Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

Low latency vision-based control for robotics

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Engineering

In

Mechatronics

At Massey University, Manawatu, New Zealand

Joshua Lues

2018

Abstract

In this work, the problem of controlling a high-speed dynamic tracking and interception system using computer vision as the measurement unit was explored.

High-speed control systems alone present many challenges, and these challenges are compounded when combined with the high volume of data processing required by computer vision systems. A semi-automated foosball table was chosen as the test-bed system because it combines all the challenges associated with a vision-based control system into a single platform. While computer vision is extremely useful and can solve many problems, it can also introduce many problems such as latency, the need for lens and spatial calibration, potentially high power consumption, and high cost.

The objective of this work is to explore how to implement computer vision as the measurement unit in a high-speed controller, while minimising latencies caused by the vision itself, communication interfaces, data processing/strategy, instruction execution, and actuator control. Another objective was to implement the solution in one low-latency, low power, low cost embedded system. A field programmable gate array (FPGA) system on chip (SoC), which combines programmable digital logic with a dual core ARM processor (HPS) on the same chip, was hypothesised to be capable of running the described vision-based control system.

The FPGA was used to perform streamed image pre-processing, concurrent stepper motor control and provide communication channels for user input, while the HPS performed the lens distortion mapping, intercept calculation and “strategy” control tasks, as well as controlling overall function of the system. Individual vision systems were compared for latency performance. Interception performance of the semi-automated foosball table was then tested for straight, moderate-speed shots with limited view time, and latency was artificially added to the system and the interception results for the same, centre-field shot tested with a variety of different added latencies.

The FPGA based system performed the best in both steady-state latency, and novel event detection latency tests. The developed stepper motor control modules performed well in terms of speed, smoothness, resource consumption, and versatility. They are capable of constant velocity, constant acceleration and variable acceleration profiles, as well as being completely parameterisable. The interception modules on the foosball table achieved a 100% interception rate, with a confidence interval of 95%, and reliability of 98.4%. As artificial latency was added to the system, the performance dropped in terms of overall number of successful intercepts. The decrease in performance was roughly linear with a 60% in reduction in performance caused by 100 ms of added latency. Performance dropped to 0% successful intercepts when 166 ms of latency was added.

The implications of this work are that FPGA SoC technology may, in future, enable computer vision to be used as a general purpose, high-speed measurement system for a wide variety of control problems.

Acknowledgements

I would first like to thank my supervisors Donald Bailey and Gourab Sen Gupta for their support throughout this process. Between their busy schedules they gave me so much of their time and shared their knowledge with me whenever I needed it. Donald guided me with his extensive expertise in image processing, and Sen steered the overall direction of the project.

My gratitude to my wife, Nadia, cannot be expressed enough. I am grateful for all the meals, late nights she spent awake in solidarity, trips to the supermarket, and encouragement. She is truly a blessing and I am so lucky to have the most incredible wife a man could ask for.

I am thankful to my brother-in-law, Ruven, and my brother, Giulio. Ruven was my personal trainer, and a heathy body leads to a healthy mind. Giulio was an inspiration to me as he is the hardest working person I know. Every Skype conversation increased my motivation level.

I can't thank my parents (biological and in law) enough for all of their financial and emotional support. It really has been wonderful having such a loving family and I couldn't have done it without each and every one of them.

Table of contents

Acronyms and conventions used ix

1. Introduction 1

1.1. Aims and objectives 3

1.2. Thesis statement 3

2. Subsystems required for low latency control of a vision-based system 4

2.1. Foosball table automation 5

2.2. Object tracking techniques 6

2.3. Computer vision-based control systems 6

2.3.1. Self-adjusting vision systems 6

2.3.2. Other methods to improve vision system robustness 7

2.3.3. Spatial calibration 8

2.3.4. Trajectory prediction and state estimation techniques 9

2.4. Image and signal processing on FPGA 11

2.4.1. Software and hardware-based image processing 11

2.4.2. Performance improvements 13

2.4.3. Power consumption and embedded capabilities 14

2.4.4. Programming and reconfigurability 14

2.5. Actuator control and sensing on FPGA 15

2.6. Conclusions 16

3. System overview, requirements and high-level architecture 17

3.1. Temporal and spatial resolution requirements – Camera parameters 17

3.2. System response 19

3.3. Interception performance 20

3.4. Power consumption 21

3.5. Modularity and versatility 21

3.6. Minimal latency from distribution of processes 22

3.7. Reconfigurability 24

3.8. Debuggability/traceability 24

3.9. Potential methods to achieve outcomes 25

3.9.1. CPU or GPU based systems 26
3.9.2. ASIC or custom hardware 27
3.9.3. FPGA 27
3.9.4. Summary of potential methods 29
3.9.5. Proposed method 30
3.10. High-level complete system architecture 30
3.10.1. Semi-automated foosball table control system 30
3.10.2. Compute subsystem 32
4. Mechanical subsystem 34
4.1. Introduction 34
4.1.1. Custom aspects of foosball table design 34
4.2. Materials 39
4.3. Design aspects meeting official specifications 40
4.4. Accurate vision system placement 42
5. Vision subsystem 44
5.1. Background 44
5.1.1. Latency in control systems 44
5.1.2. Related work 45
5.1.3. Hardware tested 46
5.2. Novel event detection latency 48
5.2.1. Aim 48
5.2.2. Methodology 48
5.2.3. Results 49
5.3. Steady state latency 50
5.3.1. Aim 50
5.3.2. Methodology 50
5.3.3. Mathematics 51
5.3.4. Results 52
5.4. Discussion 53
5.4.1. Immediate usefulness of results 53

5.4.2. Comparison of performance 53
5.4.3. Potential improvements 54
5.4.4. Automated foosball 54
5.5. Conclusions 55
5.6. Implementation of final vision system 55
5.6.1. Terasic D5M camera module 55
5.6.2. Prototypes 56
5.6.3. Foosball table interception simulation 58
5.6.4. Lens changes 58
5.6.5. Distortion correction 60
5.6.6. Final implementation 61
6. Compute subsystem 65
6.1. Terasic DE1-SoC FPGA development board 65
6.1.1. FPGA component 66
6.1.2. HPS component 67
6.1.3. SoC design 67
6.2. Motor Control 73
6.2.1. Background 73
6.2.2. Mathematical approximation 74
6.2.3. Stepper motor basic control algorithm 75
6.2.4. Stepper motor control on FPGA 78
6.2.5. Determining acceleration parameters 80
6.2.6. Kicking algorithm 80
6.2.7. Homing and sliding algorithm 81
6.2.8. Interception algorithm and image to table spatial mapping 83
6.3. Resource requirements 87
6.4. Summary 88
7. System integration 89
7.1. Electrical system 89
7.1.1. Overall electrical system architecture 89

7.1.2. 36V stage 91

7.1.3. 12V stage 91

7.1.4. Limit and homing switches 92

7.1.5. Wiring management and cabling 93

7.1.6. Field illumination 93

8. Testing and system performance 95

8.1. Aims 95

8.2. Spatial invariance for interception 95

8.2.1. Testing method 95

8.2.2. Overview of apparatus 96

8.2.3. Data analysis 97

8.2.4. Results 98

8.2.5. Discussion 98

8.3. Close-up interception performance 98

8.3.1. Testing method 99

8.3.2. Results 99

8.3.3. Discussion 99

8.4. Close-up interception performance with added latency 99

8.4.1. Method to artificially add latency 99

8.4.2. Testing method 100

8.4.3. Results 100

8.4.4. Discussion 101

8.5. Automated foosball 102

8.6. Potential improvements or additions 102

8.7. Conclusions and recommendations 102

9. Final conclusions and recommendations 103

References 104

Published work 110

Appendices 111

Appendix A 111

Appendix B 113

Appendix C 114

Appendix D 116

Appendix E 117

Appendix F 118

Appendix G 119

Appendix H 120

Appendix I 120

Appendix J 121

Acronyms and conventions used

A/D or D/A converter – Analogue to digital or digital to analogue

ASIC – Application specific integrated circuit

CCD – Charge coupled device

CMOS – Complementary metal oxide semiconductor

CPU – Central processing unit

DCS – Distributed control system

DLL – Delay locked loop

DSP – Digital signal processor

DVS – Dynamic vision sensor

FPGA – Field programmable gate array

FPS – Frames per second

GPIO – General purpose input/output

GPU – Graphics processing unit

HDL – Hardware description language

HPS – Hard processor system

IDE – Integrated development environment

MP – Megapixel

MPPA – Massively parallel processor array

MSB – Most significant bit

OS – Operating system

OTS – Off the shelf

PLL – Phase locked loop

USB – Universal serial bus

VHDL – Very high-speed integrated circuit HDL

List of figures

Figure 1-1 - Standard foosball table ... 3

Figure 2-1 - Cycle diagram representing algorithmic image processing ... 12

Figure 2-2 -Streamed image processing in FPGA hardware ... 13

Figure 3-1 - Representation of ball motion blur on foosball table – birds-eye view with foosmen hidden from view ... 18

Figure 3-2 - Foosball playing field captured from below, using the DE1-SoC and D5M camera ... 19

Figure 3-3 - One cycle of image capture through to system actuation response ... 20

Figure 3-4 - Latencies present in systems tested by Čížek et al. 2016 ... 23

Figure 3-5 - Flowchart representing the data flow, communication, and data transfers in distributed PC based system ... 23

Figure 3-6 - Possible configurations of PC based image capture and actuator control systems ... 26

Figure 3-7 - Semi-automated foosball table ... 31

Figure 3-8 - CAD model of the automated actuation modules ... 31

Figure 3-9 - Compute system input and output signals ... 32

Figure 4-1 - Render of automated modules of CAD model ... 34

Figure 4-2 - Pull out torque curve of Nema 23 bi-polar stepper motor - (Pololu, 2018) ... 35

Figure 4-3 - Render of actuation modules ... 35

Figure 4-4 - Foosball table actuation module - rotational drive assembly ... 36

Figure 4-5 - Foosball table actuation module - linear drive assembly ... 37

Figure 4-6 – Close-up of belt connector on actuation module ... 37

Figure 4-7 - From left to right a bottom-up view of the glass base, the interlock switch and the safety lid ... 38

Figure 4-8 - From left to right - bottom-up view of foosball playing field with field illumination on and off respectively ... 39

Figure 4-9 - Image of completed, vinyl wrapped semi-automated foosball table – from front right ... 39

Figure 4-10 - Render of foosball table CAD model ... 40

Figure 4-11 - Render of foosball goal on CAD model ... 41

Figure 4-12 - Semi automated foosball table right hand ball return chute ... 41

Figure 4-13 - CAD model of the placement jig for the foosball table vision system ... 42

Figure 4-14 - Manufactured calibration homing jig with rotating locking tabs ... 43

Figure 5-1 - Image capture test systems ... 47

Figure 5-2 – Novel event detection latency results for all 5 experiments including PS3 eye at both resolutions ... 49

Figure 5-3 - CAD model of object marker apparatus used in steady state latency experiment ... 51

Figure 5-4 - Steady state latency results for all 4 experiments ... 52

Figure 5-5 - Vision system development process ... 55

Figure 5-6 - Terasic TRDB-D5M camera development board 56
Figure 5-7 - Bayer RGB pattern representation 56
Figure 5-8 - Output window of simple trajectory calculation and interception coordinate simulation 58
Figure 5-9 - Long focal-length (narrow angle) lens provided with the D5M camera module 59
Figure 5-10 - Sunex DSL215 fisheye lens 59
Figure 5-11 - MATLAB output image of all calibration grid intersection points found by distortion correction algorithm 61
Figure 5-12 - Uncalibrated (left) and calibrated (right) distortion point map 61
Figure 5-13 - Representation of overall image processing system 62
Figure 5-14 - Block diagram of 3x3 windowed Bayer interpolation (demosaicing) hardware implemented in FPGA. (Bailey, 2018). Reprinted with permission. 62
Figure 5-15 - Block diagram of simplified RGB to YCbCr conversion implemented in FPGA hardware (Bailey, 2018). Reprinted with permission. 62
Figure 5-16 - Block diagram of a 5×5 morphological filter (erosion) implemented in FPGA hardware – for dilation, AND gates were replaced with OR gates. (Bailey, 2018). Reprinted with permission. 63
Figure 5-17 - Block diagram of the high-level architecture for connected components analysis (CCA) implemented in FPGA (Bailey, 2018). Reprinted with permission. 63
Figure 6-1 - Representation of vision-actuation control system data cycle from input to response........ 65
Figure 6-2 - Block diagram showing the master/slave relationship for each of the bridges between the FPGA and HPS (Altera, 2016). 66
Figure 6-3 - Example Qsys HPS component 68
Figure 6-4 - Setup options for PIO in Qsys 69
Figure 6-5 - PIO connections and addresses in Qsys 69
Figure 6-6 - C code used to send an "active" signal to the FPGA, and disable all the stepper motors temporarily 70
Figure 6-7 - VHDL code to receive data/signals from the HPS 70
Figure 6-8 – Representation of data transfer method between FPGA and HPS 70
Figure 6-10 - Address map for Qsys SoC design 71
Figure 6-11 - Development process for the motor control design 73
Figure 6-12 - Nema 23 bipolar stepper motor - 200 steps per revolution 74
Figure 6-13 - Pull out torque curve of Nema 23 bi-polar stepper motor 74
Figure 6-14 - Acceleration and deceleration equations for stepper motor ramping 76
Figure 6-15 - Stepper motor simulation output curve for input distance = 600 76
Figure 6-16 - Stepper motor simulation output curve for input distance = 1200 77
Figure 6-17 - Output curves for constant velocity (top 3) and trapezoidal velocity (bottom 3) profiles ... 77
Figure 6-18 - Stepper module entity diagram showing inputs and outputs 78
Figure 6-19 - Block diagram of simplified variable acceleration pulse generation method, implemented in FPGA 79

Figure 6-20 – Birds-eye view of foosball table CAD model with automated rods annotated 82
Figure 6-21 – Birds-eye view of foosball rods - Module 4 to 1 from left to right 84
Figure 6-22 – Birds-eye view of foosball table with positions of each foosman on each module labelled 85
Figure 6-23 - C code used to calculate differences between ball and foosmen positions 86
Figure 6-24 - Block diagram representing entire SoC system design 87
Figure 7-1 - Development process for the mechanical/electrical system 89
Figure 7-2 - Electrical system architecture for the semi-automated foosball table 90
Figure 7-3 - Panel with 2 USB-B sockets, 1 D-Sub socket, 1 IEC socket, and a high-current switch 90
Figure 7-4 - PCB layout of 40-pin expansion board created using Altium Designer 92
Figure 7-5 - PCB with pluggable screw terminals and a 40-pin header 92
Figure 7-6 - One actuation module with each of the homing/limit switches labelled 93
Figure 7-7 - Wiring conduits on left of image and electrical screw glands on right 93
Figure 7-8 - Semi-automated foosball table with field illumination LEDs switched on 94
Figure 7-9 - XL4015 DC-DC voltage regulator module with adjustable output voltage 94
Figure 8-1 - Ball release apparatus positioned at the back of foosball table (closest to human goal) 96
Figure 8-2 - Experimental setup using module 2 for interception 97
Figure 8-3 - Code snippet of algorithm used to artificially delay system by 20 frames 100
Figure 8-4 - Output graph of interception performance versus artificially added latency for straight goal shots 101

List of tables

Table 5.1 - Important novel event detection latency data 50
Table 5.2 –Important steady state latency data 52
Table 5.3 - Resource consumption on DE1-SoC of basic image processing design 57
Table 6.1 - System PIOs and their respective functions 72
Table 6.2 - Time taken and maximum velocity for 3 simulated stepper motor velocity profiles 78
Table 6.3 - Increment or decrement based on current stepper motor cycle position 79
Table 6.4 - Comparison of resource utilisation for minimal design through to full SoC design 88