu$ A	241.5 nA	1.818 $\mu$ A	67.2 $\times$	42.5 $\times$
10 ns	4.2 V	1.845 $\mu$ A	36.72 $\mu$ A	94.06 nA	2.421 $\mu$ A	19.62 $\times$	15.17 $\times$
10 ns	4.3 V	1.845 $\mu$ A	36.72 $\mu$ A	93.9 nA	2.405 $\mu$ A	19.65 $\times$	15.23 $\times$

**Table 7.2:** Simulated Current Consumption for Digital Inverter Circuits with and without Current Limiting Transistor



**Figure 7.10:** Physical Layout of the Asynchronous Digital Phase Detector Circuit

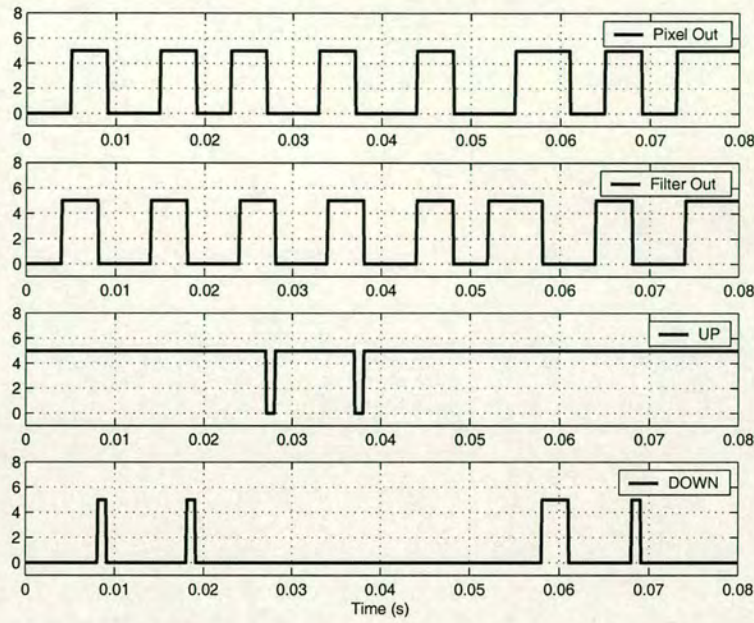
7.3.3 Digital Phase Detector: Simulated Test Results

The digital phase detector circuit was simulated using the Spectre simulation tool, with the results depicted in figure 7.11. If the pixel’s output (BPF input path) falls after the BPF’s output, the ‘down’ signal is asserted, while the ‘up’ signal asserts if the opposite is true. In both cases, the width of the output pulse is proportional to the size of the phase difference between its two inputs.

7.3.4 Comments on the Digital Phase Detector Circuit

The aim of the digital phase detector is to not only extract any phase difference between its two inputs, but to convey the direction of this phase difference. It is clear from the simulation results in figure 7.11 that this is achieved, with the two outputs being activated depending on which





**Figure 7.11:** Simulated Operation of the Digital Phase Detector

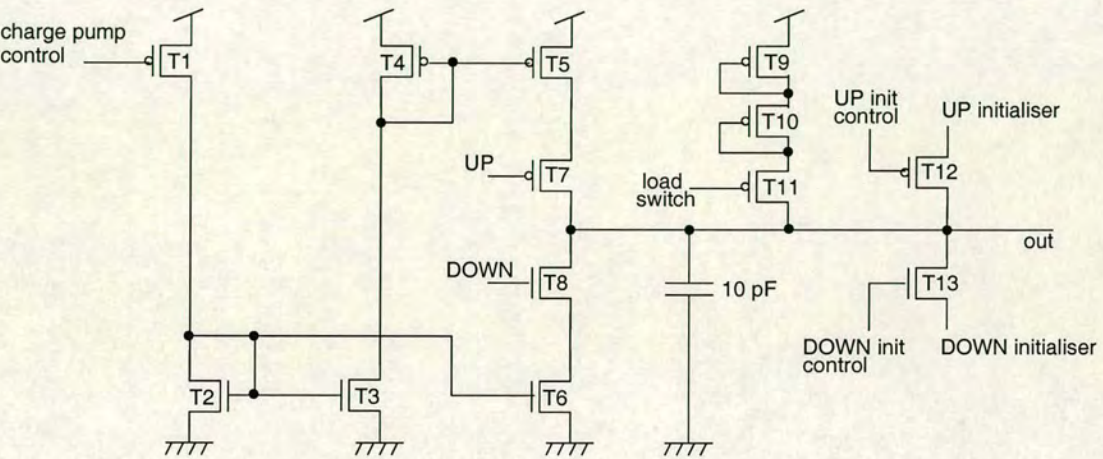
input falls first. Based on the simulation results of the digital inverter, the dynamic power reduction technique does seem to provide some benefit, with a saving of at least an order of magnitude as highlighted in table 7.2. However, this approach is only valid at low frequencies, such as those of interest in this research.

## 7.4 Charge Pump

The charge pump circuit essentially converts the 'up' and 'down' signals from the phase detector into a reference voltage, used to control the band pass filter. At its simplest, it is just a capacitor which is charged by the presence of a pulse on the 'up' signal, and correspondingly discharged by a similar pulse on the 'down' signal[111]. The capacitor's voltage is then supplied to the OTA-C band pass filter, as the input to the *gm control variable* transistor. Due to the sensitive nature of the band pass filter's control voltage, a subthreshold current is used as the charging/discharging current, meaning output voltage changes are very small. The circuit can be seen in figure 7.12. Transistors T1-T6 create the subthreshold reference current for the switching control transistors, T7 and T8. Transistors T9-T11 form a diode load for the output, which can be switched in or out of the circuit with the load switch input. The aim is to create a minimum voltage to which the capacitor can discharge, by having a continuous *drizzle* of cur-



rent from the load which begins to increase as the output voltage reduces. Creating a minimum voltage in this manner effectively translates to a maximum centre frequency for the tunable band pass filter, ensuring it does not move outwith a sensitive frequency range. Transistors T12 and T13 allow the charge pump's output to be initialised to external voltage references, thus providing some control over the band pass filter's initial centre frequency. It is also possible to effectively break the feedback loop with these transistors, by forcing the charge pump's output to an external reference. The dimensions of the transistors can be found in table 7.3.



**Figure 7.12:** Charge Pump Circuit: The capacitor is charged or discharged due to the assertion of the 'UP' or 'DOWN' control signal from the phase detector

Transistor	$\frac{W}{L}$ ( $\mu\text{m}$ )
T1	10/10
T2	5/10
T3	5/10
T4	10/10
T5	10/10
T6	5/10
T7	1/0.6
T8	1/0.6
T9	10/1
T10	10/1
T11	1/0.6
T12	1/0.6
T13	1/0.6

**Table 7.3:** Transistor Dimensions for the Charge Pump



7.4.1 Charge Pump: Power Consumption

The charge pump is biased with a subthreshold current to allow very small changes in the capacitors stored voltage value, thus allowing the band pass filter to be controlled with relatively high sensitivity. An advantage of this approach is the extremely low current consumption of the circuit. As before, an estimate of the current consumption was made with the Spectre simulation tool, with the results for typical control voltage values highlighted in table 7.4. As with other simulations of this type, the absolute values may be inaccurate, but it is still clear that the charge pump will consume power in the nW range.

Control Voltage (V)	Current Consumption
4.3	10.05 nA
4.35	2.62 nA
4.4	630 pA

Table 7.4: Simulated Charge Pump Current Consumption, for Typical values of Control Voltage

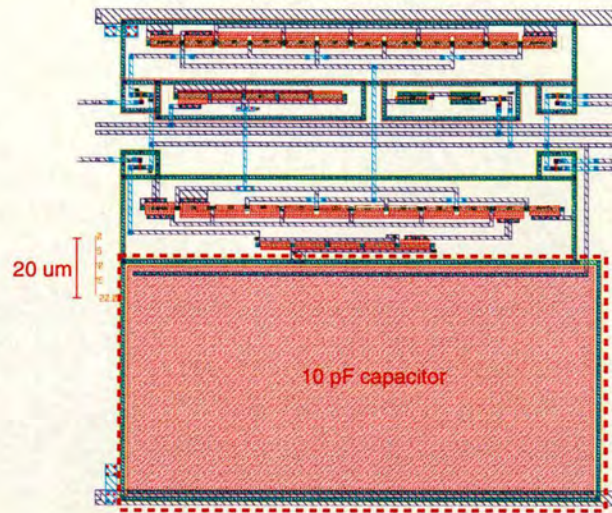
7.4.2 Charge Pump: Physical Realisation

The charge pump circuit is dominated by the 10 pF poly-poly capacitor, as the layout in figure 7.13 highlights. The circuit consumes almost 180  $\mu m$  by 175  $\mu m$ , of which the capacitor accounts for about half. The charge pump uses current mirror circuitry, which were constructed with common-centroid techniques and guard ring structures in an effort to improve matching.

7.4.3 Charge Pump: Simulated Results

The charge pump circuit was simulated using the Spectre simulation tool, with the results depicted in figure 7.14. Depending on the 'up' or 'down' control signals, the capacitor is charged or discharged with the subthreshold current. The sensitivity of the charge pump's output voltage can be varied by altering the control voltage, thus increasing or decreasing the charging/discharging current.





**Figure 7.13:** *Physical Layout of the Charge Pump Circuit*

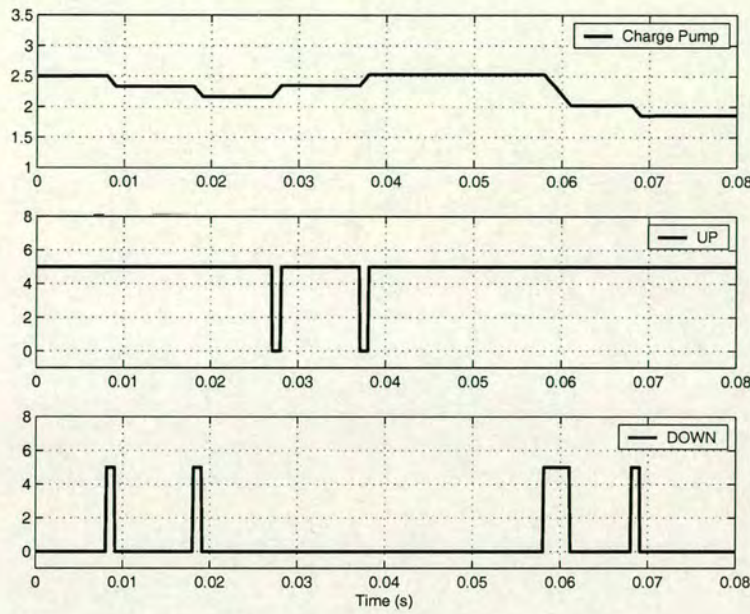
#### 7.4.4 Comments on the Charge Pump Circuit

The charge pump is a relatively simple circuit, yet its correct operation is crucial to the accuracy of the proposed system. The large 10 pF capacitance could be reduced in size, providing a benefit in implementation area, but at a cost of reduced sensitivity. Potentially, this could be countered by reducing the available charging or discharging current deeper into the sub-threshold regime. The inclusion of the switchable diode load should provide a safety factor in the form of an upper limit on the frequency to which the band pass filter can be tuned. The NMOS and PMOS initialisation transistors will also prove useful, by providing a simple means of resetting the BPF to a known centre frequency.

### 7.5 System-Level Test IC Results: Automatically Tuned BPF

The algorithm implemented on test IC three is depicted in figure 7.1. The aim is to centre the magnitude response of the band pass filter onto the fundamental frequency of the input, by comparing the phase difference between its input and output. To this end, the chip was tested to evaluate the performance of the system. Figures of merit that were examined include the range over which the band pass filter can successfully tune, the speed with which it does so and the accuracy of the filter's placement in the frequency domain. The system was characterised with inputs generated directly from a signal generator, to allow measurements with 'pure' input signals.





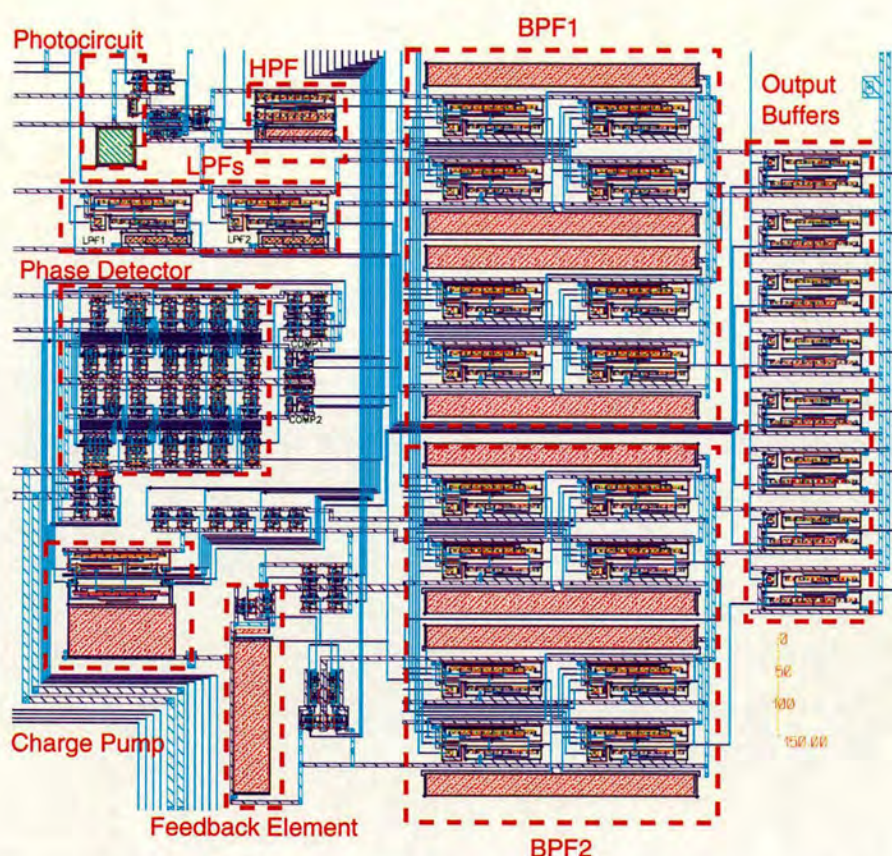
**Figure 7.14:** *Simulated Operation of the Charge Pump Circuit*

The layout of the algorithm can be seen in figure 7.15. For test purposes, a second band pass filter was included, together with eight analogue output buffers. In total, the algorithm consumes an area of  $1350\text{ }\mu\text{m}$  by  $1180\text{ }\mu\text{m}$ , with approximately half of this due to test structures, which will be unnecessary in future versions. No effort was made to minimise the area of the algorithm, as the large amounts of empty space between the processing elements confirm.

**7.5.1 Automatically Tuned BPF: Simulated Current Consumption**

The simulated current consumption for the algorithm tuning to 250 Hz is included in table 7.5. The currents were calculated using the average function in the waveform calculator as detailed in chapter four. Measurements were taken with control parameters initialised for normal operation. The table highlights the contribution of both analogue and digital current consumption separately, in an effort to gauge the effect of different parameters. From the table, it is clear that the analogue output buffers constitute the vast majority of the current drawn from the analogue supply, as they are biased in the strong inversion region of operation. With the buffers turned off, the band pass filter consumes the majority of the analogue current, which rises as the filter’s centre frequency increases. The difference between the average analogue current for 30 ms, 60 ms and 100 ms is due to the filter tuning to a higher frequency in the longer time frame. The current drawn from the digital supply is dominated by the two comparator circuits. This can be





**Figure 7.15:** Layout of the Phase Derived BPF Tuning Algorithm

significantly reduced by biasing them in the subthreshold regime, but at the price of a reduced maximum operating frequency.

The average current consumption for the system, combining analogue bias currents and dynamic digital currents can be estimated at approximately  $4.5 \mu A$  for a 250 Hz target frequency, giving a power consumption of  $22.5 \mu W$  when operated with a 5 V power supply. It is clear that digital switching currents dominate the system's consumption at low input frequencies, but as the required target frequency increases, the BPF's bias current requirements also increase as reported in table 7.1.

As expected, the current consumption increases when the algorithm tunes to the higher input frequency of 1 kHz, as the results in table 7.6 highlight. A plot of the current consumption for the 200 ms simulation can be seen in figure 7.16. It is clear that switching currents from the digital supply dominate the system's power consumption. Notice also the current drawn from the analogue supply increases until the filter is *locked* to the correct centre frequency, from



Parameter			Average Current	
Buffer Ctrl	Comp Ctrl	Sim time	I(AVdd)	I(DVdd)
1 V	1 V	30 ms	9.033 $\mu A$	7.225 $\mu A$
1 V	0.75 V	30 ms	9.033 $\mu A$	1.222 $\mu A$
0 V	0.75 V	30 ms	6.992 nA	1.221 $\mu A$
0 V	0.75 V	60 ms	13.564 nA	1.57 $\mu A$
0 V	0.75 V	100 ms	13.7 nA	4.437 $\mu A$

**Table 7.5:** Simulated Average Current Consumption for the Automatically Tuned BPF Algorithm, Tuning to 250 Hz

which point on the analogue bias current remains constant.

Parameter			Average Current	
Buffer Ctrl	Comp Ctrl	Sim time	I(AVdd)	I(DVdd)
0 V	0.75 V	100 ms	21.78 nA	11.47 $\mu A$
0 V	0.75 V	200 ms	59.95 nA	13.12 $\mu A$

**Table 7.6:** Simulated Average Current Consumption for the Automatically Tuned BPF Algorithm, Tuning to 1 kHz

### 7.5.2 Automatically Tuned BPF: Tuning Range

The tuning range of the system refers to the range of input frequencies over which the band pass filter can successfully tune. Two parameters were monitored as the input frequency was increased, the magnitude of the band pass filter's output and the fed-back value of the filter's variable control voltage. With a fixed amplitude input, generated with a signal generator, the band pass filter's output amplitude should remain relatively constant over its tuning range. The value of the variable control voltage should exhibit a linear relationship with input frequency, due to the subthreshold biasing of the band pass filter. The test was repeated for a variety of different system control parameters, as highlighted in table 7.7. All nine tests were performed with an input sinusoid of magnitude 200 mV-PP, at a DC level of 2.5 V. In addition, the BPF's reference was fixed at 2.5 V, the comparator control at 1 V, both LPFs were controlled with 4.4 V and the HPF's reference and control voltages were set at 2.5 V and 0.53 V respectively. The main parameters that were directly varied to gauge their effect on the system's performance are the charge pump's diode load, the charge pump's control voltage, the phase detector's control voltage and the value of BPF fixed control.



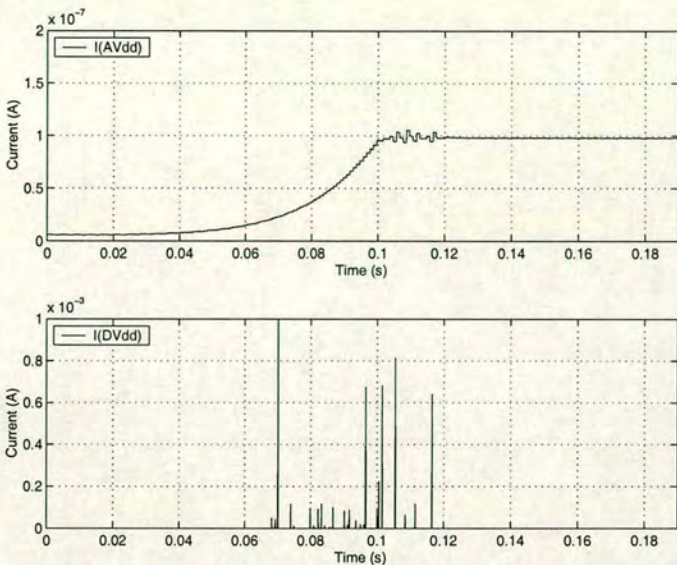


Figure 7.16: Simulated Analogue and Digital Current Consumption when Tuning to 1 kHz

	Frequency Range Test								
Parameter Value	1	2	3	4	5	6	7	8	9
BPF fixed ctrl (V)	4.33	4.33	4.33	4.33	4.33	4.33	4.4	4.25	4.25
P detect. ctrl (V)	4.2	4.2	4.2	4.2	4.1	4.1	4.2	4.2	4.2
Charge pump ctrl (V)	4.39	4.39	4.35	4.45	4.39	4.39	4.39	4.39	4.39
Diode Loadswitch (V)	0	5	0	0	0	5	0	0	5

Table 7.7: Parameter Values for the Nine Different Frequency Tuning Range Tests

7.5.2.1 Tuning Range: Band Pass Filter’s Output Magnitude

The results in figure 7.17 are grouped into four different graphs, highlighting the effect of varying each of the four system parameters mentioned earlier.

Figure 7.17 (a) depicts the effect of the diode load, which can be included or removed from the charge pump’s output by varying the ‘loadswitch’ parameter. The aim of the diode load was to limit the charge pump’s output voltage from falling below two threshold drops, effectively placing an upper limit on the maximum attainable tuning frequency. It is clear from the results that the diode load does effect the high frequency performance, with the limit being approximately 2 kHz when it is included, increasing to 5 kHz in its absence.

The results depicted in figure 7.17 (b) highlight the effect of varying the charge pump’s control voltage. Increasing this parameter forces the bias network further into the subthreshold regime,



reducing the current available to charge or discharge the capacitor, therefore effecting the sensitivity of the BPF's variable control parameter. However, this has little effect on the tuning range of the algorithm as the results confirm.

The effects of varying the phase detectors control voltage are depicted in figure 7.17 (c). When the phase detector is biased with a control voltage of 4.2 V, the maximum frequency is approximately 2 kHz. This increases to nearer 9 kHz when the bias voltage is reduced to 4.1 V, with the absence of the diode load increasing this to over 10 kHz. This is probably due to the trade-off between power consumption and speed of operation as mentioned in the section on the design of the phase detector, with the current limiting transistors impeding the high frequency performance. By reducing the control voltage, more current is available to charge the node capacitances, meaning the circuit can operate at higher frequencies.

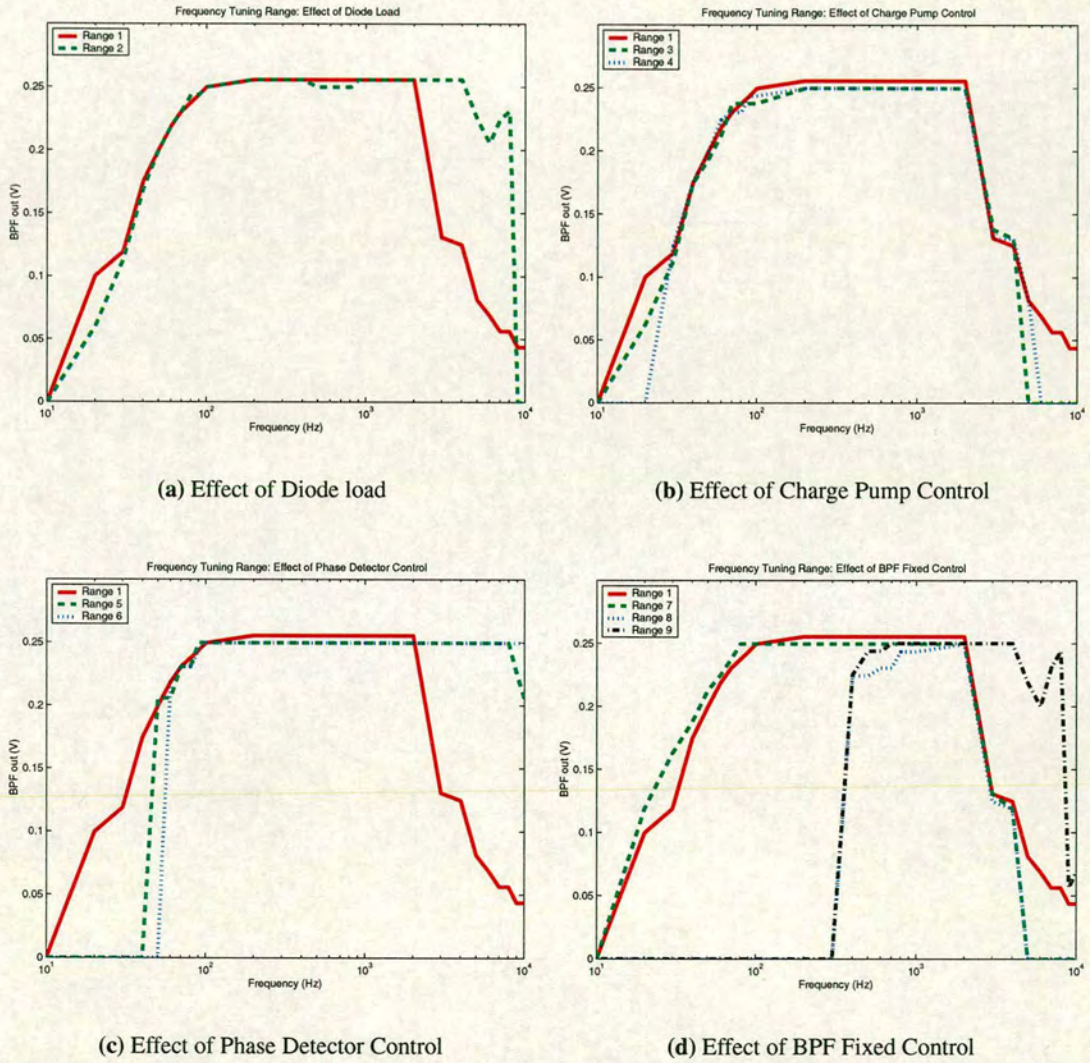
The final set of test data is plotted in figure 7.17 (d), which shows the effect of varying the fixed control voltage of the band pass filter. This parameter effectively fixes a lower limit to the band pass filter's tuning frequency. Range 1, with a fixed control voltage of 4.33 V has a lower frequency limit of approximately 90 Hz. This is reduced only slightly to 80 Hz by increasing the fixed control voltage to 4.40 V. As expected, reducing the parameter to 4.25 V for ranges 8 and 9 increases the lower tuning limit to approximately 3 kHz.

#### **7.5.2.2 Tuning Range: Band Pass Filter's Variable Control Voltage**

For all nine frequency range tests, measurements of the BPF's variable control voltage were taken, which serves to tune the filter in the frequency domain. At low frequencies, the variable control voltage should exhibit a linear relationship when plotted against log frequency, due to the subthreshold biasing of the band pass filter. In weak inversion,  $I_{ds}$  is exponentially related to  $V_{gs}$ , and directly proportional to  $g_m$ . This results in an exponential relationship between the BPF's transconductance and  $V_{gs}$  or the variable control voltage, which appears linear when plotted against the logarithm of the input frequency. As the frequency increases, the relationship may exhibit a square law characteristic as the BPF moves from weak inversion into moderate and strong inversion.

The results in figure 7.18 (a) highlight the linear relationship between variable control voltage and log frequency, confirming the band pass filter is biased in the subthreshold regime. The linear range begins at approximately 40 Hz, and stops at 2 kHz with the diode load, and nearer





**Figure 7.17:** Measured Test IC Results: Algorithm Tuning Range as a Function of BPF Output Magnitude. The BPF's output should remain constant over the tuning range



8 kHz without it.

Similar results are depicted in figure 7.18 (b), highlighting the limited effect of variations in the charge pump control voltage on the tuning range.

The tuning range for different phase detector control voltages is highlighted in figure 7.18 (c), with increased current increasing the range to over 10 kHz. The linear relationship appears to give way to a quadratic relationship above approximately 3 kHz, suggesting the filter is beginning to move out of the subthreshold regime.

Figure 7.18 (d) shows the effect of varying the value of the band pass filter's fixed control voltage. As expected, the lower frequency limit of the automatically tuning system is varied, with a minimum achievable frequency of approximately 20 Hz, corresponding to a fixed control voltage of 4.4 V.

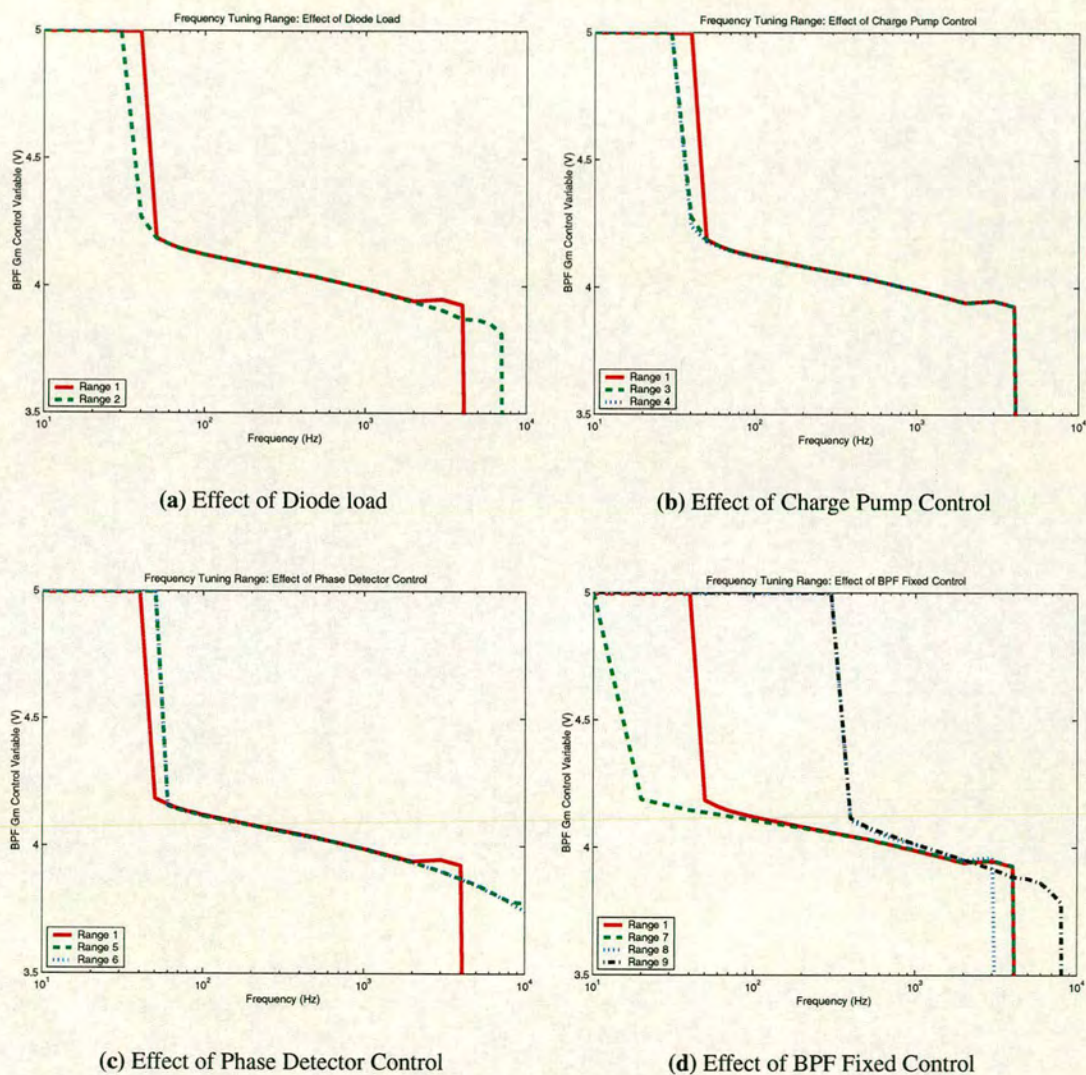
#### **7.5.2.3 Comments on the Algorithm's Tuning Range**

From the results included in figures 7.17 and 7.18, it is clear that the system is capable of successfully tuning the band pass filter from approximately 20 Hz to 10 kHz, depending on the control parameters. As expected, the diode load places an upper limit on the tuning range, while the inclusion of a fixed control voltage for the BPF performs the same function at low frequency. Varying the charge pump's control voltage has little effect on the frequency range, but the phase detector is capable of operating at higher frequencies if it is biased with more current.

Of particular interest is the linear relationship between the variable control voltage and the logarithm of the input frequency at low input frequencies, suggesting the BPF is biased in the weak inversion regime. The results in figure 7.18 (c) show that this relationship changes to a square law as the frequency is increased, meaning the filter is moving out of subthreshold. This suggests that the power consumption of the system will be linked to the required input frequency, with lower frequencies requiring less power. It would appear that the threshold of weak inversion operation is approximately 3 kHz.

Note that the values of variable control voltage reported in figure 7.18 suggest that the variable control transistor will be biased in the strong inversion region of operation. However, the aspect ratio of this transistor was selected such that the current it injects into the differential pair at





**Figure 7.18:** Measured Test IC Results: Algorithm Tuning Range as a Function of BPF Variable Control Voltage. The BPF's variable control parameter should exhibit a relatively linear relationship with frequency



low frequencies is sufficient to bias them in weak inversion. As it is the transconductance of this differential pair that governs the centre frequency, we can assume that the filter is in the subthreshold regime.

### **7.5.3 Automatically Tuned BPF: Tuning Speed**

Another parameter of interest is the speed with which the algorithm 'locks' onto the required centre frequency. Two different types of test were performed to this end. The first involves forcing the charge pump's output to a pre-determined voltage using the PMOS reset transistor (T12), effectively breaking the feedback loop. When the loop is completed, the BPF tunes from the initialisation voltage to the actual voltage for that particular input frequency. The second test involves sweeping the input frequency from 100 Hz to 1kHz within different time frames, giving an indication of the maximum rate of change of input frequency which the algorithm can process.

#### **7.5.3.1 Tuning Speed: Charge Pump Initialisation Voltage**

By varying the initialisation voltage of the charge pump before completing the feedback, the algorithm will attempt to tune to whichever input frequency is currently applied. For the purposes of these tests, the time taken to tune to 100 Hz and 1kHz were taken, giving an estimate of the speed with which the system can tune.

Table 7.8 highlights the effect of varying the initialisation voltage, combined with the diode load. For the purposes of this test, the phase detector's control voltage was fixed at 4.2 V, while the charge pump's control voltage was 4.39 V. As expected, reducing the charge pump's initialisation voltage results in an reduction in tuning delay. The diode load appears to make very little difference to the tuning speed.

The effect on tuning time of changes in charge pump control voltage are highlighted in table 7.9. In this case, the phase detector control was fixed at 4.2 V and the charge pump's output was initialised to 4.3 V. If the charge pump's control is increased, the circuit is biased deeper into the subthreshold regime, meaning the current available to charge or discharge the capacitor is reduced. The results highlight the different sensitivities available to the system, with reductions in charge pump control speeding up the tuning time considerably. However, at higher values of charge pump current, the accuracy required to tune to a low frequency target is unavailable.



Load Switch	CPump Init	100 Hz	1 kHz
0 V	4.44 V	430 ms	642 ms
5 V	4.44 V	430 ms	624 ms
0 V	4.4 V	392 ms	576 ms
5 V	4.4 V	392 ms	574 ms
0 V	4.3 V	250 ms	422 ms
5 V	4.3 V	258 ms	422 ms

**Table 7.8:** Algorithm Tuning Time: Effect of Different Charge Pump Initialisation Voltages

The system begins to take longer to tune to 100 Hz than 1 kHz, before finally failing to tune altogether for a charge pump control voltage of 4.25 V. At low frequency, any inaccuracy in the system will produce long, spurious 'up' or 'down' control signals. At extremely low values of charge pump current, such signals will have a small effect on the output voltage. However, as the charge pump's current is increased, the system will essentially *oscillate* around the correct voltage, unable to actually tune to the desired input frequency. Despite this, it is possible to increase the tuning speed of the system by a factor of over 130, simply by varying the charge pump's control voltage.

CPump Ctrl	Load Switch	100 Hz	1 kHz
4.45 V	0 V	1.08 s	2.13 s
4.45 V	5 V	1.096 s	1.95 s
4.39 V	0 V	250 ms	422 ms
4.39 V	5 V	258 ms	422 ms
4.35 V	0 V	162 ms	160 ms
4.35 V	5 V	160 ms	160 ms
4.30 V	0 V	80.80 ms	41 ms
4.30 V	5 V	76.40 ms	43.6 ms
4.25 V	0 V	-	15.6 ms
4.25 V	5 V	-	15.4 ms

**Table 7.9:** Algorithm Tuning Time: Effect of Charge Pump Control Voltage

### 7.5.3.2 Tuning Speed: Rate of change of Frequency

The second type of tuning speed test that was performed involved sweeping the system input from 100 Hz to 1 kHz, using an Agilent 33120A signal generator. It is possible to vary the time it takes for the input to sweep, allowing the response to the rate of change of input frequency to be quantified. The results in table 7.10 highlight a series of input frequency sweeps over



varying time intervals, with a charge pump control voltage of 4.39 V. For correct operation, the BPF’s variable control voltage should vary approximately from 4.128 V to 3.997 V. When the system is unable to reach this voltage in the available time frame, it can be assumed that it has reached its limit. When tuning from a lower frequency to a higher one, the system begins to fail at a time difference of less than 0.3 s, corresponding to a frequency rate of 3000 Hz/s. When tuning down in frequency, the system fails at frequency rates below 0.5 s or 1800 Hz/s.

Frequency Sweep			gm Control Variable	
Start	Stop	$\Delta t$	Initial	Final
100 Hz	1 kHz	2 s	4.128 V	3.997 V
100 Hz	1 kHz	1 s	4.128 V	3.994 V
100 Hz	1 kHz	0.5 s	4.128 V	3.997 V
100 Hz	1 kHz	0.3 s	4.128 V	3.997 V
100 Hz	1 kHz	0.2 s	4.128 V	4.013 V
100 Hz	1 kHz	0.1 s	4.128 V	4.072 V
1 kHz	100 Hz	2 s	3.997 V	4.128 V
1 kHz	100 Hz	1 s	3.997 V	4.128 V
1 kHz	100 Hz	0.5 s	3.997 V	4.128 V
1 kHz	100 Hz	0.3 s	3.997 V	4.113 V
1 kHz	100 Hz	0.2 s	3.997 V	4.100 V
1 kHz	100 Hz	0.1 s	3.997 V	4.056 V

**Table 7.10:** Algorithm Response to Changes in Input Frequency: Charge Pump Control Equal to 4.39 V

The results for the increased current available with a charge pump control of 4.3 V are included in table 7.11. As expected, the system is able to cope with much higher rate of input frequency change. When tuning up in frequency, the system fails at a time interval below 50 ms, corresponding to a frequency rate of 18 kHz/s. As the frequency sweep reduces from 1 kHz to 100 Hz, the algorithm fails at a time interval of less than 0.1 s or a frequency rate of 9 kHz/s.

7.5.3.3 Comments on the Algorithm’s Tuning Speed

It is clear from both of the the tuning speed tests that the system can be made to respond more quickly if the charge pump’s control voltage is reduced, thus providing a larger current with which to manipulate the capacitor’s charge. However, the results in table 7.9 suggest there is a trade-off between the tuning speed and the system’s ability to tune to the correct frequency. If the charge pump’s current is made too large, it appears that the system is unable to find the accuracy required to tune to lower frequencies.



Frequency Sweep			gm Control Variable	
Start	Stop	$\Delta t$	Initial	Final
100 Hz	1 kHz	2 s	4.128 V	3.997 V
100 Hz	1 kHz	1 s	4.128 V	3.997 V
100 Hz	1 kHz	0.5 s	4.128 V	3.997 V
100 Hz	1 kHz	0.3 s	4.128 V	3.997 V
100 Hz	1 kHz	0.2 s	4.128 V	3.997 V
100 Hz	1 kHz	0.1 s	4.128 V	3.997 V
100 Hz	1 kHz	0.05 s	4.128 V	3.997 V
100 Hz	1 kHz	0.01 s	4.128 V	4.078 V
1 kHz	100 Hz	2 s	3.997 V	4.128 V
1 kHz	100 Hz	1 s	3.997 V	4.128 V
1 kHz	100 Hz	0.5 s	3.997 V	4.128 V
1 kHz	100 Hz	0.3 s	3.997 V	4.128 V
1 kHz	100 Hz	0.2 s	3.997 V	4.128 V
1 kHz	100 Hz	0.1 s	3.997 V	4.128 V
1 kHz	100 Hz	0.05 s	3.997 V	4.103 V

**Table 7.11:** Algorithm Response to Changes in Input Frequency: Charge Pump Control Equal to 4.3 V

A possible solution to this trade-off involves dynamically biasing the charge pump, based on the width of the 'up' or 'down' pulses. When the target frequency is far away from the BPF's actual centre frequency at a particular time instant, the phase difference will be large, which manifests as long 'up' or 'down' control signals. At this point, the accuracy of the system is relatively unimportant, but the speed with which it moves towards the target should be maximised. As the target moves closer, the phase difference will be reduced, corresponding to shorter charge pump control signals. At this point, the system can be considered to be *fine-tuning* the BPF's centre frequency, with the accuracy of critical importance compared to the tuning speed. It follows that at large phase differences, the speed is critical which means the charge pump's current should be large, while for small phase differences, the required accuracy dominates so the current should be reduced. Such a dynamic bias arrangement could be achieved by relating the value of charge pump control to the width of the 'up' or 'down' control pulses.

#### 7.5.4 Automatically Tuned BPF: Tuning Accuracy

The aim of the system is to place a band pass filter onto the fundamental frequency of the input waveform. An obvious test to perform involves finding the accuracy with which the system



achieves this. Based on the IC test measurements for the band pass filter in figure 7.6, it is clear that the peak of the magnitude response corresponds to a phase difference of  $0^0$ . Based on this, a test was performed to calculate the phase difference between the band pass filter's input and its output as it is tuned over its entire frequency range. Another accuracy test was performed by initially tuning the system to a particular input frequency, and noting the value of variable control voltage. By forcing the charge pump's output to this value with the PMOS reset transistor, the feedback is broken but the filter has effectively been tuned by the algorithm. It is therefore possible to measure the BPF's frequency response, without the system attempting to *follow* the variations in input frequency.

#### **7.5.4.1 Tuning Accuracy: BPF's Phase Difference**

Based on both mathematical proof and measured test results, it is clear that the BPF exhibits a  $0^0$  phase difference between input and output when the magnitude response is at its peak. As a result, a measure of the BPF's phase difference over its tuning range can give an indication of the accuracy of the algorithm. The results in figures 7.19 and 7.20 depict the phase difference versus frequency, for the nine different frequency range tests in table 7.7.

Figure 7.19 (a) highlights the phase difference versus frequency as a function of the diode load, with a magnified version of the same data in (b). For the operational frequency range, the phase difference is approximately constant at between  $4^0$  and  $5^0$ . The effects of charge pump control variation are included in 7.19 (c) and (d). The results for range tests 1, 3, and 4 are very similar, with a phase difference varying between  $3^0$  and  $5^0$  for the successfully tuning frequency range. Reducing the phase detector's control voltage increases the algorithm's tuning range as highlighted in 7.20 (a), although at the cost of a very slight reduction in accuracy at low frequencies. At frequencies above 1 kHz, the accuracy is greatly improved, with an average of approximately  $4^0$ . Figures 7.20 (c) and (d) highlight the effect of BPF fixed control on the system's accuracy. For a reduction in this parameter, the accuracy fluctuates between  $5^0$  and  $10^0$ , which is inferior to larger values of BPF fixed control.

#### **7.5.4.2 Tuning Accuracy: BPF Frequency Response with Feedback Broken**

While the value of the phase difference gives an idea of the limitations of the algorithm, it is perhaps hard to relate to the accuracy with which the band pass filter is placed onto the



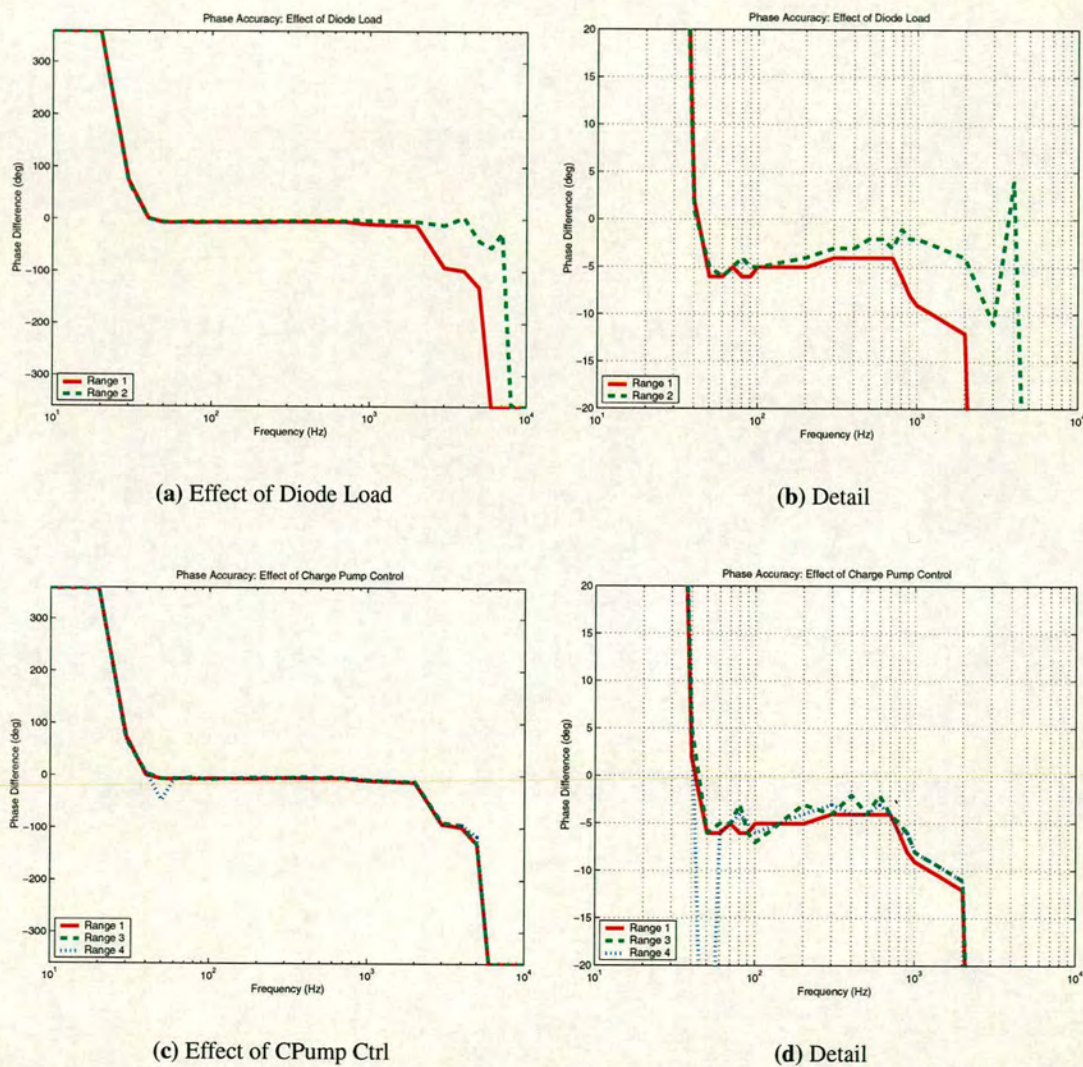


Figure 7.19: Measured Test IC Results: Algorithm Phase Difference vs Frequency



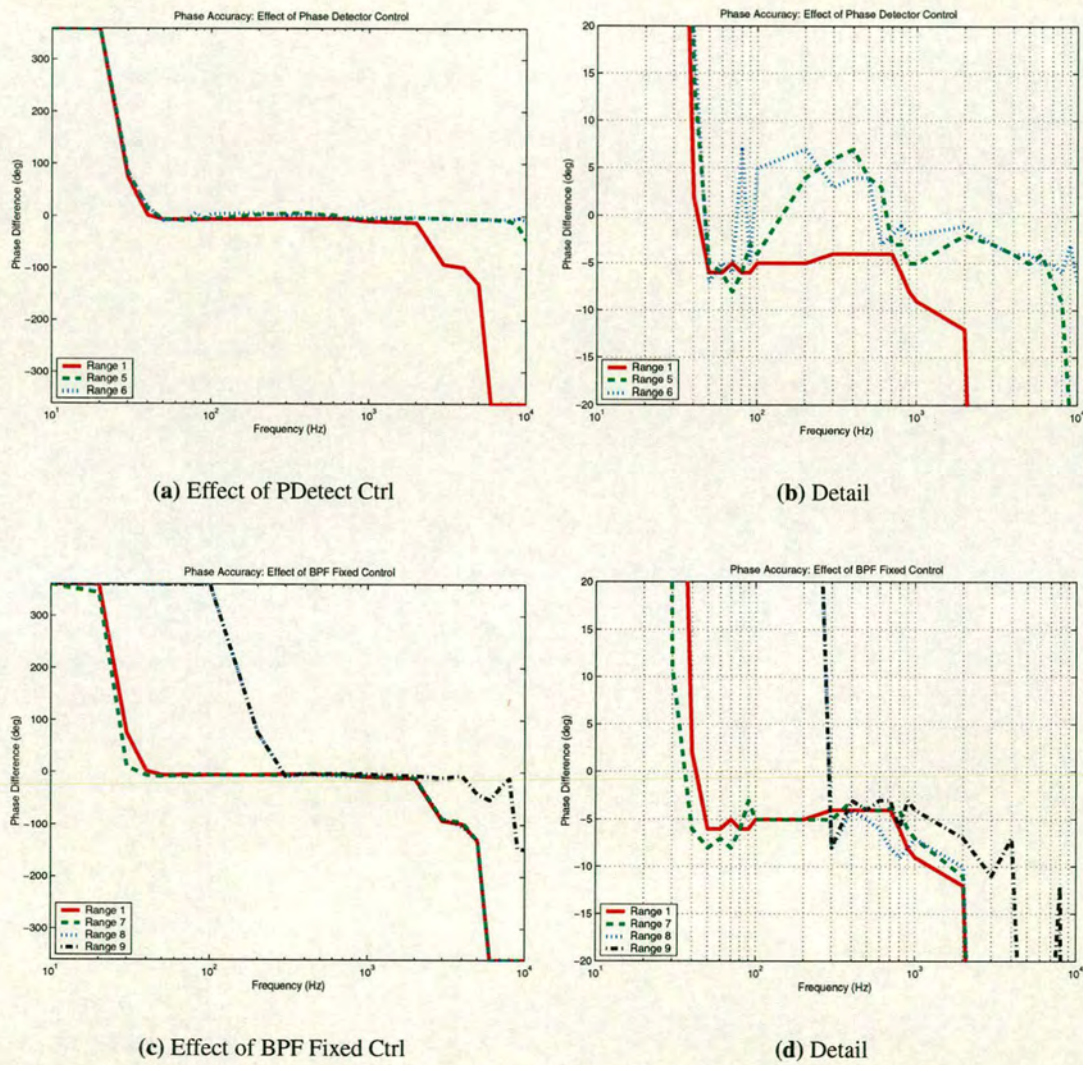


Figure 7.20: Measured Test IC Results: Algorithm Phase Difference vs Frequency



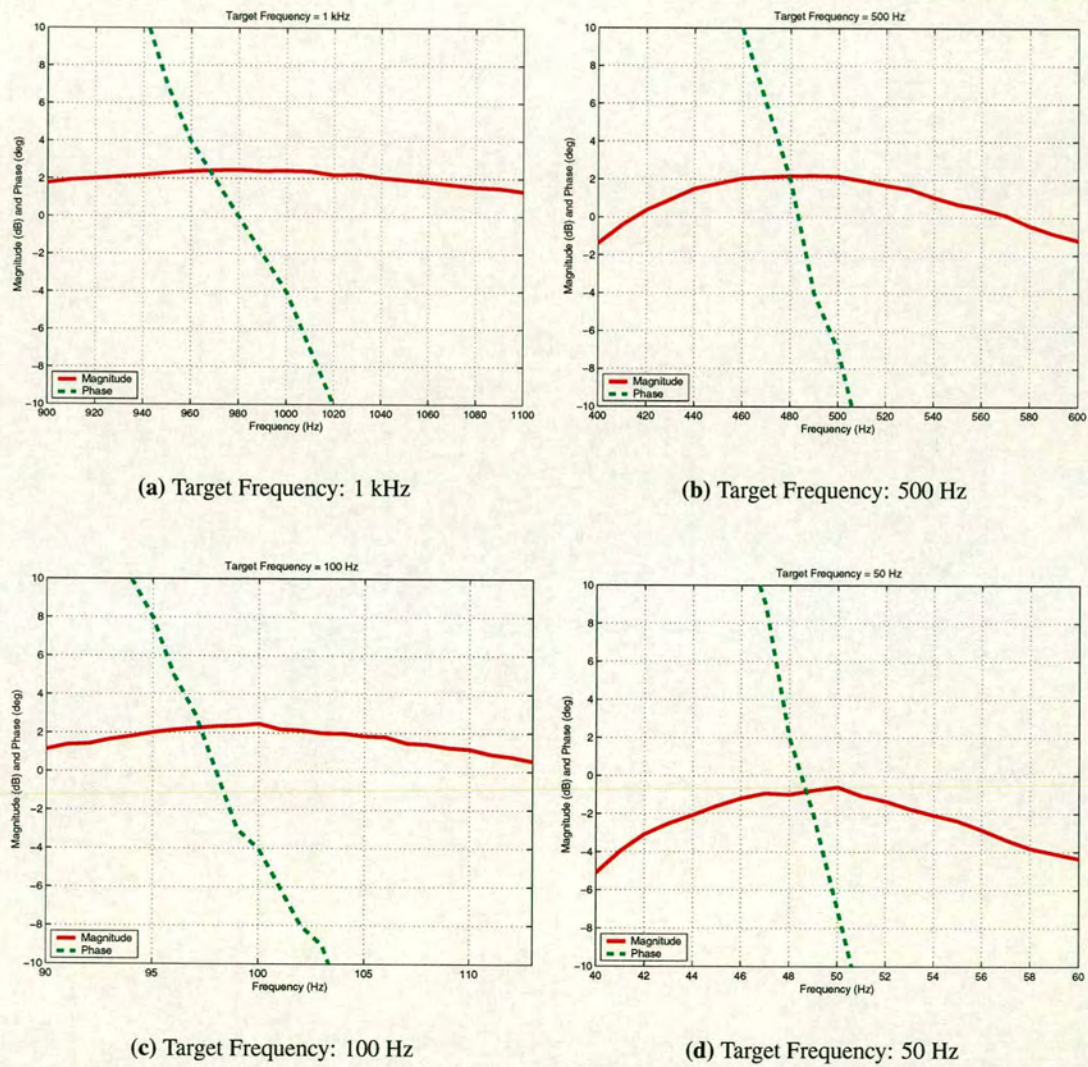
fundamental frequency of the input. A better measure is the difference between the target centre frequency and the peak of the BPF's magnitude response. To this end, a series of different tests were performed to find the BPF's frequency response for four different target frequencies. The presence of the initialisation transistors in the charge pump (T12 and T13 in figure 7.12) allows the BPF's variable control voltage to be externally manipulated. The tests were performed by first allowing the algorithm to tune to the proposed target frequency, producing a corresponding DC level on the charge pump's output. This value was then mimicked with the 'UP initialiser' external control signal, by turning on transistor T12. This effectively breaks the feedback path, but crucially, the BPF is biased as if the feedback was complete, meaning its position in the frequency domain should remain unchanged. This allows the frequency response of the BPF to be measured by simply ramping the system's input frequency, with the broken feedback stopping the system from tracking the frequency change. The approach was repeated for four different target frequencies, 1 kHz, 500 Hz, 100 Hz and 50 Hz, with each test producing a phase and magnitude bode plot highlighting the accuracy with which the filter has been tuned.

The results for each of the target frequencies are highlighted in figure 7.21. The system was setup with the parameter values for frequency range test 1, as highlighted in table 7.7. For the 1 kHz target frequency, the filter is positioned at approximately 980 Hz, based on the frequency that exhibits a  $0^\circ$  phase difference. At 1 kHz, the phase difference is approximately  $4^\circ$ , which maps well to the results in figures 7.19 and 7.20. For a 500 Hz target frequency, the filter is positioned at approximately 485 Hz, while the 100 Hz target produces a filter centred at approximately 98 Hz. Finally, the 50 Hz target frequency results in a filter positioned at approximately 48.5 Hz. Table 7.12 depicts the four different target frequencies and the percentage accuracy with which the filter has been tuned. The measured accuracy is good, with a maximum error of 3% of the desired centre frequency.

Target	Actual	$\Delta$	% Accuracy
1 kHz	980 Hz	20 Hz	2%
500 Hz	485 Hz	15 Hz	3%
100 Hz	98 Hz	2 Hz	2%
50 Hz	48.5 Hz	1.5 Hz	3%

**Table 7.12:** *Algorithm Tuning Accuracy*





**Figure 7.21:** Measured Test IC Results: Algorithm Tuning Accuracy for 1 kHz, 500 Hz, 100 Hz and 50 Hz. The Bode plots highlight the accuracy with which the BPF is tuned



#### **7.5.4.3 Comments on Algorithm Tuning Accuracy**

The measured test results prove that the system can tune a band pass filter to the fundamental frequency of the input, with a high accuracy of no worse than 3%. The phase difference versus frequency results confirm this accuracy is relatively constant over a range of approximately 20 Hz to over 10 kHz, depending on parameter values. Such results are more impressive given the subthreshold biasing of much of the system. It is clear that the direct tuning approach increases the accuracy of the system.

The phase and magnitude responses for the band pass filter in figure 7.6 highlight the relatively high gain of the phase response. Around the peak of the magnitude response, the gradient of the phase difference is high, meaning a constant phase difference of between  $4^\circ$  to  $5^\circ$  is only a small difference in terms of frequency.

### **7.6 System-Level Test IC Results: Automatically Tuned BPF with Visual Stimulus**

With the figures of merit in terms of tuning range, speed and accuracy established with an input applied directly from a signal generator, the performance of the system with a visual input was also measured. By connecting a logarithmic photodetector to the system's input, it was possible to measure the algorithm's response to a variation in input light intensity. As with the tests performed on chips one and two, an LED's intensity was modulated with a signal generator, allowing both intensity and frequency to be varied in a controlled manner.

#### **7.6.1 Automatically Tuned BPF: Tuning Range with Visual Stimulus**

The tuning range of the system was determined as before, by measuring the value of the BPF's variable control voltage versus frequency for the five different parameter setups detailed in table 7.13. For all five tests, the LED was illuminated with a 200 mV A.C. signal, superimposed onto a 1.5 V D.C. level.

The results in figure 7.22 highlight the achieved tuning range, which appear very slightly restricted at lower frequencies when compared with the function generator inputs. The reasons for this may include the frequency response of the employed LED, combined with distortion



Parameter Value	LED Frequency Range Test				
	1	2	3	4	5
BPF fixed ctrl (V)	4.33	4.33	4.40	4.40	4.40
BPF reference (V)	2.5	2.5	2.5	2.5	2.5
Comp control (V)	1.3	1.3	1.3	1.3	1.3
P detect. ctrl (V)	4.23	4.23	4.23	4.23	4.23
LPF1 control (V)	4.24	4.24	4.24	4.20	4.27
LPF2 control (V)	4.25	4.25	4.25	4.20	4.27
Charge pump ctrl (V)	4.39	4.39	4.39	4.39	4.39
HPF reference (V)	2.5	2.5	2.5	2.5	2.5
HPF control (V)	0.61	0.47	0.47	0.47	0.47
Diode Loadswitch (V)	5	5	5	5	5

**Table 7.13:** *Parameter Values for the Five Different LED Frequency Tuning Range Tests*

through the logarithmic photocircuit.

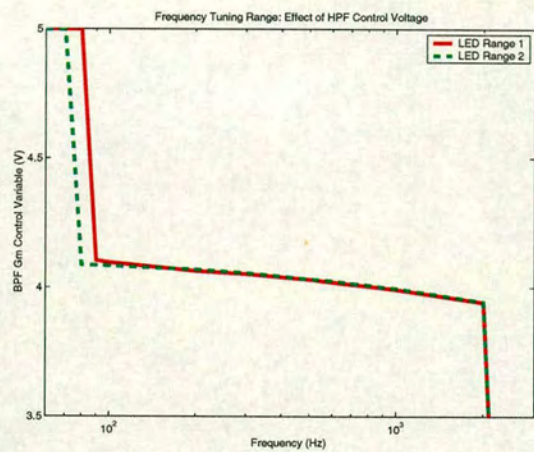
## 7.7 Conclusions on the Performance of the Automatically Tuned BPF Algorithm

From the test measurements taken from the phase-derived feedback network, it is clear that the algorithm works well. The tuning range can vary from approximately 20 Hz to over 10 kHz, depending on parameter values, while the tuning speed is also variable. It can cope with a maximum rate of change of frequency in either direction of 9 kHz/s. The accuracy with which the system tunes is also impressive, with an error of no more than 3% between the target frequency and the BPF's centre frequency. These results are more impressive given the subthreshold nature of much of the circuitry, with simulated estimates of the entire system's current consumption at approximately  $4.5 \mu A$  for a 250 Hz input.

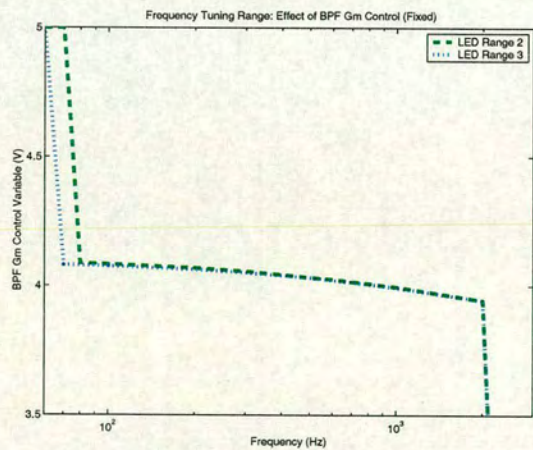
The fact that the phase-derived feedback algorithm utilises a direct tuning technique means that mismatch caused by poor subthreshold matching is minimised. The high accuracy is achieved because the band pass filter's output signal is directly employed in tuning its centre frequency.

However, the proposed algorithm is still at the test and development stage, and as such consumes more silicon area than could feasibly be integrated into a dedicated CMOS image processor. The on-chip algorithm takes an area of approximately  $1350 \mu m$  by  $1180 \mu m$ , and while much is consumed by unnecessary test circuitry such as output buffers and a second BPF, an

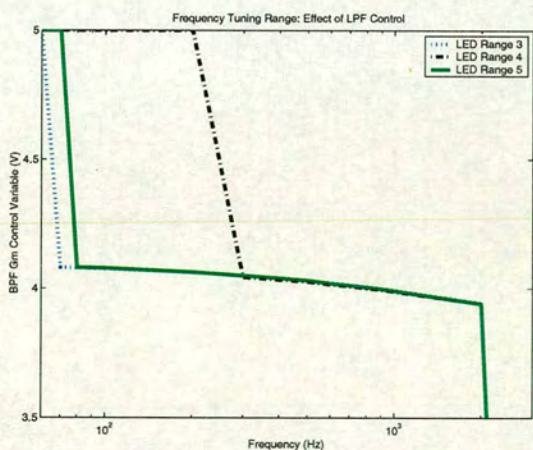




(a) Effect of HPF Control Voltage



(b) Effect of BPF Fixed Control Voltage



(c) Effect of LPF Control Voltage

Figure 7.22: Measured Test IC Results: Algorithm Tuning Range with Visual Stimulae



estimate of the actual area might reduce this by a factor of two or three at best. Much of the area is consumed by the band pass filter, which has four 10 pF capacitor as well as eight OTA structures. The phase detector is also fairly large, due mostly to the guard ring structures around each individual logic gate. Future implementations of the circuit would concentrate on reducing the area of the implementation, possibly by taking more risks with the layout regarding noise reduction techniques.

Other potential areas for improvement include a reduction in the dynamic power dissipation, possibly by replacing the digital phase detector with an analogue equivalent. The difficulty here stems from the need to find not only the magnitude of the phase difference but also its polarity, achieved with memory elements in the implemented phase detector. An analogue approach may prove more difficult to implement, but the potential advantages in terms of physical area and power dissipation warrant an investigation.

The low frequency operation of the system is limited to approximately 20 Hz when supplied with an input directly from a signal generator, as highlighted in figure 7.18. This increases slightly to approximately 70 Hz when the visual stimulus is applied. Ideally, the system would be able to operate down to almost 1 Hz, with the limiting factor in this case being the band pass filter. At such low frequencies, it appears the deep subthreshold biasing required limits the low frequency performance of the band pass filter. The reasons for the slight increase in low frequency cutoff for the visual stimulus probably stem from the employed logarithmic photodetector, and the attenuation caused by the high pass filter.

Despite these shortcomings, the performance of the system is encouraging, proving the automatically tuned BPF algorithm is the first step in the creation of a low power, pseudo-Fourier image processor



---

## Chapter 8

# Summary and Conclusions

---

The research documented in this thesis involves novel algorithms and their subsequent IC implementations for extracting temporal frequencies from visual stimulæ. This chapter aims to summarise each previous chapter, before presenting the conclusions and findings. The contributions to knowledge are highlighted along with a critical evaluation of the work undertaken. A section on future work is included, to highlight the directions that further research could take.

### 8.1 Summary

Chapter one introduced the concept of temporal frequency signatures, together with a number of target applications for a sensor capable of extracting such signatures. In addition, a number of implementation issues are discussed, with the sponsor's requirements leading to an investigation of continuous time, focal-plane computation, using transistors biased in the subthreshold region of operation. Similarities between the projects' system-level requirements and the potential advantages of biologically-inspired or neuromorphic processing are explored, leading to a design framework combining the advantages of both. Finally, the contributions to knowledge in the form of novel system level algorithms are explained.

Chapter two builds on the link between this research and the field of neuromorphic vision with a review of focal-plane approaches to spatial, temporal and hybrid spatio-temporal image processors implemented in CMOS technologies. Research of particular relevance to this project is explained in detail. The research described in this thesis could be included in the temporal processing section, as it deals only with the transient aspects of the light intensity.

Chapter three expands on the software development of potential algorithms, ranging from a wavelet style decomposition of the incident intensity variation to the adopted *pseudo-Fourier* approach. The idea involves splitting the processing into two separate tasks, firstly finding the fundamental frequency before using this to place a series of band pass filters in the frequency



domain. This idea was tested with the creation of fundamental frequency maps, as well as simulations of fixed pattern and transient noise, both common in CMOS imager implementations.

Chapter four details the development of the first test IC, which essentially converts the algorithm developed in chapter three into a circuit-level equivalent. An OTA-C high pass filter is employed to strip the DC level from the logarithmic photocircuits output, superimposing the transient information on an external reference voltage. Due to the low frequency nature of the intensity variation, the filter is biased with a subthreshold current to *pass* the relevant temporal information, while suppressing the DC level. The filter's output is then applied to a comparator, whose reference corresponds directly to the HPF's reference, ensuring the comparator's output will switch. The result is a pulse train whose frequency directly corresponds to the fundamental frequency of the incident temporal light variation. Measured test results confirm that the system can successfully extract the fundamental frequency over the input range of 1 Hz to 10 kHz, while benefiting from the low power consumption of circuits biased in the subthreshold region of operation.

Improvements to the pixel processing unit implemented on test IC one are introduced in chapter five. The original high pass filter is replaced with a low pass version, in order to extract the DC level of the photocircuit's output for use as the reference voltage for the comparator. As before, measured test IC results confirm the correct operation of the algorithm, which successfully extracts the fundamental frequency from 1 Hz to 10 kHz depending on system parameters.

Chapter six highlights the design of the *minipix* algorithm, essentially a miniaturised version of the self-referencing scheme proposed in chapter five. The system consumes an area of approximately  $60 \mu\text{m}^2$  yet is capable of accurately extracting the fundamental frequency of temporal light variations. A simulated average current consumption of less than 14 nA is consumed by the *minipix* algorithm when operating at a 1 kHz input frequency. The *minipix* algorithm was conceived as a pixel processing unit that could be realistically included in a CMOS temporal frequency image processor.

With two techniques capable of accurately extracting the fundamental frequency of the incident intensity variation, attention shifted to the second stage of the *pseudo-Fourier* algorithm developed in chapter three. Chapter seven introduces the automatically tuned BPF algorithm, which builds on the previous two test ICs to position an OTA-C BPF on the fundamental frequency of the incident intensity variation. The approach uses a feedback system to tune the



band pass filter to the correct frequency, based on the phase difference between its input and output. A BPF with a  $0^\circ$  phase difference corresponding to its centre frequency is included in a negative feedback loop, which attempts to minimise the phase difference between its input and output. As a result, the filter's centre frequency will automatically tune to the fundamental frequency of any input signal. Measured test IC results confirm that the system operates, with a tuning range of 20 Hz to 10 kHz, maximum tuning speed of 9 kHz/s and an accuracy of within 3 % of the desired centre frequency. Simulation results of the algorithm suggest a current consumption of  $4.5 \mu A$  when tuning to 250 Hz, giving a power consumption of approximately  $22.5 \mu W$  when operating from a 5 V supply voltage. The main characteristics of the automatically tuning BPF algorithm are summarised in table 8.1.

Property	Value
Tuning Range	20 Hz - 10 kHz
Tuning Rate	9 kHz/s
Tuning Accuracy	3 % of centre freq.
Simulated Current Consumption @ 250 Hz	$\approx 4.5 \mu A$
Simulated Current Consumption @ 1 kHz	$\approx 13.2 \mu A$

**Table 8.1:** *Properties of the Automatically Tuned BPF Algorithm*

## 8.2 Conclusions

In general, the research has proved the potential of creating a dedicated low-power image-processor, capable of extracting temporal frequencies from visual data. Although not as powerful as combining an imager with some form of dedicated DSP, the advantages in terms of power consumption and size of implementation are clearly evident. The approach attempts to benefit from the advantages of *neuromorphic* vision systems in the form of parallel distributed, low power pixel processing units, while using traditional *engineering* circuit techniques.

The adopted algorithm appears well suited to implementation in analogue VLSI, based on the simulations performed in chapter three. The approach is completely robust to fixed pattern noise, a common problem with CMOS imager arrays, yet is simple enough to allow a realistically sized pixel processing unit. The fundamental frequency maps demonstrate the potential of the algorithm for identifying pixels that contain temporal frequencies of interest, while ignoring those that do not.



The pixel processing units detailed in chapters four and five both accurately extract the fundamental frequency of the incident light intensity. Both techniques consume extremely low bias currents, with the comparator proving the limiting factor. For input frequencies less than approximately 3 kHz, the system can be solely biased in the subthreshold regime, producing current consumption in the low nA range. As the input frequencies increase, the comparator needs larger bias currents to cope with the increased slew rate requirements. The initial approach implemented on test IC one relies on an external reference, supplied to both high pass filter and comparator. Tests discovered that the system is extremely sensitive to this reference voltage, possibly due to attenuation through the high pass filter. An improved approach was conceived and implemented on the second test chip, using a low pass filter to extract the DC level of the photocircuit's output. This was then supplied to the comparator, producing a self-referencing system which could be more realistically implemented in a CMOS image processor. The area of this self-referencing pixel processing unit was minimised by the creation of the *minipix* algorithm, which contains phototransistor, log photoreceptor, low pass filter and comparator in an area of approximately  $60 \mu m^2$ , with a fill factor of 14.1 %. The performance of the *minipix* algorithm was comparable with the previous implementation, proving the potential of the approach for the creation of a CMOS fundamental frequency extraction image processor.

The third and final test chip developed during this research built on the findings from the first two to produce a system capable of tuning a band pass filter onto the fundamental frequency of its input. From the measured test IC results detailed in chapter seven, it is clear that the system performs well. The tuning range of approximately 20 Hz to 10 kHz can be varied depending on system parameters, but is limited at the lower end by the performance of the band pass filter itself. Ideally, the system would be able to tune down to a centre frequency of 1 Hz, but it appears that the BPF is unable to operate at such low frequencies due to the extremely small subthreshold currents required. However, the band pass filter could be replaced with a better version in future implementations, with the only stipulation being a  $0^0$  phase difference corresponding to its centre frequency.

The maximum tuning speed of the algorithm was measured at approximately 9 kHz/s. However, this parameter can be varied by changing the current available to charge or discharge the charge pump's capacitance. If the current is made too large, the system is unable to correctly tune to the correct frequency as the charge pump's output *bounces* around the correct value of BPF variable



control voltage. This suggests some trade-off between the speed of the algorithm's tuning and its corresponding accuracy. As suggested in chapter seven, the best approach may be a compromise between the two, utilising some form of dynamic charge pump biasing technique. By linking the charging or discharging current to the width of the system's 'up' or 'down' control pulses, the system will benefit from increased speed initially, giving way to increased accuracy as the filter *fine-tunes* to the desired centre frequency.

The accuracy of the system is excellent, with a tuning error of no more than 3 % of the desired centre frequency. Despite the subthreshold biasing of much of the analogue circuitry, the employed direct tuning method allow the band pass filter to be tuned with high accuracy.

The power consumption of the automatically tuned band pass filter algorithm was simulated for two different target frequencies. The system consumes approximately  $4.5 \mu A$  when tuning to 250 Hz, increasing to  $13.2 \mu A$  at 1 kHz, resulting in a power consumption of  $22.5 \mu W$  and  $66 \mu W$  respectively when operating from a 5 V supply. This current is the combined value consumed from both analogue and digital supplies, as highlighted in figure 7.16. As the analogue circuits are biased in the subthreshold regime, the current from the analogue supply is negligible compared to the digital equivalent. For example, when tuning to 1 kHz, the current drawn from the analogue supply increases to a maximum of approximately 100 nA as the filter is tuned. It is clear from the lower plot in figure 7.16 that the digital current is dominated by switching currents, caused primarily by the comparators and digital phase detector. As such, it can be concluded that the employed technique to limit switching currents in the phase detector is relatively unsuccessful. It is clear from the simulation results that the quoted values of current consumption are dominated by the digital switching currents. Therefore, future implementations may benefit from the inclusion of an analogue phase detector circuit.

The physical size of the algorithm including support circuitry and test structures is approximately  $1350 \mu m$  by  $1180 \mu m$  when implemented in a  $0.6 \mu m$  process, with the dimensions of the band pass filter dominant. An estimate of the area consumed just by the algorithm itself is closer to  $1000 \mu m$  by  $600 \mu m$ , which is still larger than could be feasibly be included in a dedicated CMOS image processor. Despite this, it is clear from the physical layout of the algorithm depicted in figure 7.15 that little effort was made to minimise the area of implementation, with a heavy emphasis on proof of concept. As such, the area could be considerably reduced in future implementations, particularly if fewer guard structures are employed and the band pass filter is implemented with smaller capacitors. Despite the large implementation area, it can be



concluded that the phase derived feedback algorithm successfully tunes a band pass filter to the fundamental frequency of the incident light intensity.

### 8.3 Contributions

The novelty in this thesis stems from the underlying subject matter, extracting temporal frequency signatures from visual data. More precisely, the algorithm's investigated in chapter three are novel with regards to their application in this research. In particular, the *flashing pixel* algorithm and the two subsequent versions that it inspired, the *half Laplacian HPF* and *no mask* algorithms are original.

The subsequent realisation of the *no mask* algorithm in analogue VLSI is novel, culminating in the creation of the *minipix* algorithm. Finally, the phase-derived feedback algorithm developed to automatically tune a BPF to the fundamental frequency of the input signal is original.

### 8.4 Critical Evaluation

The adopted approach of analogue, focal plane processing was decided upon by the requirements of the sponsor company. The need for low power processing led to an emphasis on transistors operating in the weak inversion regime. From a system perspective, a major advantage of the employed *pseudo-Fourier* algorithm is the fact that the parallel processing capabilities of the employed focal plane processing techniques minimise the potential problems of subthreshold current mismatch. The strength of the algorithm lies in each pixel operating as an independent fundamental frequency extraction unit. A common problem with biasing analogue circuits in weak inversion is mismatch between subthreshold currents. Research has shown that the variation can be as high as 20 % depending on device dimensions[94]. Circuits biased in subthreshold are also strongly dependent on ambient temperature variations. The algorithm presented here was developed to reduce the impact of such variations, by not relying on well matched current ratios. With each pixel acting independently, the need for strongly correlated subthreshold currents is reduced. The subthreshold current mismatch simulations performed on the *minipix* algorithm prove the robustness of the approach to variation in low pass filter cutoff frequency variation. Similarly, the direct frequency tuning applied to the band pass filter provides high filter accuracy of within 3 % of the desired centre frequency, despite subthreshold



biasing. By developing custom system level algorithms, it is possible to minimise the potential inadequacies of analogue signal processing when biased in the subthreshold regime.

By utilising transistors biased in the weak inversion region of operation, a considerable saving in power consumption is achieved. As previously stated, it is estimated that the *minipix* algorithm consumes a mere 14 nA when operating at 1 kHz, resulting in a power consumption of 70 nW when operating from a 5 V supply. Similarly, the analogue processing blocks of the phase derived feedback algorithm are biased with subthreshold currents, producing simulated average current consumptions of approximately 60 nA when tuning to 1 kHz, with a peak current of 100 nA. The digital switching currents increase this average considerably, as highlighted in tables 7.5 and 7.6.

This increase in the power consumption caused by digital switching currents is one area that could be improved upon. The fact that the system converts the BPF's input and output signals into pulse trains, which are subsequently compared for phase difference, means that the system will always produce such switching currents. The reason for producing the pulse trains is the ease with which any phase difference can be measured, producing a relatively robust system. However, it may be possible to produce a fully analogue system, less reliant on transistors switching between the power rails.

Another potential weakness of both the *minipix* and automatically tuning BPF algorithms is the choice of a logarithmic compression photocircuit as the means of converting the photocurrent to a voltage signal. The major difficulty with circuit level image processing algorithms is the extremely wide range of possible inputs, from bright sunlight to almost complete darkness. Such conditions place great strains on the circuitry employed to condition the photocurrent into a voltage signal that can realistically be processed by the subsequent stages. An industry standard CMOS pixel uses a variable integration period to allow for this input range, but the temporal aspects of the light intensity variation are compromised. The logarithmic photoreceptor operates by biasing diode connected load transistors in the weak inversion region of operation with the photocurrent, producing a voltage that is logarithmically compressed. This has the advantage of *shrinking* the huge input range of photocurrents into a more manageable range of output voltages, as highlighted by the simulated DC transfer characteristics in figure 4.3. The circuit was adopted in this research because of this property, combined with its simplicity and relatively compact size. However, it is severely limited in certain crucial aspects. The fact that the circuit logarithmically compresses the input signal means that large transient



input swings may become distorted. The ultimate aim is to analyse the frequency content of the temporal intensity variations, which becomes almost impossible if the photocircuit itself introduces distortion. The logarithmic compression may also result in the converse situation of small transient changes being *missed*, as the change in output voltage is too small to be detected. It is clear from the measured test IC results in chapter five that the self referencing pixel processing unit works better for some LED control voltages than others, probably due to the poor performance of the employed photocircuit. The logarithmic photocircuit also suffers from limited bandwidth, as highlighted in chapter four. Despite its strengths, all these reasons suggest that the employed photocircuit could be improved. A potential candidate is the adaptive photoreceptor developed by Delbruck[12], which is described in chapter two. However, it is interesting that Kramer et al[75] who employed Delbruck's photoreceptor as the input for their token based motion detection algorithm suggest that the major limitation of such circuitry is the difficulty in detecting temporal tokens over a wide range of input illumination. It is clear that the problems of creating continuous time circuitry to accurately condition photocurrent into a corresponding voltage are yet to be met.

## 8.5 Future Work

The ultimate aim of the project is to produce a dedicated, low-power image processor capable of extracting the fundamental frequency of the temporal light intensity variation, together with the relative strength of the first four harmonics. The circuitry developed so far extracts only the fundamental frequency, in the form of the output of the automatically tuned BPF. By simply integrating this signal, it should be possible to extract information regarding the energy present in this particular frequency band.

Shifting the BPF to tune to integer multiples of the fundamental frequency is a more complex problem. From a system perspective, there are two potential techniques for achieving this. The first involves implementing a separate band pass filter for each of the required harmonics, all tuned by a single charge pump. By ratioing the bias currents of the OTAs in each BPF, it should be possible to position each filter on a different integer multiple of the fundamental. Providing the BPFs operate in the subthreshold region of operation, the linear relationship between bias current and frequency should allow such a system to operate successfully. Potential advantages of this approach include the fact that each component of the frequency signature will be available at the same time, in the form of the output from each band pass filter. However,



the required implementation area of such an approach may limit its feasible integration into a dedicated image processor. Another potential problem with such an approach is the accuracy with which the filters would be positioned in the frequency domain. As reported, subthreshold mismatch is such that an indirect tuning method such as this may exhibit large variations in bias currents, resulting in poorly positioned filters.

A better approach involves simply tuning a single band pass filter to different multiples of the input frequency. This could be achieved by multiplying the frequency at some point in the phase-derived feedback algorithm. Possibilities include placing an analogue multiplier connected as a frequency doubler[107] before the band pass filter. In keeping with the low power requirements of the project, this multiplier could potentially be biased with subthreshold currents. However, the multiplier may introduce harmonic distortion which could corrupt the resultant frequency signature. Another approach involves doubling the frequency of one of the comparators' output pulse trains, forcing the system to tune to twice the input. Both approaches benefit from a reduction in area, coupled with an increase in accuracy as the band pass filter is tuned directly. However, a major disadvantage is that the system could only place the band pass filter at twice, four times, eight times etc the fundamental frequency, thus not performing the required Fourier decomposition of the input signal. This may be sufficient for certain applications, but remains a trade-off between processing power and circuit-level complexity. Such a system will also produce a delay in the calculation of the frequency signature, with the band pass filter tuned to a particular frequency before 'jumping' to double that frequency.

Another area for future consideration is the implementation area of the automatically tuned BPF algorithm. As previously mentioned, the total area including support and test circuitry is approximately  $1350\ \mu m$  by  $1180\ \mu m$  when implemented in a  $0.6\ \mu m$  process. However, this includes a second BPF together with output buffer circuits, which are included only for test purposes. An estimate of the area consumed by essential circuitry is approximately  $1000\ \mu m$  by  $600\ \mu m$ . Little effort was made to minimise the area, with guard structures and large gaps evident in the layout in figure 7.15. The ultimate aim for such a processor, including circuitry to facilitate harmonic tuning, is its inclusion in a dedicated CMOS frequency signature extraction image-processor. The aim is one processor per column of the pixel array, which will require a significant reduction in implementation area. Nevertheless, it should be possible to achieve such a reduction with better layout techniques and a resizing of certain key elements. For instance, the aspect ratios of the transistors in the OTA are relatively large, in an effort



to improve matching. However, the direct tuning method applied here may compensate for mismatch, allowing the use of minimum sized transistors.

The *minipix* algorithm has proved the potential of a fundamental frequency extraction pixel processing unit with a pitch size conducive to inclusion in an image-processor with reasonable resolution. Further work may involve the creation of an imager capable of producing fundamental frequency maps similar to those created with software in chapter three. By placing an integrator or some form of digital timing circuitry at the side of the minipix array, a DC level corresponding to the fundamental frequency of the visual stimulus may be extracted. Such an imager could be used as an early warning technique for a system containing the automatically tuning BPF algorithm, or as a standalone image processor in its own right.

## 8.6 Final Comments

This thesis has documented research into the design of an image processor capable of extracting frequency signatures from visual data. From its inception, an emphasis on low power, focal-plane processing techniques led the research away from powerful but costly combinations of standard imagers with dedicated DSP, towards analogue signal processing techniques with transistors biased in the weak inversion region of operation. A trade-off between the power of the solution and its corresponding power consumption has resulted in a phase derived feedback algorithm, consuming an estimated average current of  $4.5 \mu A$  when tuning to a 250 Hz input signal. Despite the complexity of the required processing, the measured IC test results appear promising, suggesting that this is the first step in the creation of a low power pseudo-Fourier temporal light intensity image-processor.



---

# Appendix A

## Dyadic Tree Algorithm

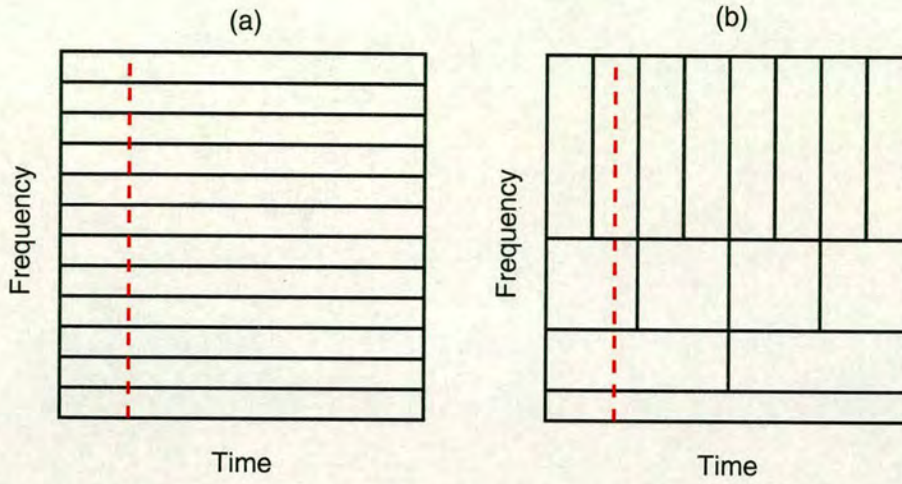
---

The purpose of this section is to explain in detail the dyadic tree algorithm, along with its potential application and subsequent simulation regarding the extraction of temporal frequency signatures from visual data.

### A.1 Wavelet Transforms

Wavelet transforms are similar to Fourier analysis in that the original signal is divided into frequency components using underlying basis functions[112]. In Fourier analysis, the employed basis functions are sines and cosines of different frequencies and different magnitudes, which combine to produce the original signal. Wavelet transforms use more complex basis functions, aimed at improving the performance of signal transformation, particularly regarding signals that exhibit spikes and discontinuities. The underlying idea behind the use of wavelets is to change the *scale* of the basis function, providing more detailed analysis. If a signal is non-periodic, the windowed Fourier transform (WFT) can be employed, where the signal is split into separate sections, termed *windows*, with each section analysed individually for frequency content. This windowing procedure splits the signal into separate time intervals, but, crucially, the size of the window remains constant. This means that there is the danger of too little information for low frequency variations, or too much data for high frequency variations. Of more use would be the ability to *change* the size of the sampling window, so that high and low frequency components can be discerned. The wavelet transform achieves this by using short, high frequency basis functions coupled with long, low frequency ones. The difference between Fourier and wavelet transforms are highlighted in figure A.1(a) and (b) respectively. The Fourier time-frequency map highlights the fact that the same sized window is used for both low and high frequencies, with potential loss of data as a result. The multi-resolution approach of the wavelet transform means that, at the time highlighted in red, there are four differently scaled basis functions, each providing different information about the signal's frequency content.





**Figure A.1:** Time-Frequency Plots for Fourier and Wavelet Transforms: (a) Fourier transform: at the selected time, highlighted in red, each frequency window has the same scale. (b) Wavelet Transform: at the selected time, there are four different windows, each of different size. For low frequency, there are long time windows while for high frequency there are short time windows. Adapted from[113]

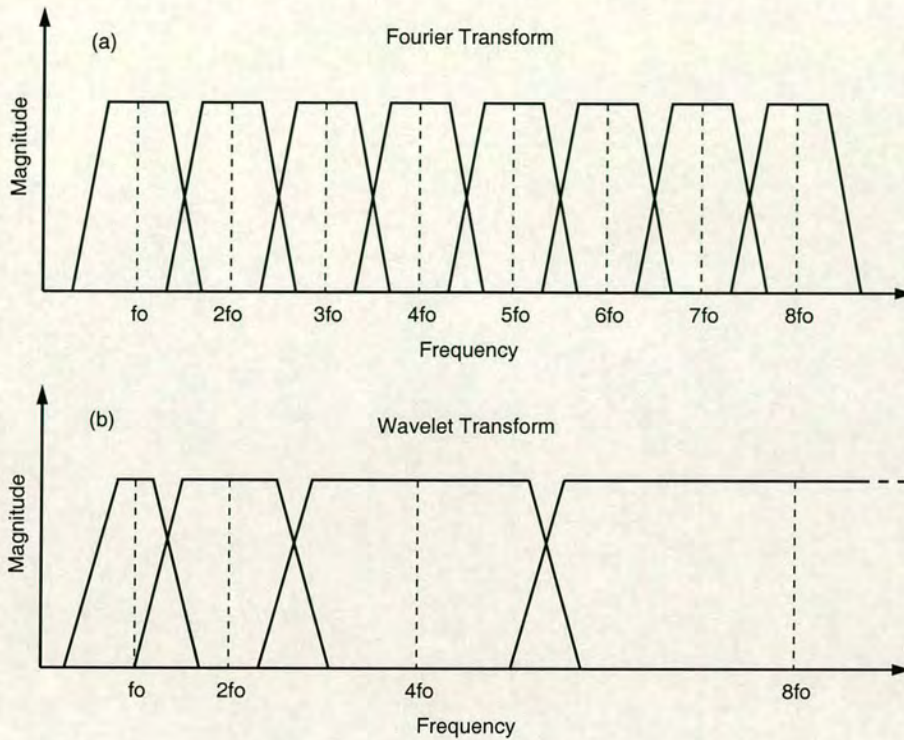
From a circuit level perspective, wavelet transforms can be viewed as a bank of logarithmically placed bandpass filters, dividing the frequency domain into bands. The bandwidth of each filter is proportional to its centre frequency, with the higher the frequency, the wider the filter's response. The Fourier transform can be thought of as a similar dissection of the frequency plane, but each filter has the same bandwidth and is uniformly located on the frequency axis. The differences are highlighted in figure A.2.

Implementations of wavelet transforms using analogue VLSI circuit techniques have been previously attempted, for a number of applications including cochlear sound processing[115], audio frequency decomposition [116–118] and radar analysis[119]. Many rely on switched-capacitor implementations of bandpass filter banks, allowing precise control of the filter time constants.

## A.2 Wavelet Processing with the Dyadic Tree Filterbank

The wavelet filterbank depicted in figure A.2 (b) can be implemented with a dyadic tree filterbank[120]. The technique uses low and high pass filters to segment the input signal into different frequency bins, as depicted in figure A.3. The signal is effectively split in two regarding the





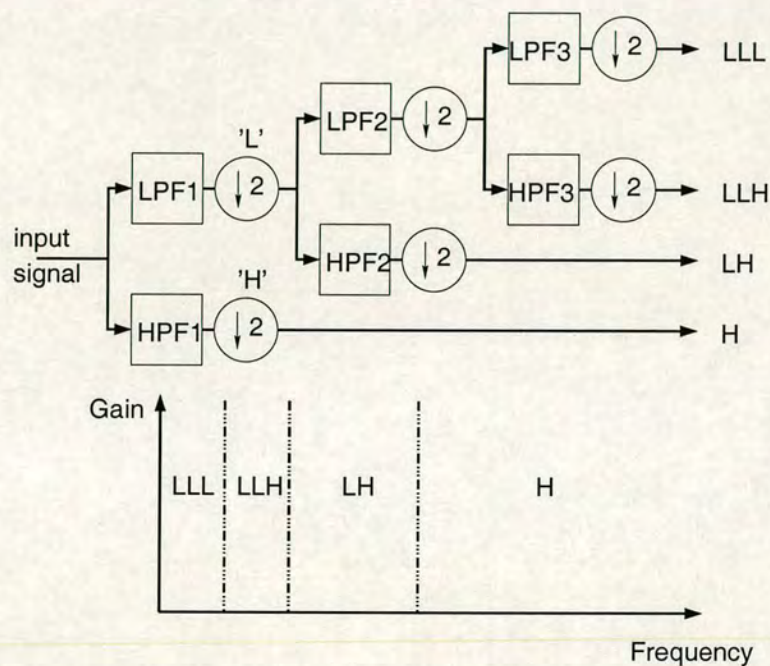
**Figure A.2:** Comparison of Fourier and Wavelet Transform Frequency Domain Division: (a) Fourier: Uses uniformly placed BPFs, with similar bandwidth. (b) Wavelet: Uses logarithmically placed BPFs, whose bandwidth is proportional to the centre frequency. Adapted from [114].

respective frequency content, with the operation then repeated on the low frequency band in the subsequent filtering stages. The three stage dyadic tree depicted here assumes sampled data filters and makes use of downsampling techniques to ease the required filter specifications. By downsampling each signal after a filtering stage, the output signal occupies the same frequency range as the original input, meaning the same filter specification can be used for subsequent filter steps. The differences between a dyadic tree with and without downsampling can be found in figure A.4. Without downsampling, the requirements for the filters in the latter stages of the tree become more and more demanding, resulting in large circuit level implementations.

For the purposes of this research, it was felt that a simple three stage dyadic tree filterbank may be sufficient to tell the difference between objects. Although not providing a true frequency domain representation of the input signal, an estimate of the energy within each of the frequency bands may allow different objects to be distinguished from each other. This trade-off between the power of the achievable processing with the requirements for simple, low power



circuit techniques is a key feature of the research described in this thesis. Subsequent filtering stages could have been added, but the increased silicon area required made this an unrealistic option. Note also that the dyadic tree assumes the use of sampled data filter structures, contrary to the requirements specified by QinetiQ. The purpose of the software simulation phases of the research was to define and then test potential candidate algorithms for the extraction of frequency signatures. As such, the algorithm was tested in software to ascertain its utility in this application. Despite the use of sampled data filtering, it was felt the benefits of the approach in terms of solving the problem outweighed the potential drawbacks.

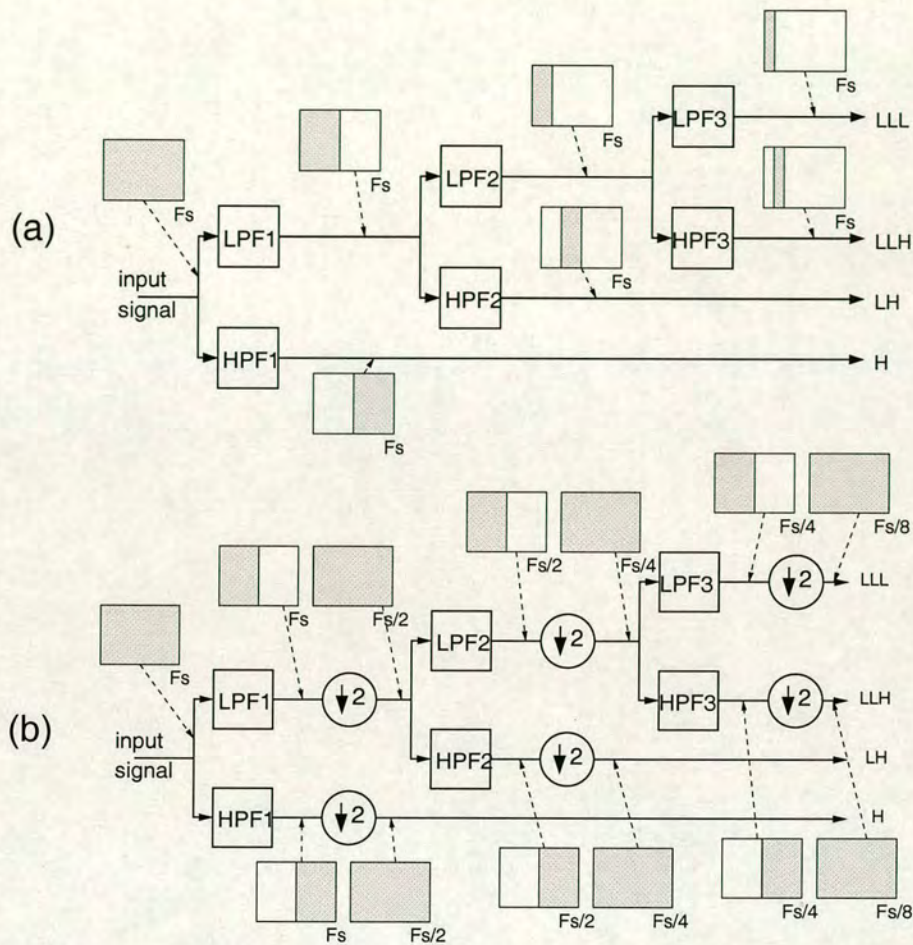


**Figure A.3:** *Dyadic Tree Filterbank: A wavelet style decomposition of the input signal is achieved with a combination of filtering and down-sampling*

### A.3 Software Simulation of the Dyadic Tree Filterbank

The aim of developing a dyadic tree filterbank is to see how it performs in classifying the transient visual stimulæ that appear in the field of vision. To this end, a series of MATLAB simulations were developed to see how powerful the tree filterbank is at distinguishing between different frequency inputs. Simulations were performed using the fan data sets, as they contains both a variable frequency (the luminescence device) and a stationary control frequency (the fan). The adopted approach was to select the same 50 frames from each of the stimulus

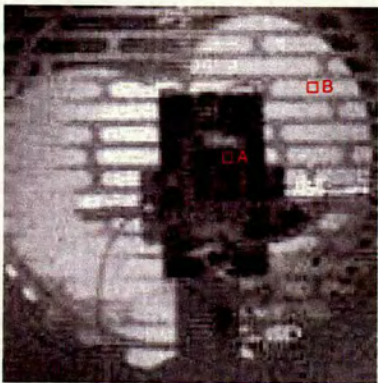




**Figure A.4:** Comparison of Dyadic Tree Filterbank with and without Downsampling: (a) Without downsampling: the latter filtering stages place strict demands on the filtering circuitry, increasing the area and cost of implementation. (b) With downsampling: The effective frequency range after each downsampling stage is halved, allowing the same filter specification to be used for each stage.

files and use this value as a pseudo sampling frequency for the simulations. Due to the difficulties in linking this computer simulation to real time, all frequency values are measured in so-called 'pseudo-frequency'. This corresponds to the number of repetitions of the intensity waveform within the 50 frames subset. Table A.1 shows the translation from real frequency to pseudo-frequency. The pseudo-frequency values are approximate and were obtained from visual inspection of the intensity waveforms. Figure A.5 shows the two pixels that were observed as the test stimulus advances in 'pseudo-time', from the first frame to the fiftieth. Pixel A corresponds to the negative luminescence device, while Pixel B represents the change in intensity caused by the fan.





**Figure A.5:** *Selected Pixels from the 'Fan' Data Sequence used to Test the Dyadic Tree Algorithm: (A) corresponds to the negative luminescence device, while (B) represents the fan itself.*

Test Stimulus	Frequency of Luminescence	Pseudo-Frequency
TEST10	10 Hz	1 units
TEST20	20 Hz	2 units
TEST30	30 Hz	3 units
TEST40	40 Hz	4 units
TEST50	50 Hz	5 units
TEST70	70 Hz	7 units
TEST90	90 Hz	9 units

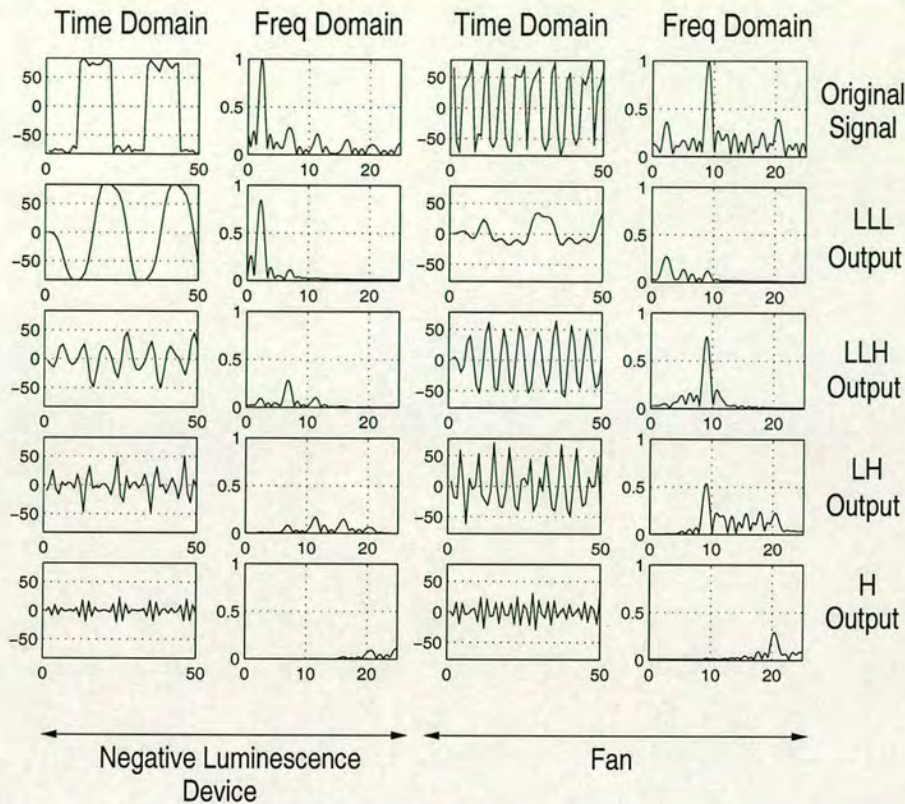
**Table A.1:** *Mapping of Pseudo-Frequencies to Real Frequencies*

For the purposes of this simulation the maximum possible pseudo input frequency to avoid aliasing was 25 units, corresponding to a real frequency of 250 Hz, half the camera’s sampling frequency. A three stage dyadic tree was implemented, with first stage cutoff frequency 20 units, second stage cutoff 10 units and final stage 5 units. A ‘loose’ range of filter cutoff frequencies was deliberately chosen, as this best reflects the likely eventuality in real-life applications. To ease the complexity of the code, downsampling was not implemented as it does not effect the simulation results, merely the ease of circuit-level implementation.

### A.3.1 Dyadic Tree Sim Results: Luminescence *Flashing* at 20 Hz.

The results of applying the dyadic tree to the fan data sequence containing a luminescence flashing frequency of 20 Hz can be seen in figure A.6. The first row corresponds to the ori-





**Figure A.6:** *Selected Time and Frequency Domain Signals from Three Stage Dyadic Tree Simulation when Tested with the 20 Hz Negative Luminescence Device: The original input and each of the four dyadic tree outputs are depicted for both pixels A and B, highlighting the decomposition of the signal into frequency bands.*

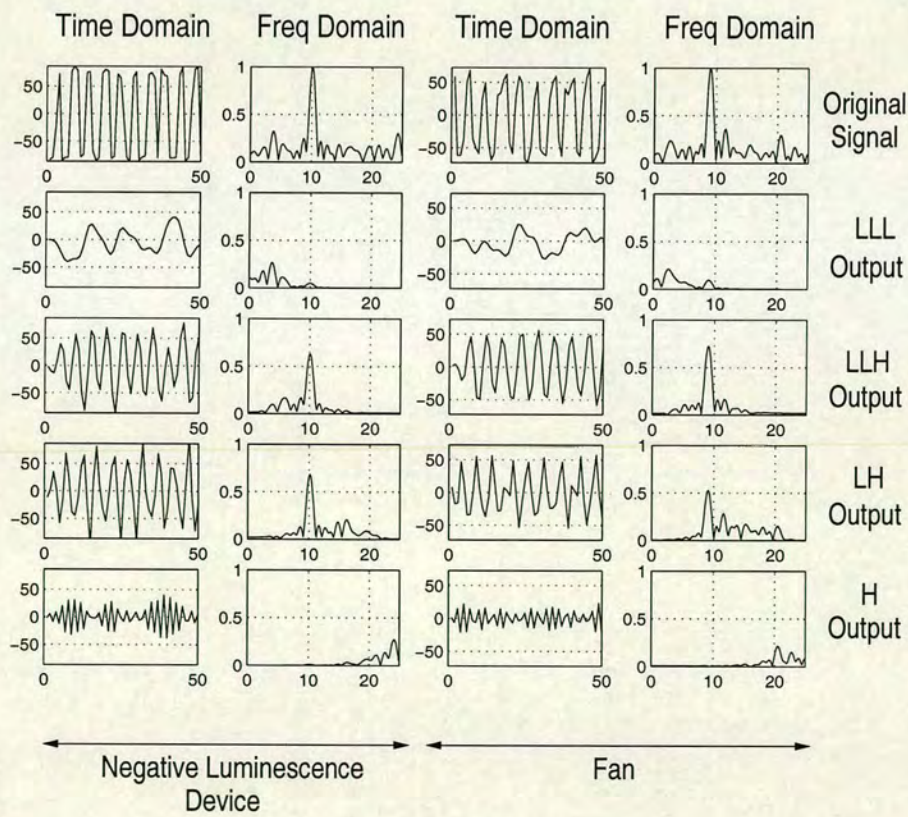
ginal, unfiltered input signal formed from the intensity change through time of the test stimulus. Columns one and two show the time domain and frequency domain representations of the negative luminescence device (pixel A in figure A.5), whilst columns three and four show the same for the fan (pixel B). The input from the negative luminescence device appears to be a square wave with a pseudo-frequency of 2 units. The frequency domain representation agrees with this, showing a fundamental at 2 frequency units and then every odd harmonic as would be expected from a square wave. The fan data appears more random, at a frequency of roughly 9 units and a first harmonic at about 20. The second row corresponds to the LLL output from the dyadic tree. Notice that the filtering has isolated the fundamental frequency of the negative luminescence input signal, with the frequency domain dominated by the two frequency unit pulse. The fan produces little at the LLL output, although the low frequency pulse shown on the original frequency domain trace does appear. As we advance down the rows, more and more high frequency detail is added until the final row, corresponding to the H output. The



frequency domain representation of the transient output signals highlights the splitting of the frequency content into different frequency bands. Taking the negative luminescence device as an example, row two (LLL) shows only the fundamental at the output. Row three (LLH) shows the third harmonic and an attenuated version of the fifth harmonic, with row four (LH) depicting the fifth and seventh harmonics. The high frequency output band in row five shows only the ninth and highest frequency harmonic. It is this decomposition of the transient input signal that may provide a solution to the problem of classification.

A.3.2 Dyadic Tree Sim Results: Luminescence *Flashing* at 90 Hz.

A similar plot for a negative luminescence frequency of 90 Hz is shown in figure A.7. This time, the frequency of the luminescence device is similar to the fan at roughly 9 units.



**Figure A.7:** Selected Time and Frequency Domain Signals from Three Stage Dyadic Tree Simulation when Tested with the 90 Hz Negative Luminescence Device: Once again, the input signal and four dyadic output signals are depicted.

The outputs from the dyadic tree show similar traces for both luminescence device and fan,



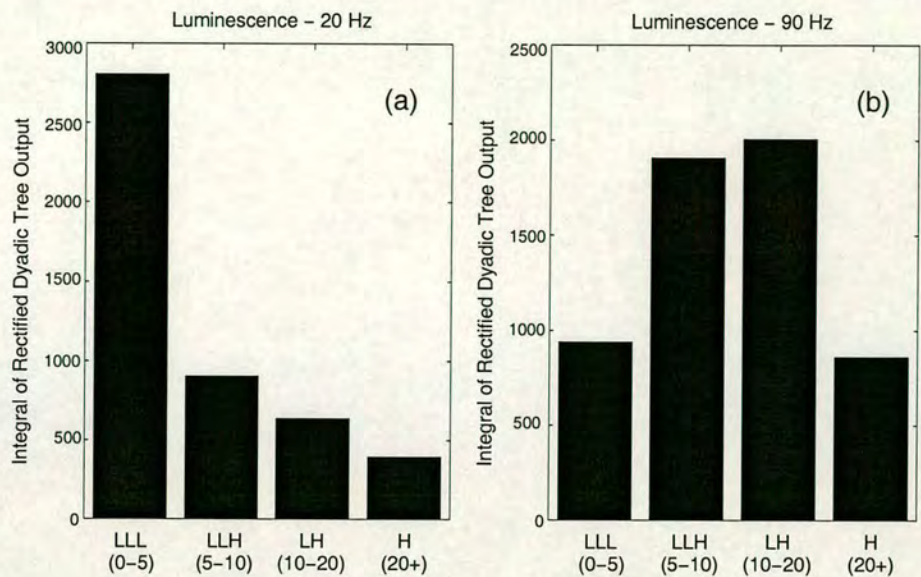
resulting in a truer test of the algorithm than figure A.6. With two inputs at similar frequencies, it is necessary to look at the harmonic content of the inputs if a positive identification is to be made.

### A.3.3 Dyadic Tree Simulations: Frequency Band Energy Content for Luminescence Device *Flashing* at 20 Hz and 90 Hz

Despite showing clearly how the original signals are split into different frequency bands, the simulation results in figures A.6 and A.7 do not allow clear discrimination between the fan and the negative luminescence device. To achieve this end, an estimate of the energy within each of the dyadic tree's outputs could be calculated, effectively producing a frequency signature. A simple way of doing this within the simulation environment is to calculate the integral of the rectified transient signal in each band. This was done for the input data described earlier, with the negative luminescence device at 20 Hz and 90 Hz. Figure A.8 (a) shows the 20 Hz bar chart, whilst (b) shows the values for the luminescence flashing at 90 Hz. As expected, the majority of the energy for the 20 Hz stimulus appears in the LLL band as this contains pseudo-frequencies zero to five units. The energy levels drop as we proceed through the bands, with the H band showing there is very little high frequency information in the signal. The 90 Hz signal corresponds to a pseudo-frequency of roughly 9 units. Figure A.8 (b) shows the energy levels for this signal, with the majority appearing in the LLH and LH bands. The fact that the 20 Hz and 90 Hz signals produce different energy signatures allows us to differentiate between the two transient inputs.

Although the method produces different integral values for different frequency inputs, another consideration is how it performs for similar inputs. In each stimulus file, the fan rotates at a constant frequency, corresponding to about 9 pseudo-frequency units. The bar charts for the fan should be similar no matter which input stimulus is used. Figure A.9 shows the outputs for the test stimulus corresponding to negative luminescence frequencies 20 Hz and 90 Hz. As expected, a comparison with the energy bands from figure A.8, shows that the fan produces a similar energy signature regardless of which test stimulus is used. It would appear that this algorithm has the ability to 'recognise' the fan whilst still providing different energy bands for the luminescence device. It is this talent that may allow us to classify particular visual stimuli from the transient signals that they produce.



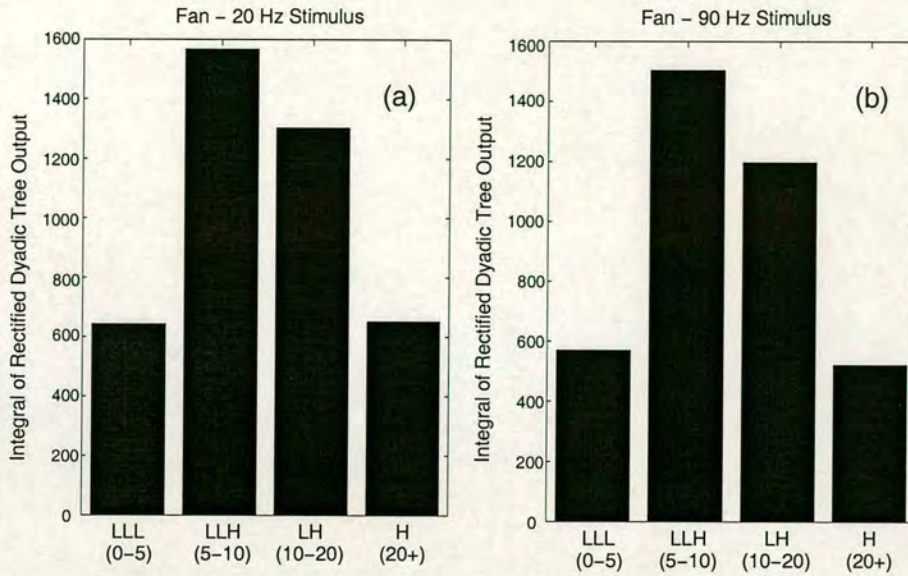


**Figure A.8:** *Energy within Dyadic Tree Output Bands for Luminescence Flashing at: (a) 20Hz, (b) 90Hz. The difference in the frequency signatures allows clear discrimination between the two.*

### A.3.4 Dyadic Tree Simulations: Frequency Band Energy Content for All Available Luminescence Device Frequencies, 10 Hz to 90 Hz

To further test the algorithm, the process was repeated with all the available stimulae, with the results depicted in figure A.10. This shows the integrated values of the dyadic tree outputs for all of the 'fan' test sequences that were supplied. It should be stressed that it is only the luminescence device that varies in frequency, from 10 Hz to 90 Hz. The fan that appears in all of the stimulae rotates with a constant frequency. This means that the outputs in figure A.10 (b) should appear constant whereas those in figure A.10 (a) should reflect the changing nature of the input frequency. The general trend highlighted in figure A.10 (a) and (b) shows that it is possible to differentiate between transient signals at different frequencies with a dyadic tree structure. By sampling the transient signals on the focal plane, AC-coupling them to remove the DC level and then passing them through a dyadic tree, we are provided with a series of transient signals whose frequency contents exist in a particular band. If these signals are then rectified before being integrated in an effort to calculate the energy in each band, it is possible to produce a 'signature' of the input signal. It is then possible to identify one particular signature from another, thus identifying one transient input signal from another.





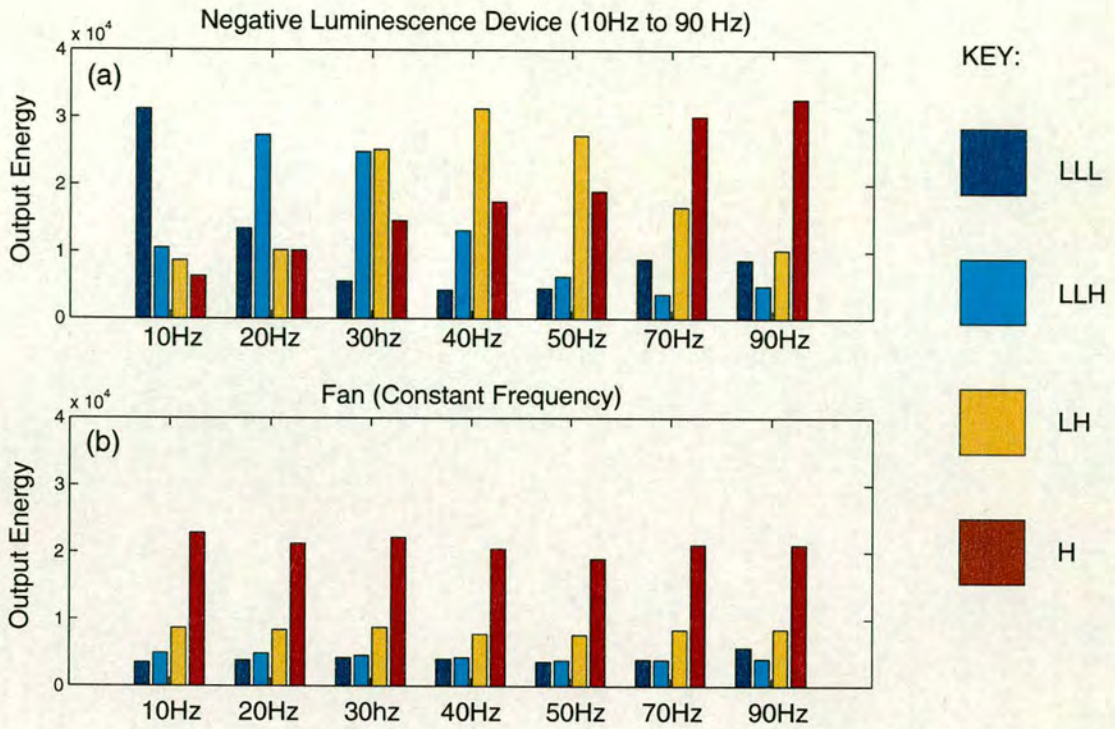
**Figure A.9:** Energy within Dyadic Tree Output Bands for Fan at a Luminescence Frequency of: (a) 20Hz, (b) 90Hz. The fan's frequency remains constant throughout the different test stimulus, reflected in the similarity of the frequency signatures.

#### A.4 Comments on the Dyadic Tree Algorithm

At first glance, the results in figure A.10 appear promising. It is clear from (a) that the different frequencies of the negative luminescence device produce different frequency signatures, allowing successful discrimination. However, the results from figure A.10 (b), showing the frequency signatures for the fan, are less impressive. The fan acts as a control frequency for the series of seven 'fan' data sequences. As such, its frequency signature should remain constant throughout, despite the change in frequency of the luminescence device. While the fan's frequency signatures appear similar, there is still considerable variation in the energy at each dyadic tree output, particularly in the higher frequency bands. This is probably due to the lack of filter *resolution* in the dyadic tree filterbank, with only the lower frequency sections subject to more filtering stages. The variation in frequency signature for the fan data suggests a limitation to the usefulness of the device for this application. If the same object produce different frequency signatures then any attempt at classification becomes extremely difficult.

In an effort to improve the high frequency resolution of the system, a *full-tree* filterbank was implemented. This is similar to the dyadic tree in figure A.3 but has both halves of the tree expanded, such that the frequency domain is split into equal bands. In effect, this implements a simple Fourier processor, providing higher resolution for the high frequency bands, at the cost



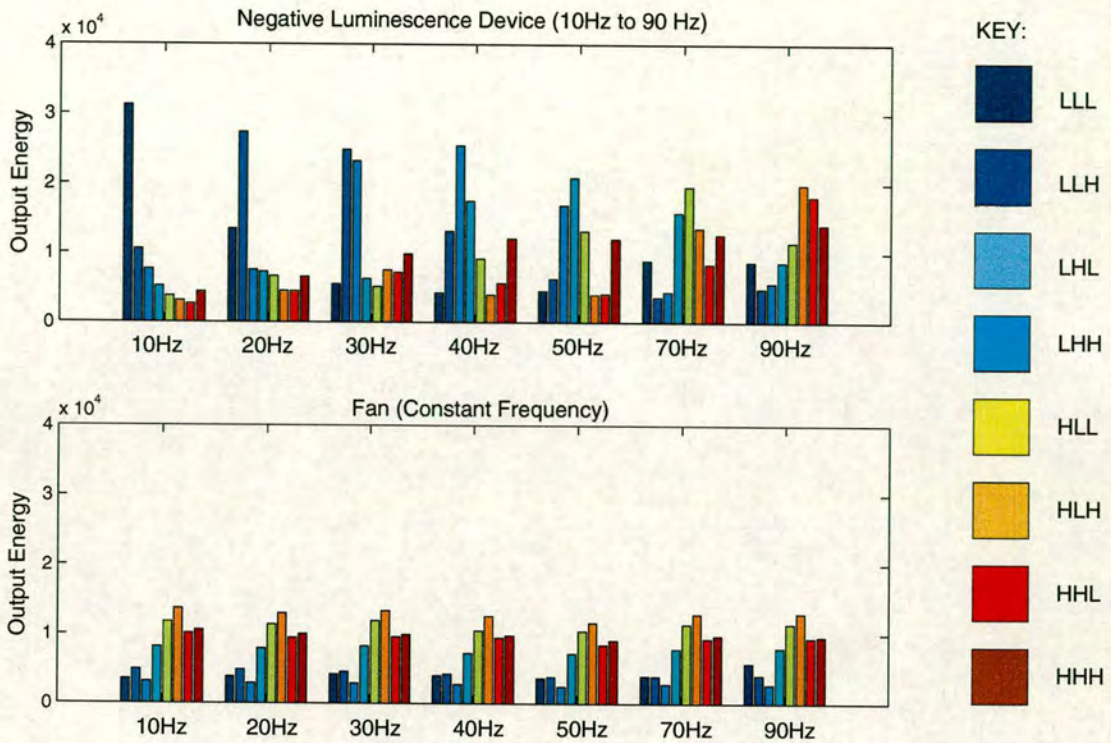


**Figure A.10:** Comparison of Dyadic Tree Output Integrals: (a) Luminescence Device, (b) Fan. The frequency signature from the fan remains relatively constant, while that of the negative luminescence device varies as expected.

of more complex processing. The same simulation setup as was used to test the dyadic tree was applied to the full tree filterbank, with the results depicted in figure A.11. The results are an improvement, but the variation in fan frequency signature is still evident. It was felt that any improvement in performance was offset by the increased size of the filterbank, particularly regarding a circuit level implementation

Another area for consideration is the frequency range of the input signals as well as the minimum frequency difference between two similar visual stimulæ. If it is the case that many stimulæ exhibit similar frequency characteristics, it will be necessary to have a tree with close cutoff frequencies in order to successfully differentiate between them. On the other hand, if the stimulæ exist over a very wide range of input frequencies then the tree's cutoffs must be well-spaced. A possible way round this problem is to construct a tunable dyadic tree that can have variable cutoff frequencies. This would allow both widely and narrowly spaced stimulæ to be detected, although at a cost of higher complexity and silicon area. Other possibilities include increasing the number of stages of the tree, from three to four or five levels. This would increase





**Figure A.11:** Comparison of Full Tree Output Integrals: (a) Luminescence Device, (b) Fan. With higher resolution in the upper frequency bands, it was hoped the full tree would improve on the results for the dyadic tree. The different frequencies of the luminescence device produce differing frequency signatures, but the fan also shows slight variation.

the resolution of the system but would once again cost extra in terms of silicon implementation.

The fact that the dyadic tree algorithm also relies on sampled data circuit techniques is another potential problem. The sponsor company stressed an emphasis on real-time processing, with the sampling of signals on the focal plane contrary to this requirement. It may be possible to use continuous time circuit techniques to implement the filterbank, but as figure A.4 highlights, the requirements for the latter filter stages are particularly strict. The size of the circuitry required to implement the dyadic tree suggested a processor per column, with each pixel supplying samples in a time-multiplexing system.

For all these reasons, the dyadic tree algorithm was deemed an interesting approach, but ultimately not suitable for this application.



---

## Appendix B

# Analogue Buffer Circuitry

---

In order to 'see' AC signals that are generated within test chips, it is essential to buffer the sensitive nodes before they are attached to external pad circuitry. Pads contain protection diodes that exhibit large capacitances, effectively creating a low pass filter that can severely attenuate or load the signals of interest. As much of the research described in this thesis is concerned with measuring the amplitude of AC signals, this document describes a simple buffer circuit that was employed for all test chips.

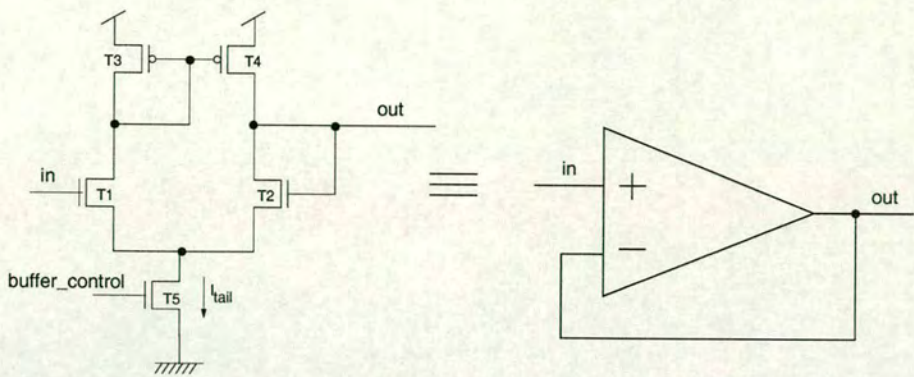
### B.1 Differential Stage

The employed buffer was a simple differential stage, connected as a voltage follower as depicted in figure B.1(a). The frequency response and common mode range for the actual buffer implemented on the chip can be seen in figure B.1(b) and (c) respectively. The effect of buffer control voltage on frequency response can be seen in figure B.1(b), with the cutoff frequency reducing from about 100kHz to 10kHz as the control voltage is reduced from 2.5V to 1V. The passband for all values of control voltages is relatively flat at 0 dB attenuation, suggesting the buffer will be effective in passing the signals of interest in this research. The common mode range of the buffer can be seen in figure B.1(c), with the positive CMR increasing as the bias current is reduced.

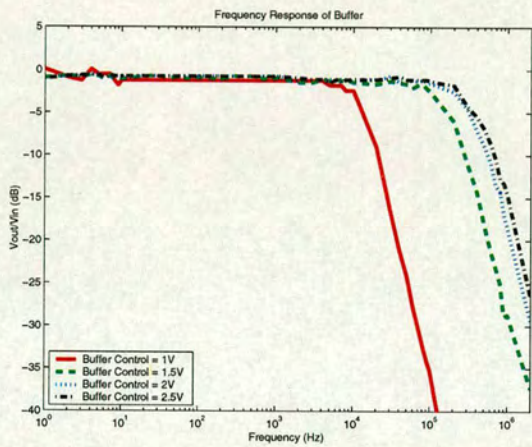
### B.2 DC Offset

The buffer circuit will exhibit some difference in the DC level between its input and output signals, commonly termed the DC offset voltage. This is due to process variations introducing mismatch between the transistors in the circuit. An effort to measure the offset through the buffer circuit when biased with a typical control voltage of 1 V. By ramping the input DC level and measuring the resultant output voltage, the results in figure B.2 were produced. It appears

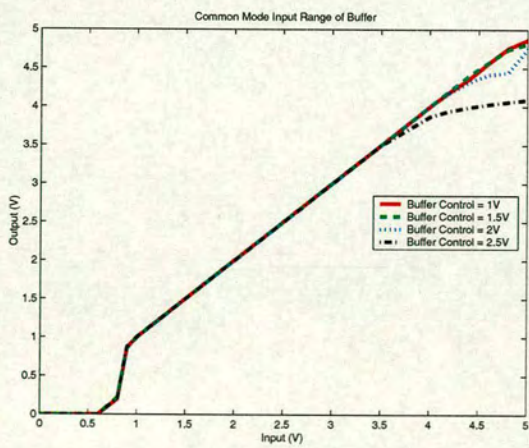




(a) Circuit Topology



(b) Frequency Response

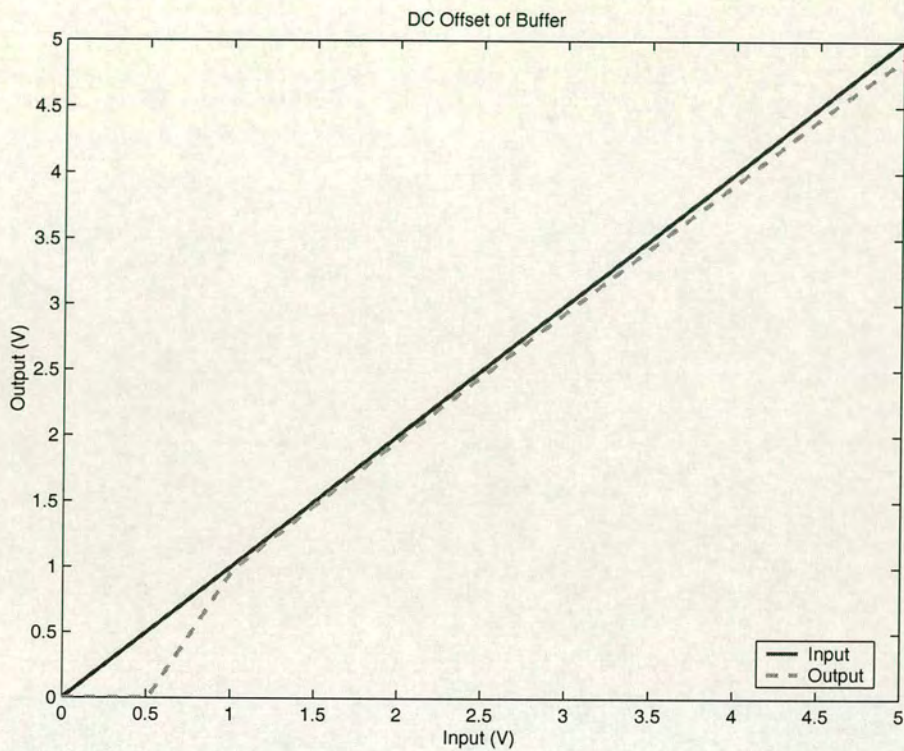


(c) Common Mode Range

**Figure B.1:** CMOS Buffer Implementation: Circuit Topology and Measured IC Test Results



that the buffer's offset increases slightly with the input DC level, from 32mV at 1V to 91mV at 4.5V.



**Figure B.2:** *Measured Test IC Results-DC Offset of the CMOS Buffer*



---

## References

---

- [1] M. Anbar, L. Milesco, A. Naumov, C. Brown, T. Button, C. Carty, and K. AlDulaimi, "Detection of cancerous breasts by dynamic area telethermometry," *IEEE Engineering in Medicine and Biology Magazine*, 2001.
- [2] B. Mulgrew, P. Grant, and J. Thompson, *Digital Signal Processing: Concepts and Applications*. MacMillan Press Ltd., 1999.
- [3] P. O. Pouliquen, A. G. Andreou, and G. Cauwenberghs, "A CMOS Smart Focal-Plane for Infra-red Imagers," in *Proceedings of the 2000 International Symposium on Circuits and Systems*, vol. 4, pp. 329–332, May 2000.
- [4] G. Betta, C. Liguori, A. Paolillo, and A. Pietrosanto, "A DSP-based FFT-Analyzer for the Fault Diagnosis of Rotating Machine Based on Vibration Analysis," in *IEEE Instrumentation and Measurement Technology Conference*, pp. 572–577, May 2001.
- [5] V. Giurgiutiu, A. Cuc, and P. Goodman, "Review of Vibration-Based Helicopters Health and Usage Monitoring Methods," in *55th Meeting of Society for Machinery Failure Prevention Technology*, 2001.
- [6] A. Moini, *Vision Chips*. Kluwer Academic Publishers, 1999.
- [7] E. R. Fossum, "CMOS Image Sensors: Electronic Camera-On-A-Chip," *IEEE Transactions on Electron Devices*, vol. 44, pp. 1689–1698, October 1997.
- [8] G. Chapinal, S. A. Bota, M. Moreno, J. Palacin, and A. Herms, "A 128 X 128 CMOS Image Sensor With Analog Memory for Synchronous Image Capture," *IEEE Sensors Journal*, vol. 2, pp. 120–127, April 2002.
- [9] C. Koch and B. Mathur, "Neuromorphic vision chips," *IEEE Spectrum*, pp. 38–46, May 1996.
- [10] A. R. Vazquez, T. Roska, and A. Andreou, "Guest Editorial: Special Issue on Bio-Inspired Processors and Cellular Neural Networks for Vision," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 46, pp. 226–228, February 1999.
- [11] V. Brajovic and T. Kanade, "A VLSI Sorting Image Sensor: Global Massively Parallel Intensity-to-Time Processing for Low-Latency Adaptive Vision," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 67–75, February 1999.
- [12] T. Delbruck and C. A. Mead, "Analog VLSI Phototransduction by Continuous-Time, Adaptive, Logarithmic Photoreceptor Circuits," *CNS Memo No. 30*, April 2, 1996.
- [13] C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley Publishing Company, 1989.



- 
- [14] C. A. Mead and M.A.Mahowald, "A Silicon Model of Early Visual Processing," *Neural Networks*, vol. 1, pp. 91–97, 1988.
- [15] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, "Analog VLSI Architectures for Motion Processing: From Fundamental Limits to System Applications," *Proceedings of the IEEE*, vol. 84, pp. 969–87, July 1996.
- [16] C. Mead, "Neuromorphic Electronic Systems," *Proceedings of the IEEE*, vol. 78, pp. 1629–1636, October 1990.
- [17] M. A. C. Maher, S. P. DeWeerth, M. A. Mahowald, and C. A. Mead, "Implementing Neural Architectures Using Analog VLSI Circuits," *IEEE Transactions on Circuits and Systems*, vol. 36, pp. 643–652, May 1989.
- [18] A. G. Andreou, R. C. Meitzler, K. Strohbehn, and K. A. Boahen, "Analog VLSI Neuromorphic Image Acquisition and Pre-processing Systems," *Neural Networks*, vol. 8, no. 7/8, pp. 1323–1347, 1995.
- [19] E. A. Vittoz, "Analog VLSI Signal Processing: Why, Where and How?," *Analog Integrated Circuits and Signal Processing*, pp. 27–44, July 1994.
- [20] A. Watson, "Why cant a computer be more like a brain?," *Science*, vol. 277, pp. 1934–36, Sept 1997.
- [21] C. Diorio, D. Hsu, and M. Figueroa, "Adaptive CMOS: From Biological Inspiration to Systems-on-a-Chip," *Proceedings of the IEEE*, vol. 90, pp. 345–357, March 2002.
- [22] R. R. Harrison, P. Hasler, and B. A. Minch, "Floating-Gate CMOS Analog Memory Cell Array," in *Proceedings of the 1998 International Symposium on Circuits and Systems*, vol. 2, pp. 204–207, 1998.
- [23] W. P. Millard, Z. K. Kalayjian, and A. G. Andreou, "Calibration and Matching of Floating Gate Devices," in *Proceedings of the 2000 International Symposium on Circuits and Systems*, vol. 4, pp. 389–392, 2000.
- [24] X. Arreguit, F. A. van Shaik, F. Bauduin, M. Bidiville, and E. Raeber, "A CMOS Motion Detector System for Pointing Devices," *IEEE Journal of Solid-State Circuits*, vol. 31, pp. 1916–1921, December 1996.
- [25] J. Giles, "Think like a bee," *Nature*, vol. 410, pp. 510–512, March 2001.
- [26] D. Marr and E. Hildreth, "Theory of Edge Detection," *Proceedings of the Royal Society of London*, vol. 207, pp. 187–217, 1980.
- [27] C. Mead, *Analog VLSI Implementation of Neural Systems*, ch. 10: Adaptive Retina, pp. 239–246. Kluwer Academic Publishers, 1989.
- [28] K. A. Boahen and A. G. Andreou, "A Contrast Sensitive Silicon retina with Reciprocal Synapses," in *Advances in Neural Information Processing 4*, vol. 4, pp. 764–72, 1991.
- [29] A. G. Andreou and K. A. Boahen, "A 48,000 pixel, 590,000 transistor silicon retina in current-mode subthreshold CMOS," in *Proc. of the 37th Midwest Symposium on Circuits and Systems*, pp. 97–102, 1995.



- [30] C.-Y. Wu and C.-F. Chiu, "A New Structure of the 2-D Silicon Retina," *IEEE Journal of Solid-State Circuits*, vol. 30, pp. 890–97, August 1995.
- [31] C.-Y. Wu and C.-F. Chiu, "A New Structure for the Silicon Retina," in *Int. Electron Devices Meeting, Technical Digest*, 1992.
- [32] C.-Y. Wu and H.-C. Jiang, "An improved BJT-Based Silicon Retina with Tunable Image Smoothing Capability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, pp. 241–48, June 1999.
- [33] H. Ikeda, K. Tsuji, T. Asai, H. Yonezu, and J.-K. Shin, "A Novel Retina Chip with Simple Wiring for Edge Extraction," *IEEE Photonics Technology Letters*, vol. 10, pp. 100–104, 1998.
- [34] H. Ikeda, K. Tsuji, T. Asai, H. Yonezu, and J.-K. Shin, "An Adaptive Silicon Retina performing an edge Extraction with a MOS-type Spatial Wiring and Smart Pixel Circuits," in *1997 Int. Conference on Solid State Devices and Materials*, pp. 386–87, 1997.
- [35] T. Yagi, Y. Hayashida, and S. Kameda, "An Analog VLSI Which Emulates Biological Vision," in *Proceedings of the 1998 Second International Conference on Knowledge-Based Intelligent Electronic Systems*, vol. 3, pp. 454–460, 1998.
- [36] T. Yagi, S. Kameda, Y. Hayashida, and L. Li, "An artificial retina with adaptive mechanisms and its application to retinal prosthesis," in *Proceedings of the 1999 International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 418–423, 1999.
- [37] S. Kameda and T. Yagi, "An Analog Vision Chip Applicable to Real-Time Image Processings in Indoor Illumination," in *Proceedings of the 2000 International Joint Conference on Neural Networks*, vol. 3, pp. 489–493, 2000.
- [38] H. Kobayashi, T. Matsumoto, T. Yagi, and K. Tanaka, "Light-Adaptive Architectures for Regularization Vision Chips," *Neural Networks*, vol. 8, no. 1, pp. 87–101, 1995.
- [39] H. Kobayashi, J. L. White, and A. A. Abidi, "An Active Resistor Network for Gaussian Filtering of Images," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 738–748, May 1991.
- [40] J. G. Harris, C. Koch, and J. Luo, "A Two-Dimensional Analog VLSI Circuit for Detecting Discontinuities in Early Vision," *Science*, vol. 248, pp. 1209–1211, June 1990.
- [41] C. Koch, W. Blair, J. G. Harris, T. Horiuchi, A. Hsu, and J. Luo, "Real-Time Computer Vision and Robotics Using Analog VLSI Circuits," in *Advances in Neural Information Processing Systems*, vol. 2, pp. 750–757, 1990.
- [42] M. Barbaro, P.-Y. Burgi, A. Mortara, P. Nussbaum, and F. Heitger, "A 100 x 100 Pixel Silicon Retina for Gradient Extraction With Steering Filter Capabilities and Temporal Output Coding," *IEEE Journal of Solid State Circuits*, vol. 37, pp. 160–172, February 2002.
- [43] D. L. Standley, "An Object Position and Orientation IC with Embedded Imager," *IEEE Journal of Solid-State Circuits*, vol. 26, pp. 1853–1859, December 1991.



- [44] B. E. Shi, "A Low-Power Orientation-Selective Vision Sensor," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 47, pp. 435–440, May 2000.
- [45] T. G. Morris, T. K. Horiuchi, and S. P. DeWeerth, "Object-Based Selection Within an Analog VLSI Visula Attention System," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 45, pp. 1564–1572, December 1998.
- [46] C. S. Wilson, T. G. Morris, and S. P. DeWeerth, "A Two-Dimensional, Object-Based Analog VLSI Visual Attention System," in *Proceedings of the 20th Anniversary Conference on Advanced Research VLSI*, pp. 291–308, 1999.
- [47] V. Brajovic and T. Kanade, "Computational Sensor for Visual Tracking with Attention," *IEEE Journal of Solid State Circuits*, vol. 33, pp. 1199–1207, August 1998.
- [48] T. Delbruck, "Silicon Retina for Autofocus," in *Proceedings of the 2000 International Symposium on Circuits and Systems*, vol. 4, pp. 393–396, 2000.
- [49] T. Delbruck, "'Bump' Circuits for Computing Similarity and Dissimilarity of Analog Voltages," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 475–479, 1991.
- [50] V. Brajovic and T. Kanade, "A VLSI Sorting Image Sensor: Global Massively Parallel Intensity-to-Time Processing for Low-Latency Adaptive Vision," *IEEE Transactions on Robotics and Automation*, vol. 15, pp. 67–75, February 1999.
- [51] T. Delbruck, *Investigations of Visual Transduction and Motion Processing*. PhD thesis, Computation and Neural Systems Program, Caltech, 1993.
- [52] S.-C. Liu, "Silicon Retina with Adaptive Filtering Properties," *Analog Integrated Circuits and Signal Processing*, vol. 18, pp. 243–254, 1999.
- [53] S.-C. Liu, "Silicon Retina with Adaptive Filtering Properties," *Advances in Neural Information Processing Systems*, vol. 10, 1998.
- [54] J. Mann, "Implementing early visual processing in analog VLSI: light adaptation," in *Proc. SPIE/SPSE Visual Information Processing: From Neurons to Chips*, vol. 1473, pp. 128–136, 1991.
- [55] J. Kramer, "An On/Off Transient Imager with Event-Driven, Asynchronous Read-Out," in *Proceedings of the 2002 International Symposium on Circuits and Systems*, vol. 2, pp. 165–168, 2002.
- [56] J. Kramer, "An Integrated Optical Transient Sensor," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 49, pp. 612–628, September 2002.
- [57] V. Gruev and R. Etienne-Cummings, "Pipelined Temporal Difference Imager," *Electronics Letters*, vol. 37, pp. 315–317, March 2002.
- [58] A. Gopalan and R. R. Harrison, "A CMOS Imager with On-Chip Temporal Filtering For Motion Pre-Processing," in *Proceedings of the 2002 International Symposium on Circuits and Systems*, 2002.



- [59] A. G. Andreou, K. Strohhahn, and R. Jenkins, "Silicon Retina for Motion Computation," in *Proceedings of the 1991 International Symposium on Circuits and Systems*, vol. 3, pp. 1373–1376, 1991.
- [60] R. C. Meitzler, K. Strohhahn, and A. G. Andreou, "A Silicon Retina for 2-D Position and Motion Computation," in *Proceedings of the 1995 International Symposium on Circuits and Systems*, vol. 3, pp. 2096–2099, 1995.
- [61] T. Horiuchi, J. Lazzaro, A. Moore, and C. Koch, "A Delay-Line Based Motion Detection Chip," in *Advances in Neural Information Processing Systems 3*, vol. 3, pp. 406–412, 1991.
- [62] T. Delbruck, "Silicon Retina with Correlation-Based, Velocity-Tuned Pixels," *IEEE Transaction on Neural Networks*, vol. 4, pp. 529–41, May 1993.
- [63] S.-C. Liu, "Silicon Model of Motion Adaptation in the Fly Visual System," in *Proc. 3rd UCSD - Caltech Symposium on Neural Computation*, June 1996.
- [64] H.-C. Jiang and C.-Y. Wu, "A 2-D Velocity and Direction-Selective Sensor with BJT-Based Silicon Retina and Temporal Zero-Crossing Detector," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 241–47, February 1999.
- [65] C.-Y. Wu and H.-C. Jiang, "An improved BJT-Based Silicon Retina with Tunable Image Smoothing Capability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, pp. 241–48, June 1999.
- [66] R. R. Harrison and C. Koch, "A Robust Analog VLSI Reichardt Motion Sensor," *Analog Integrated Circuits and Signal Processing*, vol. 24, pp. 213–229, 2000.
- [67] R. G. Benson and T. Delbruck, "Direction Selective Silicon Retina that uses Null Inhibition," in *Advances in Neural Information Processing Systems 4*, vol. 4, pp. 756–763, 1992.
- [68] R. R. Etienne-Cummings, S. A. Fernando, and J. V. der Spiegel, "Real-Time 2-D Analog Motion Detector VLSI Circuit," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 4, pp. 426–431, 1992.
- [69] R. Etienne-Cummings, S. Fernando, N. Takahashi, V. Shtonov, and J. V. der Spiegel, "A New Temporal Domain Optical Flow Measurement Technique for Focal Plane VLSI Implementation," in *Proceedings of Computer Architectures for Machine Perception, 1993*, pp. 241–250, 1993.
- [70] R. Etienne-Cummings, J. V. der Spiegel, and P. Mueller, "A Focal Plane Visual Motion Measurement Sensor," *IEEE Transactions on Circuits and Systems -I: Fundamental Theory and Applications*, vol. 44, pp. 55–66, January 1997.
- [71] S.-C. Liu and C. Mead, "Continuous-Time Adaptive Delay System," in *1994 IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 119–22, 1994.
- [72] A. Moini, A. Bouzerdoun, K. Eshraghian, A. Yakovlev, X. T. Nguyen, A. Blanksby, R. Beare, D. Abbott, and R. E. Bogner, "An Insect Vision-Based Motion Detection Chip," *IEEE Journal of Solid State Circuits*, vol. 2, pp. 279–284, February 1997.



- [73] K. Yamada and M. Soga, "A Compact Integrated Visual Motion Sensor for ITS Applications," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000*, pp. 650–655, October 2000.
- [74] J. Kramer, R. Sarpeshkar, and C. Koch, "An Analog VLSI Velocity Sensor," in *Proceedings of the 1995 IEEE International Symposium on Circuits and Systems*, pp. 413–416, 1995.
- [75] J. Kramer, R. Sarpeshkar, and C. Koch, "Pulse-Based Analog VLSI Velocity Sensors," *IEEE Transactions on Circuit and Systems II*, vol. 44, pp. 86–101, February 1997.
- [76] C. M. Higgins and C. Koch, "Analog CMOS Velocity Sensors," in *Proc. Of Electronic Imaging '97 (SPIE Vol. 3019)*, vol. 3019 of *Proceedings of SPIE*, Feb 1997.
- [77] C. M. Higgins, R. A. Deutschmann, and C. Koch, "Pulse-Based 2D Motion Sensors," *IEEE Transactions on Circuits and Systems II*, vol. 46, pp. 677–87, June 1999.
- [78] C. P. Chon, C. A. T. Salama, and K. C. Smith, "Image-Motion Detection Using Analog VLSI," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 93–96, January 1992.
- [79] R. A. Deutschmann and C. Koch, "An Analog VLSI Velocity Sensor using the Gradient Method," in *Proceedings of the 1998 International Symposium on Circuits and Systems*, vol. 6, pp. 649–652, 1998.
- [80] R. E. Cummings, Z. K. Kalayjian, and D. Cai, "A Programmable Focal-Plane MIMD Image Processor Chip," *IEEE Journal of Solid-State Circuits*, vol. 36, pp. 64–73, January 2001.
- [81] V. Gruev and R. Etienne-Cummings, "Implementation of Steerable Spatiotemporal Image Filters on the Focal Plane," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 49, pp. 233–244, April 2002.
- [82] F. Paillet, D. Mercier, and T. M. Bernaud, "Second Generation Programmable Artificial Retina," in *Twelfth Annual IEEE International ASIC/SOC Conference*, pp. 304–09, 1999.
- [83] P. Dudek and P. J. Hicks, "An SIMD Array of Analogue Microprocessors for Early Vision," in *PREP'99 Conference Proceedings, UMIST, Manchester*, pp. 359–362, 1999.
- [84] P. Dudek and P. J. Hicks, "A CMOS General-Purpose Sampled-Data Analog Processing Element," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 47, pp. 467–73, May 2000.
- [85] P. Dudek and P. J. Hicks, "A CMOS General-Purpose Sampled-Data Analogue Microprocessor," in *ISCAS 2000 - IEEE Int. Symposium on Circuits and Systems*, pp. 417–420, May 2000.
- [86] T. Ashley, C. Elliot, N. Gordon, R. Hall, A. Johnson, and G. Pryce, "Negative Luminescence from  $In_{1-x}Al_xSb$  and  $Cd_xHg_{1-x}Te$  diodes," *Infrared Physics and Technology*, vol. 36, pp. 1037–1044, December 1995.
- [87] T. Ashley, C. Elliot, N. Gordon, T. Phillips, and R. Hall, "Applications of Negative Luminescence," *Infrared Physics and Technology*, vol. 38, pp. 145–151, April 1997.



- 
- [88] P. Dudek, *A Programmable Focal-Plane Analogue Processor Array*. PhD thesis, University of Manchester Institute of Science and Technology, 2000.
- [89] K. Singh, "Noise Analysis of a Fully Integrated CMOS Image Sensor," in *IS and T/SPIE Conference on Sensors, Cameras and Applications for Digital Photography*, vol. 3650, pp. 44–51, January 1999.
- [90] A. Hastings, *The Art of Analog Layout*. Prentice Hall, 2001.
- [91] K. R. Laker and W. M. Sansen, *Design of Analog Integrated Circuits and Systems*. McGraw-Hill International Editions, 1994.
- [92] C. Toumazou, C. A. Makris, F. J. Lidgey, and D. G. Haigh, "Towards a New Generation of Analogue IC Design Architectures," in *IEE Colloquium on Analogue IC Design: Obstacles and Opportunities*, pp. 1–16, June 1990.
- [93] C. Toumazou, F. Lidgey, and D. Haigh, eds., *Analogue IC Design: the current mode approach*. Peter Peregrinus Ltd, 1990.
- [94] A. G. Andreou, K. A. Boahen, P. O. Pouliquen, A. Pavasovic, R. E. Jenkins, and K. Strohhahn, "Current-mode Subthreshold MOS Circuits for Analog VLSI Neural Systems," *IEEE Transactions on Neural Networks*, vol. 2, pp. 205–213, March 1991.
- [95] H. Schmid, "Why the Terms 'Current Mode' and 'Voltage Mode' Neither Divide nor Qualify Circuits," in *Proceedings of the 2002 International Symposium on Circuits and Systems*, vol. 2, pp. 29–32, May 2002.
- [96] Z. K. Kalayjian and A. G. Andreou, "Mismatch in Photodiode and Phototransistor Arrays," in *Proceedings of the 2000 International Symposium on Circuits and Systems*, vol. 4, pp. 121–124, May 2000.
- [97] E. Vittoz and J. Fellrath, "CMOS Analog Integrated Circuits Based on Weak Inversion Operation," *IEEE Journal of Solid-State Circuits*, vol. 12, pp. 224–231, June 1977.
- [98] M. Tabet, N. Tu, and R. Hornsey, "Modeling and characterization of logarithmic complementary metal-oxide-semiconductor active pixel sensors," *Journal of Vac. Sci. Technol. A*, vol. 18, pp. 1006–1009, May/Jun 2000.
- [99] Y. P. Tsividis, "Integrated Continuous-Time Filter Design-An Overview," *IEEE Journal of Solid-State Circuits*, vol. 29, pp. 166–176, March 1994.
- [100] G. Duzenli, Y. Kilic, H. Kuntman, and A. Ataman, "On the design of low-frequency filters using CMOS OTAs operating in the subthreshold region," *Microelectronics Journal*, no. 30, pp. 45–54, 1999.
- [101] P. M. Furth, "A Subthreshold CMOS Continuous-Time Bandpass Filter with Large-Signal Stability," *Analogue Integrated Circuits and Signal Processing*, pp. 197–205, May 1999.
- [102] C.-C. Hung, K. Halonen, M. Ismail, and V. Porra, "Micropower CMOS GM-C Filters For Speech Signal Processing," in *Proceedings of the 1997 International Symposium on Circuits and Systems*, pp. 1972–1975, June 1997.



- 
- [103] C.-C. Hung, K. Halonen, V. Porra, and M. Ismail, "Low-Voltage, Micropower Weak-Inversion CMOS Gm-C Filter," in *ICECS'96*, pp. 1178–1181, 1996.
- [104] R. L. Geiger and E. Sanchez-Sinencio, "Active Filter Design Using Operational Transconductance Amplifier: A Tutorial," *IEEE Circuits and Devices Magazine*, pp. 20–32, March 1985.
- [105] E. Sanchez-Sinencio and J. Silva-Martinez, "CMOS transconductance amplifiers, architectures and active filters: a tutorial," *IEE Proc.- Circuits Devices and Systems*, vol. 147, pp. 3–11, Feb 2000.
- [106] R. Castello, F. Montecchi, F. Rezzi, and A. Baschiroto, "Low-Voltage Analog Filters," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 42, pp. 827–840, November 1995.
- [107] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design*. Holt, Rinehart and Winston, INC, 1987.
- [108] A. J. Sutherland, A. Hamilton, D. Renshaw, and M. Glover, "Analogue VLSI for temporal frequency analysis of visual data," in *Proceedings of the 2002 International Symposium on Circuits and Systems*, vol. 3, pp. 743–746, May 2002.
- [109] P. E. Allen and D. R. Holberg, *CMOS Analog Circuit Design: Second Edition*. Oxford University Press, second ed., 2002.
- [110] D. A. Johns and K. Martin, *Analog Integrated Circuit Design*. John Wiley and Sons, 1997.
- [111] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Addison Wesley, 1992.
- [112] A. Graps, "An Introduction to Wavelets," *IEEE Computational Science and Engineering*, vol. 2, no. 2, 1995. <http://www.amara.com/IEEEwave/IEEEwavelet.html>.
- [113] R. T. Edwards, *Time-Frequency Acoustic Processing and Recognition: Analysis and Analog VLSI Implementations*. PhD thesis, The Johns Hopkins University, 1999.
- [114] O. Rioul and M. Vetterli, "Wavelets and Signal Processing," *IEEE Signal Processing Magazine*, vol. 8, pp. 14–38, Oct 1991.
- [115] J. Lin, W.-H. Ki, T. Edwards, and S. Shamma, "Analog VLSI Implementation of Auditory Wavelet Transforms Using Switched-Capacitor Circuits," *IEEE Transactions on Circuits and Systems-I Fundamental Theory and Applications*, vol. 41, pp. 572–583, September 1994.
- [116] R. T. Edwards and M. D. Godfrey, "An Analog Wavelet Transform Chip," in *IEEE International Conference on Neural Networks*, vol. 3, pp. 1247–1251, 1993.
- [117] R. T. Edwards and G. Cauwenberghs, "Analog VLSI Processor Implementing the Continuous Wavelet Transform," in *Advances in Neural Information Processing Systems 5*, pp. 692–98, 1995.



- [118] R. T. Edwards and G. Cauwenberghs, "A VLSI Implementation of the Continuous Wavelet Transform," in *Proceedings of the 1996 International Symposium on Circuits and Systems*, vol. 4, pp. 368–371, May 1996.
- [119] E. Justh and F. Kub, "Analog CMOS High-Frequency Continuous Wavelet Transform Circuit," in *Proceedings of the 1999 International Symposium on Circuits and Systems*, vol. 2, pp. 188–191, 1999.
- [120] A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands, Wavelets*, ch. 3, pp. 133–34. Academic Press, Inc, 1992.