



**UNIVERSITY OF
KWAZULU-NATAL**

**INYUVESI
YAKWAZULU-NATALI**

**AUTOMATIC CALIBRATION OF A TOOL-CHANGING
UNIT FOR MODULAR RECONFIGURABLE MACHINES**

James Collins

891115639

November 2011

**Supervisor:
Prof. Glen Bright**

CORRECTED COPY

Submitted in fulfilment of the academic requirements for the degree of Master of Science in Engineering at the School of Mechanical Engineering, University of KwaZulu-Natal.

Declarations

Declaration by Author

I, James Collins, declare that

- ii) The research reported in this dissertation, except otherwise indicated, is my original work.
- iii) This dissertation has not been submitted for any other degree or examination at any other university.
- iv) This dissertation does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
- v) This dissertation does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted then:
 - a) their words have been rewritten but the general information attributed to them has been referenced;
 - b) where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
- vi) Where I have reproduced a publication of which I am an author, co-author, or editor, I have indicated in detail which part of the publication was written by myself alone and have fully referenced such publications.
- vii) This dissertation does not contain text, graphics, or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation and in the References sections.

Signed: _____
Mr. James Collins

Date: _____

Declaration by Supervisor

As the candidate's Supervisor I agree to the submission of this dissertation.

Signed: _____
Professor Glen Bright

Date: _____

Acknowledgements

I would like to thank my supervisor, Professor Glen Bright. I am very appreciative of this opportunity and the experiences I have gained as a result of it. I am extremely grateful for his support, encouragement and leadership during this research.

To my wife, thank you for picking up the extra load at home and allowing me the hours away to do this. You continue to be the best thing that ever happened to me.

I am very thankful to the fourth year group for all their work on the tool-changing unit. They did an excellent job in providing me with a great foundation to work from.

I would like to extend my gratitude to the University of KwaZulu-Natal, the Advanced Manufacturing Technology Strategy (AMTS) and the Department of Science and Technology (DST) for the financial support and research funding.

I would also like to thank my colleagues and the staff of the School of Mechanical Engineering who have made this a most enjoyable experience and who have aided me in so many ways.

Abstract

Modern trends in customer demand have resulted in the development of a class of manufacturing system known as Reconfigurable Manufacturing Systems (RMS). Reconfigurable systems are designed around the idea that they must be able to be reconfigured in both their production capacity as well as in the machining processes they perform. A subset of the RMS paradigm is a group of machines called Modular Reconfigurable Machines (MRMs). Modular machines are built up from different hardware modules. They offer the user the possibility of only purchasing the required tooling for the specific need at the time. As reconfigurable machines are able to offer flexibility in machining functions, their ability to have easy access to a variety of machine tools would greatly influence their effectiveness and production capacity. This project presents a machine tooling system that would provide MRMs with an efficient way to change tools.

A major requirement of the unit was that it should automatically calibrate itself in terms of its position relative to the machine it was servicing. In order for the unit to realize this requirement, it needed a method that would provide it with real-time 3D tracking of the spindle with which it was interacting. Commercially available systems that offer this facility are very costly. A popular gaming controller, the Nintendo Wii remote, was used to provide the tool-changing unit with a very economical real-time 3D tracking capability. This dissertation details the design, implementation and testing of the positioning system for the tool-changing unit.

Table of Contents

1 Introduction	1
1.1 Reconfigurable Manufacturing Systems (RMS).....	1
1.2 Research Motivation.....	1
1.3 Project Objectives	3
1.4 Research Publications.....	3
1.5 Dissertation Outline	4
1.6 Chapter Summary	4
2 The Development of Reconfigurable Machines	5
2.1 Introduction.....	5
2.2 Pre-RMS Manufacturing Paradigms	5
2.3 Dedicated Manufacturing Systems.....	6
2.4 Flexible Manufacturing Systems	6
2.5 Bridging the gap – the advent of RMS.....	7
2.6 Defining RMS	8
2.7 RMT Design Considerations	9
2.7.1 Mechanical requirements	10
2.7.2 Control requirements	10
2.7.3 Ramp-up requirements	11
2.8 RMS State-of-the-art.....	11
2.9 Tool Changing Systems	13
2.10 The need for a tool-changing unit in RMS.....	15
2.10 Chapter Summary	16
3 The Tool-Changing Unit	17
3.1 Introduction.....	17
3.2 The Design Specifications of the Unit.....	17
3.3 The selection of the design concept.....	17
3.4 The core structure of the unit.....	18
3.5 The tool transfer system	19
3.6 Controlling the unit.....	20
3.6.1 System modelling approach.....	20

Table of Contents

3.6.2 Control system description.....	21
3.7 Chapter Summary	23
4 Calibration, Integrability and positioning systems	24
4.1 Introduction.....	24
4.2 Calibration and Integrability.....	24
4.3 Automatic calibration of the tool changer.....	25
4.4 Calibration Methods and Measuring Devices.....	26
4.4.1 Manual Calibration.....	27
4.4.2 Automatic tool height setter	27
4.4.3 Touch Probe Robot Calibration	27
4.4.4 Michelson Interferometer	27
4.4.5 Laser Doppler Calibration System	28
4.4.6 Theodolite Triangulation Systems.....	28
4.4.7 Tracking Laser Interferometer	29
4.4.8 Camera-aided Robot Calibration Methods	30
4.5 Discussion	30
4.6 Chapter Summary	31
5 Kinematic Considerations	32
5.1 Introduction.....	32
5.2 Forward Kinematics	33
5.2.1 Machine with 3 DOF	33
5.2.2 Machine with 4 DOF	35
5.2.3 Machine with 5 DOF	37
5.3 Inverse Kinematics.....	40
5.3.1 Machine with 4 DOF	40
5.3.2 Machine with 5 DOF	40
5.4 Chapter Summary	42
6 The Positioning System Concept	43
6.1 Introduction.....	43
6.2 Considered Systems.....	43

Table of Contents

6.2.1 Vision Systems.....	43
National Instruments Systems.....	43
Festo Vision Recognition System.....	44
6.2.2 Laser measuring devices.....	44
6.2.3 Sonar	44
6.3 The Nintendo Wii Platform.....	45
6.4 The Nintendo Wii Remote	46
6.4.1 Bluetooth Communication	47
6.4.2 Accelerometer	49
6.4.3 Infrared Camera.....	50
6.4.4 Power requirements and feedback features.....	51
6.5 Wiimote positioning in the Nintendo Wii system	52
6.6 Wiimote positioning concepts.....	55
6.6.1 Single Wiimote, Single LED	55
6.6.2 Two Wiimotes, Single LED	55
6.6.3 One Wiimote with two LEDs	56
6.7 Wiimote Location on the tool-changer	57
6.8 Chapter Summary	60
7 Software Design	61
7.1 Introduction.....	61
7.2 Initial API selection	61
7.3 Calculating the 3D position.....	62
7.3.1 Camera Characteristics	62
7.3.2 Two Wiimotes, Single LED	63
7.3.2.1 Normalizing the pixel values and ray mapping	64
7.3.2.2 The Intersection of the two rays	65
7.3.3 Single Wiimote, Two LEDs.....	66
7.4 GUI Development	68
7.4.1 Connecting.....	70
7.4.2 Read	70

Table of Contents

7.4.4 Festo Positions	73
7.4.5 Tool Change	75
7.5 Chapter Summary	76
8 Results and Discussion	77
8.1 Introduction	77
8.2 Camera Characteristics	77
8.2.1 Testing Setup.....	77
8.2.2 Horizontal Field-of-View.....	78
8.2.3 Verical Field-of-View.....	80
8.3 Positioning System Validation.....	82
8.4 Tests on the tool-changing unit	87
8.4.1 Stereo Wiimote configuration.....	87
8.4.2 Single Wiimote configuration.....	90
8.5 Cost Analysis.....	93
8.6 Discussion	95
8.7 Chapter Summary	97
9 Conclusion	98
References	101
Appendices	
Appendix A. Mechanical Calculations	106
A.1 Wii Support Weld Calculations.....	106
A.1.1 Weld 1 Calculation	107
A.1.2 Weld 2 Calculation	109
A.2 Deflection.....	110
A.3 Bolt Analysis.....	111
Appendix B. Drawings	114

Table of Contents

Appendix C. Sample 3D Position Calculations	117
C.1 Two Wiimote, single LED Calculation	117
C.2 Single Wiimote, two LEDs calculation.....	120
Appendix D. Code Listings	122
D.1 GlovePIE Script for Two Wiimotes and a single LED	122
D.2 GlovePIE Script for One Wiimote and two LEDs	123
D.3 GUI Main listing	124
D.4 Wiimote Info Listing.....	130
Appendix E. Further Results	132
E.1 Wiimote Horizontal FOV Analysis.....	132
E.2 Stereo Wiimote Robot Test Results.....	134
E.3 Single Wiimote test results on the tool-changing unit.....	139
E.4 Single Wiimote Repeatability	143
Appendix F. Quotation and Datasheet	146
F.1 National Instruments Smart Camera Quotation	146
F.2 Vishay TSAL6400 Infrared LED Datasheet	147
PDF of the dissertation	Supplementary CD
Code and related software	Supplementary CD

List of Figures

Figure 2-1: A Manufacturing Paradigms hypothesis	8
Figure 2-2: An Overview of the Automated Design Methodology	10
Figure 2-3: An example of a library of modules for Modular Reconfigurable Machines	12
Figure 2-4: Using different modules to vary a machine's functionality.....	12
Figure 2-5: Automatic tool changer for a small-sized Plano Miller.....	13
Figure 2-6: Large automatic tool changer for a vertical lathe	14
Figure 2-7: CNC milling machine with inbuilt tool carousel.....	15
Figure 3-1: Tool-changing unit concept	18
Figure 3-2: The skeletal structure of the tool changer	19
Figure 3-3: Spindle drive configuration	20
Figure 3-4: The tool-changing unit	22
Figure 4-1: The light path through the Michelson Interferometer	28
Figure 4-2: Theodolite triangulation	29
Figure 4-3: Tracking laser interferometer	30
Figure 5-1: The MR2G modular machine.....	32
Figure 5-2: 3 DOF machine.....	33
Figure 5-3: Frame allocation for 3 DOF Machine.....	34
Figure 5-4: 4 DOF machine configuration showing added rotation	36
Figure 5-5: Geometry for the tool rotation θ , about the x-axis.....	36
Figure 5-6: The 5 DOF configuration showing rotation about the y-axis	37
Figure 5-7: Geometry of the rotation about the y-axis	38
Figure 6-1: The Nintendo Wii Remote and Wii Console	46
Figure 6-2: The Wii Sensor bar showing the two LED clusters.....	46
Figure 6-3: Wii Remote	48
Figure 6-4: Wii Remote Accelerometer coordinate system.....	50
Figure 6-5: Various Wiimote positional readings	53
Figure 6-6: Horizontal Angular Determination of the Wiimote.....	54
Figure 6-7: Single Wiimote and single LED positioning.....	55
Figure 6-8: 3D Stereoscopic positioning - Two Wiimotes and one LED.....	56

List of Figures continued

Figure 6-9: 3D positioning with a single Wiimote	56
Figure 6-10: Wiimotes mounted on either side of gripper.....	58
Figure 6-11: Wiimotes mounted above the Carousel.....	59
Figure 6-12: Wiimote FOV interaction with the MRM spindle.....	60
Figure 7-1: Wiimote stereo vision setup.....	63
Figure 7-2: Steps to determine the stereoscopic 3D position.....	64
Figure 7-3: Single Wiimote distance calculation	66
Figure 7-4: Distance determination with Single Wiimote	68
Figure 7-5: Steps required in the automatically calibrated tool-change.....	69
Figure 7-6: Tool Changer GUI –Opening screen.....	70
Figure 7-7: Full GUI with 2 Wiimotes connected.....	71
Figure 7-8: Wiimote information panel.....	71
Figure 7-9: GUI main screen showing No LED condition	72
Figure 7-10: GUI showing 3D position.....	72
Figure 7-11: Festo FCT Configuration.....	74
Figure 7-12: Festo FCT showing the position table and motion testing panel	75
Figure 7-13: GUI window showing conditions allowing for a tool change	76
Figure 8-1: Wiimote positioning system frame of reference	78
Figure 8-2: Positioning System Testing Setup	78
Figure 8-3: Wiimote Horizontal FOV Test results.....	80
Figure 8-4: Wiimote Vertical FOV results.....	82
Figure 8-5: Error in the X direction compared to the Robot X coordinate.....	85
Figure 8-6: Error Y vs Robot Y.....	85
Figure 8-7: Error distribution of the primary axes with respect to Robot Z.....	86
Figure 8-8: Stereo Wiimote positioning errors	88
Figure 8-9: Wii Error vs Axis distance along entire axis	89
Figure 8-10: The repeatability of the Stereo Wiimote System	89
Figure 8-11: Single Wiimote arrangement.....	90
Figure 8-12: Single Wiimote Errors vs Machine Axis position.....	91
Figure 8-13: Repeatability spread of the single Wiimote axes.....	93
Figure 8-14: Wiimote positioning system cost effectiveness.....	95

List of Figures continued

Figure A-1: Wii Support welds and forces	106
Figure A-2: Weld 1 moment diagram with tube profile.....	107
Figure A-3: Moment diagram for weld 2.....	109
Figure A-4: Deflection in Wii Mounting.....	110
Figure B-1: Wii mounting support	114
Figure B-2:Wii Mounting Plate.....	115
Figure B-3: Wii mounting bracket.....	115
Figure B-4: Carousel Modification.....	116
Figure E-1: Graphs of the x right values	133
Figure E-2: Variations in the FOV along the z-axis	134
Figure E-3: Comparison between Festo and Single Wii Errors.....	142

List of Tables

Table 6-1: Wiimote reverse engineering status.....	47
Table 6-2: Basic Wiimote Identification report.....	48
Table 6-3: Wiimote reports.....	49
Table 8-1: Horizontal FOV readings.....	79
Table 8-2: Vertical FOV readings	81
Table 8-3: Stereo Wiimote error distribution from all results.....	83
Table 8-4: Error distribution in region of interest (200 mm to 400 mm from Wiimote)	84
Table 8-5: Error distribution in the region between 500 mm & 750 mm from the Wiimotes	84
Table 8-6: Repeatability test for the Stereo Wiimote System.....	90
Table 8-7: Error distribution of the Single Wiimote tests	92
Table 8-8: Repeatability of Single Wiimote axes	92
Table 8-9: Wiimote Positioning System costs compared to other systems.....	94
Table A-1: Weld pattern for weld 1	108
Table A-2: Weld pattern for weld 2	109
Table E-1: Further analysis of the Horizontal FOV results.....	132
Table E-2: Stereo Wiimote Robot Test Results.....	134
Table E-3: Single Wii Z Error.....	139
Table E-4: Single Wiimote and Festo Error comparison	140
Table E-5: Single Wiimote X and Y Errors.....	142
Table E-6: Single Wiimote Z repeatability.....	144
Table E-7: Single Wiimote X repeatability	144
Table E-8: Single Wiimote Y repeatability	145

List of Acronyms and Abbreviations

2D	Two dimensional
3D	Three dimensional
API	Application Programming Interface
CCD	Charged Coupled Device
CNC	Computer Numerically Controlled
DMS	Dedicated Manufacturing Systems
DMT	Dedicated Machine Tools
DOF	degree-of-freedom
FCT	Festo Configuration Tool
FMS	Flexible Manufacturing Systems
FOV	Field of View
fps	frames per second
GUI	Graphical user interface
HID	Human Interface Device
IDE	Integrated Development Environment
IR	Infrared
LDDM	Laser Doppler Displacement Meters
LED	Light Emitting Diode
MI	Michelson Interferometer
MR2G	Mechatronics and Robotics Research Group
MRM	Modular Reconfigurable Machine
PC	Personal Computer
PSD	Position Sensing Device
RAM	Random Access Memory
RMS	Reconfigurable Manufacturing Systems
RMT	Reconfigurable Machine Tool
ROM	Read-only Memory
SDP	Service Discovery Protocol
TV	Television
VGA	Video Graphics Array

Nomenclature

A	Area
A_c	Clamped area
A_s	Tensile stress area of a bolt
b	Breadth of weld pattern, breadth of plate
{B}	Base coordinate frame
B	Weight of tube
d	Depth of weld pattern, diameter , distance apart of LEDs or Wiimotes
\hat{d}	Unit direction vector
E	Young's modulus
F	Force
F_e	Joint separation force
F_i	Bolt pre-tension force
F_s	Thread stripping force
g	Acceleration of gravity = 9.81m/s ²
G	Shear Modulus
h	Height from base plate to the centre of the tool arm / height of weld
h_{max}	Maximum height from the base plate to the centre of the tool arm
I	Moment of Inertia
I_u	Unit moment of Inertia
J	Polar moment of Inertia
K_b	Bolt stiffness
K_c	Joint stiffness
L	Length along the beam
M, M1, M2	Moments
o	Origin point
p	distance between two dots on Wiimote camera
P_{TA}, P_{TB}	Tool position points
P	Weight of Wii mounting plate
P_g	Goal Point
${}^S P_o$	Starting position of the tool with respect to the station frame
${}^S P_T$	Position of the tool with respect the station frame
${}^{SR} P$	The point of the tool with reference to frame {SR}
R	Ray equation

Nomenclature continued

r	radius of rotation of work table
r_c	radius from the centre point of r to the corner of the work table
${}^S R_S$	The rotation matrix going from $\{SR\}$ to $\{S\}$
S_u	Tensile Strength
S_y	Yield Strength
S_{ys}	Shear Strength
$\{S\}$	(Work) Station coordinate frame
$\{SR\}$	New position of station frame $\{S\}$ after rotation
s_b	Worktable width
s_l	Worktable length
t	Thickness
t_1, t_2	Ray scale values
T	Applied Torque
$\{T\}$	Tool coordinate frame
t_o	Tool offset
x_c	The distance tool column has moved in the x-direction
x_p	Wiimote pixel value in the x-direction
\hat{x}	Normalised x pixel value
x'	Unit ray x-value
y	distance from neutral axis
y_p	Wiimote pixel value in the y-direction
\hat{y}	Normalised y pixel value
y'	Unit ray y-value
y_t	The distance the work table has moved in the y-direction
W	Weight of Wiimotes
z	Distance from Wiimote/s to IR LED
z'	Unit ray z-value
α	Angle of rotation of the work table, half angle subtended between two IR LEDs and the Wii remote camera
β	Angle from centre line of the work table to the corner
β_p	Angular field of view per pixel
Δy	Change in position on the y-axis
Δz	Change in position along the z-axis

Nomenclature continued

θ	Angle of rotation, angle of twist, Wiimote vertical field of view
θ_{py}	Vertical angle subtended by the centre point of the two LED circuit and the centre of the Wiimote camera
φ	Wiimote horizontal field of view
φ_{px}	Horizontal angle subtended by the centre point of the two LED circuit and the centre of the Wiimote camera
σ_b	Bending stress
τ	Shear stress

1

Introduction

1.1 Reconfigurable Manufacturing Systems (RMS)

The concept of Reconfigurable Manufacturing Systems (RMS) was initiated by researchers in the mid to late 1990s [1]. The paradigm emerged as a result of the need for manufacturers to respond to the varying demands that were, and continue to be placed on them. The current manufacturing climate requires producers to be able to adjust to constant changes in both the volume and mix of products. Reconfigurable Manufacturing Systems aim to meet these needs by being able to be reconfigured in terms of the processes they perform as well as their output capacity.

Within the paradigm of reconfigurable systems, a family of machine tools known as Modular Reconfigurable Machines (MRMs) was developed [2]. This type of machine is built up from different hardware modules that are put together to provide the necessary amount of machining capability needed at any particular time. If a more complex machine is needed for a new process, additional modules may be added to facilitate the extra requirements. Similarly, if a product requires less functionality, then modules may be removed to make a simpler machine or even used somewhere else in the plant.

Another advantage which may be very beneficial for developing countries is that end users of these machines can start out purchasing a very simple machine or a few modules. As the business expands, additional modules may be purchased to increase the capacity and/or functionality. This type of machine would therefore reduce the initial capital outlay of a start-up business and hence reduce the capitalisation risk of potential business owners.

1.2 Research Motivation

Reconfigurable manufacturing systems are designed at the outset to offer adaptability in both hardware and software. The machine tools that make up these systems must be able to deliver an array of functionality and processing capability. A selection of tools is therefore required to perform these operations. This necessitates rapid tool-changing to

promote an efficient and cost effective manufacturing process. Along with this, the reconfiguration of the manufacturing equipment gives rise to the need for an in-process calibration procedure to ensure machining accuracy and timely part production. Reconfigurable machine tools will not be able to effectively function without an adaptable tool supply system.

Modern manufacturing equipment typically includes a tool-changing unit built into the system. In the field of modular reconfigurable machines, tool-changing units will be designed as one of the modules available to potential customers. It is therefore important for tool-changing units to be effectively integrated into the manufacturing system. Other machining modules, such as those that add an extra degree of freedom, will predominantly have hardware interfaces with the system. A tool-changing unit may not have this advantage, as it may be cumbersome and unnecessary to design a hardware interface for a module that does not specifically require rigidity constraints within the system.

The Mechatronics and Robotics Research Group (MR²G) of the School of Mechanical Engineering, of which the author is part, has developed both a modular reconfigurable manufacturing machine [3] as well as a tool-changing unit designed to service modular machines [4]. The tool-changing unit is a standalone unit that is able to service a machine without relying on any other structural support.

One of the key characteristics of Reconfigurable Manufacturing Systems is the integrability of the components that make up the system [5]. Integrability emphasizes the ease with which different modules can be joined to or separated from the rest of the system.

As the tool-changer is a structurally independent module, it cannot rely on mechanical interfaces to secure its position relative to the machine it is servicing. A system needs to be developed whereby the tool-changer can automatically sense its position compared to the machine or spindle it is interacting with. Also, modular machines are designed to be reconfigured. The positioning system must therefore be able to adjust to a change in configuration. The quicker this system responds the more integrable the unit becomes.

A real-time 3D position detecting system was necessary for the tool-changer to accurately sense its position relative to the work station it was interacting with and hence provide it with an automatic calibration capability. If the unit could automatically calibrate itself to various configurations, reconfiguration time would be reduced. This would allow for the advantages of reconfigurable manufacturing systems to be exercised.

The main problem with respect to the tool-changer and its possible application in industry would be its ability to quickly adapt to a variety of configurations of machinery. The unique characteristic of the tool-changer is its ability to service different machines. If a system was not developed for the unit to be able to rapidly and automatically adapt to different configurations or machines, the reconfigurability of the system would be compromised.

The required system would also need to allow the unit to interface with machining stations with a minimum of hardware connections in order to reduce setup time. This would allow for the tool-changer to service more than one machine or configuration.

Another advantage of one tool-changer having the ability to service more than one machine would be that it would reduce the cost of the equipment to the end user.

1.3 Project Objectives

The objectives of the project were to:

- Research the current development of Reconfigurable Machine Systems and Modular Machines
- Design modifications to the tool-changing unit that can be integrated into systems that will allow autonomous calibration and positioning detection
- Research positioning and calibration methods and equipment
- Provide the machine with in-process calibration capability to ensure machining accuracy and minimal down time after process or machine reconfigurations
- Test and verify the tool-changing unit positioning system.

1.4 Research Publications

- Padayachee, J. , Davrajh, S., Collins, J., Bright, G., *The Development of Reconfigurable Manufacturing Equipment for Product Mass Customization*, International Conference on Competitive Manufacturing (COMA '10), 3-5 February 2010, Stellenbosch, South Africa, Pages 291-296
- Collins, J. , Bright, G., *A Tool-Changing Unit for Modular Reconfigurable Manufacturing Systems*, 25th ISPE International Conference on CAD/CAM, Robotics and Factories of the Future; 14-16 July 2010, Pretoria, South Africa
- Collins, J.E.T., Bright, G., *Automatic Tool-Changing within the Reconfigurable Manufacturing Systems Paradigm*, South African Journal of Industrial Engineering Nov 2011 Vol 22 No.2, p. 68-79.

- Collins, J., Bright, G., *Automatic Calibration of a Tool-Changing Unit for Reconfigurable Manufacturing Systems (RMS) using Nintendo Wii Remotes*, 26th ISPE International Conference on CAD/CAM, Robotics and Factories of the Future, 26-28 July 2011, Malaysia.

1.5 Dissertation Outline

Chapter 1: Provides the background to the research, its objectives and the resultant research publications.

Chapter 2: Describes the manufacturing developments that resulted in RMS and gives an overview of the RMS design requirements and considerations. It also summarises RMS research to date.

Chapter 3: Presents an overview of the development of the tool-changing unit.

Chapter 4: Discusses calibration and integrability with respect to the tool-changer. Several positioning and calibration systems are also presented.

Chapter 5: Details the kinematic calculations necessary in determining the position of the tool relative to the work table of the MRM.

Chapter 6: Presents the development of the concept of using Wii Remotes as a position sensing system for the tool-changing unit.

Chapter 7: Details the software development required to provide the unit with an automatic calibration capability.

Chapter 8: Presents the testing, results and discussion of the positioning system.

Chapter 9: Concludes the research and provides suggestions for further work in this field.

1.6 Chapter Summary

This initial chapter introduced the reader to Reconfigurable Manufacturing Systems and modular machines. It gave the motivation for an automatically calibrated tool-changing unit within this paradigm. It stated the objectives of the research as well as providing a list of the generated research publications. Finally, it presented an outline of the dissertation.

2

The Development of Reconfigurable Machines

2.1 Introduction

This chapter presents the manufacturing paradigms and events that resulted in the development of Reconfigurable Manufacturing Systems. The shortcomings of both dedicated and flexible manufacturing systems are outlined along with the key features and design considerations of RMS. The chapter ends with the current progress that has been made in RMS research along with a presentation of a variety of tool changing systems.

2.2 Pre-RMS Manufacturing Paradigms

Manufacturing system development started with a job shop environment where general purpose machines were used to produce a large variety of parts at a relatively low production rate and developed into using dedicated machines that produce a very small range of parts, but at a high volume in order to deliver the cheapest part possible. In the eighties, flexible manufacturing systems began to emerge to be able to meet the needs of product and technology changes and constantly varying market demands [6].

Job shops require a considerable amount of human involvement whereas modern manufacturing trends are constantly searching for ways in which to minimize the amount of human interaction required for production.

The predominant manufacturing systems in use today come in the form of either Dedicated Manufacturing Systems (DMS) or Flexible Manufacturing Systems (FMS). Reviewing these systems and how their shortcomings led to the RMS paradigm will be investigated.

2.3 Dedicated Manufacturing Systems

Mass production materialized as the first major manufacturing paradigm at the beginning of the 20th century [7]. As the need for greater quantities of products grew and customers were demanding cheaper products, dedicated machines were developed to produce high quantities of products at reduced production costs. These machines or dedicated machine tools (DMT) are designed around a specific product and are limited to produce only a small selection of products or even only one specific part. In order to reduce the complexity and hence the cost of the machines they are designed with only the required degrees of freedom in order to manufacture the specific part. In essence the concept is to create a machine with the 'bare bones' essentials to get the job done. As a result, DMTs are the least expensive machines in terms of initial capital outlay [8]. They are very robust and produce parts at high capacity [1]. This high production rate is often achieved through the use of a number of tools working on the product at the same time e.g. 'gang drilling.'

Dedicated manufacturing systems work well when production volumes are high and are even more effective if lifecycles of the parts they make are long. However, due to their dedicated design in both hardware and software, any modifications or changes are costly or not economically viable at all.

Modern trends in customer demand are requiring manufacturers to produce to the paradigm of mass customization. Mass customization is a response to the fluctuating market trends and unstable economies. Production volume, as well as the variety of products needed to satisfy demand is constantly changing. These factors do not bode well for dedicated systems as they are the most economically resistant to change in demand.

2.4 Flexible Manufacturing Systems

As the limitations of dedicated systems became apparent, an effort was made to offer a more general purpose manufacturing system that could cope with variations in mix and volume of parts. FMS was developed to accommodate both predicted and unknown changes in production requirements. This is advantageous in a market place with the all rapid changes in both technology and customer desires.

The major components of FMSs are Computer Numerically Controlled machines, more commonly known as CNC machines. CNC machines are expensive, general purpose machines that are very flexible in their manufacturing capabilities. Unlike DMTs they use only one tool at a time and thus do not have the production capacity that is available with dedicated systems. Unlike DMTs which are designed around a specific part, CNC machines

are designed without knowledge of the specific part and thus have a large amount of functionality built in.

Although FMSs and CNC machines offer great flexibility and are able to adjust to product and part redesign they have some distinct disadvantages, which are [9]:

- **Expensive** – CNC machines are expensive due to their complexity in being able to deliver flexibility
- **Underutilized** – In order to offer flexibility CNC machines are designed to offer multiple functions and in many cases not all of these functions are used resulting in the machines not being used to their full potential
- **Propriety software and control** – Although CNC machines are programmable their software and control systems are usually machine or manufacturer specific and thus cannot be easily modified by the user

2.5 Bridging the gap – the advent of RMS

It became apparent that although DMTs could deliver mass production capacities and CNC machines could offer flexibility there had to be some manufacturing paradigm that could offer flexibility without the underutilized expense of CNC machines.

Modern markets are in such a state of flux that manufacturing methods have to evolve to systems that can keep up with all the constant changes. A hypothesis given by Hu (2005) [10] is shown in figure 2-1, positions RMS between DMS and FMS. While FMS could offer a general flexibility, the idea behind RMS was to be able to offer a customized flexibility that could adapt and change to meet the varying demands.

In the mid to late nineties Professor Yorem Koren from the University of Michigan began formalizing RMS concepts and the first reconfigurable machine tool (RMT) was patented by him in 1999 [11].

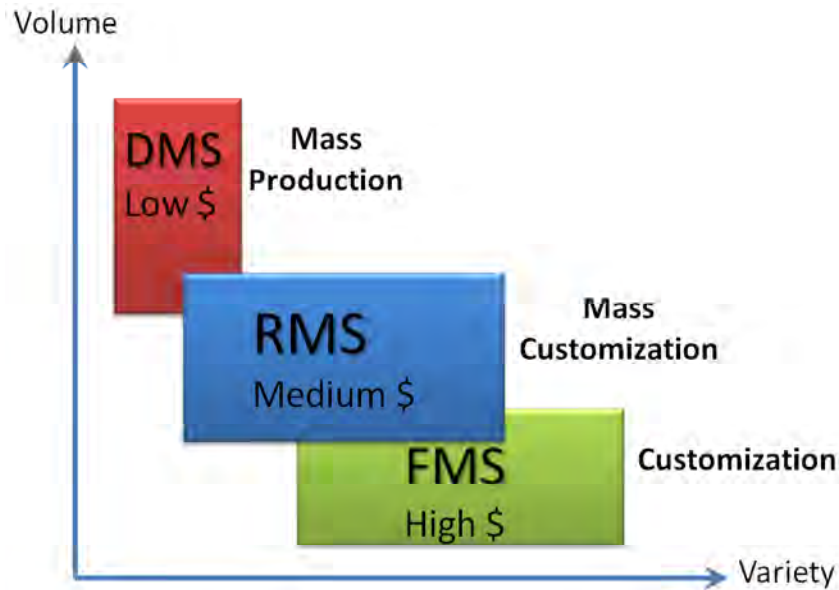


Figure 2-1: A Manufacturing Paradigms hypothesis

2.6 Defining RMS

One of the most concise definitions describing RMS is provided by the Engineering Research Center for Reconfigurable Machining Systems (University of Michigan). It posts the following:

“ Exactly the capacity and functionality needed, exactly when needed” [12].

A more complete definition is offered by Koren et al [1] and is as follows:

“A Reconfigurable Manufacturing System is designed at the outset for rapid change in its structure, as well as its hardware and software components, in order to quickly adjust its production capacity and functionality within a part family in response to sudden changes in market or regulatory requirements.”

The ability to be reconfigurable is the characteristic that sets RMS apart from previous paradigms. Koren et al [1] define the key characteristics of reconfigurable systems to be the following [9], [13]:

Modularity - all systems components, both hardware and software, are designed to be modular.

Integrability - Systems and components are designed to be easily integrated together. This facilitates the current building of systems as well as the adding of future components and new technology.

Customization – the systems capability and flexibility (hardware and controls) are designed to match the application (product family).

Convertibility – Conversion is concerned with the changeover required between batches of production. Changes may include tools, programs and fixtures and in reconfigurable systems these conversions are expected to be short.

Diagnosability – the ability to quickly calibrate newly reconfigured systems to produce quality parts. It also is the ability to quickly identify the sources of quality and reliability in the system.

Scalability – speaks to the ability of the system to increase the capacity of production. It is related to convertibility in that scalability is dependent on the systems ease of conversion to different throughput demands.

Modularity, integrability and diagnosability are concerned with minimizing the amount of time and effort required for reconfiguration, whereas customization and convertibility are concerned with reducing the cost. The combination of all these characteristics is what determines the ease and cost of reconfigurable systems [1].

RMSs are designed to integrate the best of DMS and FMS i.e. robust volume production along with the flexibility to adjust to new production requirements. As such a RMS may contain both DMTs and CNC machines as well as RMTs where needed. There may be some operations that would be constant over the lifetime of the manufacturing system and a DMT would work well. Other operation requirements may change significantly and CNC machines might be a good option to meet those requirements [14].

2.7 RMT Design Considerations

RMT design can be a complex procedure, due to the modular nature of the components that make up these tools. There may be many possible permutations for a good solution to a specific manufacturing requirement. Moon and Kota have developed a method for the systematic design of RMTs [15, 16].

Figure 2-2 shows that the first step in the design of an RMT is to define the task requirements. The task requirements indicate the machining functions and processes that are needed to manufacture the part or family of parts. The design theory, which uses a combination of screw and graph theory, matches the tasks to machine modules in the library and a set of viable solutions is generated. The solutions are then evaluated by

dynamic stiffness to provide a sub set of most feasible machine tools. In summary, the design process consists of the following three stages [15]:

- 1) Conceptual Design
- 2) Module Selection
- 3) Design Evaluation

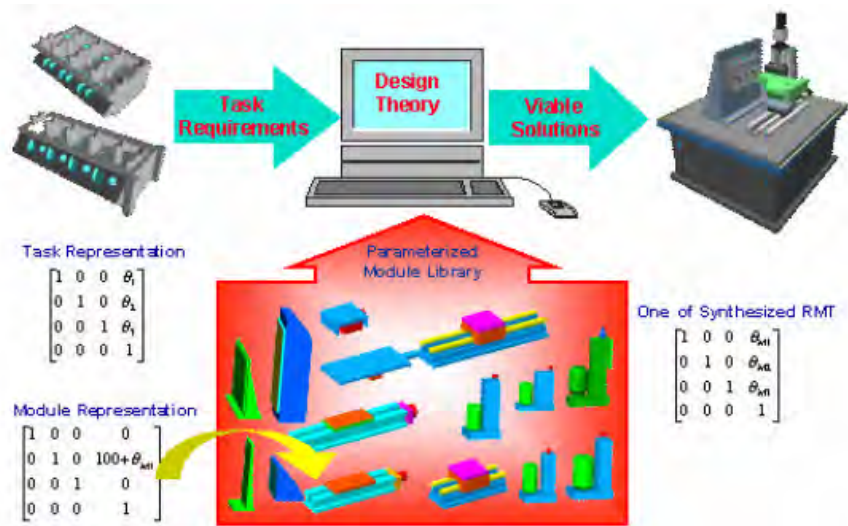


Figure 2-2: An Overview of the Automated Design Methodology

2.7.1 Mechanical requirements

In designing the mechanical aspects of machine tools, one must consider the tool's ability to produce the specified motions for the operation and its ability to produce parts to the required quality and accuracy. These considerations are satisfied by two key properties in mechanical design: kinematic viability and structural stiffness [14]. Kinematic viability is the property that enables the tools to perform the necessary motions to produce the needed features of the part. Structural stiffness is the property that reduces the presence of errors in the manufacturing process [17].

2.7.2 Control requirements

The control components of DMTs are, like the rest of the machine, designed very specifically around the production of a specific part. They are therefore highly customized with no extra complexity and they are not economically modified or upgraded. Typical CNC controllers are very complex in order to offer the flexibility required of this type of machine. This comprehensive functionality can often be underutilized.

In RMT control design, the concept of open-architecture control is necessary. Open-architecture control architecture is modular in design. This allows for hardware and software modules to be added or removed and the controller can be economically reconfigured. RMT controllers can therefore be customized to suit the present requirements and then later be reconfigured as changes occur or new technology is added [14].

2.7.3 Ramp-up requirements

Due to the rapidly changing market demands in the current global economy, manufacturers may only have short windows of opportunity in which to capitalize on a new product. This requires them to minimize the lead time required to produce new parts. Ramp-up may be defined as the time it takes for a new or reconfigured machine or system to reach the required production capacity and quality [1]. If RMTs are to make their desired impact on the manufacturing industry it is important that ramp-ups are achieved quickly.

The ramp-up process will involve calibration of machining system parameters and inspection of the parts to verify their quality. One of the most likely sources of errors in RMTs will be due to incorrect or faulty assembly of the multiple modular units. A machining system's diagnosability is critical in helping to minimize machine ramp-up. A variety of sensory systems will be needed to give accurate feedback and thereby promote good diagnosability. Examples of sensors that would be useful would be: sensors for static deflections, vibration sensors, a tool wear monitoring system and a tool position monitoring system. Accurate diagnostics, along with effective methods for error compensation and calibration of the machine must be applied to RMT design to ensure an efficient ramp-up process [18].

2.8 RMS State-of-the-art

A comprehensive state-of-the-art survey performed by Bi et al in 2007 [19] gives a clear breakdown of the development of reconfigurable machines including the research emphasis up to that point. The paper categorizes the work done, into the areas of:

- 1) Reconfigurable machining systems
- 2) Reconfigurable fixturing systems
- 3) Reconfigurable assembling systems
- 4) Reconfigurable material handling systems

5) Reconfigurable inspecting and calibrating systems

It is evident from these points that extensive research has been conducted in the field of Reconfigurable Manufacturing Systems and Reconfigurable Manufacturing Tools. Although this work has been done, there is still a need to develop formal design methodologies for these types of tools [20].

In the modular machine arena, libraries of mechanical components have predominantly focused on motion and function modules as shown in figure 2-3 and figure 2-4 [21]. Motion modules allow the degrees of freedom of the machine to be varied from a single degree of freedom such as in a drilling machine to a full six degrees of freedom. Function modules deliver the machining process functions such as milling, boring, drilling and turning.

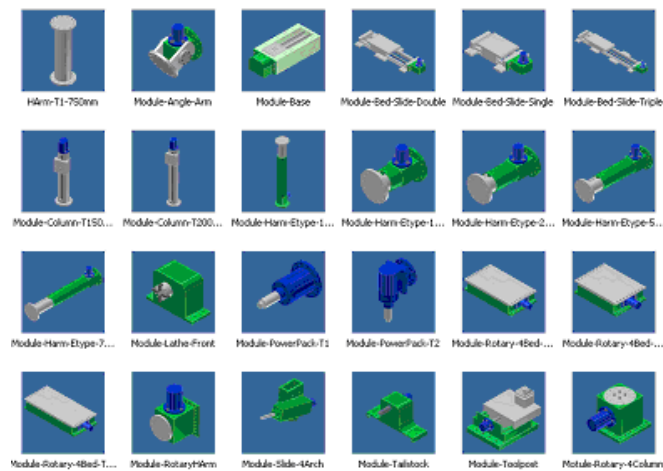


Figure 2-3: An example of a library of modules for Modular Reconfigurable Machines

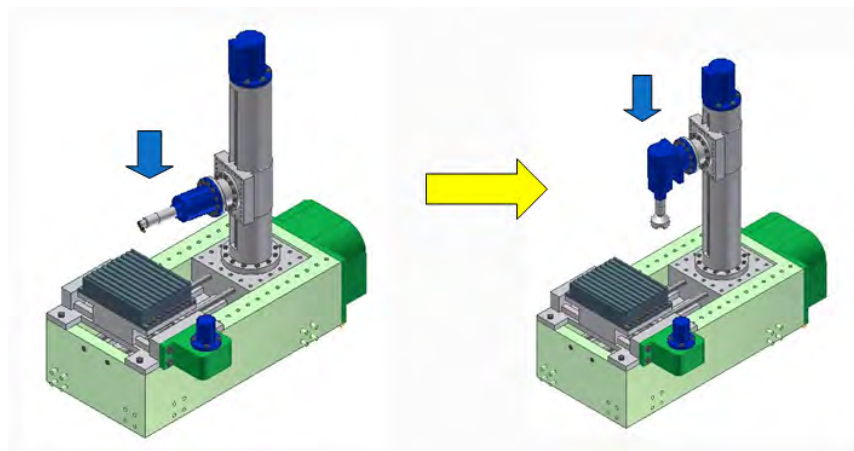


Figure 2-4: Using different modules to vary a machine's functionality

A third category of modules, called accessory modules, are those parts such as clamps and stabilizers. They might not have an active role in the cutting process, but they are critical in ensuring successful production. Tool storage and exchange systems may fall into this category but the author suggests that they would form part of a fourth set of modules called auxiliary modules. Auxiliary modules may also not form a direct part of the machining process but would add extra functionality to the machines, which would aid in the efficiency and quality of production. These modules might also include tool monitoring and quality control modules.

In the South African context, research has been completed by Estment et al [7] in the area of adding an automatic tool changer and spindle to a Reconfigurable Machine Tool. The machine tool was a gantry type CNC machine delivering 2.5 axis machining. The emphasis of the research was the successful integration of these two new modules with the existing machine. Mach 3 CNC programming software was used to generate the required G code. The tool changer module was mounted directly to the gantry machine bed utilising part of the machine that was not required for the workspace.

2.9 Tool Changing Systems

Tool changing systems and tool storage facilities come in a variety of forms and sizes depending on the application and the number of tools required. Tool storage systems vary from a small carousel storing about ten tools to large storage banks which can store several hundred tools. Figures 2-5 [72] and 2-6 [73] show examples of large tool storage systems. The system in figure 2-6 can hold 250 tools.

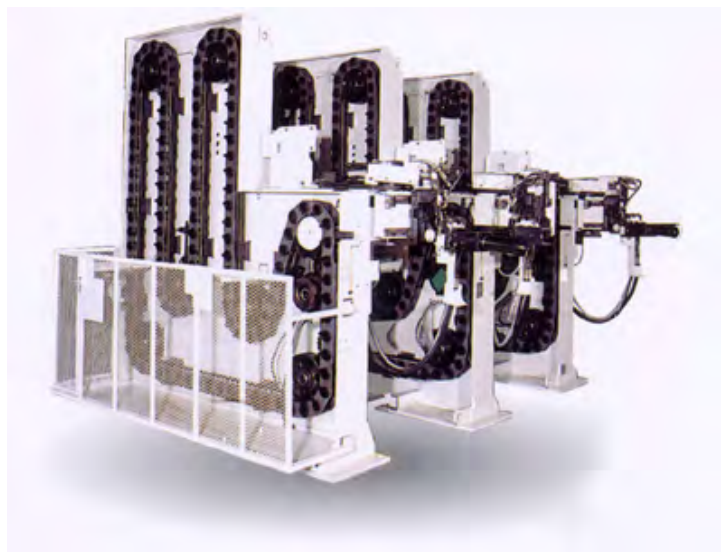
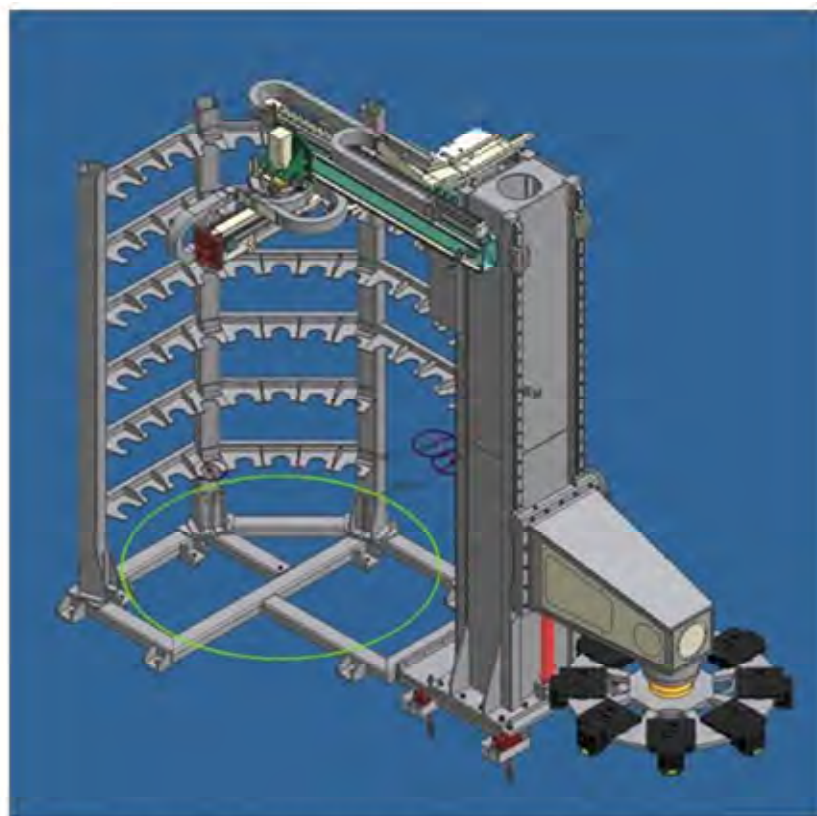


Figure 2-5: Automatic tool changer for a small-sized Plano Miller



Shown here is the machine's tool storage and retrieval system. A two-position tool arm loads and unloads the tools, and the next cutter automatically moves to the "ready" position for faster change times.

Figure 2-6: Large automatic tool changer for a vertical lathe.

Many CNC machines contain a rotating carousel housed near the main working area of the machine. An example of this is shown in figure 2-7. The carousel design provides an efficient tool storage and exchange system. It provides quick access to all the tools, but it is limited by the capacity of tools it can house. The system in figure 2-6 uses a combination of a carousel design to house frequently used tools as well as a larger storage bank to allow access to a large selection of tools.

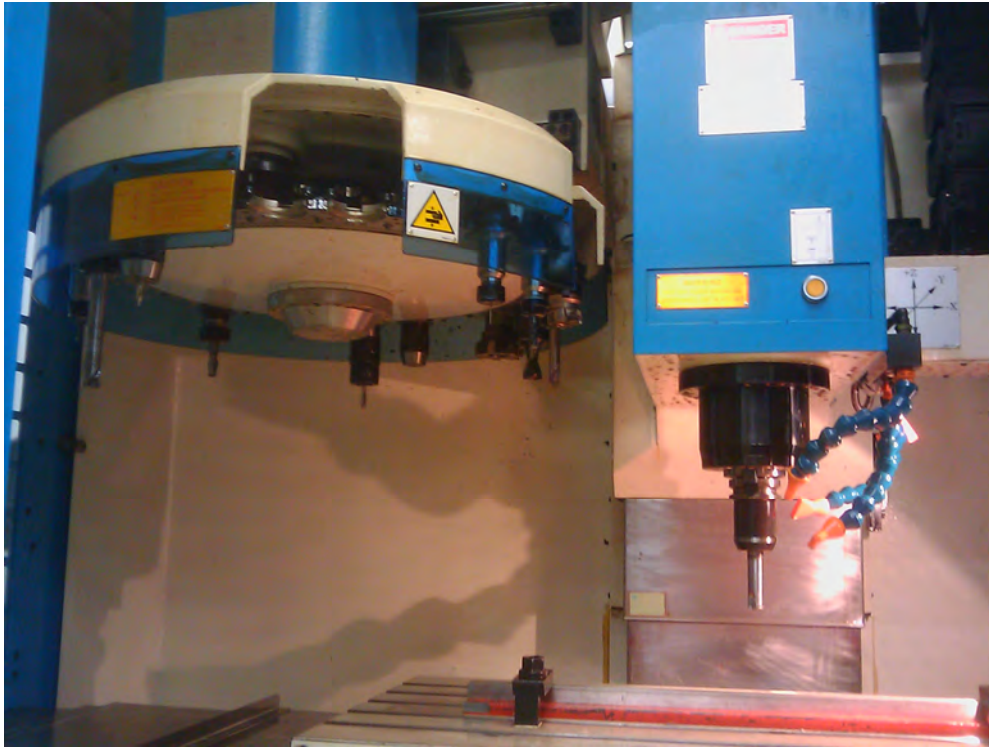


Figure 2-7: CNC milling machine with inbuilt tool carousel

Tool-changing systems vary from arrangements where the spindle is required to move to the storage magazine to systems that provide delivery of the required tool to the spindle.

2.10 The need for a tool-changing unit in RMS

Previously it was indicated that modular machines will require auxiliary modules that may be designed differently from those that service other types of manufacturing equipment. An example of this would be tool-exchange and storage systems. The typical approach to modern day equipment is to build tool-storage into the machine. This could be due to the popularity of CNC machines which offer an all-in-one solution.

Allowances have to be made within modular machines for the possibility of a small or simple machine being needed for a workstation. The machine may require access to a selection of tools and therefore the need for a tool storage system. This was the premise on which the MR²G approached the design of the tool-changing unit. The unit was to be a self-contained and self-supporting module which could service a variety of machines and machine configurations.

2.10 Chapter Summary

This chapter gave a brief history of the manufacturing events that led up to the development of reconfigurable manufacturing systems and machines. It presented dedicated systems and flexible systems as precursors to reconfigurable systems. The RMS describing characteristics and design considerations were discussed. The chapter concluded with an overview of the state-of-the-art of RMS research and a brief discussion of tool changing systems.

3

The Tool-Changing Unit

3.1 Introduction

This chapter presents the development of a structurally, self-sufficient tool-changing module within the RMS paradigm. The design specifications along with the chosen concept are shown and the constructed tool-changer is presented. The chapter concludes with a discussion of the control system of the unit.

3.2 The Design Specifications of the Unit

The first specification to be decided upon was the working space that the tool changer should cover. After a survey of typical production machines, it was concluded that the tool changer should be able to cover 60% of machine working area sizes. This characteristic would represent a good portion of commercially available machines. This requirement translates into a working area for the tool changer of 600 x 300 x 450 mm above the worktable. The tool changer also had to occupy no more than 1.5 m³.

The next consideration was the carrying capacity of the unit, along with the requirement for the speed of the tool exchange. The unit was required to house at least six industry standard tool holders and perform a tool change within 30 seconds.

Initially, it was thought to design the module around a BT-30 tool holder. Upon further investigation, it was found that the BT-40 holder was more common in the South African manufacturing environment. The tool changer was therefore designed to handle a tool weight of up to 5 kg and have an accuracy of 2 mm at the tool insertion point.

3.3 The selection of the design concept

Various methods of exchanging tools are commercially available in the manufacturing environment. Five conceptual designs were generated and considered for the development of the unit. A rotating carousel, a design that is widely implemented on CNC machines, was the emerging concept.

A graphical depiction of the structure and motion of the tool-changing concept is shown in figure 3-1 [22]. The BT-40 tool holders were suspended from the carousel using pull studs. Magnets were used to keep them in place during carousel rotation. The carousel, linear actuators, and gripper arm were capable of providing:

- 1) storage and appropriate selection of the required tools
- 2) translational motion to exchange the tools between the carousel and the manufacturing spindle
- 3) rotation to allow for spindles that may be at an angle
- 4) vertical motion to allow flexibility for different tool holders and for mounting the tool in the spindle and retrieving it.

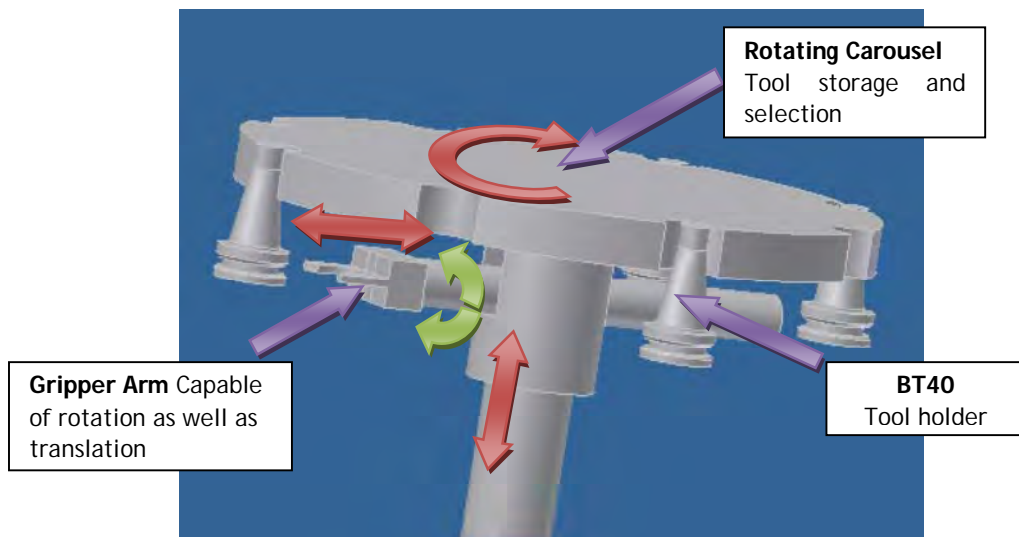


Figure 3-1: Tool-changing unit concept

3.4 The core structure of the unit

The tool changer was designed to be an autonomous module within a reconfigurable manufacturing system. It was therefore important to design a unit that would provide adequate stability during the tool-changing process. The tool changer would also need to be self-reliant, because it would be critical that it not depend on other structures for stability. A structurally-autonomous module would offer the user the advantage of flexibility: the unit could easily be used on several different machines without concern about the structural surroundings. No hardware interface would be required between the tool changer and the machine it was interacting with, as the tool-changing unit would form a complete module in the RMS environment.

The final design used a 6 mm-thick mild steel ‘rolled can’ to form the core of the structure. The ‘can’ would produce the structural strength of the machine as well as provide room for some of the inner mechanical workings of the unit. It was mounted on four adjustable legs that provided stability as well as the necessary height adjustment. The skeletal structure can be seen in Figure 3-2 (adapted from [4]).

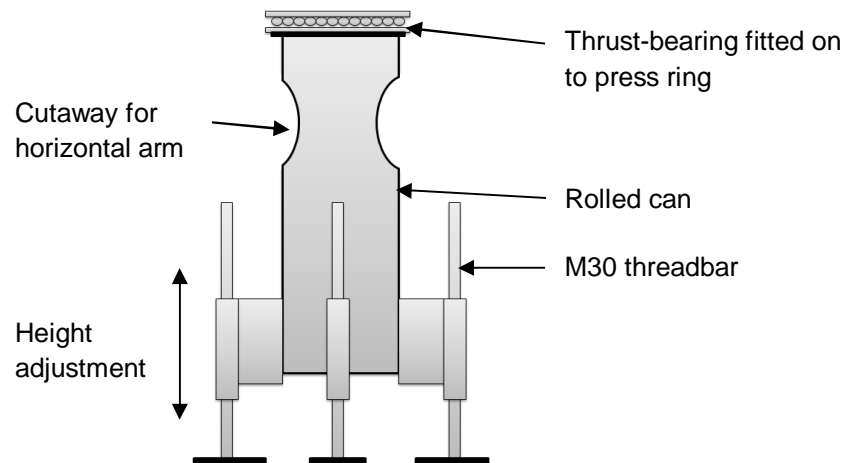


Figure 3-2: The skeletal structure of the tool changer

3.5 The tool transfer system

The tool transfer system was responsible for the required motions of the tools to and from the carousel and the spindle. Four degrees of freedom were needed from the tool-changing unit in order to transfer the tool effectively. The first required motion would be the rotation of the carousel. This provided for the exchange of the different tools to and from the gripper position. The second would be the needed horizontal motion to move the tool away from the storage carousel and towards the spindle. The tool also needed some form of vertical actuation to lift the tool into the spindle – the third degree of freedom.

A unique feature of this tool-changing unit was that it offered a fourth motion not commonly found in other tool-changing systems. This was the rotation of the tool about the horizontal axis so that the tool-changing unit could be used with a machine that had a spindle at a non-orthogonal fixed angle. This added flexibility increased the unit’s ability to interact with a greater range of machines.

In terms of the structural setup of the system, a thrust bearing was placed at the top of the core structure of the unit. The carousel was then mounted on the bearing and a geared

12 V DC motor was used to drive the carousel. A DC motor was also used to drive the worm gear that would provide the rotation of the gripper arm.

For the horizontal and vertical actuation, two linear actuators manufactured by Festo were used. The actuation was delivered through electric ball screw spindle drives. DC servo motors were chosen to drive the spindles due to their small size and relatively high torque. Figure 3-3 [23] shows the configuration of the two spindle drives and their directions of actuation. The larger horizontal drive delivered 300 mm of translation, while the vertical drive provided the necessary 100 mm to ensure adequate insertion of the tool into the spindle.

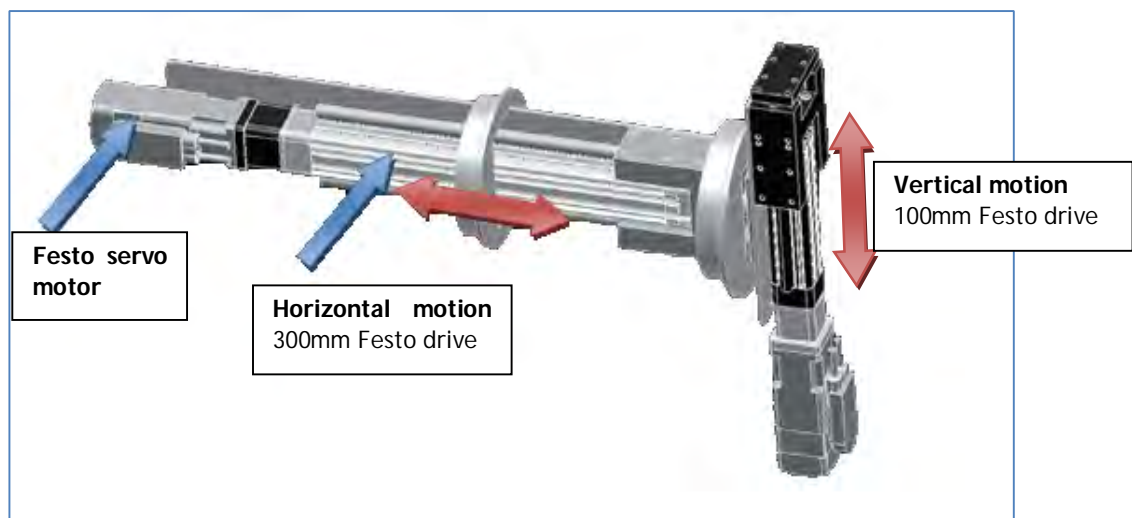


Figure 3-3: Spindle drive configuration

3.6 Controlling the unit

The control system played a vital role in integrating all the aspects of the unit so that it functioned successfully. It also acted as the link between the user and the machine, allowing the user to interact with the module and produce the desired results.

3.6.1 System modelling approach

The accurate modelling of a system was a significant step in ensuring accurate control. It was also important to try to model the system as efficiently as possible, to avoid excessive computation.

There are several approaches that can be taken when attempting to model a robotic system. The focus of the model is to be able to identify the position of the end effector.

One of the approaches is to view the position of the end effector as a cumulative output of the positions of the various degrees of freedom (inputs) of the system. In order to model the system, a mathematical function containing all of the inputs is used to calculate the output. This approach may lead to a complex mathematical function, requiring significant computing power. Although that may be seen as a disadvantage, there are applications where it is beneficial. This option only requires a single control loop, and the end effector error is seen as a mean error distributed through the entire system. This is advantageous in situations where there are excessive degrees of freedom, or where several different combinations of the inputs give rise to the same output. It is also a useful approach when working with degrees of freedom containing similar hardware with similar tolerances.

An alternative approach – and the one that was used for the tool changer – differs from the previous concept in that each degree of freedom is considered as its own system with its own errors. This model separately examines the error generated by each degree of freedom, and reduces the errors individually. The model works well in a system where the degrees of freedom are not directly linked together in affecting the output position. An example of such a system is one with three or less orthogonal axes. The mathematical calculations are, as a result, significantly reduced; but the system requires a separate control loop for each degree of freedom.

This type of model was a good solution for the tool changer, as each degree of freedom was independent of the others in its effect on the desired output. To have each error calculated separately was useful in this application, since the unit consisted of degrees of freedom of varying tolerances. The high accuracies of the Festo drives could be taken advantage of, and would not be affected by the other less accurate parts of the unit.

3.6.2 Control system description

A requirement of the tool-changing unit's control system was that it be efficiently integrated into a broader manufacturing system. The control architecture used commercially-available software and hardware in its development.

User interaction with the unit was achieved through the use of a graphical user interface (GUI) programmed to run on a Windows XP PC. The PC then used serial (RS232) and USB connections to communicate with two Atmel AVR Atmega 32 microcontrollers. The microcontrollers were used to control the two 12 V DC wiper motors that drove the carousel and the worm gear that generated the rotation of the horizontal axis. The switch that powered the gripper solenoid for locking the grippers was also controlled by the

microcontrollers. One of the microcontrollers was used for the pulse counting of the encoders and the angular velocity data generation routines. The other was used for the pulse width modulation (PWM) and the PID controller routines. Incremental encoders were used to provide the required feedback from the motors.

The Festo drives were coupled with proprietary controllers that initially had to be set up through the Festo configuration tool (FCT). The FCT could be used to generate a position table as well as velocity profiles. Once this data was stored on the controllers, digital I/O was used for the PC to communicate which point the drives should move to. The two Festo controllers were also linked to the PC via RS232.

Figure 3-4 [22] below shows the tool-changing unit.

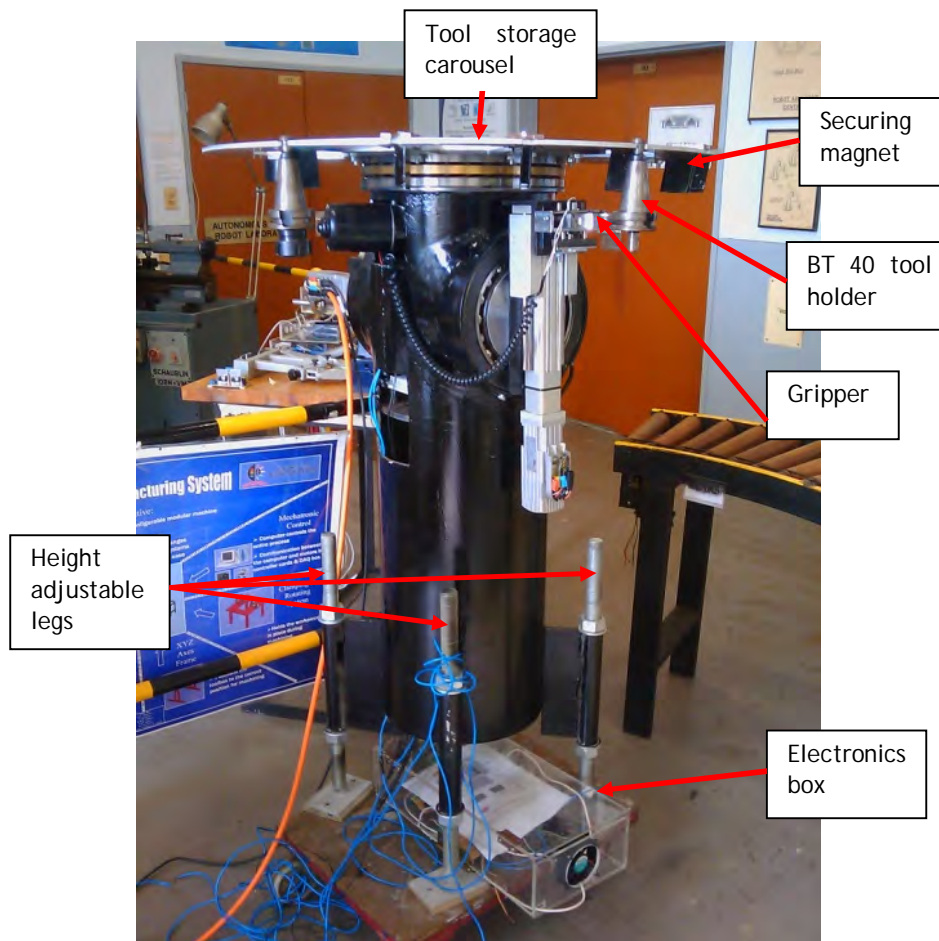


Figure 3-4: The tool-changing unit

3.7 Chapter Summary

This chapter presented the development of the tool-changing unit. It started out with the design specifications imposed on the unit and gave an overview of how those constraints were implemented in making a self contained unit capable of tool-storage and exchange. The hardware was shown and the control philosophy discussed.

4

Calibration, Integrability and positioning systems

4.1 Introduction

Calibration and integrability are inter-linked concepts which enable the efficient integration of the tool-changing unit into the broader reconfigurable manufacturing system. These concepts are discussed in this chapter as well as how they relate to the tool-changing unit. The chapter then goes on to describe several calibration methods and positioning systems.

4.2 Calibration and Integrability

In order for the unit to deliver automatic tool-changing in a reconfigurable manufacturing system environment, the core RMS characteristic of integrability with respect to the tool changer needed to be addressed. One method of applying this characteristic practically was to enable the unit to have an automatic calibration capability.

Calibration may be defined as the procedure of establishing the relationship between a measuring device and a known set of units or standard. In essence, it determines the accuracy of an instrument or device. It may also be the process of adjusting equipment so that it performs to a required specification or standard.

Integrability defines the efficiency with which different components in a reconfigurable system can be added to each other or to the system as a whole. Its definition, according to Mehrabi [9] and ElMaraghy [6], implies integration with the current system as well as an ability to be integrated into future systems with future technologies. Therefore, for a higher level of integrability, one not only has to consider the current configuration of a system, but future configurations also have to be taken into account. Designs of reconfigurable machines need to be forward-thinking, and highly reconfigurable modules will be able to adapt to future changes without having to be rebuilt or redesigned.

Integrability affects the time required to reconfigure a machine, which in turn affects the life-cycle cost of the equipment [8]. It is concerned with both the hardware and the software components of a module [1].

The tool changer is a mechatronic system. There are two primary factors that affect the accuracy of mechatronic tools. They are:

- 1) the precision of the mechanical components;
- 2) the ability of the software and control architecture to adjust and correct errors [24].

The tool changer is required to form part of a reconfigurable manufacturing system – a system designed to be changed and adapted to suit different manufacturing demands. It can therefore be seen that the level of integrability of the unit will depend largely on its ability to be efficiently re-calibrated to a different configuration or machine.

A typical robot calibration process takes place in four stages [25], [26]. A brief description of each stage is given below:

- a) **Modelling** – The robot/machine must first be mathematically modelled in order to relate its functioning to its control system
- b) **Measurement** - The end effector positions are then measured and compared with the predicted positions of the mathematical model
- c) **Identification** – This is the process of identifying how the degrees of freedom or joint angles affect the position of the end effector
- d) **Compensation** – The software commands are then reprogrammed to allow for accurate correlation between the user's input and the resulting output.

4.3 Automatic calibration of the tool changer

The tool changer needed to be able to adjust to a new reconfiguration of equipment, or it may even be required to service a different machine altogether. As mentioned earlier, its integrability will be affected by its ability to be re-calibrated to the new environment. Re-calibration must not only be possible, but also efficient. The more efficiently the unit is able to be calibrated, the more integrable it becomes.

Accurate sensors are the key to calibrating the unit. Often, calibration is performed with the use of additional sensors or specialised equipment. As this equipment is not usually required for a machine's day-to-day running, it is a suitable solution for machines that do not need to be calibrated very often. In a reconfigurable system, however, it may be

difficult to predict how often calibration may be needed. The solution is therefore to have the sensors built into the machine. The number and accuracy of the sensors will mainly be limited by financial constraints [27].

There is an additional benefit to having an automatically-calibrated machine: it could be programmed offline [28]. This would significantly reduce the down-time of equipment during a reconfiguration. It would also have the knock-on effect of reducing ramp-up time; and a new product might gain a competitive edge by being brought to the market sooner.

In industry, a majority of robots are programmed using the 'teach' method. The end effector is placed in position for each of its required motions. These positions are then recorded and used to produce the program for the desired function of the machine. While the benefit of this procedure is that an inverse kinematic model is not required to produce a program, teaching the robot is very time-consuming. This method may be advantageous in circumstances where the machine task is repetitive, but for reconfigurable systems it may lead to excessive amounts of equipment down-time.

In terms of the tool-changing unit, one of the limitations in the initial design had a significant effect on its ease of re-calibration and hence its integrability. The machine had to be taught the required positions in order to effect a tool change. This would mean that the unit would have to be positioned in exactly the same position relative to each machine tool, or reconfiguration. That might not be possible in every circumstance, in which case the unit would have to have a new position table configured. This would drastically increase reconfiguration time.

High integrability of the unit would allow it to be positioned within a reasonable working range of the machine it would be interacting with, and for it to detect its necessary working positions automatically.

4.4 Calibration Methods and Measuring Devices

Calibration is typically a lengthy process. The calibration procedure could therefore result in significant downtime of manufacturing equipment. Calibration is however a very critical procedure in ensuring the accuracy of products. As cumbersome as the process can be, not ensuring that equipment is correctly calibrated could have very costly consequences. This section gives an overview of some of the methods and instruments used to calibrate robots and other manufacturing equipment.

4.4.1 Manual Calibration

One of the most time consuming methods of calibration is when it is done manually. An example of this could be setting the tool offsets on a CNC milling machine. Each time a new tool is used it would be lowered to the machining bed and the offset measured between the bed and the tool head. The tool offset would then be entered into machine's program. Although this takes a lot of time, it is 'real time' process in that the machine is calibrated for each tool as it is about to be used. The accuracy of this method is dependent on the operator's ability to use the measuring equipment and input the correct data into the program.

4.4.2 Automatic tool height setter

An automatic process of the above is accomplished with the use of a tool height setter. A height setter is a touch probe that is mounted to the work table. When a tool change occurs, the new tool is lowered until it touches the probe and the offset is recorded. This is dramatically more time efficient than the manual method.

4.4.3 Touch Probe Robot Calibration

A touch probe can be used to calibrate a robot by using a probe as the end effector tool. The probe makes contact with a predetermined set of calibration points and the robot's internal sensor measurements are compared with dimensions of the reference points [29].

4.4.4 Michelson Interferometer

The Michelson Interferometer (MI), invented in the late 1880s, uses a single light source, a combination of stationary and movable mirrors and a photo detector to precisely measure position (see Figure 4-1 [31]). As the light is split and combined it produces fringe patterns accurate to half a wavelength of the light source [30].

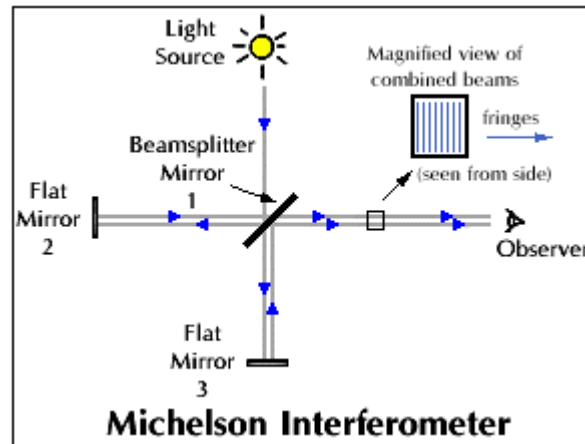


Figure 4-1: The light path through the Michelson Interferometer

Originally, a white light source was used but today helium-neon lasers are used. Currently, there are two types of laser Michelson interferometers in use. One uses a single frequency laser beam and the other type mixes two laser beams to minimize ‘noise’ effects that are present in the single frequency models. These measuring devices, although very accurate, require precision optical components and are bulky. This necessitates the removal of most machine covers in order to perform the calibration [32].

4.4.5 Laser Doppler Calibration System

Laser Doppler displacement meters (LDDMs), which use 1950’s microwave and 1970’s laser technology, are used to perform this type of calibration. LDDMs are more compact than MIs, therefore machine covers do not need to be removed in order to perform the calibration process. Also, LDDMs do not require the special optics of interferometers and are not subject to polarization and stray light issues. They are also simpler to set up than MIs [32]. LDDMs are able to measure displacement, velocity and acceleration and may be used as vibration sensors.

4.4.6 Theodolite Triangulation Systems

These systems are very well described by their name. Multiple electronic theodolites are used along with triangulation geometry to give the position of a target in 3D space. Figure 4-2 [33] shows the concept in calculating the position of a robot tool centre point.

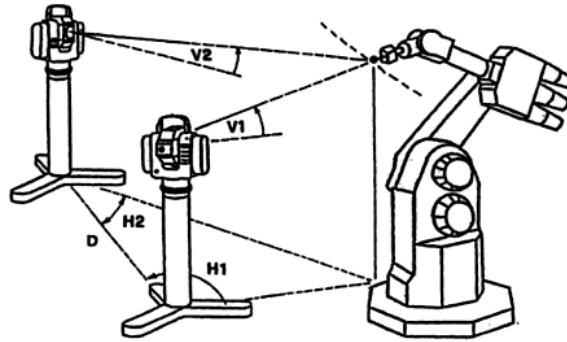


Figure 4-2: Theodolite triangulation

4.4.7 Tracking Laser Interferometer

As opposed to the theodolite method which can only take static measurements, laser trackers are able to take both static and dynamic measurements. Laser trackers are also able to provide both distance and angular displacements giving the polar coordinates of a target. Only one instrument is needed whereas theodolite triangulation requires multiple instruments and, in the case of manual systems, multiple operators. [33]

A plane mirror that can move about the vertical and horizontal axes is used to direct the laser beam of a standard laser interferometer towards a retro-reflector (the target). The laser beam is reflected by the retro-reflector back along its original path. As the beam returns into the measuring instrument, it is split, with some of it going to a position sensing device (PSD) and some going to the interferometer. The PSD is used to generate an output signal which is used to correct the position of the mirror so that the beam stays centered on the reflector. One can calculate the position of the target from the angles of the mirror and the distance measured by the interferometer. Figure 4-3 [33] shows the layout of a tracking laser interferometer. [33]

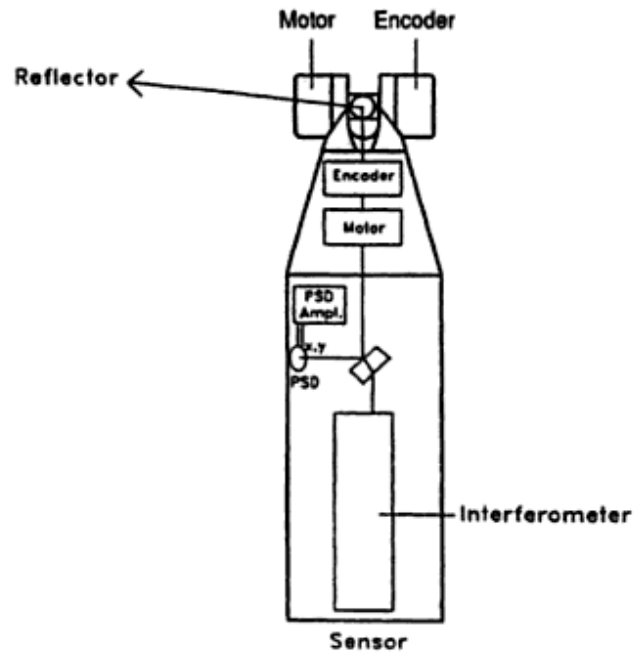


Figure 4-3: Tracking laser interferometer

4.4.8 Camera-aided Robot Calibration Methods

Cameras are also used in robot calibration. Robot calibration is possible with only a single camera [34], but two or more cameras are also used. The camera or cameras are either mounted to a fixture outside of the robot's working area or onto the end effector itself. Image processing techniques are used to translate the 2D images received from the camera or cameras into 3D coordinates. Some methods require the cameras to be calibrated prior to using them for robot calibration but there are methods [35] that require only images of a reference frame or 'ground-truth length' and a calibration object in order to calibrate the robot. Camera-aided calibration can achieve accuracies of between 0.2 mm and 0.4 mm with off-the-shelf cameras [35]. With more expensive equipment an overall accuracy of 0.05 mm has been achieved [36].

4.5 Discussion

A significant factor in enabling the integrability of the tool-changer was its ability to be able to adjust to reconfigurations of equipment or machinery. Integrability is a key characteristic of reconfigurable systems and a major contributor to the success of the tool-changer as part of a machining system that is constantly changing.

The calibration of the unit with regard to its position, compared to the spindle it was servicing, was the primary factor in determining its ability to adjust to a reconfiguration. It was necessary to develop a positioning system that required minimal hardware interfacing to reduce down time of equipment. Positioning systems that required little or no hardware contact were considered and presented in this chapter as options to providing the tool-changer with an efficient automatic calibration capability.

4.6 Chapter Summary

This chapter discussed the concepts of calibration and integrability and how they apply to the tool-changing unit. Some of the positioning and calibration systems that are available are discussed.

5

Kinematic Considerations

5.1 Introduction

As a module within the RMS paradigm the tool-changing unit must interact with other modules and machines. The modular machine developed by the MR²G was to be the case study for the tool-changing unit and positioning system. Figure 5-1 [3] shows the modular machine.

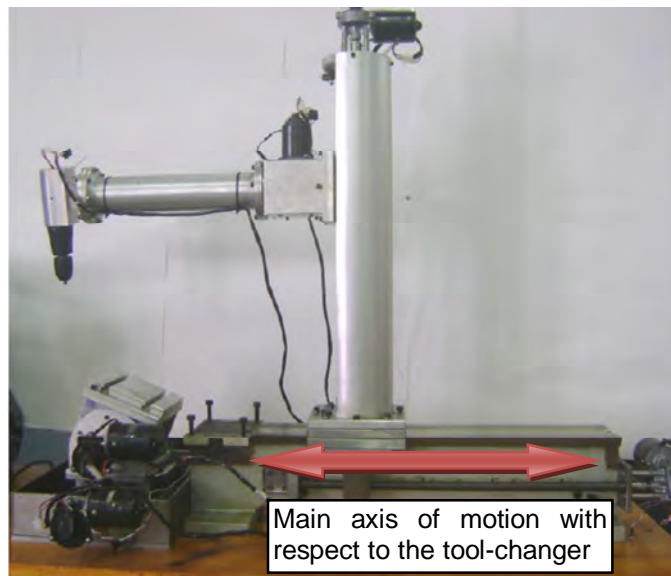


Figure 5-1: The MR²G modular machine

In terms of determining position relative to the 'client' machine, the tool-changer could use sensing systems built onto it as a separate module. These systems would detect the position of the unit relative to the machine. There may have to be a way of identifying the machine with some form of transmitter or receiving device mounted on it.

Another approach would be to get position readings from the machine it was interacting with. If the tool-changing unit was initially set up at a known location from the machine, the values generated by, or calculated from the machine itself could provide the unit with the necessary information to enable a successful tool change.

Kinematics is the study of motion without consideration of the forces behind the motion. We use kinematics to determine position, velocity and acceleration of moving parts. Kinematic analysis allows us to determine the position of the spindle of the machine.

This chapter presents the kinematic analysis of the modular machine in three of its configurations: 3 degree-of-freedom (DOF), 4 DOF and 5 DOF modes.

5.2 Forward Kinematics

Forward kinematics gives us the position of the tool tip relative to a frame of reference. If we know what the translations and rotations of our machine joints are, forward kinematics tells us where the tool is relative to our workstation.

5.2.1 Machine with 3 DOF

In consideration of the forward kinematics, the simplest and therefore the first case to examine, is the 3 DOF machining configuration. The 3 DOF case, as shown in figure 5-2, supports only linear actuation of the tool in the x, y and z directions. From figure 5-2, it can be seen that these linear actuations are provided in the z-direction, by the vertical motion of the tool arm relative to the base plate, in the x-direction by the horizontal motion of the tool column and, in the y-direction by the horizontal motion of the work table.

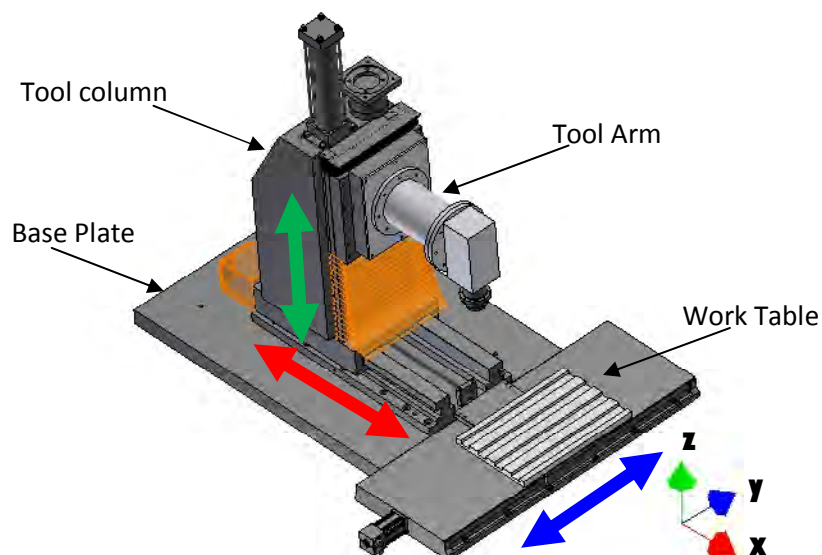


Figure 5-2: 3 DOF machine

The first step is to define the frames of reference. See figure 5-3. The frame of most interest is the (work) station frame denoted by {S}. This frame describes the position relative to the work surface. Machining processes will require the coordinates of the tool relative to this frame in order to machine parts. The base frame {B} is attached to a non-moving part of the base of the machine. Lastly, a frame is attached to the tool head and this frame shall be called the tool frame {T}.

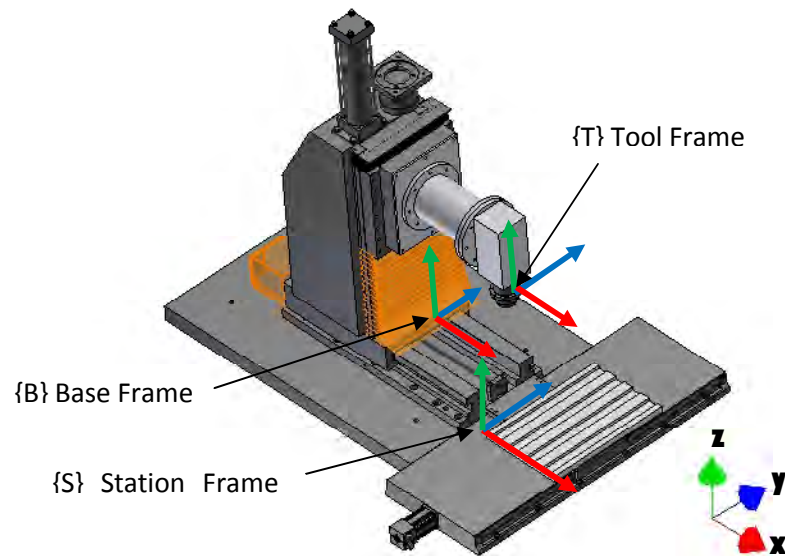


Figure 5-3: Frame allocation for 3 DOF Machine

The following constants and variables will be used

Constants

s_b – width of the of the worktable in the x direction

s_l – length of the worktable in the y direction

t_o - tool offset. The distance from the centre of the tool arm to the origin of the tool frame.

h_{max} – Maximum height from the base plate to the centre of the tool arm

Variables

h – height from base plate to the centre of the tool arm

x_c – the distance tool column has moved in the x-direction

y_t – the distance the work table has moved in the y-direction

Assuming the tool starts at the position shown in figure 5-3, which is the tool, set at the maximum height. The starting position of the tool, with reference to the station frame {S}, is:

$${}^S P_o = \begin{bmatrix} 0 \\ \frac{sl}{2} \\ h_{max} - t_o \end{bmatrix} \quad (5.1)$$

The system is relatively simple at this point and since translation is the only type of motion the forward kinematics is easy to calculate. Given any height of the tool arm h , any motion in the x-direction (x_c) and motion in the y-direction (y_t) our Cartesian coordinates of the tool frame {T} with respect to frame {S} will be:

$${}^S P_T = \begin{bmatrix} x_c \\ \frac{sl}{2} - y_t \\ h - t_o \end{bmatrix} \quad (5.2)$$

5.2.2 Machine with 4 DOF

For the 4 DOF machine, a rotational joint is added to the tool arm as shown in figure 5-4. The rotation is about the x-axis.

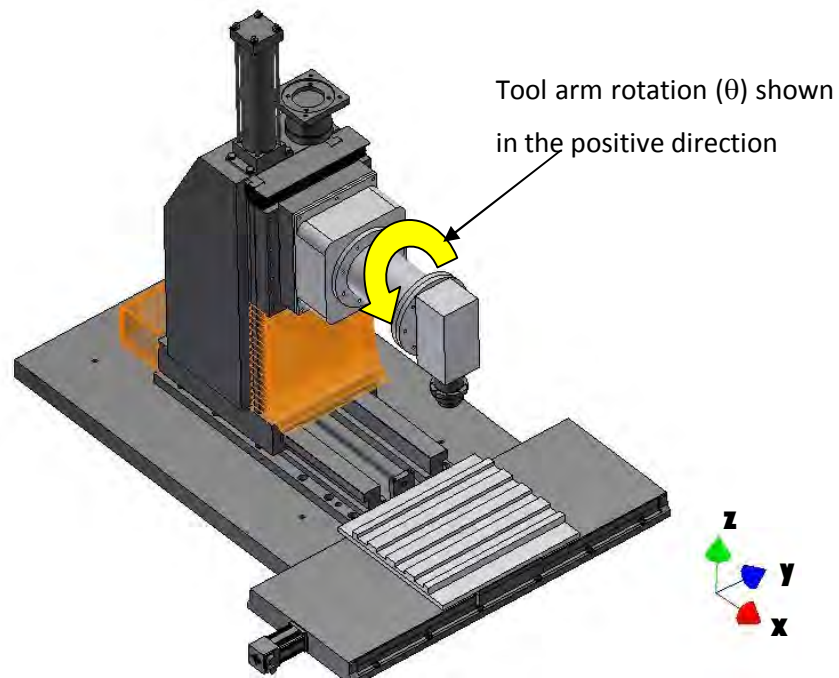


Figure 5-4: 4 DOF machine configuration showing added rotation

Although a rotation matrix could be used to express how a change in the rotation angle (θ) effects the position of the tool, geometry will be used to calculate the new tool position (P_{TB}). Figure 5-5 shows the geometry.

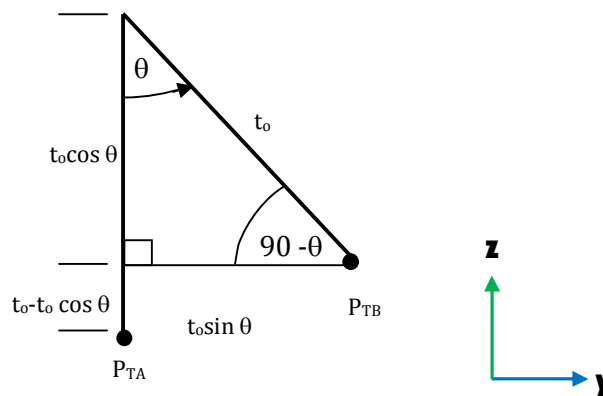


Figure 5-5: Geometry for the tool rotation θ , about the x-axis

The rotation θ causes a change in the y and z coordinates of the tool as the tool moves from P_{TA} to P_{TB} .

The change in the y coordinate: $\Delta y = t_0 \sin \theta$ (5.3)

And the change in the z coordinate: $\Delta z = t_0 - t_0 \cos \theta$ (5.4)

∴ The new y coordinate is given by:

$$y_B = \frac{s_l}{2} - y_t + t_o \sin \theta \quad (5.5)$$

And

$$z_B = h - t_o + t_o - t_o \cos \theta \quad (5.6)$$

$$\therefore z_B = h - t_o \cos \theta \quad (5.7)$$

Including the new rotation DOF about the x-axis, the Cartesian coordinates of the tool relative to the station frame origin are:

$${}^s P_{TB} = \begin{bmatrix} x_c \\ \frac{s_l}{2} - y_t + t_o \sin \theta \\ h - t_o \cos \theta \end{bmatrix} \quad (5.8)$$

And the tool would have an angle of $(90 - \theta)^\circ$ to the work table.

5.2.3 Machine with 5 DOF

To generate the fifth DOF for the machine, the work table is mounted such that it can rotate about the y-axis on a radius r. Figure 5-6 shows the fifth DOF.

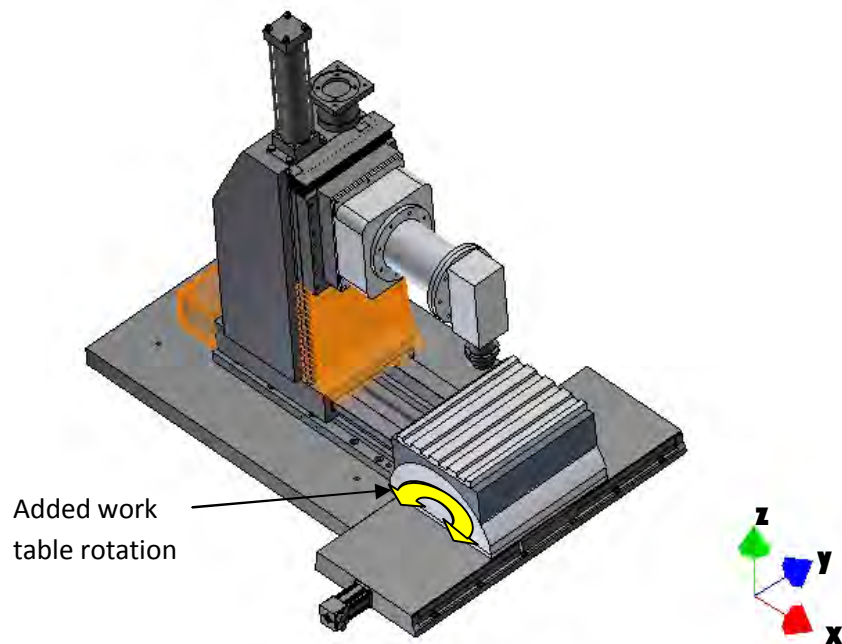


Figure 5-6: The 5 DOF configuration showing rotation about the y-axis

The following designations apply to Figure 5-7.

r = radius of rotation of work table

r_c = radius from centre point of r to the corner of the work table i.e. the radius about which the frame $\{S\}$ rotates

β = the angle from the centre line of the work table to the corner on which $\{S\}$ is mounted

$\{SR\}$ = the new position of frame $\{S\}$ after the rotation through angle α .

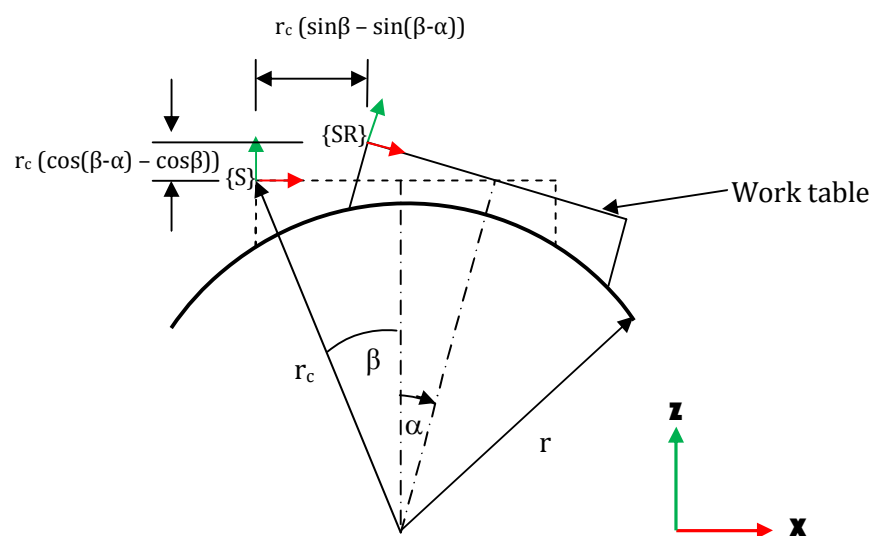


Figure 5-7: Geometry of the rotation about the y-axis

From Figure 5-7 it can be seen that $\{S\}$ has been translated and rotated to $\{SR\}$. We will use a homogeneous transform to describe this change in position and hence give the coordinates of the tool tip relative to the new position of the work table i.e. relative to $\{SR\}$. The transform is generated from:

$$\begin{bmatrix} {}^{SR}P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{SR}R_S & {}^{SR}P_{SORG} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^S P \\ 1 \end{bmatrix} \quad (5.9)$$

Where ${}^{SR}P$ is the point of the tool with reference to frame $\{SR\}$

${}^{SR}R_S$ is the rotation matrix going from $\{SR\}$ to $\{S\}$

${}^{SR}P_{SORG}$ is the vector that locates the origin of $\{S\}$ relative to the origin of $\{SR\}$

And ${}^S P$ is the point relative to {S}

We have
$${}^{SR}R = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (5.10)$$

$${}^S P_{SRORG} = \begin{bmatrix} r_c(\sin \beta - \sin(\beta - \alpha)) \\ 0 \\ r_c(\cos(\beta - \alpha) - \cos \beta) \end{bmatrix} \quad (5.11)$$

Now

$${}^{SR}P_{SRORG} = -{}^{SR}R {}^S P_{SRORG} \quad (5.12)$$

Using $s\alpha$ for $\sin \alpha$, and $c\alpha$ for $\cos \alpha$, we get

$${}^{SR}P_{SRORG} = \begin{bmatrix} s\alpha r_c(c(\beta - \alpha) - c\beta) - c\alpha r_c(s\beta - s(\beta - \alpha)) \\ 0 \\ -s\alpha r_c(s\beta - s(\beta - \alpha)) - c\alpha r_c(c(\beta - \alpha) - c\beta) \end{bmatrix} \quad (5.13)$$

$$\begin{aligned} \therefore \begin{bmatrix} {}^{SR}P \\ 1 \end{bmatrix} &= \\ \begin{bmatrix} c\alpha & 0 & -s\alpha & s\alpha r_c(c(\beta - \alpha) - c\beta) - c\alpha r_c(s\beta - s(\beta - \alpha)) \\ 0 & 1 & 0 & 0 \\ s\alpha & 0 & c\alpha & -s\alpha r_c(s\beta - s(\beta - \alpha)) - c\alpha r_c(c(\beta - \alpha) - c\beta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^S P \\ 1 \end{bmatrix} & \end{aligned} \quad (5.14)$$

From equation (5.8)

$${}^S P = \begin{bmatrix} x_c \\ \frac{sl}{2} - y_t + t_o \sin \theta \\ h - t_o \cos \theta \end{bmatrix}$$

$$\begin{aligned} \therefore \begin{bmatrix} {}^{SR}P \\ 1 \end{bmatrix} &= \\ \begin{bmatrix} c\alpha & 0 & -s\alpha & s\alpha r_c(c(\beta - \alpha) - c\beta) - c\alpha r_c(s\beta - s(\beta - \alpha)) \\ 0 & 1 & 0 & 0 \\ s\alpha & 0 & c\alpha & -s\alpha r_c(s\beta - s(\beta - \alpha)) - c\alpha r_c(c(\beta - \alpha) - c\beta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ \frac{sl}{2} - y_t + t_o \sin \theta \\ h - t_o \cos \theta \\ 1 \end{bmatrix} & \end{aligned} \quad (5.15)$$

This gives the position of the tool relative to the work table.

5.3 Inverse Kinematics

If we desire to place the tool in a certain position in space inverse kinematics are used to determine what joint translations and/or rotations are required to get the tool to that position. Calculating the inverse kinematics, is usually a more complex procedure than calculating the forward kinematics because there may be more than one solution for any given goal point.

Examining the 4 DOF case as its solution encapsulates both the 3 and 4 DOF cases.

5.3.1 Machine with 4 DOF

The task is to find the joint values for any given goal point, P_g with the desired coordinates x_g, y_g, z_g . From (5.8) it can be seen that to solve variables: h, y_t, x_c , and θ the following equations are considered.

$$x_g = x_c \quad (5.16)$$

$$y_g = \frac{sl}{2} - y_t + t_o \sin \theta \quad (5.17)$$

$$z_g = h - t_o \cos \theta \quad (5.18)$$

Initially, it would appear that there are three equations with four unknowns, but the configuration of this machine has some properties which make the inverse kinematics a solvable task. The machine is designed for processes such as milling, drilling and boring. For these processes, the angle θ will be predefined and in most cases, it will constant for each specific machining process. This leaves us with only three equations with three unknowns. Solving for the three remaining variables gives:

$$x_c = x_g \quad (5.19)$$

$$y_t = \frac{sl}{2} + t_o \sin \theta - y_g \quad (5.20)$$

$$h = z_g + t_o \cos \theta \quad (5.21)$$

5.3.2 Machine with 5 DOF

When the fifth degree of freedom is added, the procedure is somewhat more complex. Our goal point is now defined with respect to a possibly rotated work surface. If we define the goal point as

$${}^{SR}P_g = \begin{bmatrix} x_{gsr} \\ y_{gsr} \\ z_{gsr} \end{bmatrix} \quad (5.22)$$

Where, the subscript sr refers to the fact that these goal points are relative to the work table which can rotate. Since our axes of motion are predominantly with reference to the fixed station frame {S} we need to transform the goal point into

$${}^S P_g = \begin{bmatrix} x_{gs} \\ y_{gs} \\ z_{gs} \end{bmatrix} \quad (5.23)$$

so that we can more easily solve for the axes of motion.

$$\begin{bmatrix} {}^S P_g \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{SR}R & {}^S P_{SRORG} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{SR}P_g \\ 1 \end{bmatrix} \quad (5.24)$$

We have that

$${}^{SR}R = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (5.25)$$

$${}^S P_{SRORG} = \begin{bmatrix} r_c(\sin \beta - \sin(\beta - \alpha)) \\ 0 \\ r_c(\cos(\beta - \alpha) - \cos \beta) \end{bmatrix} \quad (5.26)$$

$$\therefore \begin{bmatrix} {}^S P_g \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & r_c(\sin \beta - \sin(\beta - \alpha)) \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & r_c(\cos(\beta - \alpha) - \cos \beta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{gsr} \\ y_{gsr} \\ z_{gsr} \\ 1 \end{bmatrix} \quad (5.27)$$

$$= \begin{bmatrix} x_{gsr} \cos \alpha + z_{gsr} \sin \alpha + r_c(\sin \beta - \sin(\beta - \alpha)) \\ y_{gsr} \\ -x_{gsr} \sin \alpha + z_{gsr} \cos \alpha + r_c(\cos(\beta - \alpha) - \cos \beta) \end{bmatrix}$$

From (5.8) we know that for any point

$${}^S P = \begin{bmatrix} x_c \\ \frac{s_l}{2} - y_t + t_o \sin \theta \\ h - t_o \cos \theta \end{bmatrix}$$

∴ Assuming that θ and α are predefined and solving for our other axes of motion we get:

$$x_c = x_{gsr} \cos \alpha + z_{gsr} \sin \alpha + r_c (\sin \beta - \sin(\beta - \alpha)) \quad (5.28)$$

$$y_t = \frac{s_t}{2} + t_0 \sin \theta - y_{gsr} \quad (5.29)$$

$$h = t_o \cos \theta - x_{gsr} \sin \alpha + z_{gsr} \cos \alpha + r_c (\cos(\beta - \alpha) - \cos \beta) \quad (5.30)$$

The accurate determination of the position of the tool tip is imperative in programming the tool to be able to produce parts to the required specifications. In reconfigurable systems though, these calculations are also useful in generating accurate models of different configurations of equipment. These models can be used in the design of machine tools but also in describing how different modules within reconfigurable systems interact or interface.

With regard to the tool-changer, the position of the spindle could be determined from several options:

- 1) The tool-changer could contain sensors or devices to determine the position of the spindle
- 2) The machine tool could use its position determination from the calculations above and communicate this to the tool-changer through the system if the two had a communication protocol.
- 3) Sensors outside of the two systems could be installed such as overhead cameras that communicate with each system and provide the required position information
- 4) A combination of the above solutions could be used to provide for safeguards and double checking.

The kinematic calculations are included to show an understanding of how position may be determined in the larger reconfigurable machining system and to provide a foundation upon which more comprehensive positioning and calibration systems may be developed in the future.

5.4 Chapter Summary

This chapter worked through the forward and inverse kinematics of the MRM in 3, 4 and 5-axis configurations. These calculations help determine the position of the tool tip relative to the work table.

6

The Positioning System Concept

6.1 Introduction

As part of a modular machining system, the tool-changer would enable the system to gain maximum advantage from being reconfigurable if it could easily adjust to changes in the configuration of the machining centre. It was imperative that the tool-changer had an efficient and user-friendly method of determining the position of the spindle with which it was to interact. This was not only necessary at the initial setup of the system but also after any changes were made. The more easily the tool-changing unit could adapt to any alterations in the system setup, the more reconfigurable the system became.

Several commercially available positioning systems were considered in the development of the final concept. This chapter presents the considered systems and then goes on to present the final concept that was chosen for implementation using Nintendo Wii remotes. The technical specifications of the Wii system, along with the features that make it suitable for position sensing are discussed. The chapter ends with several of the arrangements that may be considered for a positioning system using Wii remotes.

6.2 Considered Systems

6.2.1 Vision Systems

National Instruments Systems

National Instruments is known for providing systems that are easy to implement and user friendly. In researching a solution for determining the spindle positions, National Instruments presented with a “Smart Camera” system. These systems are typically used for vision inspection and pick-and-place applications. They have also been used to provide an autonomous robot with object detection and navigation capabilities [37] and a 3D adaptive welding system [38].

The quoted camera, which was the cheapest in the range, could offer a resolution of 640X480 (VGA) at 60 fps. This was a monochrome camera using CCD technology [39].

The main drawback of this system was its prohibitive cost. A quote (shown in Appendix F.1) for the camera, related software and required accessories came in at over R34 000. Apart from the fact that the system was so expensive it also offered far more capability than was required to detect position. The excessive functionality were not required, and due to the high cost, it was not considered

Festo Vision Recognition System

Festo, a well-known automation company, also offered vision systems. These systems provided for similar implementations to those from National Instruments. The systems were however mainly geared to 2D applications. The system would be able to detect the location and orientation of the spindle but extra sensors would be needed to detect the distance the tool-changer was from the spindle. The price of these systems was even more restrictive as the lower end solutions start at about R50 000 [40]. Once again the systems offered greater functionality than was necessary.

6.2.2 Laser measuring devices

Another option was to try laser distance measuring devices.

Acuity offered a device that could provide a distance measurement within 2 mm and a resolution of 0.1 mm. The cheapest device (AR1000) priced at about R2 000 could provide measurements from about 100 mm to a distance of up to 30 m. The application did not require the 30 m range but the device that offered a closer range, the AR700, was twice the price and only had a range of 1.27 m.

These devices could be useful in determining the distance from the spindle but they would be ineffective in determining the orientation of it. If they were combined with one of the vision systems a very accurate result could be obtained. The cost of the combined system would negate its implementation.

Concepts have been developed using laser technologies to provide 2D and 3D positioning but the application was more suited to a fixed work station [42].

6.2.3 Sonar

Sonar has been used to detect distance and has been used in obstacle avoidance for autonomous systems. The possibilities of using these devices for the position detection

were investigated. Sonar devices were relatively inexpensive (R280) compared to the previous options. It did require the purchase of an Arduino board to get the readings which would have cost an additional R700.

The device could only provide a resolution of 1 inch (25.4 mm) which would not be accurate enough for the positioning system. Although it could detect the presence of objects at very close range, it gave the same reading (of 6-inches) until the object was further than 6 inches (152.4 mm) away. The maximum range of the sonar was 254 inches (about 6.5 m) [41].

Although a sonar system may be economical, it could not provide the solution for the tool-changer. The tested sensors could provide an idea of the position of the spindle but it would be difficult to provide the orientation. That, along with its poor resolution, disqualified sonar as a possible solution.

6.3 The Nintendo Wii Platform

The technological advancements of the video game industry, along with the wide availability of the systems, indicated that these systems could possibly be used as an economical solution to the position sensing problem of the tool changer.

The Nintendo Wii system proved to be the best option as it was the cheapest and had been released for a longer time period than some of the newer technologies, so it was established in its design and performance. The longevity also meant that the author could further reduce the cost with the use of the second hand market.

An additional reason that the Nintendo system emerged as the optimal choice, was that it offered wireless communication through Bluetooth. This made the implementation of the system easier as it would not require the concerns of extra power supplies or cabling.

A positioning system using the Wii could be built for up to 1/50th of the price of some of the commercially priced vision systems.

The main component of the position sensing system of the Wii platform is the Wii Remote or Wiimote. The remote uses a 3-axis accelerometer along with an infrared (IR) camera to allow the console to 'read' the motions. A 'sensor bar' containing a 5 IR LED cluster at each end is mounted above or below the screen and is plugged into the console. As the remote is moved, the console picks up the relative motion of the camera in the remote compared to the stationary LEDs to determine the movement [46].

The latest version of the Wiimote called the Wii Remote Plus incorporates gyroscopes to further enhance the gaming experience. A separate module, the Wii Motion Plus, can be added to the previous generation Wiimote to offer this added functionality. The console is shown in figure 6-1 [47] and the sensor bar is shown in figure 6-2 [48].



Figure 6-1: The Nintendo Wii Remote and Wii Console












Figure 6-2: The Wii Sensor bar showing the two LED clusters

6.4 The Nintendo Wii Remote

The Wii Remote was the only part of the Nintendo system that was required in order to provide the tool-changer with a position sensing capability. This section therefore gives an overview of the hardware specifications of the Wiimote along with some of the details regarding its Bluetooth communication capability.

Nintendo has not released the technical details of the Wii system or remote but its popularity and innovativeness has resulted in the successful reverse engineering of the hardware by several online communities [49, 50]. Table 6-1 [55] shows the progress of the reverse engineering process. This table indicates the information which is retrievable from the Wiimote and which could be used by the developer.

Table 6-1: Wiimote reverse engineering status

<u>Bluetooth Communication</u>		Connecting to the Wii Remote and listening for connections works.
<u>Core Buttons</u>		All working.
<u>Accelerometer</u>		All working.
<u>IR Camera</u>		All working.
<u>Power Button</u>		All working.
<u>Speaker</u>		All working.
<u>Player LEDs</u>		All working.
<u>Status Information</u>		Battery and extension info in Status Report
<u>Extension Controllers</u>		Official extensions are supported, however many 3rd party extensions are not understood.

Legend	 Perfect or near-perfect	 Usable but not complete	 Unusable
---------------	---	---	--

6.4.1 Bluetooth Communication

The Wii Remote is a wireless user input device much like a keyboard or a mouse and this is confirmed by its Bluetooth specification as a Human Interface Device (HID). It offers two-way communication between itself and a host. Although the original intended host was the Wii console, communications can also be set up with a PC using an appropriate Application Programming Interface (API). The open source community has developed APIs that allow it to interact with all the major operating systems [52].

The chip used for the Bluetooth communication is a Broadcom 2042 commonly used in mouse and keyboard devices. The chip is an all-in-one solution offering the profile, application and the Bluetooth protocol stack on one chip. It uses Bluetooth 2.0 and includes an on-board 8051 processor and RAM/ROM memory [53].

No authentication or encryption requirements are necessary for the Wii Remote to connect to a host. The connection to a host is achieved by placing the Wiimote in discoverable mode by simultaneously pressing buttons 1 and 2 simultaneously on the device (Fig. 6-3) [51]. The Wiimote will wait in this state, with all 4 LEDs blinking, for 20 seconds. If the driver on the host does not connect to the remote during this time it will turn off [54].

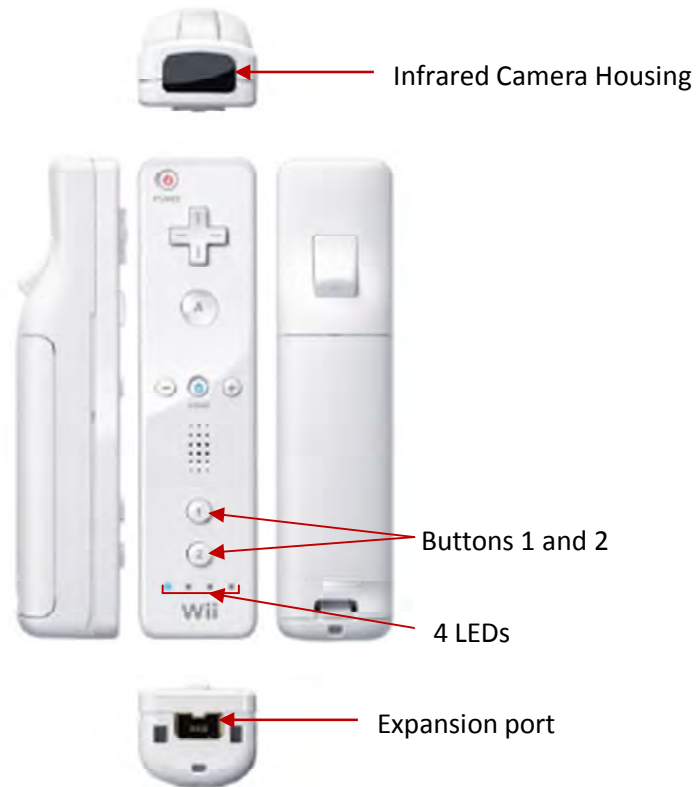


Figure 6-3: Wii Remote

As an HID peripheral, Wiimotes are self-describing using an HID descriptor block. This block provides the information required to identify the Wiimote and its status by the host. The HID descriptor block is called by the Service Discovery Protocol (SDP) which is the Bluetooth standard [46].

The descriptor block provides the basic identification information shown in Table 6-1 [54] along with many other details.

Table 6-2: Basic Wiimote Identification report

Name	Nintendo RVL-CNT-01
Vendor ID	0x057e
Product ID	0x0306

The information in table 6-2 is what uniquely identifies the Wiimote with regards to all other Bluetooth devices. Every Wiimote also has its own mac address which enables the host to identify each remote separately during communication [46].

The HID descriptor block contains an array of other reports which are useful when interacting with the Wiimote. Table 6-3 [55] shows the reports used by the Wiimote. In the convention shown below in the table, an ‘Output’ report is one sent from the host to the Wiimote and an ‘Input’ report is sent to the host from the remote.

The freely available API’s decode the packets sent from the Bluetooth module and provide the user with the values read from the reports. No decoding was necessary by the user. API’s are available as independent modules such as GlovePIE. Libraries in several programming languages which read the values from the Wii remotes are also available. Calculations on the values can either be done within the API or the programming library can be used to develop a more comprehensive user interface.

Table 6-3: Wiimote reports

Report Type	Report ID(s)	Size in bytes	Function
Output	0x11	1	Player LEDs
Output	0x12	2	Data Reporting mode
Output	0x13	1	IR Camera Enable
Output	0x14	1	Speaker Enable
Output	0x15	1	Status Information Request
Output	0x16	21	Write Memory and Registers
Output	0x17	6	Read Memory and Registers
Output	0x18	21	Speaker Data
Output	0x19	1	Speaker Mute
Output	0x1a	1	IR Camera Enable 2
Input	0x20	6	Status Information
Input	0x21	21	Read Memory and Registers Data
Input	0x22	4	Acknowledge output report, return function result
Input	0x30-0x3f	2-21	Data reports

6.4.2 Accelerometer

A source of motion detection of the Wiimote is provided by the Analog Devices, ADXL330 three-axis accelerometer. The integrated circuit is rated to measure accelerations of ± 3 g and have a sensitivity of 10% [56].

The chip uses silicon springs which support a micromechanical structure to detect the acceleration with respect to gravity. It does not measure the acceleration directly, instead it measures the forces on the springs due to the mass of the structure. This means that when the Wiimote is at rest, the upward force due to gravity (1 g) is indicated by the circuit. To get a zero reading the Wiimote has to be free falling. The values are generated by the chip from taking readings of the differential capacitance due to the movement of the mass. These readings are converted to a voltage and then digitised [54]. This method of obtaining the accelerometer readings allows the circuit to give angles of tilt values.

In terms of the layout in the Wiimote, the accelerometer uses the coordinate system as shown below in figure 6-4 [55].

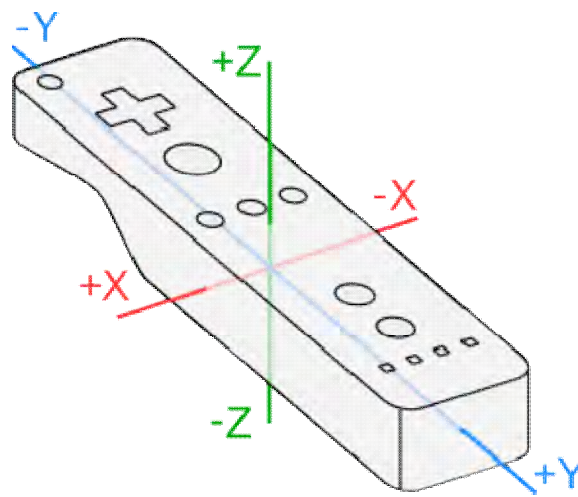


Figure 6-4: Wii Remote Accelerometer coordinate system

6.4.3 Infrared Camera

The IR camera is mounted on the front of the remote and is the key to position detection of the Wiimote relative to the screen. The camera offers an output resolution of 1024 x 768 but is apparently a 128x96 camera with an 8x sub pixel analysis [55]. It has an inbuilt image processing capability which allows it to monitor up to 4 IR 'blobs' or hotspots' at the same time.

One of the main advantages of this camera is its speed. It operates at 100Hz which over three times as fast as a webcam of similar price. The resolution of a webcam would also typically only be 640 x 480 [52] which is over two-and-a-half times less than the Wiimote camera.

The camera uses Charged Coupled Device (CCD) technology with an IR filter that allows it to sense light at wavelengths of greater than 800nm. Any IR source can be tracked by the

Wiimote but IR LEDs are particularly suited for tracking by the camera [46]. Several IR LEDs have been tested and reports are that those emitting 940nm are detected with twice the intensity of similar LEDs with a 850nm spec. The 940nm LEDs are however, not as successful at close range [55].

In terms of the field of view (FOV) of the camera, there are differing reports, with horizontal values varying from 33° to 45° [55, 52] and vertical values varying from 23° to 31.5° [55, 57].

Another advantage of the Wiimote camera is that rather than sending the entire image across the Bluetooth link it only sends the information regarding the IR hotspots. This reduces the wireless information traffic and hence the risk of bottlenecks occurring in the transmission [46].

6.4.4 Power requirements and feedback features

The Wiimote uses 2 AA size batteries which can power the unit for between 20 and 40 hours depending upon the usage [52].

Several feedback features are available to the user of the Wiimote. The 4 LEDs at the bottom end of the remote are used to indicate

- a) The battery level – When placed in discoverable mode, the number of LEDs blinking gives the level of the batteries with each of the 4 LEDs indicating 25% of the power i.e. if 4 LEDs are blinking the batteries are at full power.
- b) The player number – During game play one LED shines constantly representing the number assigned to the Wiimote by the host. The first Wiimote results in the first LED being on the second Wiimote has the second LED etc.

The LEDs are fully controllable by the host and may represent anything that the developer desires [55].

The Wii Remote also has a rumble or vibrate feature which is accessible to the developer through the available APIs.

Lastly, the device has an in built speaker which can be used for sound effects or vocal feedback. Not all APIs have successfully got this to work without some distortion though.

6.5 Wiimote positioning in the Nintendo Wii system

The Nintendo Wii system uses the combination of the sensor bar and the IR camera in the Wiimote to determine the position of the remote compared to the TV screen. The sensor bar, which is about 20 cm long, has two clusters of IR LEDs which are read as two IR hotspots by the optical sensor of the Wiimote. The two clusters are a known distance apart and the Nintendo Wii system can therefore calculate the 3D position of the remote relative to the screen using triangulation. The pixel readings from the camera will show the distance apart in pixels of the two hotspots. This value combined with the knowledge of the fixed value of the distance between the clusters can give the distance the remote is away from the screen [58].

Although the angle of the Wiimote can be determined by the accelerometers, the optical system is also able to detect the angular twist of the device by calculating the angle between the IR hotspots. Furthermore, the two clusters enable the system to detect slow forward and backward motion of the Wiimote [59]. For faster back-and-forth motions the accelerometers are used [60]. These features are shown in figure 6-5 [59].

The remote offers several variations of the IR report, some of which also include the size of the IR hotspots [55]. This data can be used to determine the horizontal angular position of the remote as shown in figure 6-6 [58].

It is clear that the Wiimote is a very versatile positioning sensor. The following section presents the positioning options that could be used in the tool-changer application.

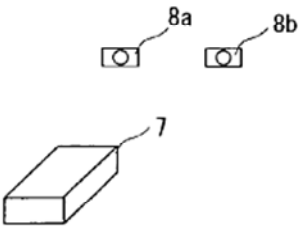
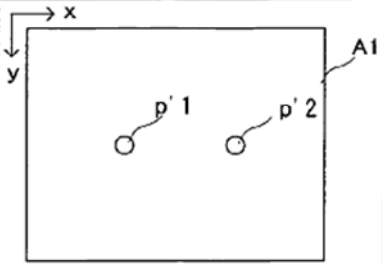
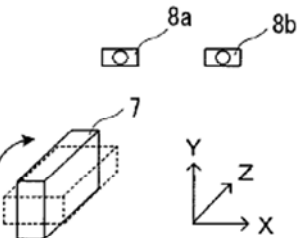
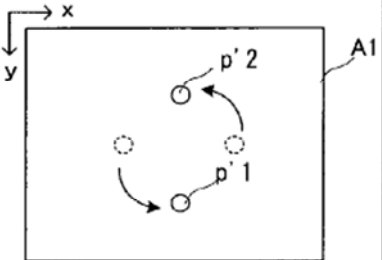
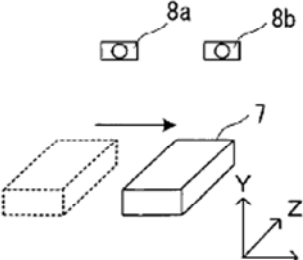
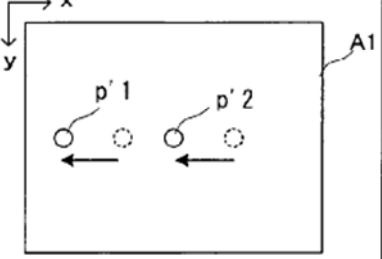
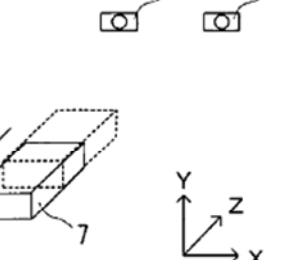
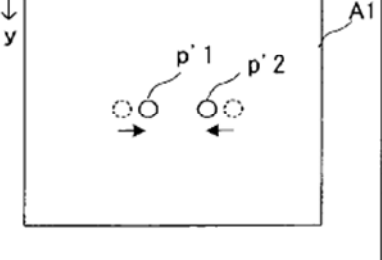
	STATE OF CONTROLLER	MARKER COORDINATE POINTS
STATE A		
STATE B		
STATE C		
STATE D		

Figure 6-5: Various Wiimote positional readings

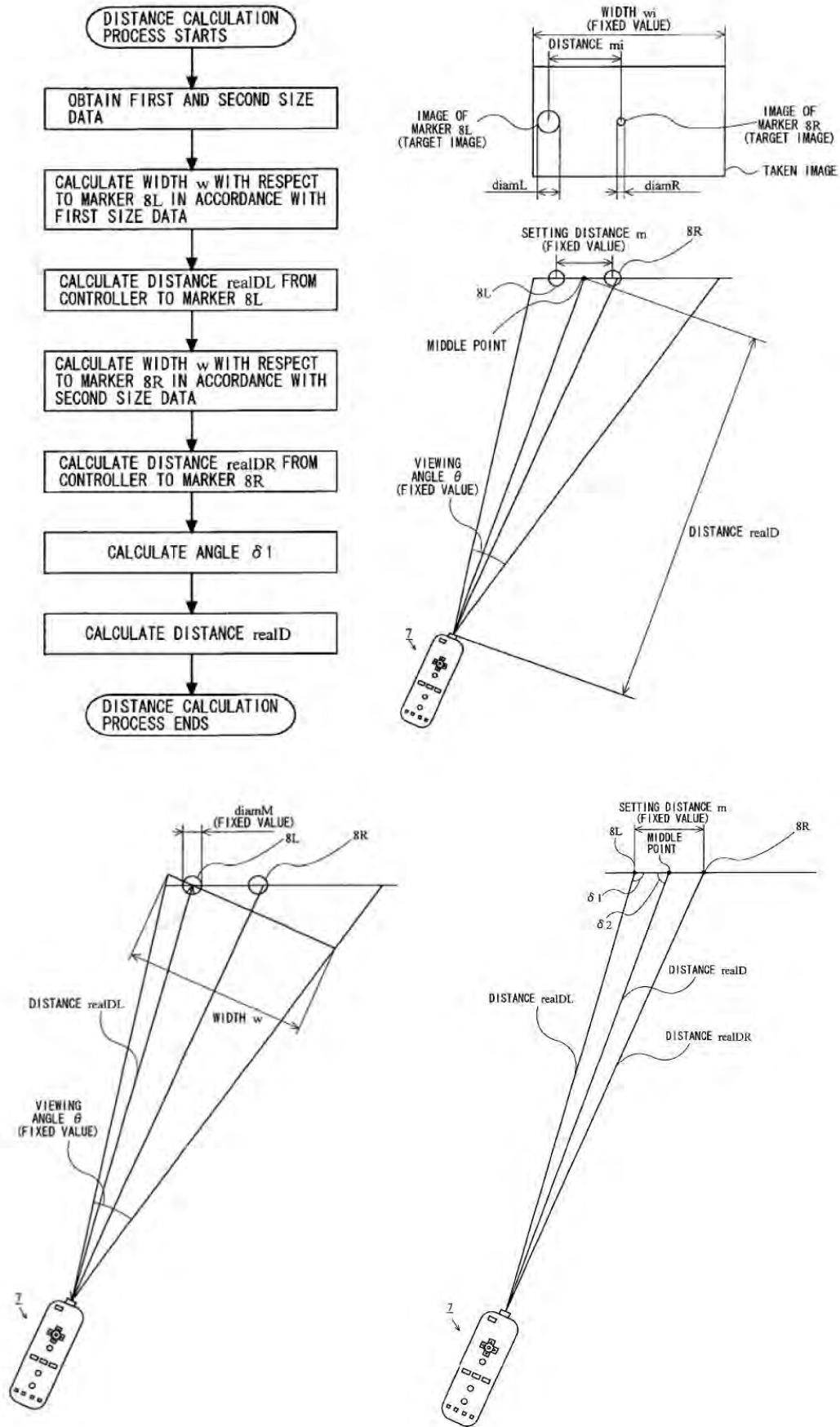


Figure 6-6: Horizontal Angular Determination of the Wiimote

6.6 Wiimote positioning concepts

6.6.1 Single Wiimote, Single LED

This is the most basic configuration and the Wiimote is only able to return the x and y pixel values (x_p , y_p) of the LED that is in vision (as is shown in figure 6-7). This is only useful in the case that the distance from the LED is known. There may be cases where this could be useful but for the tool-changing unit application a more complete solution is required.

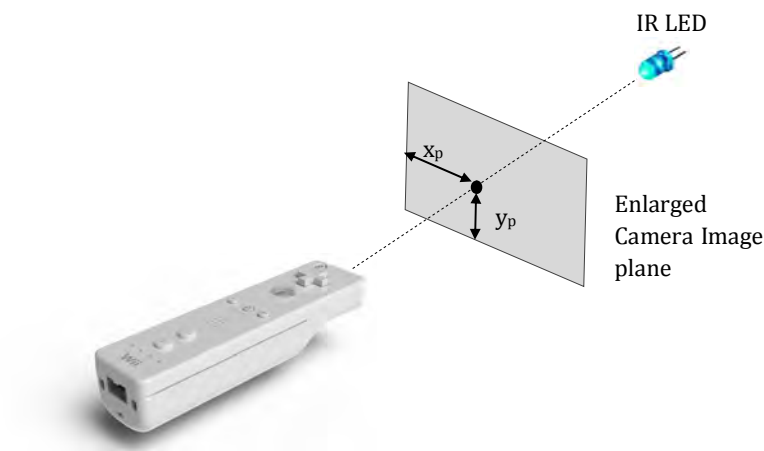


Figure 6-7: Single Wiimote and single LED positioning

6.6.2 Two Wiimotes, Single LED

By using two Wiimotes, the coordinates of the LED can be calculated in three dimensions. This setup represents stereoscopic vision similar to that used by the human eyes, as is shown in figure 6-8 [61].

In this case, the Wiimotes are set a known distance apart, dx and are parallel to each other. The pixel values which are recorded are in 2D, but since there are two pictures of the same hotspot, the 3D position of the LED can be calculated.

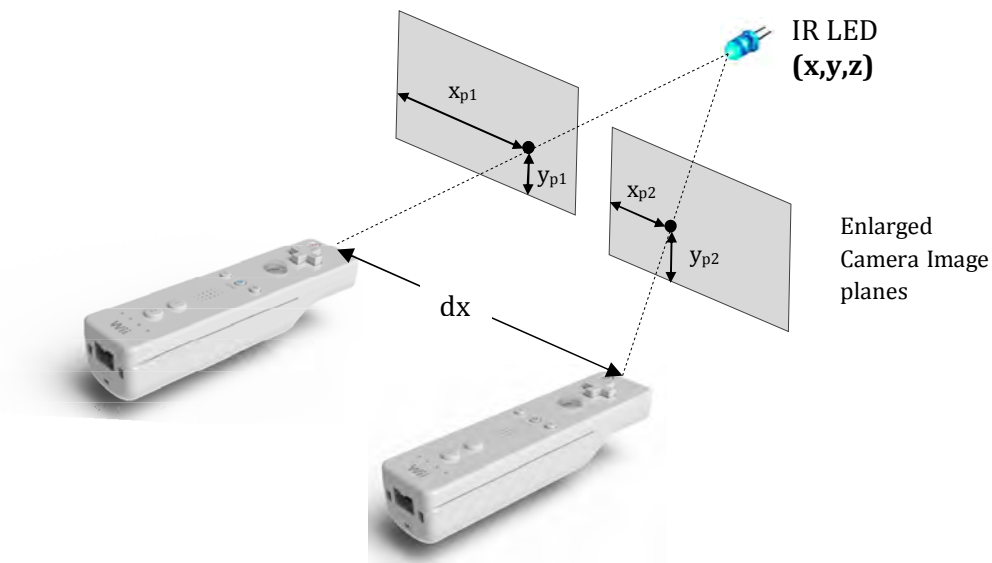


Figure 6-8: 3D Stereoscopic positioning - Two Wiimotes and one LED

6.6.3 One Wiimote with two LEDs

In this third case, only a single Wii Remote is needed, as shown in figure 6-9. Two IR LEDs with a fixed distance between them are used. The camera picks up the two pixel points (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}) . As the Wiimote moves away from the LEDs, the distance between the two points on the camera image plane decreases. This is a linear relationship if the effects of camera lens distortion do not come in to play. Typically, lens distortion is most severe at close range [46].

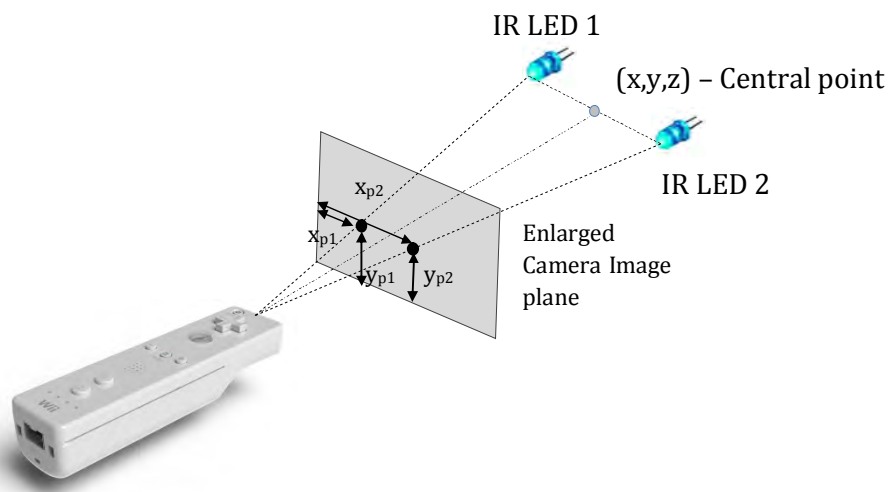


Figure 6-9: 3D positioning with a single Wiimote

The concept relies on the two LEDs being aligned to the camera image plane. If this is not the case then accuracy will be compromised [61]. Note that this system does not give the position of either of the LEDs but rather, the position of the midpoint between them.

The advantage of this system is that since only one remote is used, setup is simpler. One does not have to be concerned with making sure that the two remotes in the previous concept are perfectly aligned and the correct distance apart.

Other advantages are that it is economically more effective in that only one remote has to be purchased. The mathematics to work out the 3D position is also not as complicated as in the previous case.

A significant advantage of this system is that it could also be used to determine the angle of the LED's. This setup was particularly useful to the tool-changer as it could be used to provide the angle of the spindle and therefore take advantage of the additional degree-of-freedom provided by the tool-changer. (See Chapter 7 section 7.3.3, p.67)

Other research has determined that this concept might not be as robustly accurate as the stereo Wiimote system [61].

Both the stereo Wiimote system and the single Wiimote with dual LED system were tested for the positioning system of the tool changing unit.

6.7 Wiimote Location on the tool-changer

In order for the Wiimotes to accurately record the position of the spindle with which the tool-changer was interacting, they needed a stable mounting on the tool-changing unit. The initial thought was to mount them alongside the gripper, one Wiimote on each side as shown in figure 6-10. This would place the Wiimotes about 100 mm apart.

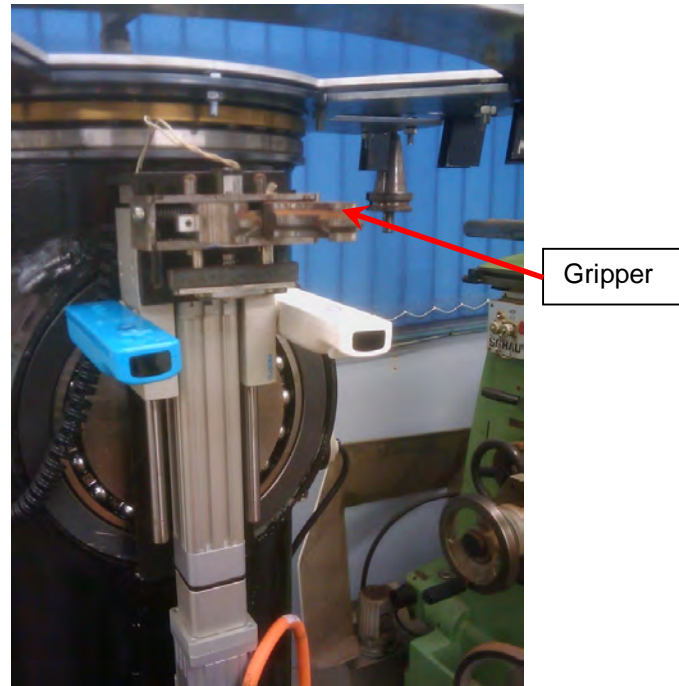


Figure 6-10: Wiimotes mounted on either side of gripper

The advantage of this setup would be that since the Wiimotes would be attached to the gripper arm they would move as the arm moved. This would mean that the Wiimote positioning system would provide constant real-time tracking of the target relative to the gripper.

Unfortunately, the field of view restrictions of the IR cameras meant that this location would limit the range and effectiveness of the positioning system. The cameras would not be able to 'see' the spindle in all the required locations. This location concept for the Wiimotes had to be rejected.

It was decided therefore, to place the Wiimotes above the tool holding carousel. This setup required the manufacture of a mounting support. The mounting was required to be designed in such a way that could be fixed to the structure of the tool-changer but still allow the carousel to freely rotate. It was imperative that any motion of the carousel would not change the position of the Wiimotes.

This alternative arrangement would allow the Wiimotes to read the position of the spindle relative to the the tool-changing unit, as shown in figure 6-11. It would still offer a real-time reading but the value would not change as the gripper arm moved. The tool-changing unit would have to rely on the feedback provided from the Festo drives themselves to validate the position of the gripper during a tool change. Although this configuration

would not allow the Wiimote system to give a real-time gripper-spindle position, the Festo drives were accurate enough to provide the necessary information.

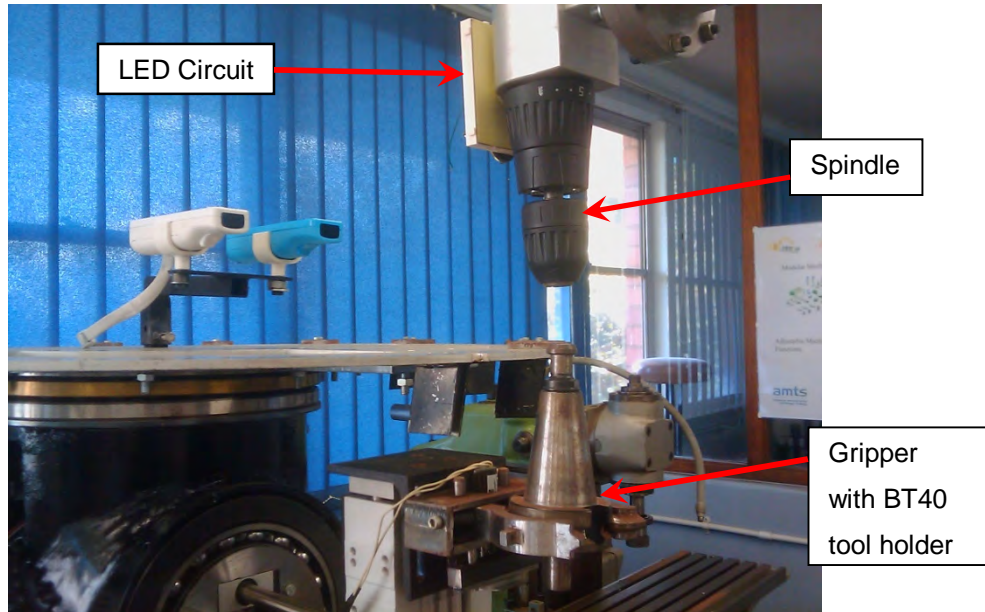
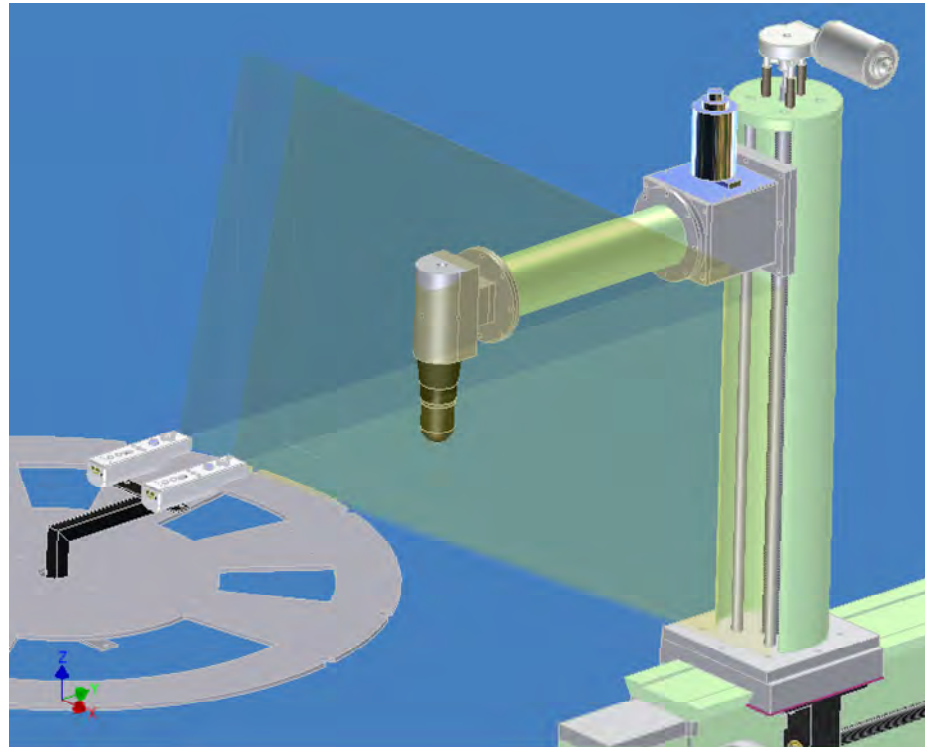
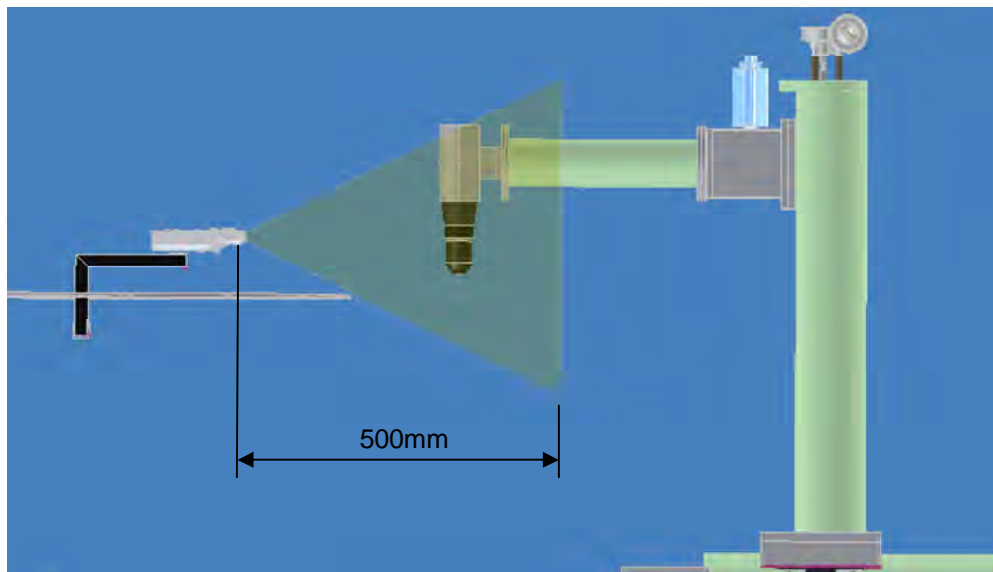


Figure 6-11: Wiimotes mounted above the Carousel

The big advantage of this setup was that the Wiimotes would have maximum range of view with respect to the spindle. This is shown in figure 6-12 where the FOVs of the Wiimotes are shown with respect to the MRM. Note that in the figure, the FOVs are only drawn out to a distance of 500mm.



a) Isometric view



b) Side View

Figure 6-12: Wiimote FOV interaction with the MRM spindle

6.8 Chapter Summary

In this chapter, the author presented the various commercially available options considered in solving the positioning of the tool-changer. It then went on to describe how the concept of using the Wiimotes for the positioning challenge emerged. The technical aspects of the Wii system were presented along with possible positioning concepts using the Wiimote.

7

Software Design

7.1 Introduction

In order for the Wiimotes to be integrated into the tool-changing system, software had to be developed to allow for the integration. Readings needed to be captured from the Wiimotes, which needed to be translated into a 3D position. The position needed to be used to issue commands to the tool-changer.

This chapter presents the positioning of the Wiimotes on the tool changer and the development of the software required to integrate the Wiimote positioning system into the broader tool-changing unit control system.

7.2 Initial API selection

The capturing of readings from the Wiimotes onto a PC was required. As stated in chapter 6, the open source community has reverse engineered the Wii remote and developed API's that allow for this possibility. A PC running the Windows XP operating system was chosen to be used as the host, since Windows is still the most common operating system in use [63]. Connection between the Wiimotes and the PC was provided through the inbuilt Bluetooth capability of the remotes and the Microsoft Bluetooth Stack on the PC.

A suitable API needed to be chosen that would allow for the capturing of data from the Wiimotes and to manipulate it. The API chosen in the initial implementation was an application called GlovePie. GlovePIE is a freely available Programmable Input Emulator developed by Carl Kenner [64]. Version 0.43 of the software was used for this research.

The development of the emulator was initiated by Carl's desire to allow the P5 Glove device to interact with the computer rather than using a keyboard or mouse. GlovePIE's capabilities were later expanded to include other devices including the Wii remote [65]. The emulator converts input from the Wiimote and maps those commands to commands that would be issued by more traditional devices such as a joystick, mouse or keyboard.

GlovePIE uses a scripting language to make this possible. Users have mostly used GlovePIE to allow for novel interaction in games.

GlovePIE was chosen as the first test bed for the Wiimote positioning system because it was simple to install, light-weight and had a quick learning curve. It did have difficulty however, in connecting more than one Wiimote. There seemed to be no set recipe to connect two Wiimotes. It was a bit of a hit-or-miss affair. It was possible to connect both Wiimotes in every use, but it required a bit of patience.

The emulator provided access to the remotes functions and outputs. Calculations were performed on the readings received, within GlovePIE, and the resulting position of the IR LED could be displayed within the application.

7.3 Calculating the 3D position

The infrared cameras on the remotes output an x and y pixel value of the IR LED. Calculations have to be performed on these values to give the 3D position of the LED. This section shows how the 3D position of the LEDs is calculated from the camera readings.

7.3.1 Camera Characteristics

The first step in the process was to determine the camera characteristics that affect its view of the world and hence the correlation to position in 3D space. For this research the cameras were modeled to be pinhole cameras and no effects of lens distortion were considered.

The specifications of the camera that were required to perform the calculations are the horizontal and vertical field of vision angles. As previously mentioned, research into these specifications yielded a variety of results. It was therefore necessary to perform tests to determine these values. After testing and comparing with other research the following camera angles were used in the calculations:

Horizontal field of view (φ) = 41°

Vertical field of view (θ) = 31°

7.3.2 Two Wiimotes, Single LED

The first positioning setup to be tested was the case using two Wiimotes and one LED. This configuration was the first one implemented because it was suggested to be the most accurate of the configurations [46, 61].

The IR cameras on the two Wii remotes were used to provide a stereoscopic 3D position sensing system. A single IR LED was used as the target point representing the position of the spindle. The Wii remotes were set parallel to each other with an offset in between them. Each remote is able to record an (x_p, y_p) pixel position of the IR LED. The centre point of the left most Wii remote was considered as the origin with the right remote a distance of 100mm away in the x-direction. The intersection of the two rays originating from the two remotes gives the 3D position of the IR LED. Figure 7-1 shows the principle.

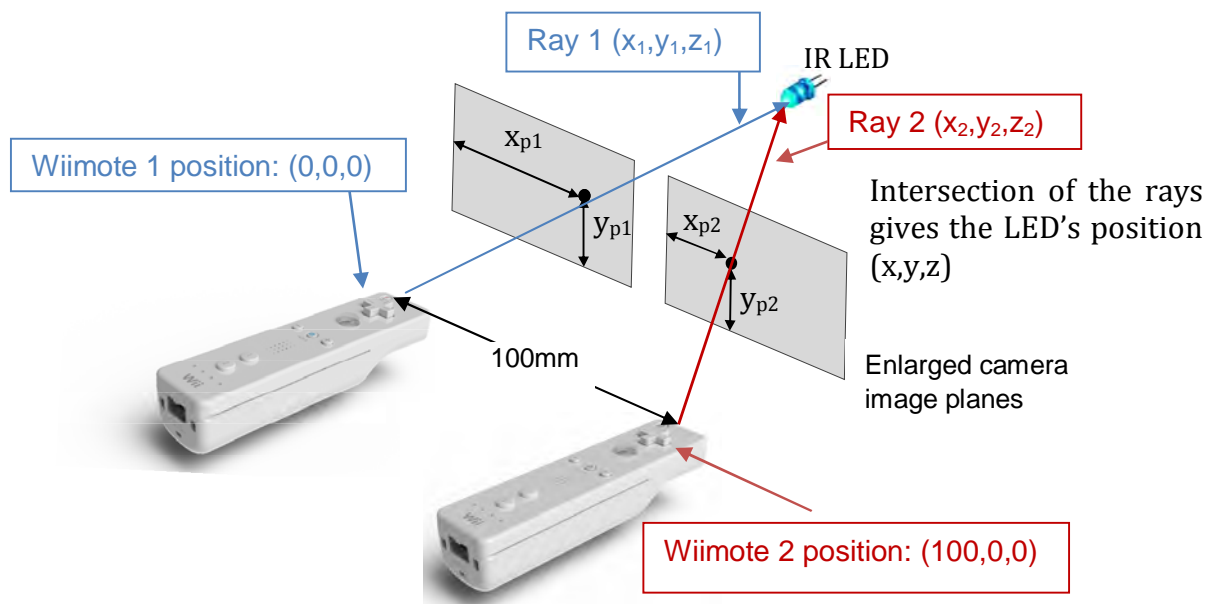


Figure 7-1: Wiimote stereo vision setup (adapted from [71])

In terms of the setup on the tool changer, the two remotes were onto the tool changer as shown in the previous chapter, and the IR LED was mounted onto a suitable reference point on the spindle.

Figure 7-2 shows the steps required in order to determine the 3D position of the single LED source in this setup.

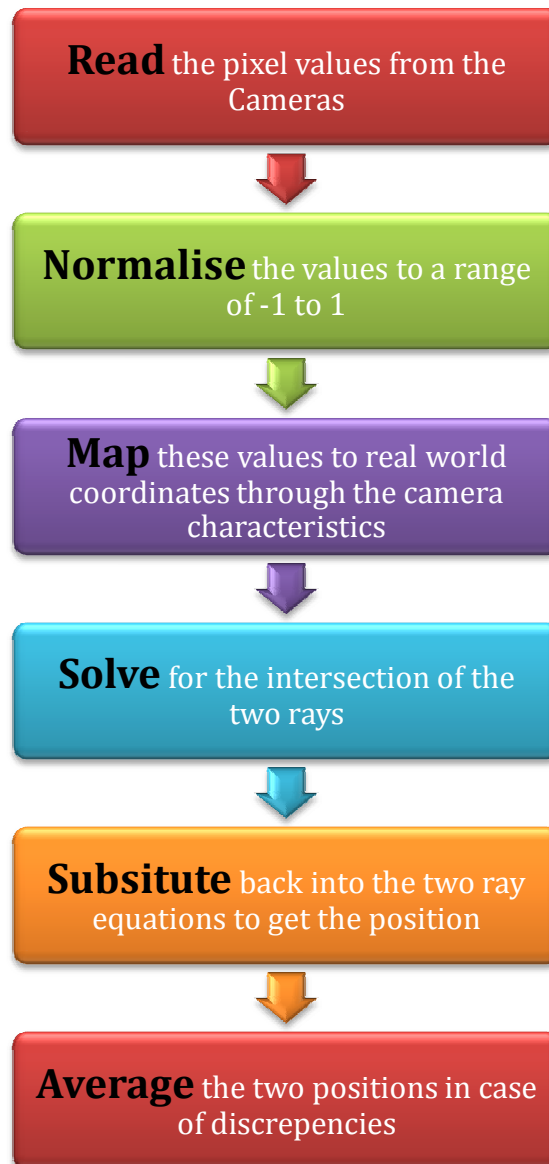


Figure 7-2: Steps to determine the stereoscopic 3D position

7.3.2.1 Normalizing the pixel values and ray mapping

A convenient mathematical model in working with the pixel values (x_p, y_p) is to normalize them with respect to the horizontal and vertical resolution of the camera [46]. The normalized values \hat{x} and \hat{y} would be in the range $[-1:1]$. That would mean that a normalized point of $(0,0)$ would be the centre of the field of vision of the camera.

As previously mentioned the stated resolution of the Wii remote camera is 1024x768. The remotes used in this research only delivered pixel values of 1016 x 760. The normalizing equations are therefore:

$$\hat{x} = 2 \frac{x_p}{1016} - 1 \quad (7.1)$$

$$\hat{y} = 2 \frac{y_p}{760} - 1 \quad (7.2)$$

These unit values must then be translated into real world coordinates through the camera characteristics. A unit ray starting at the lens origin and going through $[x',y',z']^T$, can be generated. Using the FOV angles of the camera the parameters are calculated as follows:

$$x' = \hat{x} \cdot \tan\left(\frac{\varphi}{2}\right) \quad (7.3)$$

$$y' = \hat{y} \cdot \tan\left(\frac{\theta}{2}\right) \quad (7.4)$$

$$z' = 1 \quad (7.5)$$

7.3.2.2 The Intersection of the two rays

Since the two remotes are parallel to each other, no adjustment or rotation matrix is required. Each ray is therefore defined by an origin point \mathbf{o} , and a unit direction vector \mathbf{d} [66]. Expressing the rays in this way results in:

$$\begin{aligned} R_1(t_1) &= \mathbf{o}_1 + \mathbf{d}_1 t_1 \\ R_2(t_2) &= \mathbf{o}_2 + \mathbf{d}_2 t_2 \end{aligned} \quad (7.6)$$

where:

$$\mathbf{o}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{d}_1 = \begin{pmatrix} x'_1 \\ y'_1 \\ 1 \end{pmatrix}$$

and

$$\mathbf{o}_2 = \begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{d}_2 = \begin{pmatrix} x'_2 \\ y'_2 \\ 1 \end{pmatrix}$$

The units for the calculations are mm.

Solving for t_1 and t_2 at the intersection point gives:

$$t_1 = \frac{\det(\mathbf{o}_2 - \mathbf{o}_1 \ \mathbf{d}_2 \ \mathbf{d}_1 \times \mathbf{d}_2)}{|\mathbf{d}_1 \times \mathbf{d}_2|^2} \quad (7.7)$$

$$t_2 = \frac{\det(\mathbf{o}_2 - \mathbf{o}_1 \ \mathbf{d}_1 \ \mathbf{d}_1 \times \mathbf{d}_2)}{|\mathbf{d}_1 \times \mathbf{d}_2|^2} \quad (7.8)$$

Substituting back into the equations in (7.4) gives the position of target LED. If there are inaccuracies in the physical setup of the remotes e.g. they are not exactly parallel, this algorithm will generate the closest points on the two rays. In this case, the target's coordinates can be given by the average of the coordinates of the two solutions.

Appendix C.1 shows a sample calculation of how this is implemented with the system.

Appendix D.1 shows the GlovePIE script listing for this setup.

7.3.3 Single Wiimote, Two LEDs

Research presented this setup as having some anomalies in determining distance from the Wiimote [61]. This configuration was considered as it would offer the added capability of being able to determine the angle of the spindle. It also offered a simpler setup in that there was only one Wiimote and the alignment issues of the two Wiimote arrangement would not be present. Figure 7-3 [67] shows the single Wiimote arrangement in terms of calculating the distance (z) that the remote would be from the LED centre point.

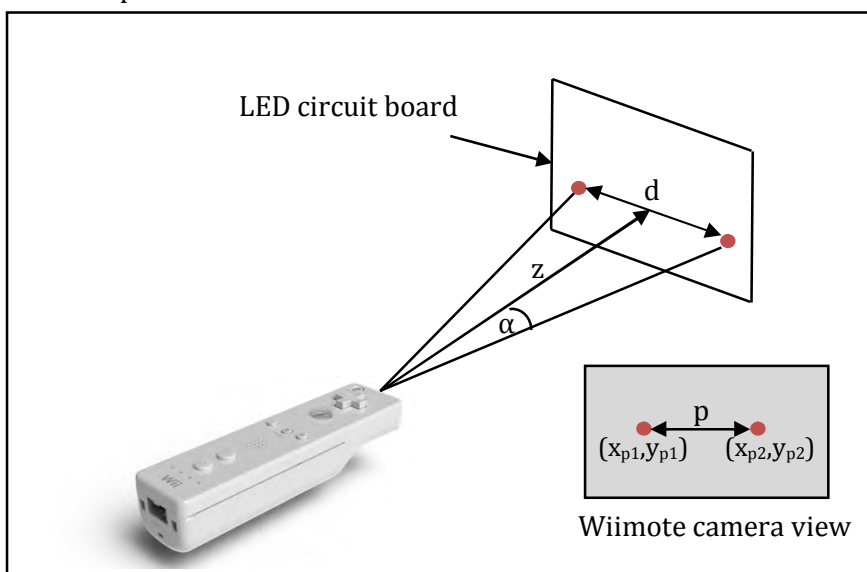


Figure 7-3: Single Wiimote distance calculation

From trigonometry, the distance z that the Wii remote is from the LEDs is given by:

$$z = \frac{d}{2 \tan(\alpha)} \quad (7.9)$$

Where d is the known distance between the IR LEDs and α is the angle subtended by the LEDs and the Wii camera.

The first step in finding α , is to determine the distance apart (p) in pixels between the two points (x_{p1}, y_{p1}) and (x_{p2}, y_{p2}) and this is given by:

$$p = \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2} \quad (7.10)$$

An angle per pixel (β_p) is related according to the resolution of the camera and the camera field of views (FOVs). In the cameras used, 1016 pixels horizontally with the horizontal FOV (φ) = 41° and 760 pixels vertically with the vertical FOV (θ) = 31°, gives:

$$\beta_p = \frac{\frac{\varphi}{1016} + \frac{\theta}{760}}{2} = \frac{\frac{41}{1016} + \frac{31}{760}}{2} = 0.0406^\circ \text{ per pixel} \quad (7.11)$$

The angle created by the LEDs is 2α . Relating that to the values calculated in equation 7.11 gives:

$$2\alpha = p\beta_p = \beta_p \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2}$$

Therefore:

$$\alpha = \frac{\beta_p \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2}}{2} \quad (7.12)$$

Substituting back into equation (7.9) yields the distance (z) between the camera and the LEDs.

A calculation is shown in Appendix C.2 using sample values. The calculations also give an example of the angular calculation.

The GlovePIE script listing for this arrangement is shown in Appendix D.2.

The flowchart shown in figure 7-4 shows the progression from reading the pixel values to determining the distance in the z -direction.

To calculate x and y coordinates, the methodology differs slightly from the stereo Wiimote setup in that the two pixel values are averaged first not last. The pixel distance from the centre of the camera (508,380) is then calculated before plugging in the angular FOV.

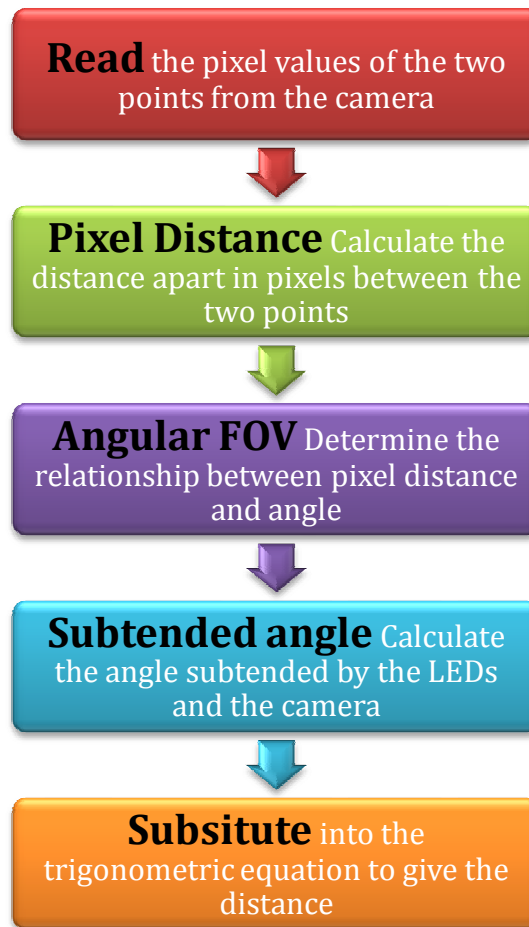


Figure 7-4: Distance determination with Single Wiimote

7.4 GUI Development

GlovePIE was a suitable interface for testing the positioning concept but an integration with the rest of the control system was required for a more user-friendly and complete interface.

The chosen development environment was Microsoft Visual Studio using Visual C# 2010 Express. There were several reasons for this choice. Visual Studio Express is a freely available Integrated Development Environment (IDE) which would reduce development costs. C# is designed for a graphical interface. Visual C# offered a user friendly graphical development environment as well as some of the programming power available to a C or C++ programmer.

Another reason that Visual C# was chosen was that a language was required in which libraries for accessing Wiimotes and their functions had been written. Developers have written libraries for interfacing a PC and a Wiimote for several programming languages [68]. Although there were other options, C# offered a quicker learning curve for the intended application.

The managed library developed by Brian Peek [69] was selected as the basis for reading data from the Wiimotes in the GUI. Version 1.7 of the WiimotLib was used. Aspects of the Wii Device Library v 1.2 [70] were also used in the GUI development.

Figure 7-5 shows the flow of processes in the automatically calibrated tool-change.

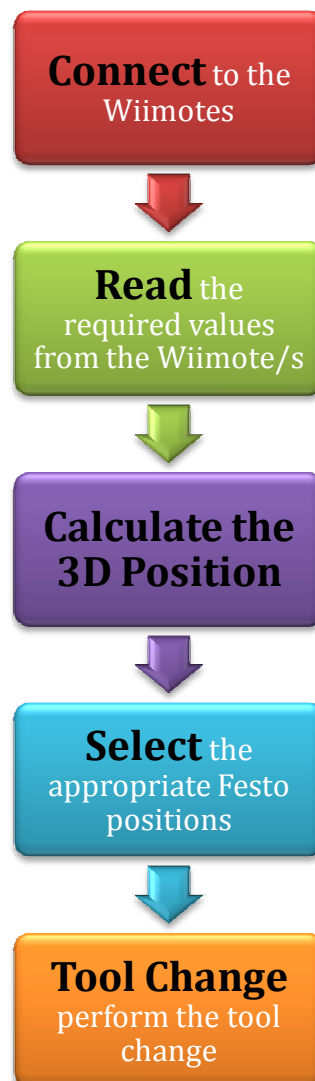


Figure 7-5: Steps required in the automatically calibrated tool-change

7.4.1 Connecting

The first requirement of the GUI was that it would provide an efficient means of connecting the Wiimotes to the PC via the Bluetooth stack. As described in section 6.4.1 the Wiimotes can be placed in discoverable mode by simultaneously pressing the 1 and 2 buttons on the Wiimote. Each Wiimote needed to be added as a Bluetooth device to the Microsoft Bluetooth stack by placing it in discoverable mode and following the add device wizard. The 'Don't use a passkey' option had to be selected in the final option window.

Once the Wiimotes were added as devices, they could be connected to the GUI by using the Connect button at the top right corner in the Main Window interface (fig. 7-6).



Figure 7-6: Tool Changer GUI - Opening screen

Once the connect button is pressed the program attempts to connect to any available Wiimotes.

7.4.2 Read

As the GUI connects to each Wiimote, it vibrates the remote to confirm that it has connected. It then opens an information panel for that remote. Figure 7-7 shows the full GUI open with two Wiimotes connected.

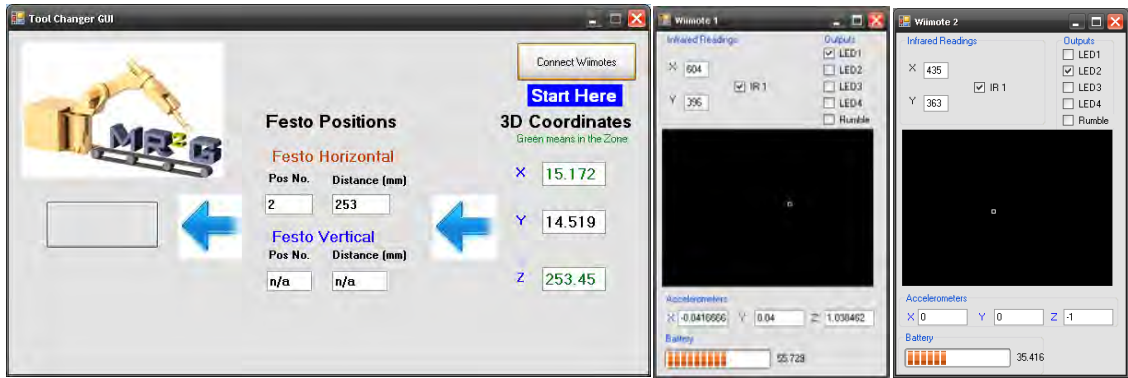


Figure 7-7: Full GUI with 2 Wiimotes connected

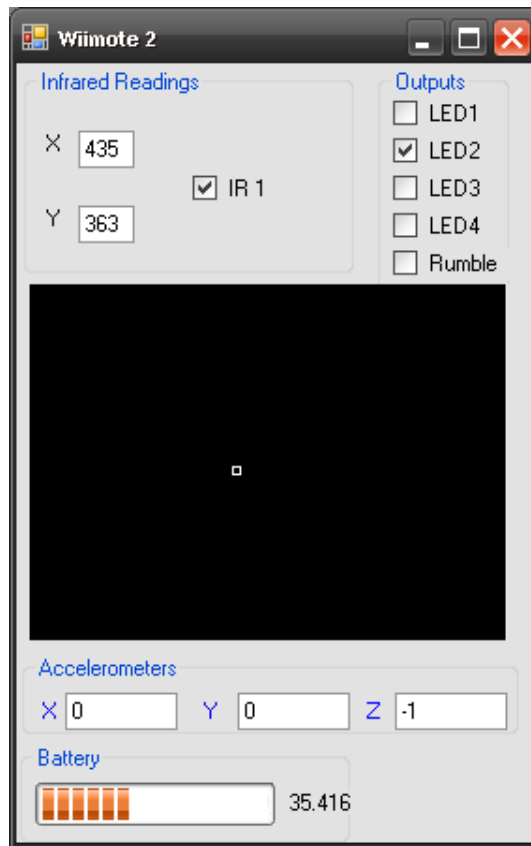


Figure 7-8: Wiimote information panel

A Wiimote information panel is shown in Figure 7-8. The black square represents the camera image plane of the Wiimote and the small white square is a graphical representation of the position of the IR LED. The Infrared Readings section outputs the pixel values of the LED location and the checkbox confirms the presence of an IR source. The accelerometers are helpful in setting up the Wiimotes since they give an indication of the angle of the Wii remote. The values shown in the figure (0, 0, -1) are the values given when the remote is horizontal such as resting on a table. The battery level of the remote is shown as a percentage at the bottom of the panel. When an IR source is not sensed by both

or one of the remotes the GUI indicates this by outputting “No LED” in the 3D Coordinates text boxes as shown in Figure 7-9.



Figure 7-9: GUI main screen showing No LED condition

7.4.3 Position in 3D

If IR sources are sensed by both Wiimotes the program is able to calculate the 3D position and outputs the values in the text boxes as shown in Figure 7-10.



Figure 7-10: GUI showing 3D position

7.4.4 Festo Positions

The tool-changer was required to be in the 'Green Zone' in order to be able to effect a tool change. If it was not within these coordinate ranges the coordinates were shown in black in the GUI and this meant the the spindle was out of range of the Festo arms. Figure 8-10 shows a GUI state where the X and Z coordinates are in range of the Festo actuators. The X coordinate is the right and left alignment of the tool-changer with the spindle and the Z coordinate is the distance the tool-changer is from the spindle. With the X and Z coordinates in range in can be seen from the figure that the program is able to select the appropriate position from the position table of the Festo horizontal arm. Since the Y value or vertical coordinate is not in range, a position is not able to be selected for the vertical actuator and this is indicated by the 'n/a' values in the text boxes.

Although the Festo actuators were very accurate they had one limitation. They were designed to be used in a repeatable application. That is, they were designed to be used in situations where the required positions of motion could be programmed at setup and the machines could be left to repeat those motions. This meant that they had a limited number of set points and could not be programmed on-the-fly to go to any required location.

In order to facilitate the application of the tool-changer, where the spindle would not necessarily be positioned in a repeatable location, the Festo arms' usual method of operation had to be adapted to provide for the more dynamic environment.

Festo provided the Festo Configuration Tool (FCT) in order to set up the two Festo actuators. The FCT allowed for the configuration of parameters, setting up of a position table and testing of routines. The position table allowed for the definition of up to 64 positions and velocity profiles. It was the position table that would provide the key to allow the tool-changer to adapt to the different positions of the spindle.

The horizontal arm had a total working range of 300 mm. However, only the range from 252 mm to 294 mm could be used to exchange tools as the gripper arm only cleared the tool storage carousel at 252 mm. This meant that a position point could be set every millimetre within the working range. This would provide for more than enough accuracy since the tool-changer was easily expected to be able to make a tool change if it was within 5 mm of the spindle centre point. It was of course also necessary to have several other points such as the home point of the arm and the point at which the gripper arm picked up a tool.

The vertical arm had a total working range of 100 mm. If interacting with a real spindle accommodating BT-40 tool holders, almost all of the 100 mm would be used in the insertion or removal of the tool. In the test environment of this research, the BT-40 tool holder was only required to touch the end of the spindle. This allowed for the testing of various vertical positions of the spindle incorporating the entire range of the arm. In this case there were not enough points in the table to allow for a set point every millimetre. It was decided to store points every 2 mm.

The FCT setup and position table are shown in figure 7-11 and figure 7-12. In figure 7-12 the Manual Move panel at the bottom of the tool allowed for testing of the position and running of several position sequences.

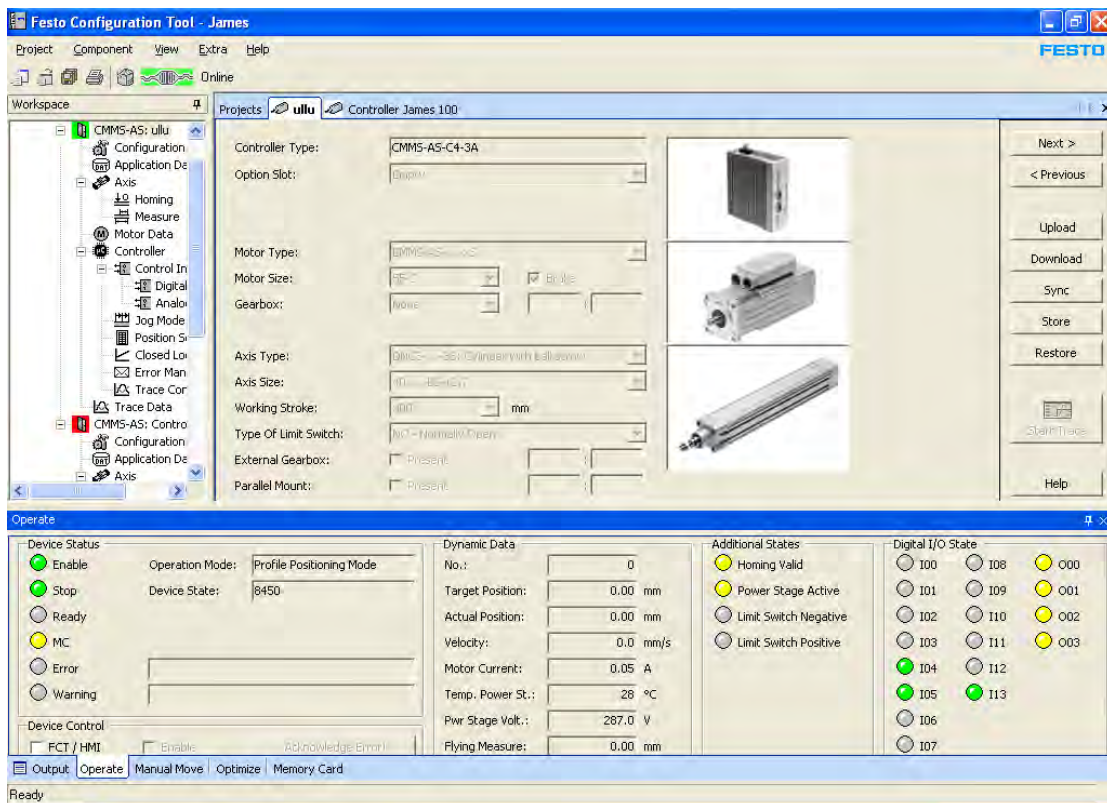


Figure 7-11: Festo FCT Configuration

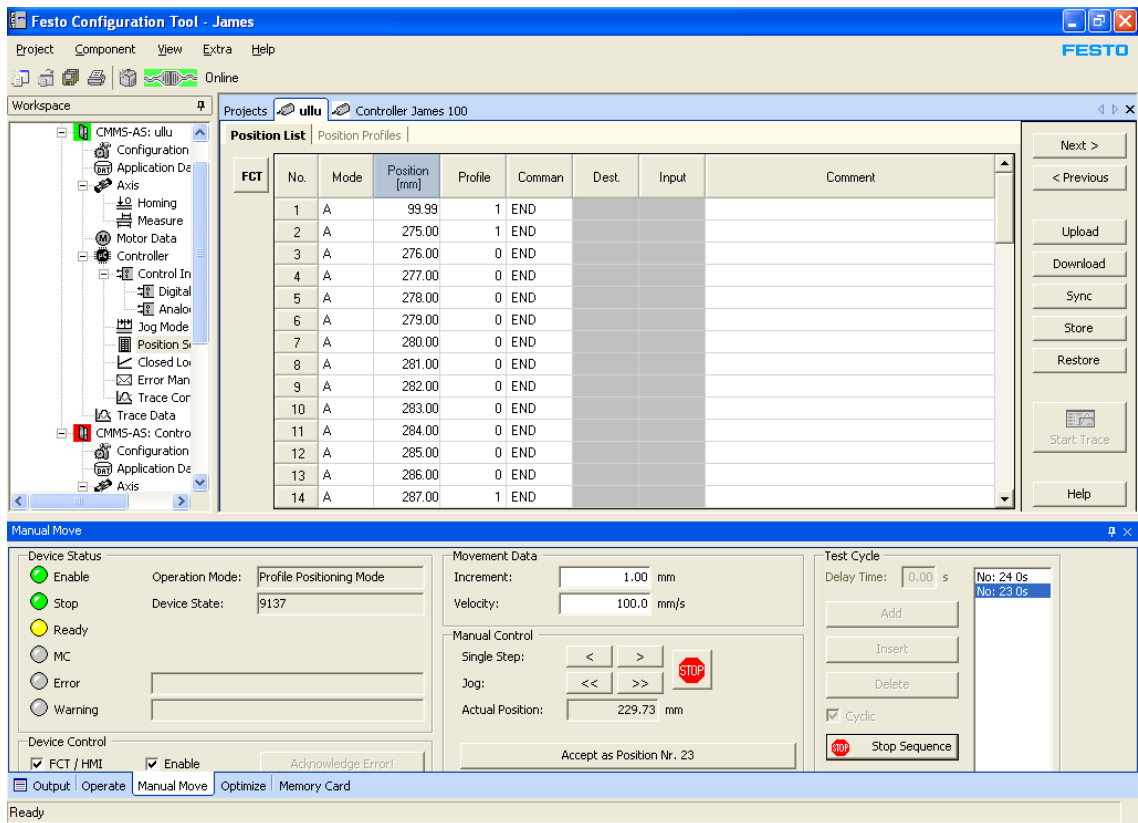


Figure 7-12: Festo FCT showing the position table and motion testing panel

Once the position table was setup and stored on the Festo controller, Digital I/O could be used to tell the arms which position to move to. This allowed for the motion of the arms without requiring the FCT to be open

7.4.5 Tool Change

In order for a tool change to be possible, the unit had to be in the correct range of the X, Y and the Z coordinate. It would only be in this case that the GUI would give set point positions for both the horizontal and vertical arms. In the previous screenshots of the main window of the GUI (figure 7-6 for example), it can be noticed the tool change button at the left of the screen is greyed out. When both arms have valid set points, the button becomes green indicating that a tool change can be made (Figure 7-13).

Code listings for the GUI Main Window and the Wiimote Info Panel are shown in Appendix D.3 and D.4. All program files are included in the accompanying CD.



Figure 7-13: GUI window showing conditions allowing for a tool change

7.5 Chapter Summary

This chapter presented the development of the software required to provide the tool-changing unit with an automatic calibration system. A robust API called GlovePIE was used to provide the initial testing platform. Once the positioning concept was proven, a GUI was developed using Visual C# 2010 Express. This provided a more user-friendly application.

8

Results and Discussion

8.1 Introduction

A variety of tests were carried out to provide evidence of the success and limitations of the automatic positioning system for the tool-changer. This chapter presents the tests, along with the accompanying results of the different experiments. The tests began with the determination of the camera characteristics. The accuracy of the Wiimote positioning concept was then tested using a FANUC robot as a benchmark. Finally, the system was tested while mounted onto the tool-changing unit.

8.2 Camera Characteristics

It was noted earlier in section 6.4.3 that other research into the use of Wiimotes presented differing values of the camera field of views. Tests were therefore needed to determine the characteristics of the cameras in use for this research.

8.2.1 Testing Setup

Testing of the position sensing system was carried out using a FANUC M-10iA robot. A simple battery powered circuit operating the IR LED was mounted onto the end effector of the robot. For determining the camera fields of view, a single remote was placed on a table. The location of the table allowed for tests to be carried out in a range of 0 to 750mm in the direction of the pointing Wii remote, the z-axis. The robot was also able to move to the extremities of the field of vision of the remote, both horizontally, the x-axis, and vertically, the y-axis. Figure 8-1 shows the frame of reference with respect to the Wiimote.

A custom frame of reference with respect to the position of the Wii remotes was programmed into the robot. This allowed for the highly accurate (to 0.001 mm) measuring system on the robot to confirm the position of the LED in relation to the remotes.

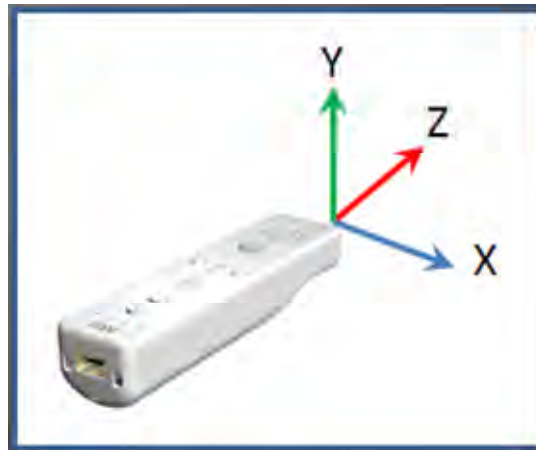


Figure 8-1: Wiimote positioning system frame of reference

The readings from the remotes were taken using the Bluetooth and GlovePIE setup mentioned previously. The maximum pixel values and distance moved by the robot were recorded when the robot reached the camera's limit. The values of the camera's limits in the X and Y-directions were recorded at several distances along the Z-axis. The arrangement of the testing configuration of the two Wiimote positioning system is shown in figure 8-2 [71]. The camera FOV tests were carried out with only one remote on the table and not two as shown in the picture.



Figure 8-2: Positioning System Testing Setup

8.2.2 Horizontal Field-of-View

The horizontal FOV was calculated using the results obtained by the maximum value read by the camera when moving the robot to the right and the left of the Wiimote. The results rounded to the nearest mm are shown in Table 8-1.

Table 8-1: Horizontal FOV readings

Distance (z) from Wiimote (mm)	X left (mm)	X right (mm)	Max Pixel Value	Adjusted X right (mm)
25	unreliable	4	962	5
30	unreliable	6	964	7
41	unreliable	17	1016	17
51	unreliable	21	1016	21
70	unreliable	28	1016	28
81	39	32	1016	32
100	46	39	1016	39
150	65	58	1016	58
201	84	80	1016	80
271	111	106	1015	106
300	123	117	1015	117
351	141	129	1008	130
400	159	136	974	141
500	195	151	920	167
600	231	177	907	198
750	284	220	892	250

It can be seen from the above table that the Wiimote had some trouble in taking reliable readings across the entire range of points. In particular, it was not possible to get consistent readings on the left side when the LED was closer than 80 mm to the Wiimote. This agreed with other research [46] in the field. It was however, interesting to note that on the right-hand-side values could be retrieved down to 25mm from the Wiimote. This could possibly be due to the characteristics of the LED used for the test.

The orange values in the pixel readings indicate where the maximum pixel value read was below the maximum possible value. The table shows that this behaviour got worse the further out you went, but only on the right-hand-side. On the left where the minimum pixel value (0) was recorded, this behaviour did not occur and consistent readings were taken to the maximum robot distance of 750 mm.

The column at the right-hand-side of the table gives the values of the right side adjusted by extrapolating the values out, assuming the camera would take readings out to 1016 pixels.

Figure 8-3 shows a graphical representation of the values. The graph includes linear trend lines as an indication of the linearity of the results. It can be seen that the results for the left side are almost perfectly linear. On the right side slight variations are seen along the length of the graph. Noticeable variations are noticed when the LED is close to the camera and about 400 mm away.

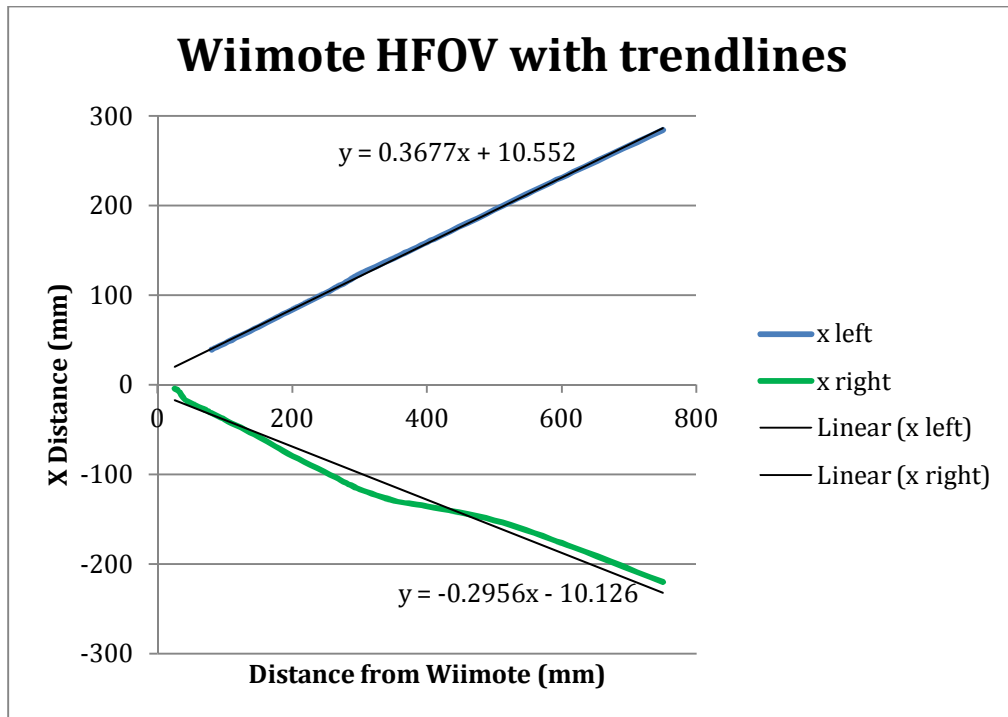


Figure 8-3: Wiimote Horizontal FOV Test results

Using the gradient of the trendlines to calculate the angles gives 40.4° for the left side and 32.9° for the right hand side. Using the adjusted values of X_{right} gives an angle of 36.8° .

Further analysis of the results is shown in Appendix E.1.

The left side exhibited the most linear behaviour. The value it produced was close to the value of 41° which was the most common figure used in other research. It was therefore decided to use 41° as the value for the calculations.

8.2.3 Verical Field-of-View

In this case, the robot arm containing the LED was moved up and down to the extremities of the camera's view at various distances from the Wiimote.

Table 8-2: Vertical FOV readings

Distance (z) from the Wiimote (mm)	Y up	Y down	Y down adjusted	Max. Pixel Reading	Min. Pixel reading	xpix max	xpix min	xpix variation
60	17	-26	-26	760	0			
70	20	-29	-29	760	0			
80	22	-32	-32	760	0			
100	27	-39	-39	760	0			
125	34	-46	-46	760	0			
150	42	-54	-54	760	0	521	535	14
200	55	-68	-68	760	0	518	534	16
250	68	-83	-83	760	0	519	530	11
300	81	-99	-99	760	0	516	529	13
350	94	-113	-113	760	0	516	529	13
400	107	-128	-128	760	0	516	529	13
500	134	-158	-158	759	0	512	528	16
600	160	-187	-187	759	0	512	529	17
750	200	-207	-230	759	38	512	529	17

The vertical FOV test produced more consistent results than the horizontal one. Only the one exception, which is indicated by the figure in orange, was recorded. At this point the minimum pixel reading on the camera only went down to 38 and not 0 as in the other cases. Again, this value was extrapolated to make for the adjusted value of -230 as opposed to -207.

There was some deviation in the x co-ordinate pixel values as shown in the last three columns of the table. This could indicate a minor error in misalignment of the Wiimote with the custom user frame of the robot. The x co-ordinate pixel values were only measured for distances above 150 mm.

It could also be noted that successful readings were able to be recorded from 60 mm proximity to the camera.

When these values were plotted (figure 8-4), the graphs showed a linear relationship. The gradient values of the two lines were much closer in the up and down movement of the robot as opposed to the horizontal readings. This can be confirmed below by the similarity of the two FOV values.

Up
Down

Gradient = 0.265
Gradient = 0.2968

FOV = 29.7°
FOV = 33.1°

Average Vertical FOV 31.4°

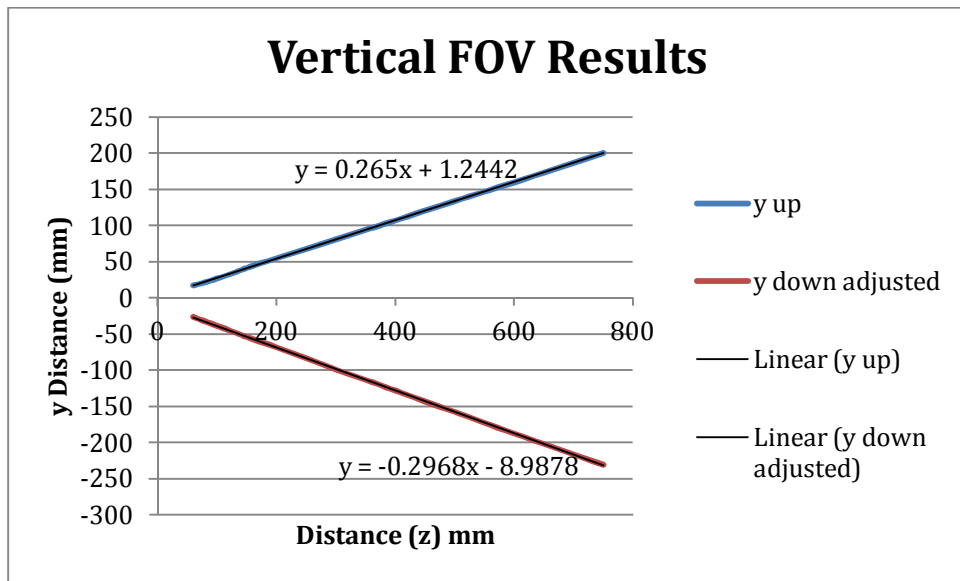


Figure 8-4: Wiimote Vertical FOV results

The average value of 31° agreed with the most common value given in other research examined by the author [46].

8.3 Positioning System Validation

Once the camera characteristics were established, the accuracy of the positioning system could be tested. The testing setup is shown in Figure 8-2. The Wii remotes were set 100 mm apart on the table and aligned manually with a ruler. Once again, the readings were taken from the Wiimotes using GlovePIE. These readings were then compared the position given by the FANUC robot, and the accuracy of the Wii remote positioning system could be verified.

More than 200 readings were taken during the test. Previous research [46, 61] indicated that the stereo Wiimote system would only be accurate between 20 mm or 25 mm. The results that the author recorded established that the Wiimote system could perform more accurately. Table 8-3 shows the error distribution achieved over the entire range of readings. Each axis is represented in the table as well as the absolute distance error which is the 3D distance between the target point and the Wii measurement.

As mentioned earlier, it was expected that accuracy within 5 mm would easily allow for a successful tool change to occur. In fact, due to the tapered design of the BT-40 tool holder errors of 10 mm or 15 mm could be tolerated.

Table 8-3: Stereo Wiimote error distribution from all results

	x	y	z	Abs
	direction	direction	direction	Distance
Average Error (mm)	0.03	1.40	0.87	5.98
Largest Positive Error	3.97	12.23	27.56	39.44
Largest Negative Error	-4.71	-19.51	-39.42	0.41
Within 1 mm	63%	54%	23%	9%
Between 1 mm & 5 mm	37%	33%	40%	48%
Between 5 mm & 10 mm	0%	10%	22%	25%
Between 10 mm & 15 mm	0%	2%	9%	10%
Between 15 mm & 20 mm	0%	0%	3%	4%
Between 20 mm & 30 mm	0%	0%	2%	1%
Greater than 30 mm	0%	0%	0%	1%
Total	100%	100%	100%	100%

For the tool-changing unit the main region of interest would be between 200 mm and 400 mm from the Wiimotes. It was in this region that some good results were achieved with 75% of the absolute distance error being under 5 mm. Of the 133 readings taken in this region, there was a session of 60 readings in which 88% of the absolute distance error was less than 5 mm and 15% were less than 1 mm. The results for this region are summarised in table 8-4.

It may be seen from table 8-4 that 96% of the results in this zone would be within 10 mm of the target point. These results were pretty impressive for a system that was manually aligned with a ruler.

Table 8-4: Error distribution in region of interest (200 mm to 400 mm from Wiimote)

	x	y	z	Abs
	direction	direction	direction	Distance
Average Error (mm)	0.00	0.87	0.88	3.71
Largest Positive Error	3.10	8.05	12.90	14.21
Largest Negative Error	-2.65	-4.12	-10.95	0.41
Within 1 mm	71%	65%	31%	14%
Between 1 mm & 5 mm	29%	31%	49%	61%
Between 5 mm & 10 mm	0%	4%	17%	21%
Between 10 mm & 15 mm	0%	0%	4%	4%
Total	100%	100%	100%	100%

An increase in errors occurs with greater distances. This would be expected as the errors would be compounded the further away the LED was from the remotes. Table 8-5 shows the error distribution in the range of 500 mm to 750 mm away from the Wiimotes.

Table 8-5: Error distribution in the region between 500 mm and 750 mm from the Wiimotes

	x	y	z	Abs
	direction	direction	direction	Distance
Average Error (mm)	0.22	2.77	0.01	11.10
Largest Positive Error	3.97	12.23	27.56	39.44
Largest Negative Error	-4.71	-19.51	-39.42	1.43
Within 1mm	43%	23%	5%	0%
Between 1 mm & 5 mm	57%	43%	25%	20%
Between 5 mm & 10 mm	0%	25%	33%	33%
Between 10 mm & 15 mm	0%	7%	18%	23%
Between 15 mm & 20 mm	0%	2%	10%	15%
Between 20 mm & 30 mm	0%	0%	7%	5%
Greater than 30 mm	0%	0%	2%	3%
Total	100%	100%	100%	100%

The table shows that in this zone only 53% of the absolute distance errors were within 10 mm, which is over a 40% decrease compared to the region of interest. It is noted that 3% of the errors were greater than 30mm.

Following are several graphs that graphically depict the error distributions.

Figure 8-5 shows the distribution of the X errors compare to the X values given by the robot. It can be seen that the majority of readings were taken at distances of $x = 20$ mm, 50 mm and 80 mm. This was due to the fact that the tool-changer would be positioned between the 100 mm difference of the two Wiimotes in order to perform a tool change. This region would therefore have the highest priority. At these points of interest it is evident that there is an even spread of errors.

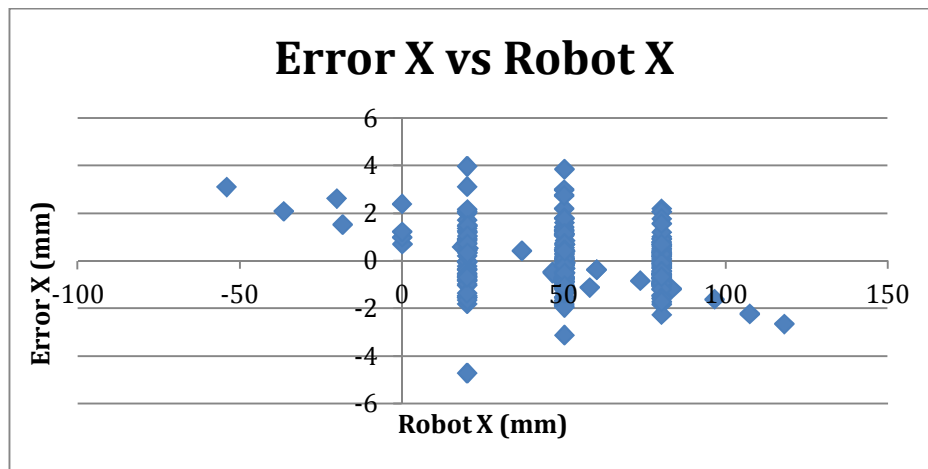


Figure 8-5: Error in the X direction compared to the Robot X coordinate

In figure 8-6 it can be seen that the Y error is also evenly distributed. The negative y-values do show a slightly larger error as seen by the left side of the graph. The one point, giving a -20 mm error, could perhaps be regarded as an erroneous reading.

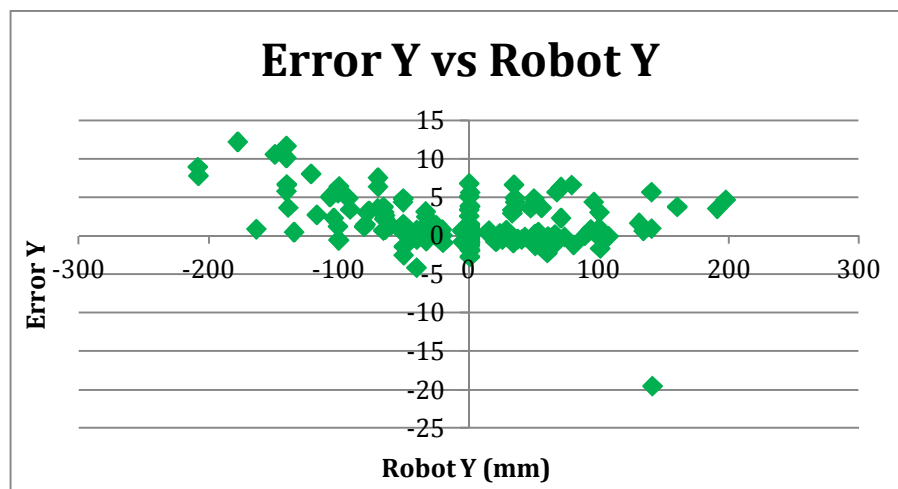
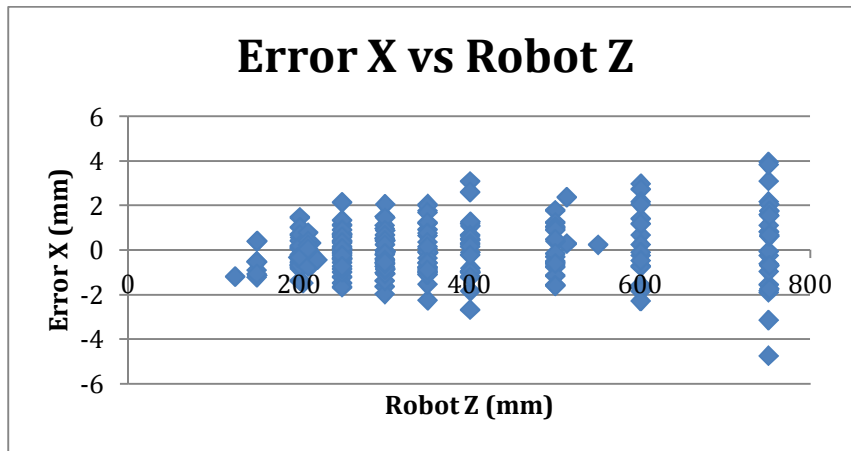
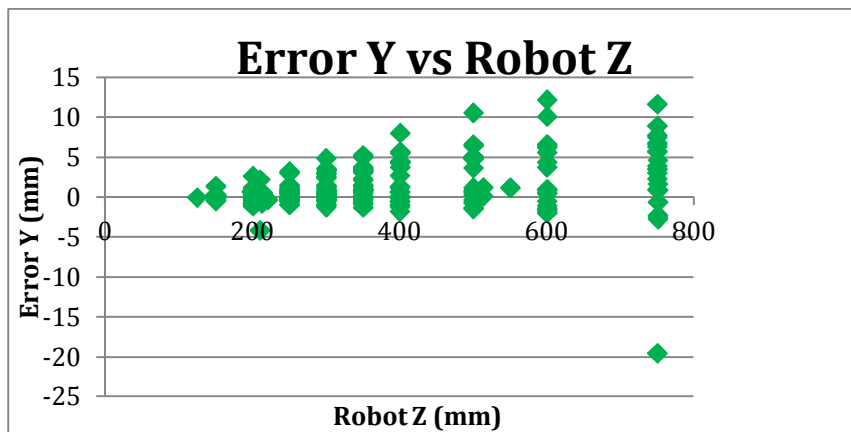


Figure 8-6: Error Y vs Robot Y

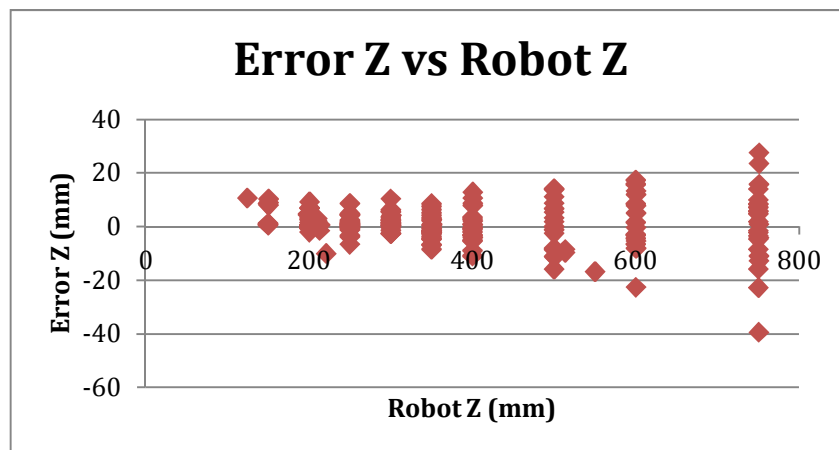
Figure 8-7 shows the error distribution of each axis with respect to the distance (z) from the Wiimotes. It is evident in all cases that the errors increase the further the distance is from the remotes. Table E-2 in Appendix E shows a table of results from the tests.



a) Error in the X-direction



b) Error in the Y-direction



c) Error in the Z-direction

Figure 8-7: Error distribution of the primary axes with respect to Robot Z

From the previous tables and graphs it is clear that the x-axis or horizontal direction provided the greatest accuracy. This is somewhat surprising since the testing of the FOV in this direction produced varying results and the FOV used in the calculations was a constant. The y-axis provided the next best accuracy with the z-axis delivering the least accurate results. It would be expected that the z-axis would be the least accurate as its values are derived from the x and y pixel readings.

8.4 Tests on the tool-changing unit

The Wiimote positioning system was then tested on the tool-changing unit itself. Two configurations were tested on the unit. The first one was the stereo vision setup using two Wiimotes. The second configuration involved the single Wiimote arrangement using two LEDs.

8.4.1 Stereo Wiimote configuration

The setup of this configuration is shown in figure 6-11.

A disadvantage of the tool-changing unit was that it did not provide for any lateral movement of the spindle. The unit had to be lined up directly with the spindle. The GUI developed for the positioning system made the setup user-friendly in that the aligning of the unit could be confirmed by the “Green Zone” indication of the X-coordinate.

The tool-changer could therefore accommodate variations in the position of the spindle along the Y and Z-axes only. The MRM with which the tool-changer was interacting also did not allow for lateral movement of the spindle, so the MRM and the tool-changer allowed for a useful configuration.

Although the MRM axis had a range of 400 mm, the Festo arm could only reach out to 140 mm. The MRM axis is shown in figure 5-1. Tests were conducted to 140 mm and the differences between the Festo, Wii and axis values were noted. Figure 8-8 illustrates the results graphically.

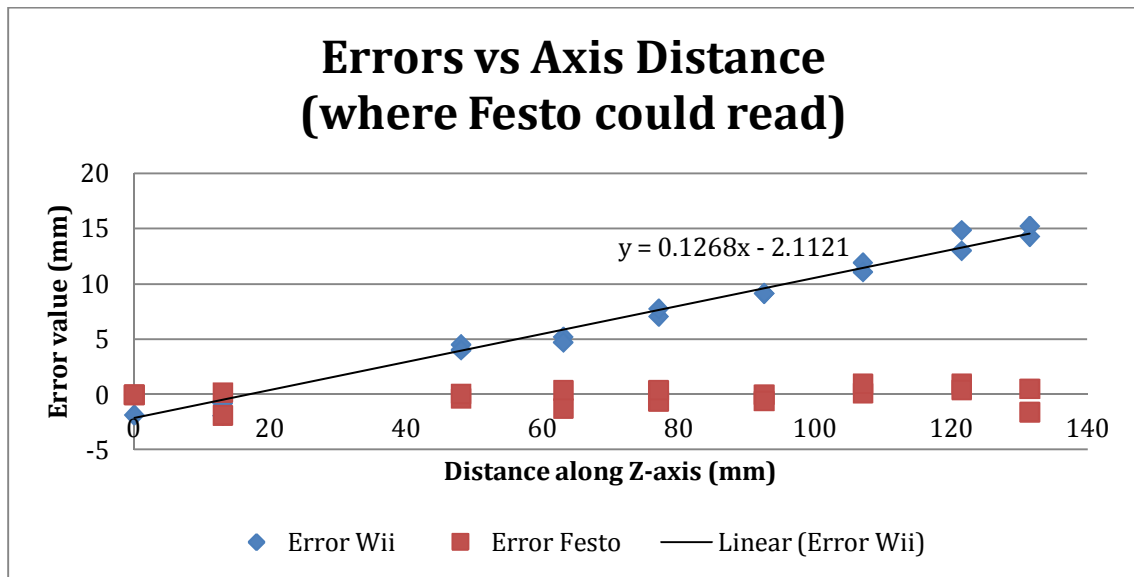


Figure 8-8: Stereo Wiimote positioning errors

Figure 9-8 shows the linear trend of the increase in the Wii error as the distance along the z-axis increases. The offset between the Wii position and the axis starting point was 216 mm therefore at the 131.5 mm Festo limit, the Wiimotes gave a reading of 362 mm. The maximum Wii error of 14 mm at the end point agreed favourably with the results achieved in the previous tests shown in table 8-4.

Although the Festo difference is called an error, the Festo axes were designed very accurately and the differences between the Festo and the machine axis are more likely to be attributed to the play in the axis of the machine.

Since the Wii error showed a linear progression, the software could easily be programmed to account for this increase in error. This adjustment would enable the Wii positioning system to deliver more accurate results.

It should also be noted that alignment of the Wiimotes could be improved. As shown in figure 6-11, the Wiimotes were placed in plastic holders. Although these holders enabled easy access to the Wiimotes, they did allow for some play in the remote position. The inbuilt accelerometers were used to aid in the setup, but their accuracy did not always detect physical error of the Wiimotes. The parallel alignment of the remotes was again done manually with rulers. The location of the remotes above the carousel made it a little more difficult to perform this alignment. A more rigid mounting for the Wiimotes could improve accuracy.

The Wii system was also tested to the limit of the machine axis. This would take the Wiimotes into the less accurate zone. The 400 mm axis limit translated into a distance of 616 mm from the Wiimotes. The results are shown in Figure 8-9.

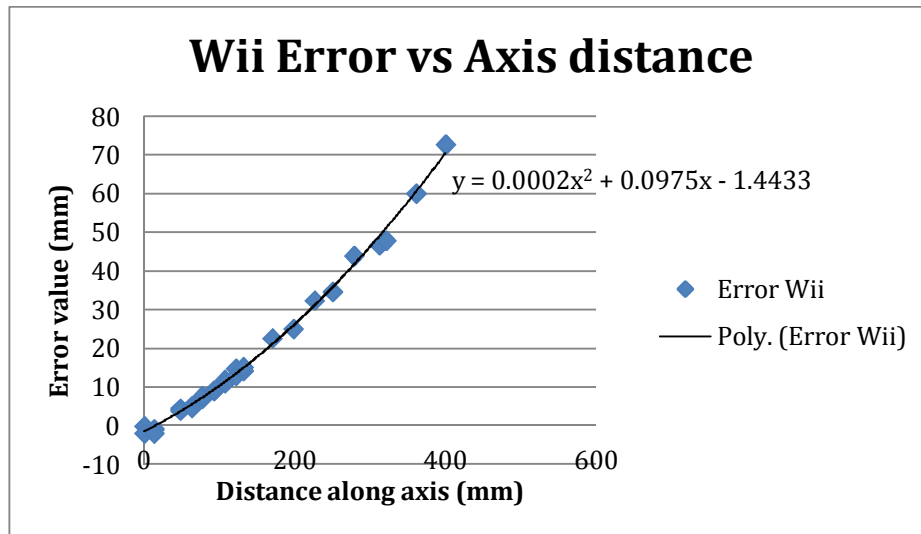


Figure 8-9: Wii Error vs Axis distance along entire axis

The proximity of the error to the trend line would allow the software to compensate for the increase in error over the length of the axis.

In terms of repeatability, the tests included an outward run and a return run along the same points of the machine axis. Figure 8-10 shows these results and gives an indication of the repeatability of the Wiimote positioning system. The flexibility of the machine axis also detracts from these findings and repeatability would be improved on a more rigid machine.

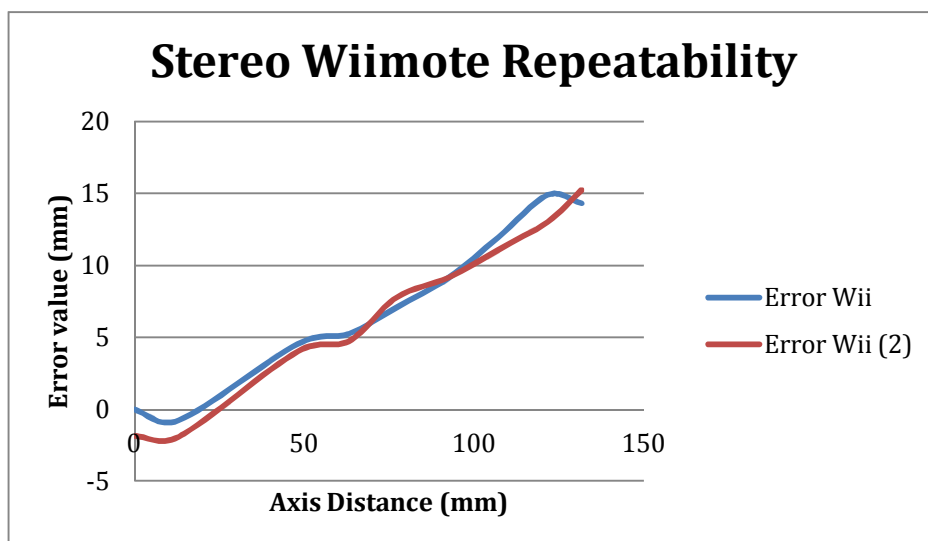


Figure 8-10: The repeatability of the Stereo Wiimote System

The results of the repeatability test are tabulated in table 8-6 to show the difference between the first and second runs. The table shows that the difference between the two errors ranges from a minimum of -0.03 mm to a maximum value of 1.85 mm. As indicated in the table, the average deviation is 0.55 mm.

Table 8-6: Repeatability test for the Stereo Wiimote System

Distance on Axis	Error Wii	Error Wii (2)	Difference in Errors
0	0	-1.82	1.82
13	-0.76	-1.88	1.12
48	4.55	4.04	0.51
63	5.24	4.74	0.5
77	7.08	7.77	-0.69
92.5	9.15	9.18	-0.03
107	11.93	11.1	0.83
121.5	14.87	13.02	1.85
131.5	14.3	15.24	-0.94
Average Difference in Error = 0.55 mm			

8.4.2 Single Wiimote configuration

The single Wiimote configuration is shown in Figure 8-11.

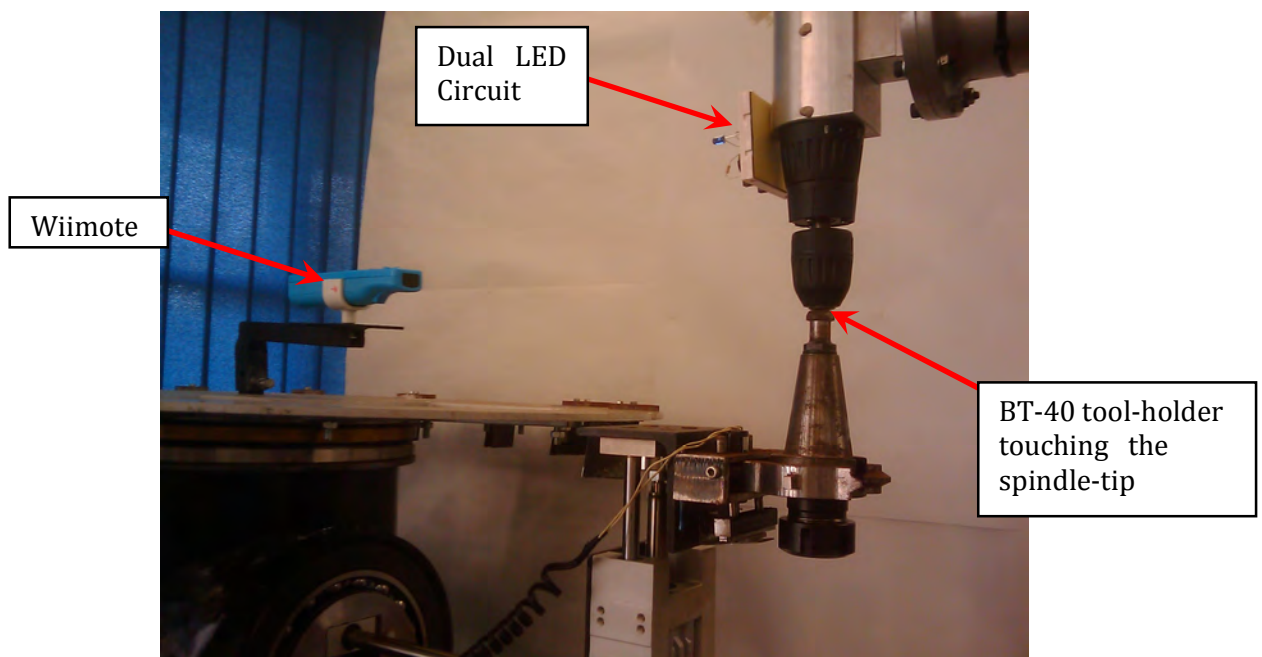


Figure 8-11: Single Wiimote arrangement

As can be seen from the photograph, the single Wiimote was located centrally on the Wii mounting bracket. The dual LED circuit was mounted at a convenient point on the spindle head.

Tests were carried out in the region where a tool change was possible. This was a narrow zone extending from where the gripper arm could clear the carousel and deliver a tool, to the limit of its extension, being from 252 mm to 294 mm of the Festo arm extension. Along the machine axis it was from 83.5 mm to 125.5 mm.

The spindle was located at several positions along the z-axis and moved vertically up and down to test the variation along the y-axis. At each point the Festo arms were manoeuvred such that the BT-40 tool holder would touch the tip of the spindle as shown in figure 8-11. The Festo coordinates, along with manual measurements taken on the machine axes provided the verification of the Wiimote readings

Figure 8-12 shows the Wii error and the Festo errors with respect to the position along the axis (z-coordinate). Once again the Festo variations could be due to the inaccuracies in the machine axis. The Wii errors are evenly distributed across the range with no obvious correlation evident. The offset of the remotes to the start of the axis was 228 mm in this case. This would mean that at 125 mm along the axis, the Wiimote would be reading the LEDs to be 353 mm.

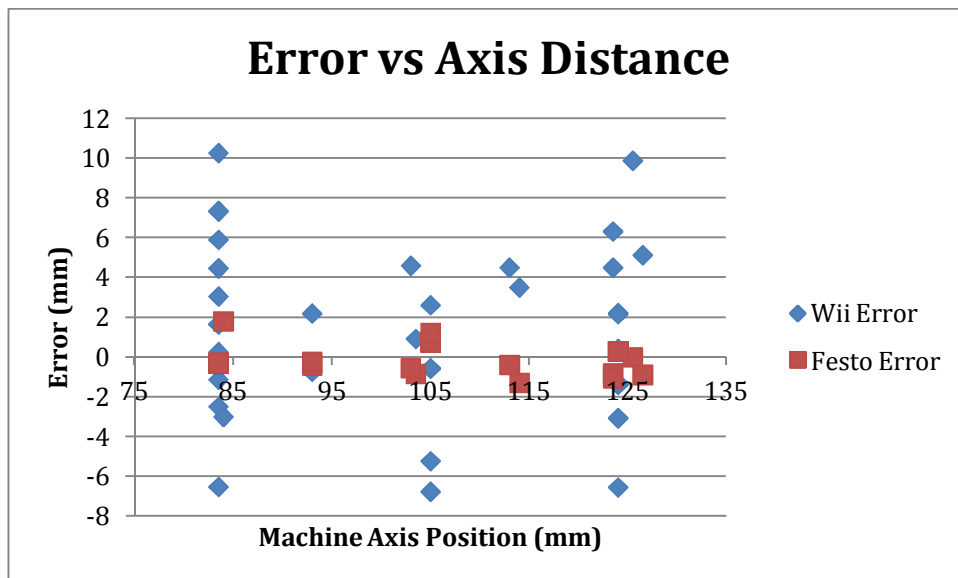


Figure 8-12: Single Wiimote Errors vs Machine Axis position

Table 8-7 shows the error distribution for the various axes of the Wiimote along with the absolute distance error. The results show that the maximum error was just over 10 mm and 98% of the results along the tested machine axis are within 10 mm. These figures

verify that the single Wiimote system provides a very acceptable solution for the tool-changer to make effective tool changes. Further tables are shown in Appendix E.3.

Table 8-7: Error distribution of the Single Wiimote tests

	x	y	z	Abs
	direction	direction	direction	Distance
Average Error (mm)	0.18	0.33	1.48	3.9
Largest Positive Error	2.42	2.5	10.27	10.29
Largest Negative Error	-1.72	-1.13	-6.75	0.48
Within 1 mm	80%	86%	16%	7%
Between 1 mm & 5 mm	20%	14%	55%	64%
Between 5 mm & 10 mm	0%	0%	27%	25%
Between 10 mm &15 mm	0%	0%	2%	5%
Total	100%	100%	100%	100%

Comparing the single Wiimote system with the stereo one, it can be observed that the single system provided more accurate results in the tool changing zone. The greatest error along the axis, in this zone in the single Wiimote system, was just over 10 mm whereas in the stereo system the maximum error was almost 15 mm. The single Wiimote system was also significantly easier to setup and provided the added advantage of giving the angle of the spindle.

With regard to repeatability, Table 8-8 summarises the repeatability performance of the different Wiimote axes. These values were derived from repeated points in the testing. Once again the x-direction is the most consistently accurate. Interestingly though, although the z-direction had the greatest range of deviation, the average deviation in this direction was the least.

Table 8-8: Repeatability of Single Wiimote axes

	x	y	z
	direction	direction	direction
Average Deviation (mm)	0.143	-0.399	-0.092
Maximum Deviation (mm)	0.82	0.48	4.21
Minimum Deviation (mm)	-0.48	-1.67	-5.22

For reference, more detailed tables are found in Appendix E.4. The following graph (figure 8-13) graphically depicts those results. The deviation of the repeated points is shown for each axis of the Wiimote. The graph clearly shows the narrowness of the x and y deviations compared to the z deviation.

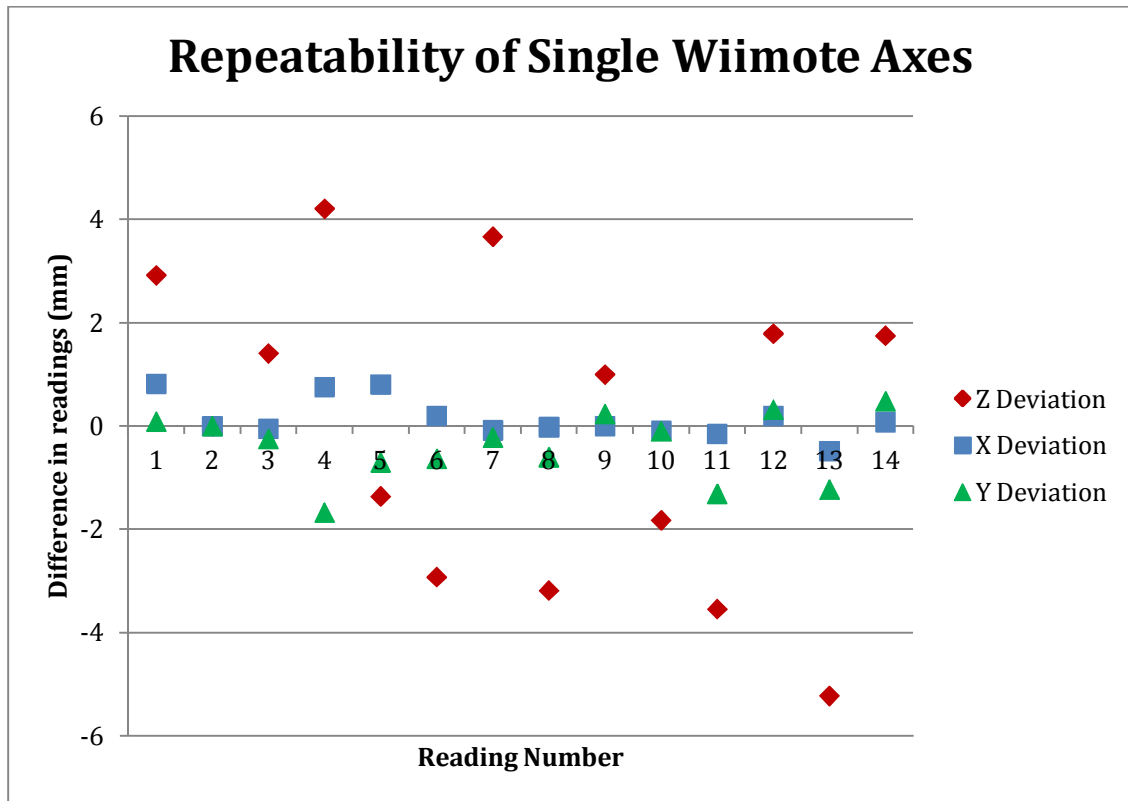


Figure 8-13: Repeatability spread of the single Wiimote axes

8.5 Cost Analysis

One of the major advantages of the Wiimote positioning system is its low cost compared to other commercially available systems. Table 8-9 shows the comparative cost of some of the different solutions researched.

As can be seen from the table, the Wiimote positioning system is a very economical solution to real-time 3D position sensing. Figure 8-14 depicts graphically how cost-effective the Wiimote system is.

Table 8-9: Wiimote Positioning System costs compared to other systems

	System Cost	Dual Wii % of cost	Single Wii % of cost	Comments
Festo				
Compact vision system SBOC-Q-R1B and accompanying software	R55,193	2%	1%	Could identify features of the spindle but would need another device to provide the distance. Proprietary software.
National Instruments				
NI Smart camera and software	R34,094	4%	2%	Same as above
Acuity Laser				
AR1000	R 2,000	64%	27%	Can only measure distance. Would have to find a way to locate the spindle tip.
Dual Wiimote System				
Wiimote 1	R 250*			*bought used
Wimote 2	R 650			
Wiimote mounting brackets	R 380			
LED Circuit	R 86			
Dual Wiimote Total	R 1,280	100%	42%	
Single Wiimote System				
Wimote	R 250*			*bought used
Wiimote mounting bracket	R 190			
LED circuit	R 99			
Single Wiimote Total	R 539	237%	100%	Cheapest and easiest to setup. Offers all required features

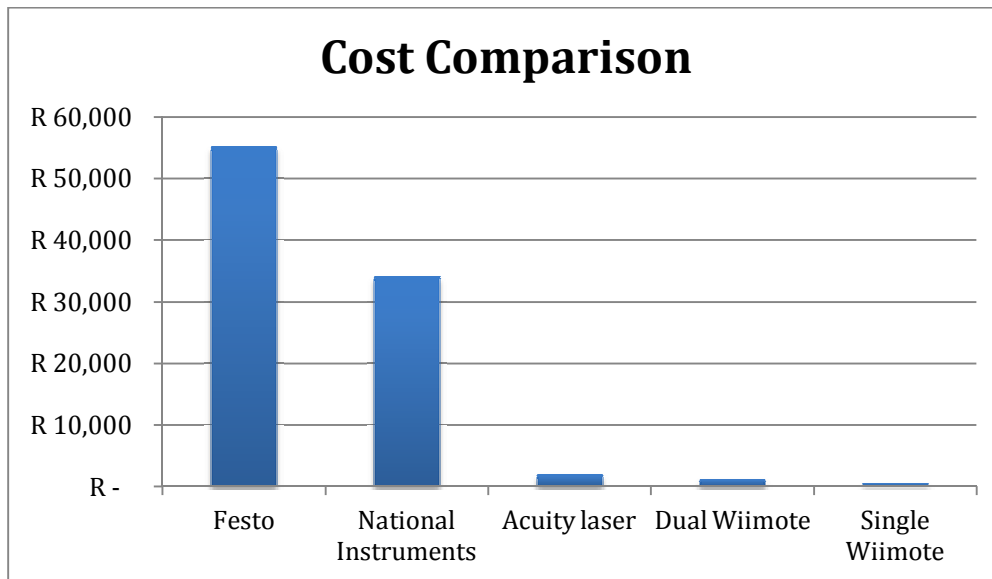


Figure 8-14: Wiimote positioning system cost effectiveness

8.6 Discussion

The initial testing of the Wiimotes determined the characteristics of the infrared camera housed within each Wiimote. These characteristics were essential to mapping the values recorded by the cameras to real world coordinates. The onboard position sensing system on a FANUC robot was used to provide the measuring device for determining the field of view of the camera. An infrared LED mounted on the robot end effector provided a target for the Wiimote camera to read. The GlovePIE API was able to read the values from the Wiimotes to the computer. The camera exhibited almost linear results in the horizontal FOV and almost perfectly linear results in the vertical direction. A horizontal FOV value of 41° was decided on based on the tests and other research. The tests in the vertical direction yielded a value of 31° .

Once the camera characteristics were known, these values could be used in determining the effectiveness of the Wiimote position sensing system. The stereo Wiimote variation was the first test subject. The FANUC robot and GlovePIE combination was once again used to verify if the Wiimote system would be accurate enough to be used on the tool-changer.

Errors of up to 15 mm could be tolerated for an effective tool-change to occur. The results achieved exceeded expectations. Within the tool-changing zone of the unit 75% of the error was less than 5 mm and 96% of the results fell within 10mm. As the spindle was moved further out past 500 mm greater errors were realized but this would not affect the tool-changer as it would not be able to reach the spindle at these distances.

Variations were also noticed with respect to the different axes of the Wiimote. The horizontal direction or the x-axis showed the most accurate results. This was somewhat surprising since the horizontal FOV showed more variation in the tests determining the camera characteristics.

Tests were then conducted with the Wiimotes mounted onto the tool-changing unit. Two variations of the Wiimote positioning system were used in the tests.

The first variation to be tested was the stereo Wiimote setup. It exhibited a progressively large error as the spindle moved away from the tool-changer. Within the tool-change zone this error showed a maximum value of almost 15 mm which was in agreement with the previous results. This error grew in a linear trend along the axis of the machine. Greater error could therefore be achieved by programming this trend into the software calculations. The Wiimotes were manually aligned and the Wiimote mountings exhibited some play. Greater accuracy could be achieved if these factors were minimized. A repeatability test of the stereo system showed an average deviation of 0.55 mm in the readings.

The next set of tests used the single Wiimote with Dual LEDs variation. The advantage of this configuration was that it could also determine the angle of the spindle. This was useful in that use could be made of the tool-changer's ability to present tools at an angle. The tool-changer was unfortunately limited in that it could not move the tool in the lateral direction. This meant that if a spindle was at an angle from the vertical the whole unit would have to be moved in order for a tool-change to be possible.

The single Wiimote configuration was also simpler to setup as alignment of multiple Wiimotes was not required. The results achieved with the single Wiimote were slightly better than the stereo Wiimote with the largest error in the tool-change zone only being 10 mm compared to the stereo system's 15 mm.

The tests conducted verified that the Wiimote positioning system could provide an accurate method for the tool-changer to sense the position of the spindle. This would give it the capability to perform effective tool-changes and adapt to changes in configuration.

The single Wiimote configuration gave more accurate results and provided the additional information of the angle of the spindle. It was also the cheapest option and compared to commercially available systems it was extremely cost-effective. The Wiimote system proved to be a very acceptable solution to the problem of automatically calibrating the tool-changing unit.

8.7 Chapter Summary

The testing of the Wiimote positioning system was presented in this chapter. Initially, a FANUC robot setup was used to determine the camera characteristics. The robot was then used to verify the accuracy of the Wii positioning system. Tests were then conducted with the Wiimotes mounted onto the tool-changing unit. The results showed that the Wiimote system performed well enough to allow the tool-changer to perform effective tool changes. The chapter ended with a cost analysis of the positioning system and a final discussion.

9

Conclusion

The modern day manufacturing environment along with the global economy is demanding manufacturers to be even more flexible in terms of the variety and capacity of parts they produce. To keep a competitive edge, they must economically be able to adjust to the varying and specific demands of customers. As technology improves, customers are demanding new products at a quicker rate.

The concept of modular machines that offer customised flexibility has developed as a result of these trends. Not only could they potentially reduce the initial cost for the entrepreneur, but they also provide an array of options for growth.

This research addresses the need for these machines to have tool-changing systems as separate and independent modules within the framework. A tool-changing unit for modular machines was presented. The unit was independently stable and could store and provide at least 8 tools. It could perform a tool change in under 30 seconds.

One of the challenges of a separate tool-changing module is its ability to sense the position of the spindle with which it must interact. This ability was shown to be a significant influencing factor in the integrability of the unit. Real-time position sensing of the unit would greatly improve its performance and ease of adjustment after any configuration change.

To enable modular machines to offer economically viable solutions, innovative techniques will be required to reduce cost. The research presented a novel positioning system for a tool-changing unit within the MRM paradigm. Nintendo Wii remotes were used as the basis for this positioning system. At the front of each Wiimote is an infrared camera with the ability to sense up to four infrared sources.

Two configurations of the Wiimote system were tested. The first configuration involved using two Wiimotes in a stereo vision arrangement. A single IR LED was used as the reference point on the spindle of the test MRM.

The second configuration required only one Wii remote but two LEDs. This configuration was cheaper and simpler to setup than the stereo system. The disadvantage of this system was that it would not work if the LED circuit was rotated away from the remote. It would require that the circuit plane remain parallel to the camera plane at all times for accurate readings. It did however offer the added capability of being able to sense the angular rotation of the spindle in the plane parallel to the camera. This added attribute was especially beneficial as one of the unique features of the tool-changing unit was that it could provide a tool to a rotated spindle.

Both configurations were determined to be successful positioning systems for the tool-changing unit with 95% of the measurement errors being less than 10 mm in the tool changing zone. The configurations were both manually aligned and with this rudimentary setup the single Wiimote system was slightly more accurate. A more rigid setup procedure would be likely to produce even better results.

Using the Wiimote system, the unit would be able to automatically calibrate itself with respect to the spindle position. It would be able to determine the position of the spindle at all times and adjust the parameters of its motion actuators when a tool change was required. The developed GUI provided a user-friendly environment for the positioning system to interact with the spindle. It provided feedback as to whether the tool changer was within the 'tool-changing zone' and automatically selected the required Festo actuator parameters.

Finally, a cost comparison of the Wiimote system as opposed to other commercially available systems was performed. The Wiimote system proved to be extremely economical compared to other systems and it offered capabilities that the other systems could not.

Further work

At the time of writing, the tool-changing unit control system was a separate unit to that of modular machine that it interacted with. Although the Wiimote positioning system provided the required amount of capability for the tool-changer to interact with the MRM, further work could be done in integrating the control systems. This would allow for the tool-changer to verify position readings from the Wiimote system with position readings from the machine axes themselves. This would provide an added safety factor.

At present, research is being done by the MR²G in the area of a modular 'plug-and-play' control system for MRMs. This work could aid in the development of a universal control system for MRMs and their auxiliary modules such as the tool-changing unit.

One disadvantage of the tool-changing unit was that it did not provide for lateral adjustment of the tool changing arm. This meant that the tool-changer had to be aligned to the MRM axis during setup. It also meant that if the spindle was at an angle the unit would have to be realigned to accommodate the angular rotation of the arm. Modifying the tool-changer such that it provided this lateral movement would greatly enhance its effectiveness.

To develop the positioning system further, additional configurations of the Wiimote system could be researched. The infrared camera on each Wiimote is able to sense up to four IR indicators. Configurations using multiple Wiimotes and multiple LEDs could be used to enhance accuracy, add extra features or provide a greater field of vision.

The positioning system provided the tool-changer with a real-time position sensing capability. This capability could be used to enable the development of a mobile tool-changer that could service several machines. Tool changers typically spend most of their time idle or underutilised and one tool-changing unit delivering tools to multiple work stations would increase capital efficiency.

References

- [1] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H., *Reconfigurable manufacturing systems*, in *Annals of the CIRP* 48(2). 1999. p. 527-540.
- [2] Padayachee, J., Bright, G., *Modular Reconfigurable Machines for Reconfigurable Manufacturing Systems*, 24th ISPE International Conference on CAD/CAM, Robotics and Factories of the Future, 29-31 July 2008, Japan.
- [3] Padayachee, J., *Development of a Modular Reconfigurable Machine for Reconfigurable Manufacturing Systems*, Masters Dissertation, 2010, University of KwaZulu-Natal.
- [4] Chowdhury, S., Olivier, J., Rodger, M., Stephen, W. *Modular Tool-Changing Unit for Reconfigurable Manufacturing Machines*, 4th year Research Project, 2009, University of KwaZulu-Natal.
- [5] Katz, R., *Design principles of reconfigurable machines*, International journal of Advanced Manufacturing Technology, 34(5), 2005, p. 430-439.
- [6] ElMaraghy, H.A., *Flexible and reconfigurable manufacturing systems paradigms*, International Journal of Flexible Manufacturing Systems, Vol. 17, No. 4, October 2005, p. 261-276.
- [7] Setchi, R.M., Lagos, N., Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review, Industrial Informatics, INDIN 04, 2nd IEEE International Conference, June 2004, p.529-535.
- [8] Dashchenko, A.I. (Ed.), *Reconfigurable Manufacturing Systems and Transformable Factories*, Springer, Germany, 2006, p. 38, 82-83.
- [9] Mehrabi, M.G., Ulsoy, A.G., Koren, Y., *Reconfigurable Manufacturing Systems: Key to Future Manufacturing*, Journal of Intelligent Manufacturing, Vol. 11, No. 4, 2000, p.403-419.
- [10] Hu, S.J., *Paradigms of Manufacturing – a panel discussion*, 3rd Conference on Reconfigurable Manufacturing, Michigan, USA, May 2005.
- [11] Koren, Y., Kota, S., *Reconfigurable Machine Tool*, US Patent # 5,943,750, issue date 31 Aug 1999.
- [12] *Engineering Research Center for Reconfigurable Machining Systems*. Available from: <http://www.nsf.gov/pubs/2000/nsf00137/nsf00137l.htm> Accessed: 7 Sep 2011.
- [13] Mehrabi, M.G., Ulsoy, A.G., Koren, Y., *Reconfigurable Manufacturing System and Their Enabling Technologies*, International Journal of Manufacturing Technology and Management, Vol. 1, No. 1, 2000, p. 113-130.
- [14] Landers, R.G., Min, B.K., Koren, Y., *Reconfigurable Machine Tools*, CIRP Annals – Manufacturing Technology, Vol. 50, Issue 1, 2001, p.269-274.

- [15] Moon, Y., Kota, S., *Generalized Kinematic Modeling of Reconfigurable Machine Tools*, Journal of Mechanical Design, Vol. 124, 2002, p.47-51.
- [16] Moon, Y., Kota, S., *Design of Reconfigurable Machine Tools*, Journal of Manufacturing Science and Engineering, Vol. 124, Issue 2, 2002, p480-483.
- [17] Xing, B., Bright, G., Tlale, N., Potgieter, J., *Reconfigurable Modular Machine Design for Reconfigurable Manufacturing System*, Univeristy of KwaZulu-Natal.
- [18] Padayachee, J., *Reconfigurable Machines for Reconfigurable Manufacturing Systems – A Literature Survey*, 2007, University of KwaZulu-Natal.
- [19] Bi, Z.M., Lang, S.Y.T., Verner, M., Orban, P. *Development of reconfigurable machines*, International Journal of Advanced Manufacturing Technology, 39(11-12), 2008, p.1227-1251.
- [20] Ahuett, H.,Aca, J.,Molina, A. *A Directed Evolution Modularity Framework for Design of Reconfigurable Machine Tools*, Cooperative Design, Visualization, and Engineering, Vol. 3675, Springer 2005, p.243-252.
- [21] Padayachee, J., Masekamela, I., Bright, G.,Tlale, N.S., Kumile, C.M., , *Modular Reconfigurable Machines Incorporating Modular Open Architecture Control*, 15th International Conference on Mechatronics and Machine Vision in Practice. Auckland, New Zealand, 2008, p. 127-132.
- [22] Collins, J., Bright, G., *A Tool-Changing Unit for Modular Reconfigurable Manufacturing Systems*, 25th ISPE International Conference on CAD/CAM, Robotics and Factories of the Future; 14-16 July 2010, Pretoria, South Africa
- [23] Collins, J., Bright, G., *Automatic Tool-Changing within the Reconfigurable Manufacturing Systems Paradigm*, South African Journal of Industrial Engineering Nov 2011 Vol 22 No.2, p. 68-79.
- [24] Wang, L. and Xi, J. (eds), *Smart devices and machines for advanced manufacturing*, Springer, 2008.
- [25] Ellata, A., Gen, L., Zhi, F., Daoyuan, Y. & Fei, L. *An overview of robot calibration*, Information Technology Journal, 3(1), 2004, p. 74-78.
- [26] Bernhardt, R. & Albright, S. (eds), *Robot calibration*, Springer, 1993.
- [27] Katz, R. & Moon, Y., *Virtual arch type reconfigurable machine tool design: Principles and methodology*, The University of Michigan NSF ERC for RMS, 2000.
- [28] Albada G., Visser A., Lagerberg J. & Hertzberger L., *A portable measuring system for robot calibration*, Proceedings of the 28th international symposium on automotive technology and automation – dedicated conference on mechatronics and efficient computer support for engineering, 1995, p. 441-448.
- [29] Zhong, X., Lewis, J., Nagy, F., *Autonomous robot calibration using a trigger probe*, Robotics and Autonomous Systems 18 1996, p. 395-410.

- [30] *Predictive Maintenance and Calibration techniques*. Available from: <http://www.mmsonline.com/articles/predictive-maintenance-and-machine-tool-calibration-techniques.aspx>. Accessed: 9 September 2011.
- [31] *SPACE.com -- Interferometry 101*. Available from: http://www.space.com/scienceastronomy/astronomy/interferometry_101.html. Accessed: 18 June 2009.
- [32] *Predictive Maintenance and Calibration techniques*. Available from: <http://www.mmsonline.com/articles/predictive-maintenance-and-machine-tool-calibration-techniques.aspx>. Accessed: 12 September 2011.
- [33] Bernhardt, R., Albright, S.L., (eds), *Robot Calibration*, Springer, 1993.
- [34] Motta, J., de Carvalho, G., McMaster, R.S., *Robot calibration using a 3D vision-based measurement system with a single camera*, Robotics and Computer Integrated Manufacturing 17, 2001, p. 487–497.
- [35] Zhuang, H., Roth, Z.S., Wang, K., *Robot calibration by mobile camera systems*, Journal of Robotic Systems, Vol. 11, Issue 3, 1994, p. 155 – 167.
- [36] Meng, Y., Zhuang, H., *Autonomous robot calibration using vision technology*, Robotics and Computer-Integrated Manufacturing 23, 2007 , p. 436–446.
- [37] *Winning an International Robotics Competition with LabVIEW and NI Machine Vision Hardware*. Available from: <http://sine.ni.com/cs/app/doc/p/id/cs-13096>
Accessed: 19 September 2011.
- [38] *Developing the WiseWELDING Machine Vision System for Adaptive Robotic Welding*. Available from: <http://sine.ni.com/cs/app/doc/p/id/cs-12389>.
Accessed: 19 September 2011.
- [39] *Smart Cameras for Embedded Machine Vision – Datasheet*. Available from: http://www.ni.com/pdf/products/us/cat_ni_1742.pdf.
Accessed: 19 September 2011
- [40] *Festo online Catalogue*. Available from: <http://www.festo.com/pnf/en-za/za/products/catalog> Accessed: 19 September 2011.
- [41] *LV-MaxSonar –EZO Data Sheet*. Available from: http://www.maxbotix.com/documents/MB1000_Datasheet.pdf Accessed: 19 September 2011.
- [42] Shaik, A., Tlale, N.S. and Bright, G., *2 DOF resolution adjustment laser position sensor*. 15th International Conference on Mechatronics and Machine Vision in Practice. Auckland, New Zealand, 2 - 4 December 2008.
- [43] *A few stats before we head into E3*
Available from: <http://majornelson.com/2011/06/03/a-few-stats-before-we-head-into-e3/> Accessed: 21 September 2011
- [44] *Console Wars*.
Available from: http://en.wikipedia.org/wiki/Console_wars#cite_note-35
Accessed: 21 September 2011

- [45] *Consolidated Sales Transition by Region. Nintendo Co., Ltd.* Available from: http://www.nintendo.co.jp/ir/library/historical_data/pdf/consolidated_sales_e1106.pdf Accessed 21 September 2011
- [46] Wronski, M., *Design and Implementation of a Hand Tracking Interface using the Nintendo Wii Remote*, 2008, University of Cape Town.
- [47] *Wii Price and Launch Date?* Available from: <http://ployer.com/archives/nintendo/> Accessed: 21 September 2011
- [48] *File:Nintendo Wii Sensor Bar.jpg.* Available from: http://en.wikipedia.org/wiki/File:Nintendo_Wii_Sensor_Bar.jpg Accessed: 22 September 2011
- [49] *Wii Homebrew.* Available from: http://en.wikipedia.org/wiki/Wii_homebrew Accessed: 22 September 2011
- [50] *Wii Reverse Engineered.* Available from: http://www.wiire.org/Main_Page Accessed: 22 September 2011
- [51] *Nintendo Wii Remote Controller for Wiimote.* Available from: http://www.szprice.com/products/Nintendo-Wii-Remote-Controller-for-Wiimote_1360.html Accessed: 22 September 2011
- [52] Lee, J.C., *Hacking the Nintendo Wii Remote*, IEEE Pervasive Computing, Vol. 7, No. 3, July 2008, p. 39-45.
- [53] *BCM2042 – Advanced Wireless Keyboard/Mouse Bluetooth Solution.* Available from: <http://www.broadcom.com/products/Bluetooth/Bluetooth-RF-Silicon-and-Software-Solutions/BCM2042> Accessed 23 September 2011.
- [54] *Wiimote.* Available from: http://homepage.mac.com/ianrickard/wiimote/wiili_wimote.html#Communication Accessed: 23 September 2011.
- [55] *Wiimote.* Available from: <http://wiibrew.org/wiki/Wiimote> Accessed: 23 September 2011.
- [56] *Small, Low Power, 3-Axis ± 3 g iMEMS® Accelerometer – ADXL330. Data Sheet.* Available from: http://www.analog.com/static/imported-files/data_sheets/ADXL330.pdf Accessed: 26 September 2011.
- [57] *How to drive a Camera's Vertical Field of view based on the pixel size.* Available from: <http://www.wiimoteproject.com/general-discussion/how-to-drive-camer'a-vertical-field-of-view-based-on-the-pixel-size/?action=printpage> Accessed: 27 September 2011.
- [58] *Image Processing Apparatus and Storage Medium Storing Image Processing Program.* Nintendo Patent Application US 2007/0211027, September 2007, Fig. 16, 17 and p.10-11, Fig. 20-23.
- [59] *Coordinate Calculating Apparatus and Coordinate Calculating Program.* Nintendo Patent Application US 2007/0211026, September 2007, Fig. 16.
- [60] *Wii Remote.* Available from: http://en.wikipedia.org/wiki/Wii_Remote Accessed 27 September 2011.

- [61] Petric, T., Gams, A., Ude, A., and Zlajpah, L., *Real-time 3D marker tracking with a Wiimote stereo vision system: application to robotic throwing*. Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop. June 2010, p. 357-362.
- [62] Case, J., Chilver, A.H. and Ross, T.F., *Strengths of Materials and Structures – Fourth Edition*, Arnold, 1999, p.387.
- [63] *OS Platform Statistics*. Available from: http://www.w3schools.com/browsers/browsers_os.asp Accessed 19 October 2011
- [64] *GlovePIE Home*. Available from: <http://glovepie.org/glovepie.php> Accessed 6 November 2011
- [65] *Revolutionary: Introducing GlovePIE*. Available from: <http://www.joystiq.com/2007/06/12/revolutionary-introducing-glovepie/> Accessed: 20 Oct 2011
- [66] Glassner, A.,(Editor), *Graphics Gems*, Academic Press, USA, 1995, p304.
- [67] *Distance Measurements with the Wiimote*. Available from: <http://wiiphysics.site88.net/physics.html> Accessed 26 Oct 2011
- [68] *Wiimote Libraries*. Available from: <http://sites.google.com/site/crayon3d/Home/wiimote-libraries> Accessed 28 Oct 2011
- [69] *Managed Library for Nintendo's Wiimote*. Available from: <http://channel9.msdn.com/coding4fun/articles/Managed-Library-for-Nintendos-Wiimote> Accessed 28 Oct 2011.
- [70] *Wii Device Library*. Available from: http://wiibrew.org/wiki/Wii_Device_Library Accessed 28 Oct 2011.
- [71] Collins, J., Bright, G., *Automatic Calibration of a Tool-Changing Unit for Reconfigurable Manufacturing Systems (RMS) using Nintendo Wii Remotes*, 26th ISPE International Conference on CAD/CAM, Robotics and Factories of the Future, 26-28 July 2011, Malaysia.
- [72] *ATC for small-sized Plano Miller*. Available from: http://www.tradekorea.com/product-detail/P00011357/ATC_for_small_sized_Plano_Miller.html. Accessed 14 Mar 2012
- [73] *VTL Devlivers Powerful, Heavy Cuts*, Available from: <http://www.mmsonline.com/articles/vtl-delivers-powerful-heavy-cuts>. Accessed 14 Mar 2012

A

Mechanical Calculations

A.1 Wii Support Weld Calculations

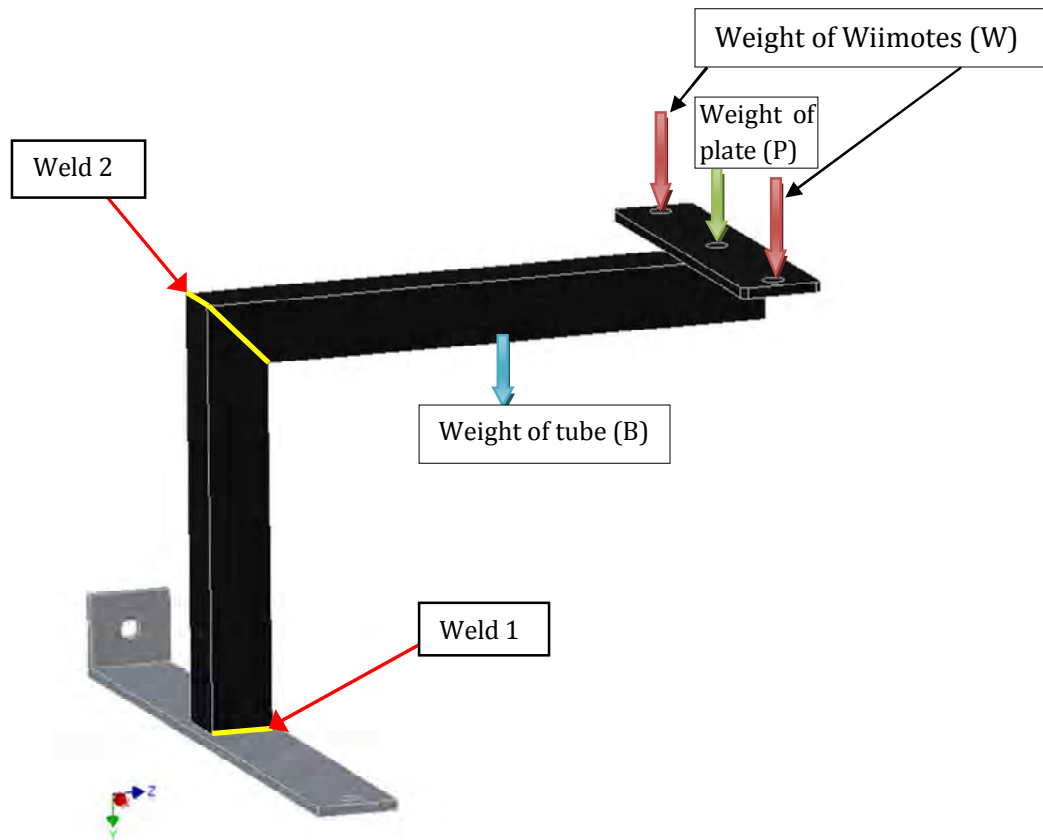


Figure A-1: Wii Support welds and forces

A.1.1 Weld 1 Calculation

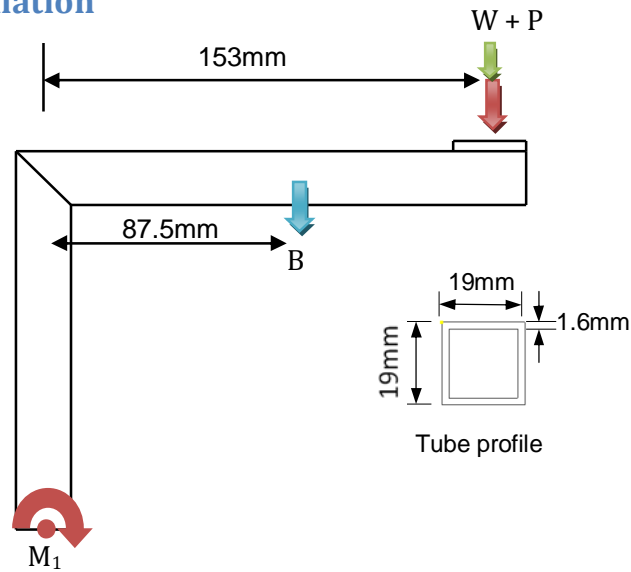


Figure A-2: Weld 1 moment diagram with tube profile

Material: Mild Steel – density = 7850 kg/m^3
 Acceleration of gravity: $g = 9.81 \text{ m/s}^2$
 Allowable shear stress of E60 electrode weld: 124 MPa
 Mass of each Wiimote plus mounting accessories and batteries: 160 grams

Find the required height of the weld.

Drawings for the various components are shown in Appendix B.

Mass = density x volume

$$\therefore \text{Mass of plate} = (25 \times 3 \times 116) \times 10^{-9} \times 7850 = 0.068 \text{ kg}$$

$$\text{Mass of Wiimotes} = 2 \times 160 \times 10^{-3} = 0.32 \text{ kg}$$

$$\text{Mass of tube} = 156 \times ((19 \times 19) - (15.8 \times 15.8)) \times 10^{-9} \times 7850 = 0.136 \text{ kg}$$

$$\therefore \text{Weight of the plate} \quad P = 0.67 \text{ N}$$

$$\text{Weight of the Wiimotes} \quad W = 3.14 \text{ N}$$

$$\text{Weight of the tube} \quad B = 1.33 \text{ N}$$

Moment = Force x distance

$$\therefore M_1 = (0.67 + 3.14) \times 153 + 1.33 \times 87.5 = 699.3 \text{ N.mm}$$

Bending stress

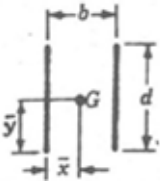
$$\sigma_b = \frac{My}{I}$$

Now

$$I = 0.707h \times I_u \text{ where } h \text{ is the height of the weld}$$

The tube was only welded on 2 sides as shown in Table A-1.

Table A-1: Weld pattern for weld 1

Weld Pattern	Area of Weld	Unit Moment of Inertia
	$A = 1.414hd$	$I_u = \frac{d^3}{6}$

$$\therefore I = 0.707h \times I_u = 0.707h \times \frac{19^3}{6} = 808.2h \text{ mm}^4$$

$$\sigma_b = \frac{699.3 \times 9.5}{808.2h} = 8.22h^{-1} \text{ MPa}$$

There is no direct or shear stress on this weld but we have to convert the bending stress to a shear stress to be able to relate it the allowable shear stress of the weld.

$$\tau_{max} = \frac{1}{2}((\sigma_{resultant})^2 + 4(\tau)^2)^{\frac{1}{2}}$$

\therefore

$$\tau_{max} = \frac{1}{2}((8.22h^{-1})^2 + 4(0)^2)^{\frac{1}{2}} = 4.11h^{-1} \text{ Mpa}$$

Setting that equal to the E60 allowable shear stress gives

$$\tau_{max} = 124 \text{ MPa} = 4.11h^{-1}$$

\therefore The minimum allowable height of the weld is: **h = 0.033mm**

A.1.2 Weld 2 Calculation

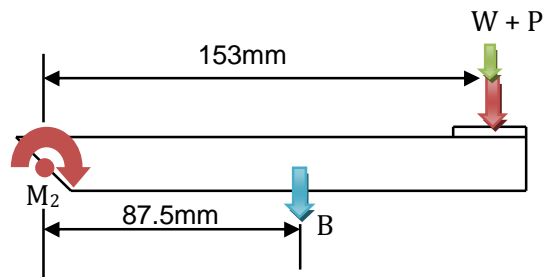


Figure A-3: Moment diagram for weld 2

In this case, there is the stress caused by the moment but there is also a direct shear component.

Since the weld is angled at 45° the components of the weights need to be taken with respect to the angle of the weld.

Also, since the weld is at an angle, the length of the weld is longer than the size of the tube.

Length of weld (d)

$$d = \frac{19}{\cos 45^\circ} = 26.87 \text{ mm}$$

Table A-2 shows the weld pattern for weld 2 which enables the calculation of the weld area.

Table A-2: Weld pattern for weld 2

Weld Pattern	Area of Weld	Unit Moment of Inertia
	$A = 1.414h(b+d)$	$I_u = \frac{d^2}{6}(3b + d)$

$$\therefore A = 1.414h(19 + 26.87) = 64.86h \text{ mm}^2$$

Direct shear stress

$$\tau = \frac{F}{A} = \frac{B \cos 45^\circ + (W + P) \cos 45^\circ}{A} = \frac{1.33 \cos 45^\circ + 3.81 \cos 45^\circ}{64.86h} = 0.06h^{-1} \text{ MPa}$$

Now

$$M_2 = M_1 = 699.3 \text{ N.mm}$$

And

$$I = 0.707h \times I_u = 0.707h \times \frac{d^2}{6} (3b + d) = 0.707h \times \frac{26.87^2}{6} ((3 \times 19) + 26.87) \\ = 7135.3h \text{ mm}^4$$

$$\sigma_b = \frac{My}{I} = \frac{699.3 \times 9.5}{7135.3h} = 0.93h^{-1} \text{ MPa}$$

Combining the stresses

$$\tau_{max} = \frac{1}{2} ((\sigma_{resultant})^2 + 4(\tau)^2)^{\frac{1}{2}}$$

∴

$$\tau_{max} = \frac{1}{2} ((0.93h^{-1})^2 + 4(0.06h^{-1})^2)^{\frac{1}{2}} = 0.47h^{-1} = 124 \text{ Mpa}$$

∴ Weld height for weld 2 is: **h = 0.0038 mm**

A.2 Deflection

Figure A-4 shows how the angular twist of the base plate due to the applied moment results in the deflection at the end of the mounting.

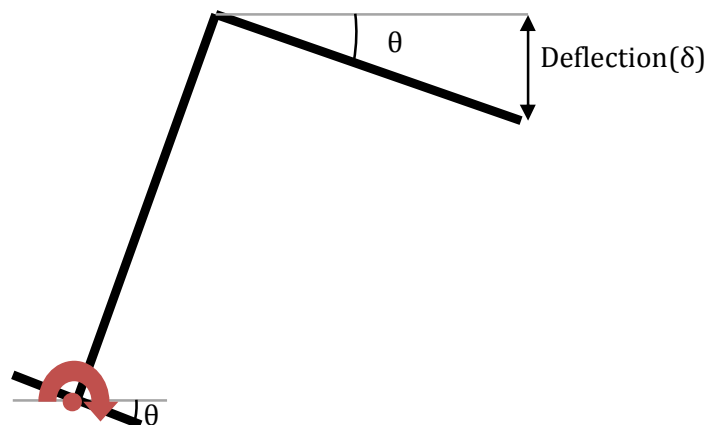


Figure A-4: Deflection in Wii Mounting

Now

$$\theta = \frac{TL}{GJ}$$

Where

$$G = 80 \text{ GPa}$$

$$L = \frac{191}{2} = 95.5 \text{ mm}$$

$$T = \frac{M_1}{2} = \frac{699.3}{2} = 349.65 \text{ N.mm}$$

$$J = \frac{1}{3}bt^3 = \frac{1}{3} \times 25 \times 3^3 = 225 \text{ mm}^4$$

∴

$$\theta = \frac{349.65 \times 95.5}{80000 \times 225} = 0.001855 \text{ radians} = \mathbf{0.1063^\circ}$$

Now

$$\sin \frac{\theta}{2} = \frac{\delta/2}{175}$$

∴ The deflection (δ) is:

$$\delta = 350 \sin 0.05315 = \mathbf{0.325 \text{ mm}}$$

A.3 Bolt Analysis

The critical bolt location shown in Figure 7-7 is analysed for different modes of failure below.

The bolt used in this location was an M6 stainless steel A2-70 specification bolt. This meant that it had the following physical properties:

Tensile Strength (S_u) = 700 MPa Yield Strength (S_y) = 450 MPa

Youngs Modulus (E) = 195 GPa Shear Strength (S_{ys}) = 311 MPa

Tensile Stress Area (A_s) = 20.1 mm² Nut thickness (t) = 5mm

Recommended tightening torque = 7.3Nm

Max Shear force the bolt can withstand:

$$F = A_s \times S_{ys} = 20.1 \times 311 = \mathbf{6.25 \text{ kN}}$$

Thread stripping load:

$$F_s = \pi d(0.75t)S_{ys} = \pi \times 6 \times (0.75 \times 5) \times 311 = \mathbf{22 \text{ kN}}$$

Joint separation force:

$$T = 0.2F_i d$$

∴

$$F_i = \frac{T}{0.2d} = \frac{7.3}{0.2 \times 0.006} = 6.08 \text{ kN}$$

Clamped area A_c :

$$A_c = d^2 + 0.68dg + 0.065g^2 = 6^2 + 0.68 \times 6 \times 9 + 0.065 \times 9^2 = 83.25 \text{ mm}^2$$

Now

$$K_b = \frac{A_b E_b}{g} = \frac{20.1 \times 193 \times 10^3}{9} = 4.31 \times 10^5$$

And

$$K_c = \frac{A_c E_c}{g} = \frac{83.25 \times 80 \times 10^3}{9} = 7.4 \times 10^5$$

∴

$$F_e = \frac{K_b + K_c}{K_c} F_i = \frac{4.31 + 7.4}{7.4} \times 6.08 = \mathbf{9.62 \text{ kN}}$$

Force due to applied load:

The moment at the centre of the base bracket is 699.3Nmm. This moment is carried by 2 bolts. The bolt head is 10mm wide.

Moment = force x distance

∴

$$Force = \frac{Moment}{distance} = \frac{\frac{699.3}{2}}{\frac{10}{2}} = \mathbf{69.9\ N}$$

B

Drawings

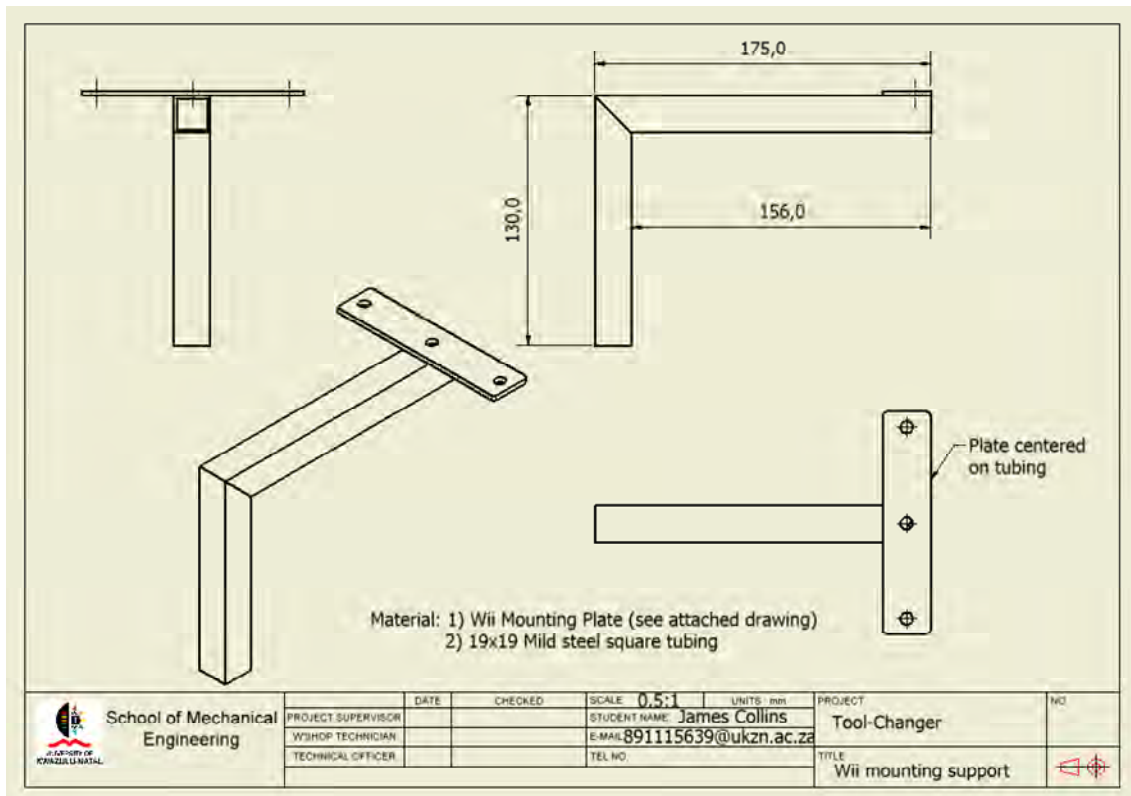


Figure B-1: Wii mounting support

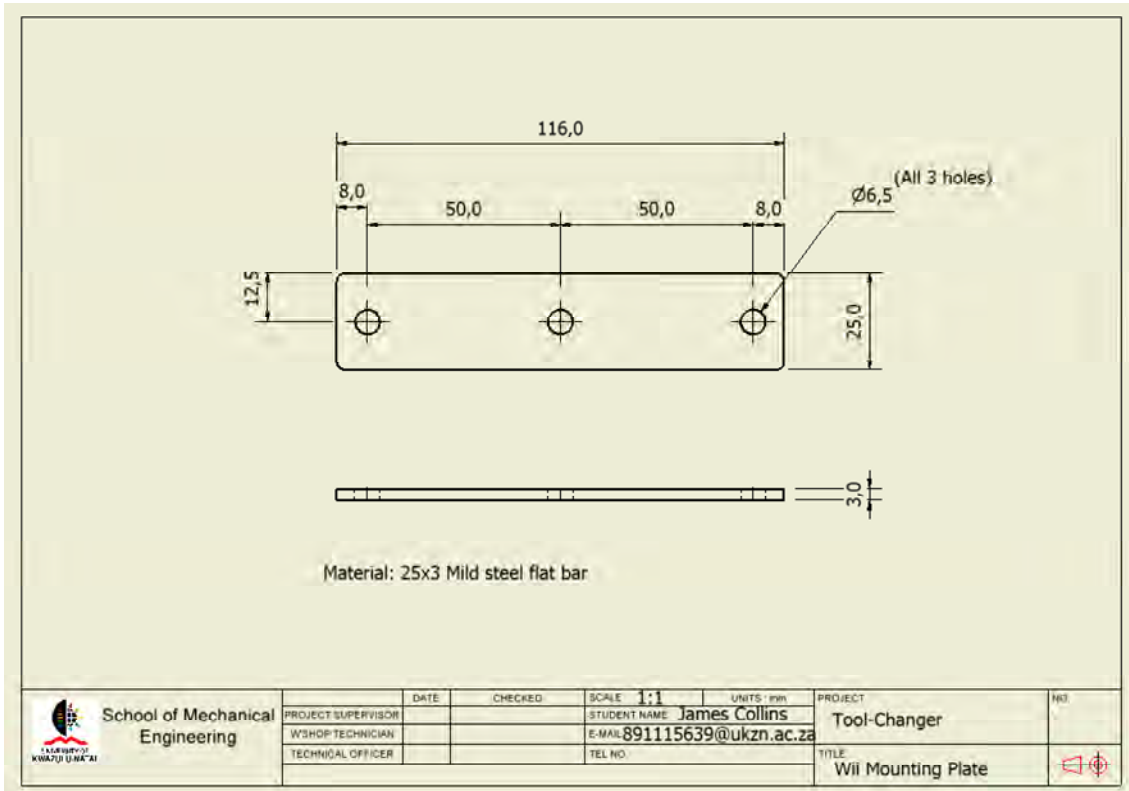


Figure B-2:Wii Mounting Plate

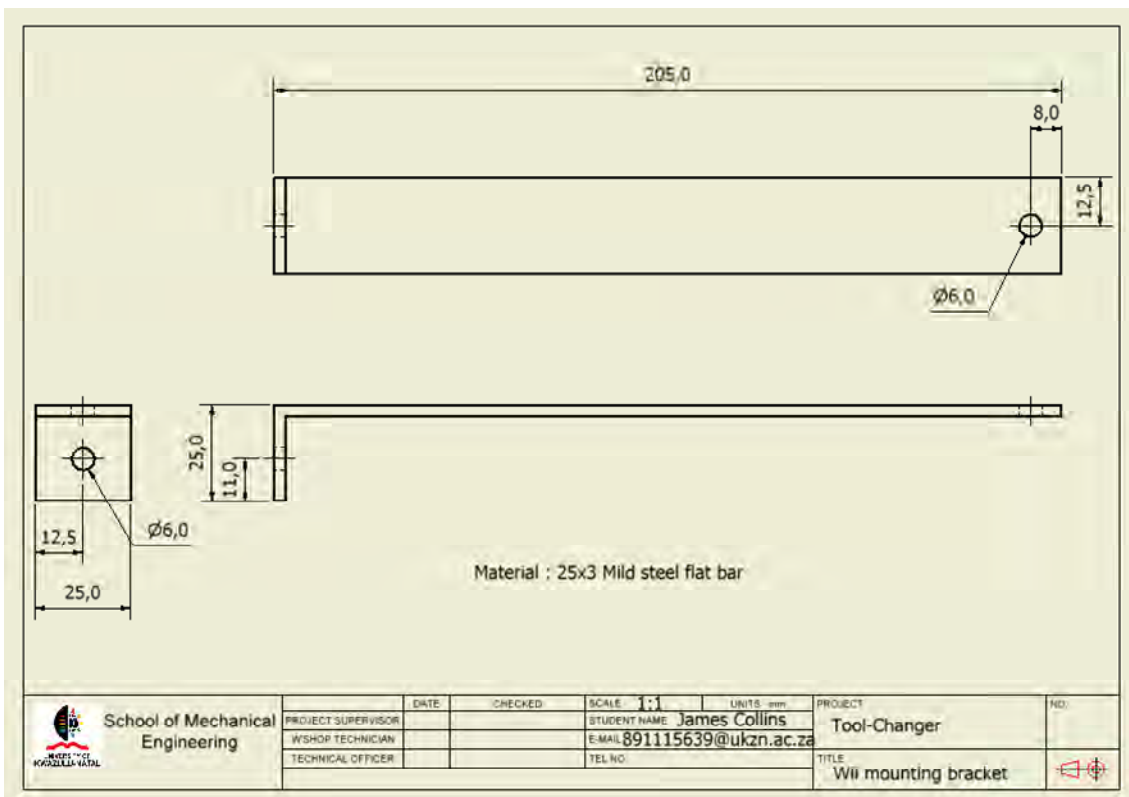


Figure B-3: Wii mounting bracket

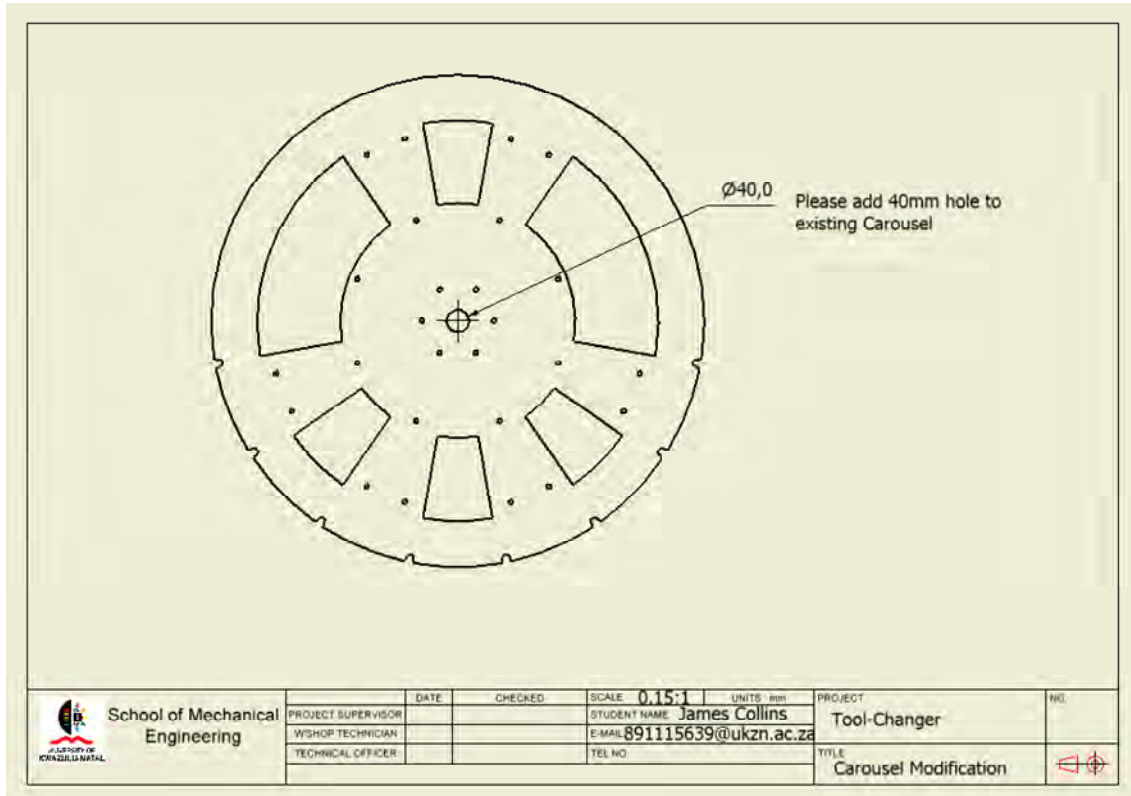


Figure B-4: Carousel Modification

C

Sample 3D Position Calculations

C.1 Two Wiimote, single LED Calculation

Calculate the 3D position of the LED given the following Wiimote pixel readings:

Wiimote 1: $x_{p1} = 835$ and $y_{p1} = 624$

Wiimote 2: $x_{p2} = 171$ and $y_{p2} = 652$

Normalising

$$\hat{x}_1 = 2 \frac{835}{1016} - 1 = 0.644$$

$$\hat{y}_1 = 2 \frac{624}{760} - 1 = 0.642$$

$$\hat{x}_2 = 2 \frac{171}{1016} - 1 = -0.663$$

$$\hat{y}_2 = 2 \frac{652}{760} - 1 = 0.716$$

Mapping

$$x'_1 = \hat{x}_1 \cdot \tan\left(\frac{\varphi}{2}\right) = 0.644 \tan(20.5) = 0.241$$

$$y'_1 = \hat{y}_1 \cdot \tan\left(\frac{\theta}{2}\right) = 0.642 \tan(15.5) = 0.178$$

$$x'_2 = \hat{x}_2 \cdot \tan\left(\frac{\varphi}{2}\right) = -0.663 \tan(20.5) = -0.248$$

$$y'_2 = \hat{y}_2 \cdot \tan\left(\frac{\theta}{2}\right) = 0.716 \tan(15.5) = 0.199$$

Solving for the Intersection

Ray equations:

$$R_1(t_1) = \mathbf{o}_1 + \mathbf{d}_1 t_1$$

$$R_2(t_2) = \mathbf{o}_2 + \mathbf{d}_2 t_2$$

where:

$$\mathbf{o}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{d}_1 = \begin{pmatrix} x'_1 & 0.241 \\ y'_1 & 0.178 \\ 1 & 1 \end{pmatrix}$$

and

$$\mathbf{o}_2 = \begin{pmatrix} 100 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{d}_2 = \begin{pmatrix} x'_2 & -0.248 \\ y'_2 & 0.199 \\ 1 & 1 \end{pmatrix}$$

Solutions are:

$$t_1 = \frac{\det(\mathbf{o}_2 - \mathbf{o}_1 \quad \mathbf{d}_2 \quad \mathbf{d}_1 x \mathbf{d}_2)}{|\mathbf{d}_1 x \mathbf{d}_2|^2}$$

$$t_2 = \frac{\det(\mathbf{o}_2 - \mathbf{o}_1 \quad \mathbf{d}_1 \quad \mathbf{d}_1 x \mathbf{d}_2)}{|\mathbf{d}_1 x \mathbf{d}_2|^2}$$

Since we are using unit values for our z coordinates:

$$\mathbf{d}_1 x \mathbf{d}_2 = \begin{pmatrix} y'_1 - y'_2 & -0.021 \\ x'_2 - x'_1 & -0.489 \\ x'_1 y'_2 - y'_1 x'_2 & 0.092 \end{pmatrix}$$

And

$$|\mathbf{d}_1 x \mathbf{d}_2|^2 = (-0.021)^2 + (-0.489)^2 + (0.092)^2 = 0.248$$

The t_1 numerator =

$$\det \begin{pmatrix} 100 & -0.248 & -0.021 \\ 0 & 0.199 & -0.489 \\ 0 & 1 & 0.092 \end{pmatrix} = 100 \times 0.199 \times 0.092 - 1 \times (-0.489) \times 100 = 50.73$$

The t_2 numerator =

$$\det \begin{pmatrix} 100 & 0.241 & -0.021 \\ 0 & 0.178 & -0.489 \\ 0 & 1 & 0.092 \end{pmatrix} = 100 \times 0.178 \times 0.092 - 1 \times (-0.489) \times 100 = 50.54$$

∴

$$t_1 = \frac{50.73}{0.248} = 204.6$$

And

$$t_2 = \frac{50.54}{0.248} = 203.8$$

Substituting

$$x_1 = 0 + x'_1 \cdot t_1 = 0 + (0.241 \times 204.6) = 49.3mm$$

$$y_1 = 0 + y'_1 \cdot t_1 = 0 + (0.178 \times 204.6) = 36.4mm$$

$$z_1 = 0 + z'_1 \cdot t_1 = 1 \cdot t_1 = 204.6mm$$

And

$$x_2 = 0 + x'_2 \cdot t_2 = 100 + (-0.248 \times 203.8) = 49.5mm$$

$$y_2 = 0 + y'_2 \cdot t_2 = 0 + (0.199 \times 203.8) = 40.6mm$$

$$z_2 = 0 + z'_2 \cdot t_2 = 1 \cdot t_2 = 203.8mm$$

Averaging

$$x = \frac{x_1 + x_2}{2} = \frac{49.3 + 49.5}{2} = \mathbf{49.4mm}$$

$$y = \frac{y_1 + y_2}{2} = \frac{36.4 + 40.6}{2} = \mathbf{38.5mm}$$

$$z = \frac{z_1 + z_2}{2} = \frac{204.6 + 203.8}{2} = \mathbf{204.2mm}$$

This gives the position of the LED as 49.4mm to the right of the centre of Wiimote1, 38.5mm above the centre of Wiimote1 and 204.2mm away from the Wiimote.

C.2 Single Wiimote, two LEDs calculation

Calculate the 3D position of the LED centre point given the following Wiimote pixel readings and a distance (**d**) of 50mm between the LEDs:

$$x_{p1} = 574, y_{p1} = 427 \quad \text{and} \quad x_{p2} = 575, y_{p2} = 652$$

Z Coordinate

Pixel Distance

$$p = \sqrt{(574 - 575)^2 + (427 - 652)^2} = 225$$

Angular FOV

$$\beta_p = 0.0406^\circ \text{ per pixel} - \text{from equation (8.8)}$$

Subtended angle

$$\alpha = \frac{\beta_p \cdot p}{2} = \frac{0.0406 \times 225}{2} = 4.57^\circ$$

∴ from equation (8.7)

$$z = \frac{d}{2 \tan(\alpha)} = \frac{50}{2 \tan(4.57)} = \mathbf{313mm}$$

X Coordinate

$$\text{Average of } x \text{ values: } x_p = \frac{x_{p1} + x_{p2}}{2} = \frac{574 + 575}{2} = 574.5$$

The centre of the camera is located at pixel 508 on the x-axis so, taking the difference from the origin or centre point of the camera gives us the x distance in pixels:

$$x_{px} = x_p - 508 = 574.5 - 508 = 66.5 \text{ pixels}$$

Plugging this through the camera gives the horizontal angle to be:

$$\varphi_{px} = x_{px} \cdot \frac{\varphi}{1016} = 66.5 \times \frac{41}{1016} = 2.68^\circ$$

Now

$$\frac{x}{z} = \tan(\varphi_{px})$$

∴

$$x = z \cdot \tan(\varphi_{px}) = 313 \tan(2.68) = \mathbf{14.7mm}$$

Y Coordinate

Using the same methodology as for the x coordinate except that the y-origin is at 380 pixels:

$$\text{Average of } y \text{ values: } y_p = \frac{y_{p1} + y_{p2}}{2} = \frac{427 + 652}{2} = 539.5$$

$$y_{px} = y_p - 380 = 539.5 - 380 = 159.5 \text{ pixels}$$

$$\theta_{py} = y_{py} \cdot \frac{\theta}{760} = 159.5x \frac{31}{760} = 6.51^\circ$$

$$y = z \cdot \tan(\theta_{py}) = 313 \tan(6.51) = \mathbf{35.7mm}$$

Angle of the Spindle

The angle of the spindle to the horizontal is given by:

$$\theta_{spindle} = \arctan\left(\frac{y_{p1} - y_{p2}}{x_{p1} - x_{p2}}\right) = \arctan\left(\frac{427 - 652}{574 - 575}\right) = \mathbf{89.75^\circ}$$

D

Code Listings

D.1 GlovePIE Script for Two Wiimotes and a single LED

```
// Constants
var.sigma = 41
var.theta = 31
    var.hres = 1016
    var.vres = 760
    var.dx = 100
    var.zo = 1

// Getting pixel values from wiimotes
var.count = Wiimote.Count
var.w1px = wiimote1.dot1x
var.w1py = wiimote1.dot1y
var.w2px = wiimote2.dot1x
var.w2py = wiimote2.dot1y

//Calculating x and y values from the pixel readings
var.x1 = ((2*var.w1px/var.hres)-1)*tan(var.sigma/2)
var.y1 = ((2*var.w1py/var.vres)-1)*tan(var.theta/2)
var.x2 = ((2*var.w2px/var.hres)-1)*tan(var.sigma/2)
var.y2 = ((2*var.w2py/var.vres)-1)*tan(var.theta/2)

//Calculating t1 and t2
var.dc1 = var.y1-var.y2
var.dc2 = var.x2-var.x1
var.dc3 = var.x1*var.y2 - var.y1*var.x2
var.magd = sqrt(var.dc1) + sqrt(var.dc2) + sqrt(var.dc3)
var.t1det = (var.dx*var.y2*var.dc3 - var.zo*var.dc2*var.dx)
var.t1 = var.t1det/var.magd
var.t2det = (var.dx*var.y1*var.dc3 - var.zo*var.dc2*var.dx)
var.t2 = var.t2det/var.magd

//Calculating actual position
var.xa = var.x1*var.t1
var.ya = var.y1*var.t1
var.za = var.t1

var.xb = var.dx + var.x2*var.t2
var.yb = var.y2*var.t2
var.zb = var.t2
```

```
var.x = (var.xa+var.xb)/2
var.y = (var.ya+var.yb)/2
var.z = (var.za+var.zb)/2

// Outputting position to screen
debug = "X,Y,Z Position: " + var.x + ", " + var.y + ", " + var.z
```

D.2 GlovePIE Script for One Wiimote and two LEDs

```
// Constants
var.sigma = 41 //HFOV
var.theta = 31 //VFOV
var.hres = 1016
var.vres = 760

var.dled = 50 //distance between LEDs in mm
var.zo = 1

// Getting pixel values from wiimotes
var.count = Wiimote.Count
var.w1px = wiimote1.dot1x
var.w1py = wiimote1.dot1y
var.w1px2 = wiimote1.dot2x
var.w1py2 = wiimote1.dot2y

//Differences between pixel values
var.diffx = var.w1px - var.w1px2
var.diffy = var.w1py - var.w1py2

// Calculating Z
var.p = (sqrt(sqrt(var.diffx)+ sqrt(var.diffy)))
var.alpha = (var.sigma/var.hres + var.theta/var.vres)* var.p/4

var.z = var.dled/(2*tan(var.alpha))

//Calculating Y
var.ypix = (var.w1py + var.w1py2)/2
var.ypxo= var.ypix-380
var.yangle = var.ypxo* var.theta/760

var.y = var.z*tan(var.yangle)

// Calculating X
var.xpix = (var.w1px + var.w1px2)/2
var.xpxo = var.xpix-508
var.xangle = var.xpxo*var.sigma/1016

var.x = var.z*tan(var.xangle)
```

```
// Calculating the angle of the spindle
var.angle = aTan(var.diffy/var.diffx)

// Outputting position to screen
debug = "X,Y,Z Position: " + var.x + ", " + var.y + ", " + var.z
```

D.3 GUI Main listing

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;
using WiimoteLib;
using System.Threading;
using System.Diagnostics;

namespace WiimoteTest
{
    public partial class MainForm : Form
    {
        // map a wiimote to a specific state user control dealie
        Dictionary<Guid, WiimoteInfo> mWiimoteMap = new Dictionary<Guid,
WiimoteInfo>();
        WiimoteCollection mWC;

        public MainForm()
        {
            InitializeComponent();
        }

        private void MainForm_Load(object sender, EventArgs e)
        {

        }

        private void btnConnect_Click(object sender, EventArgs e)
        {
            // find all wiimotes connected to the system
            mWC = new WiimoteCollection();
            int index = 1;

            try
            {
                mWC.FindAllWiimotes();
            }
            catch (WiimoteNotFoundException ex)
```

```

        {
            MessageBox.Show(ex.Message, "Wiimote not found error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        catch (WiimoteException ex)
        {
            MessageBox.Show(ex.Message, "Wiimote error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Unknown error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    foreach (Wiimote wm in mWC)
    {
        // create a new user control
        WiimoteInfo wi = new WiimoteInfo(wm);
        this.Controls.Add(wi);

        //create a new form for each Wiimote
        WiimoteForm myForm1 = new WiimoteForm();
        int leftpos = 700 + ((index-1) * 285);
        myForm1.Left = leftpos;
        myForm1.Top = 50;
        myForm1.Show();
        myForm1.Text = "Wiimote " + index;
        myForm1.Controls.Add(wi);

        // setup the map from this wiimote's ID to that control
        mWiimoteMap[wm.ID] = wi;

        // connect it and set it up as always
        wm.WiimoteChanged += wm_WiimoteChanged;
        //wm.WiimoteExtensionChanged += wm_WiimoteExtensionChanged;

        wm.Connect();
        //if(wm.WiimoteState.ExtensionType != ExtensionType.BalanceBoard)
        wm.SetReportType(InputReport.IRExtensionAccel,
            IRSensitivity.Maximum, true);

        wm.SetLEDs(index++);

        //User feedback rumble to confirm connection
        wm.SetRumble(true);
        Thread.Sleep(500);
        wm.SetRumble(false);
    }
}

void wm_WiimoteChanged(object sender, WiimoteChangedEventArgs e)
{
    //Updating the User Controls
    WiimoteInfo wi = mWiimoteMap[((Wiimote)sender).ID];
    wi.UpdateState(e);

    //Getting Wiimote States
    Wiimote wm1 = mWC[0];
    Wiimote wm2 = mWC[1];
}

```

```

WiimoteState wm1s = wm1.WiimoteState;
WiimoteState wm2s = wm2.WiimoteState;

////////////////////////////////////
//
//Calculating the 3D position
//
////////////////////////////////////

if (wm1s.IRState.IRSensors[0].Found &&
wm2s.IRState.IRSensors[0].Found)
{
//Constants
float sigma = 41; //Horizontal Field of View
float theta = 31; //Vertical Field of View
float halfsigma = sigma / 2;
float halftheta = theta / 2;
double hsigmarad = halfsigma * (Math.PI/180); //Converting to
radians
double hthetarad = halftheta * (Math.PI / 180);
int hres = 1016; //Horizontal resolution of the camera
int vres = 760; // Vertical resolution of the camera
int dx = 35; // Distance between Wiimotes
int zo = 1; //Unit z vector for calculations

//Getting the pixel values from the Wiimotes

//Wiimote1 - X and Y
double w1px = wm1s.IRState.IRSensors[0].RawPosition.X;
double w1py = wm1s.IRState.IRSensors[0].RawPosition.Y;

//Wiimote2 - X and Y
double w2px = wm2s.IRState.IRSensors[0].RawPosition.X;
double w2py = wm2s.IRState.IRSensors[0].RawPosition.Y;

//Calculating normalized X and Y values from the Pixel values
double x1test = (2 * w1px / hres) - 1;
double x1norm = (2 * w1px / hres) - 1;
double x1a = x1norm * Math.Tan(hsigmarad);

double x1 = ((2 * w1px / hres) - 1) * Math.Tan(hsigmarad);
double y1 = ((2 * w1py / vres) - 1) * Math.Tan(hthetarad);

double x2 = ((2 * w2px / hres) - 1) * Math.Tan(hsigmarad);
double y2 = ((2 * w2py / vres) - 1) * Math.Tan(hthetarad);
//Debug.WriteLine("x {0} y {1} x2 {2} y2 {3}", x1norm, x1a,
x1test, y2);

//Calculating t1 and t2
double dc1 = y1 - y2;
double dc2 = x2 - x1;
double dc3 = x1 * y2 - y1 * x2;
double magd = dc1 * dc1 + dc2 * dc2 + dc3 * dc3;
double t1det = dx * y2 * dc3 - zo * dc2 * dx;
double t1 = t1det / magd;
double t2det = dx * y1 * dc3 - zo * dc2 * dx;
double t2 = t2det / magd;

//Calculating the actual position

```

```
double xa = x1 * t1;
double ya = y1 * t1;
double za = t1;

double xb = dx + x2 * t2;
double yb = y2 * t2;
double zb = t2;

double x = (xa + xb) / 2;
double y = (ya + yb) / 2;
double z = (za + zb) / 2;

string posX = x.ToString();
string posY = y.ToString();
string posz = z.ToString();

//Festo Strings
string fposhs = "n/a";
string fdists = "n/a";
string fposvs = "n/a";
string fheights = "n/a";

int fposv = 0;
int fheight = 0;
int fposh = 0;
int fdisth = 0;

//Setting the Text color for the Green Zones

if (x>15 && x<25)
{
    tbxCalcX.ForeColor = System.Drawing.Color.Green;
}
else
{
    tbxCalcX.ForeColor = System.Drawing.SystemColors.ControlText;
}
if (y > 4 && y <= 94)
{
    tbxCalcY.ForeColor = System.Drawing.Color.Green;
}
else
{
    tbxCalcY.ForeColor = System.Drawing.SystemColors.ControlText;
}
if (z > 252 && z <= 294)
{
    tbxCalcZ.ForeColor = System.Drawing.Color.Green;
}
else
{
    tbxCalcZ.ForeColor = System.Drawing.SystemColors.ControlText;
}

this.SetText(posx, posY, posz);
```



```
    }
}

//Printing out Position Values
delegate void SetTextCallBack(string s,string u,string v);

public void SetText(string txt,string txt2,string txt3)
{
    if (this.InvokeRequired)
    {
        SetTextCallBack txtCall = new SetTextCallBack(SetText);
        this.Invoke(txtCall, txt, txt2,txt3);
    }
    else
    {
        tbxCalcX.Text = txt;
        tbxCalcY.Text = txt2;
        tbxCalcZ.Text = txt3;
    }
}

//Printing out Festo Position
delegate void SetTextCallBack2(string s, string u, string v, string w);

public void SetText2(string txt, string txt2, string txt3, string txt4)
{
    if (this.InvokeRequired)
    {
        SetTextCallBack2 txtCall = new SetTextCallBack2(SetText2);
        this.Invoke(txtCall, txt, txt2, txt3,txt4);
    }
    else
    {
        tbxPosH.Text = txt;
        tbxDistH.Text = txt2;
        tbxPosV.Text = txt3;
        tbxDistV.Text = txt4;
    }
}

private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
{
    if (mWC != null)
    {
        foreach (Wiimote wm in mWC)
            wm.Disconnect();
    }
}
}
```

D.4 Wiimote Info Listing

```

/////////////////////////////////////////////////////////////////
//
//     Adapted by James Collins
//from Brian Peeks Wiimote Tester
//(http://www.brianpeek.com/)
//
/////////////////////////////////////////////////////////////////

using System;
using System.Drawing;
using System.Drawing.Imaging;
using System.Windows.Forms;
using WiimoteLib;

namespace WiimoteTest
{
    public partial class WiimoteInfo : UserControl
    {
        private delegate void
UpdateWiimoteStateDelegate(WiimoteChangedEventArgs args);
        private delegate void
UpdateExtensionChangedDelegate(WiimoteExtensionChangedEventArgs args);

        private Bitmap b = new Bitmap(254, 190,
PixelFormat.Format24bppRgb);
        private Graphics g;
        private Wiimote mWiimote;

        public WiimoteInfo()
        {
            InitializeComponent();
            g = Graphics.FromImage(b);
        }

        public WiimoteInfo(Wiimote wm) : this()
        {
            mWiimote = wm;
        }

        public void UpdateState(WiimoteChangedEventArgs args)
        {
            if (mWiimote != null) BeginInvoke(new
UpdateWiimoteStateDelegate(UpdateWiimoteChanged), args);
        }

        private void chkLED_CheckedChanged(object sender, EventArgs e)
        {
            mWiimote.SetLEDs(chkLED1.Checked, chkLED2.Checked,
chkLED3.Checked, chkLED4.Checked);
        }

        private void chkRumble_CheckedChanged(object sender, EventArgs e)
        {
            mWiimote.SetRumble(chkRumble.Checked);
        }

        private void UpdateWiimoteChanged(WiimoteChangedEventArgs args)
        {

```

```

        WiimoteState ws = args.WiimoteState;

        //Accelerometer readings
        txtbxAccelX.Text = ws.AccelState.Values.X.ToString();
        txtbxAccelY.Text = ws.AccelState.Values.Y.ToString();
        txtbxAccelZ.Text = ws.AccelState.Values.Z.ToString();

        //LED Status
        chkLED1.Checked = ws.LEDState.LED1;
        chkLED2.Checked = ws.LEDState.LED2;
        chkLED3.Checked = ws.LEDState.LED3;
        chkLED4.Checked = ws.LEDState.LED4;

        //Image for IR visualisation
        g.Clear(Color.Black);
        UpdateIR(ws.IRState.IRSensors[0], tbxIR1X, tbxIR1Y, chkFound1,
Color.White);

        pbIR.Image = b;

        pbBattery.Value = (ws.Battery > 0xc8 ? 0xc8 :
(int)ws.Battery);
        lblBattery.Text = ws.Battery.ToString();
    }

    private void UpdateIR(IRSensor irSensor, TextBox tbxIRX, TextBox
tbxIRY, CheckBox chkFound, Color color)
    {
        chkFound.Checked = irSensor.Found;

        if(irSensor.Found)
        {
            tbxIRX.Text = irSensor.RawPosition.X.ToString(); //+ ", " +
irSensor.Size;
            tbxIRY.Text = irSensor.RawPosition.Y.ToString();
            int y = irSensor.RawPosition.Y / 4;
            int ypoint = 187 - y; // To make pixel go up when the LED goes up

            //Drawing point on screen
            g.DrawRectangle(new Pen(color),
(int)(irSensor.RawPosition.X / 4), ypoint,
irSensor.Size+4,
irSensor.Size+4);
        }
    }

    public Wiimote Wiimote
    {
        set { mWiimote = value; }
    }
}

```

E

Further Results

E.1 Wiimote Horizontal FOV Analysis

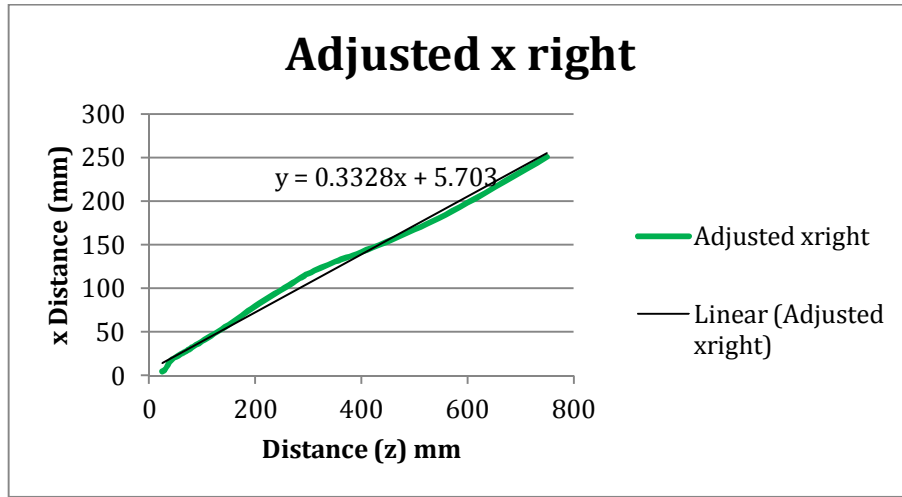
Table E-1 shows a more in depth analysis of the results. The gradient and hence the FOV is analysed at each point. The adjusted values of the right hand side are used and the table includes only values where results were obtained on both sides (i.e. $z > 80\text{mm}$).

Table E-1: Further analysis of the Horizontal FOV results

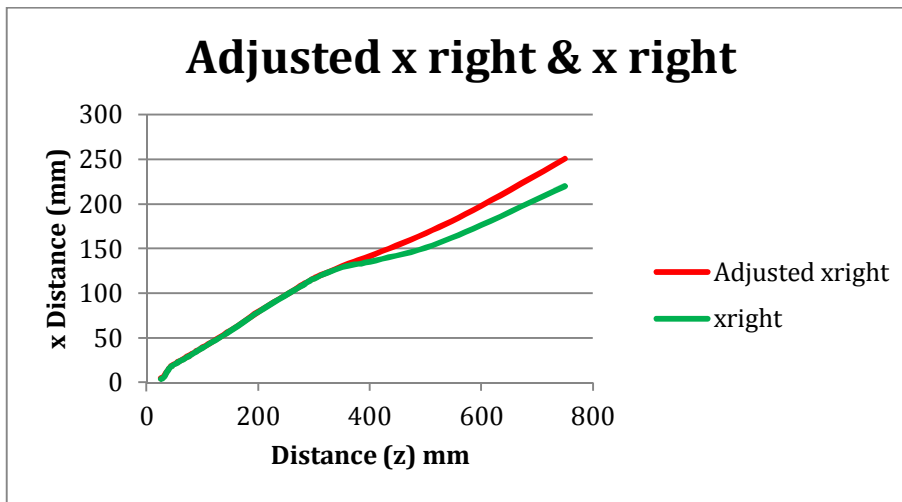
Distance (mm)	x left (mm)	x right (Adjusted)	x left gradient	x right gradient	x left FOV (°)	x right FOV (°)	Combined FOV (°)
81	39	32					
100	46	39	0.353	0.368	38.9	40.4	39.6
150	65	58	0.375	0.380	41.1	41.7	41.4
201	84	80	0.378	0.424	41.4	46.0	43.7
271	111	106	0.381	0.377	41.7	41.3	41.5
300	123	117	0.426	0.364	46.1	40.0	43.0
351	141	130	0.357	0.267	39.3	29.9	34.6
400	159	141	0.358	0.226	39.4	25.4	32.4
500	195	167	0.361	0.255	39.7	28.6	34.2
600	231	198	0.363	0.308	39.9	34.3	37.1
750	284	250	0.352	0.351	38.8	38.7	38.8
				Averages	40.6	36.6	38.6

Figure E-1 Shows graphs of the x right values.

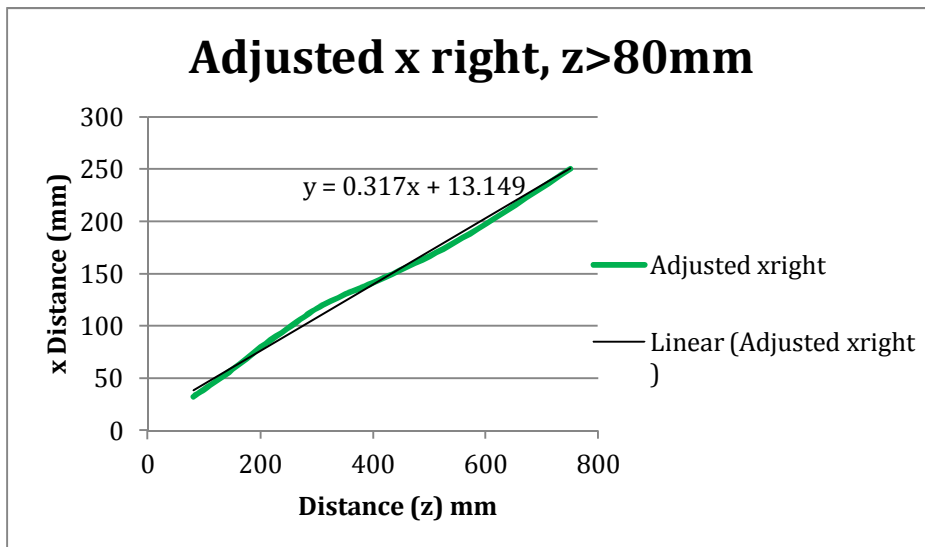
- Shows the plot of the adjusted x right values over the full range of readings along with a linear trend line. The line gives a slope of 0.3328 and FOV of 36.8°.
- Shows the difference between the raw values and the adjusted values. It can be noticed that the major departure at readings taken greater than 350mm.
- Shows the adjusted values only over the range where values were captured going in both directions i.e. where the distance was greater than 80mm. In this case the slope of the trend line is 0.317 which gives an FOV of 35.2°. The adjusted values give a closer linear relationship of the FOV compared to the raw values as seen by the closer proximity to the trend line.



a) The adjusted x right graph with a trend line.



b) The difference between the raw x right values and the adjusted values



c) Plot of the adjusted x right values further than 80mm from the Wiimote

Figure E-1: Graphs of the x right values

The final plot in this section (Figure E-2) shows the variation in the FOV as the LED was moved away from the Wiimote. The values for both the left hand side and the right hand side are shown as well as a combined value which is the average of the two. The numerical values are listed in Table E-1. The graph shows clearly how the left side (the blue line) had a more consistent result around 40°.

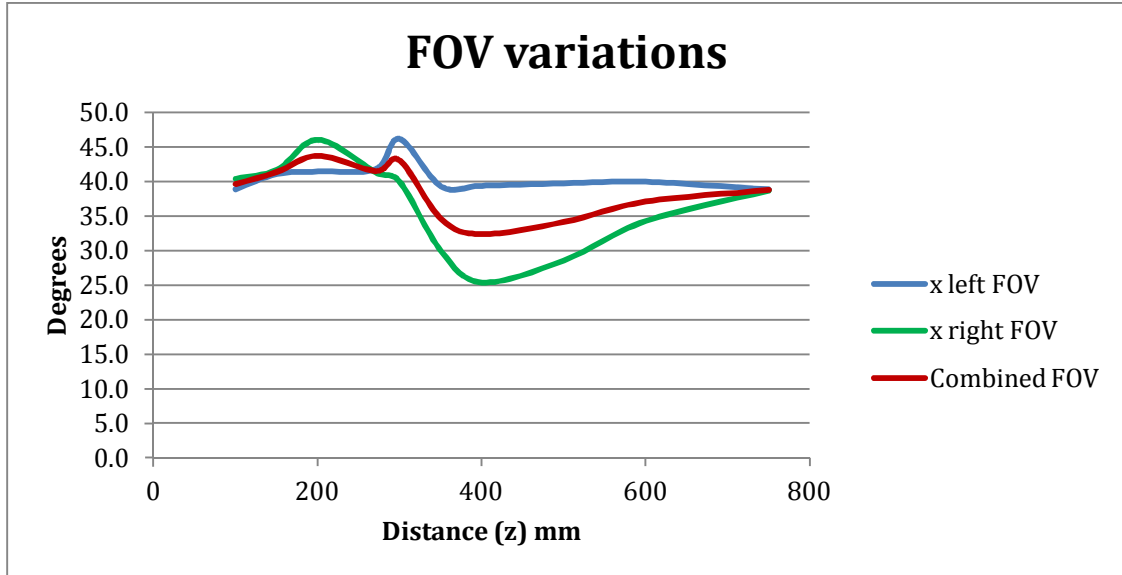


Figure E-2: Variations in the FOV along the z-axis

E.2 Stereo Wiimote Robot Test Results

Table E-2: Stereo Wiimote Robot Test Results (mm)

Number	Robot x	Robot y	Robot z	Error X	Error Y	Error Z	Error Dist
1	-0.024	0.01	513.534	2.394	0.16	-8.504	8.836
2	50.096	0.012	513.534	0.314	1.248	-9.664	9.749
3	50.096	0.014	550.207	0.264	1.226	-16.797	16.844
4	50.288	0.022	600.128	0.692	0.888	-22.618	22.646
5	50.289	0.033	750.656	0.851	0.947	-39.416	39.437
6	50.289	0.052	500.384	0.411	0.848	-11.284	11.323
7	50.289	0.059	400.445	-0.079	0.651	-0.345	0.741
8	50.289	0.062	350.908	-0.089	0.988	0.862	1.314
9	50.29	0.066	300.986	-0.05	0.854	4.294	4.378
10	50.291	0.069	250.105	-0.321	0.721	4.935	4.998
11	50.291	0.073	200.76	-0.311	0.487	6.81	6.834
12	50.292	0.076	149.879	-0.492	0.334	8.551	8.572
13	50.19	0.228	124.633	-1.16	-0.008	10.637	10.700
14	57.895	0.149	150.588	-1.115	0.361	8.932	9.009
15	36.952	0.155	150.59	0.428	-0.075	10.33	10.339
16	46.491	0.151	150.598	-0.511	0.149	8.042	8.060
17	46.456	0.149	200.713	-0.466	0.631	6.877	6.922
18	73.49	0.189	200.713	-0.83	0.981	7.137	7.252

Number	Robot x	Robot y	Robot z	Error X	Error Y	Error Z	Error Dist
19	18.437	0.217	200.722	0.593	-0.137	9.208	9.228
20	50.189	0.239	250.071	-0.129	0.261	4.539	4.548
21	83.195	0.371	250.073	-1.175	1.059	4.347	4.626
22	0.06	0.385	250.075	0.71	-0.935	8.665	8.744
23	20.009	0.387	250.075	0.451	-0.357	4.255	4.294
24	60.015	0.392	250.074	-0.365	0.578	4.496	4.548
25	60.015	0.396	300	-0.365	0.574	3.75	3.811
26	96.43	0.432	300	-1.61	1.108	2.72	3.349
27	-18.334	0.444	300	1.524	-0.854	5.78	6.038
28	0.002	0.458	300	0.998	-0.928	1.58	2.087
29	20.451	0.462	300	0.539	-0.352	3.43	3.490
30	80	0.466	300	-0.99	0.844	2.23	2.582
31	80	0.472	350.3	-0.95	0.948	0.45	1.416
32	107.271	0.488	350.3	-2.221	1.522	-2.1	3.415
33	-36.556	0.515	350.3	2.086	-1.255	2.69	3.628
34	0.014	0.51	350.3	1.226	-0.78	-1.28	1.936
35	80.079	0.518	350.3	-0.969	0.882	-0.46	1.389
36	80.08	0.526	400.024	-0.95	0.734	-1.904	2.251
37	117.926	0.55	400.025	-2.646	1.38	-5.295	6.078
38	-54.184	0.578	400.027	3.104	-1.778	-0.977	3.708
39	-20.183	0.586	400.027	2.623	-0.966	-5.157	5.866
40	20.138	0.589	400.027	1.212	-0.509	-3.227	3.484
41	20.06	100.156	600.318	-1.54	1.014	-5.398	5.704
42	20.054	100.149	750.078	-1.524	3.061	-1.598	3.774
43	20.059	190.798	750.013	-4.709	3.542	1.927	6.199
44	20.06	-208.313	750.02	3.97	7.803	-2.13	9.010
45	20.06	-140.341	750.021	3.12	6.691	-3.461	8.154
46	20.059	-70.245	750.022	0.851	3.565	8.328	9.099
47	20.058	0.61	750.023	-0.218	3.91	7.217	8.211
48	20.067	70.691	750.026	-0.587	2.319	-4.636	5.217
49	20.067	140.2	750.027	-1.797	0.96	-8.497	8.738
50	50.096	140.203	750.028	-3.116	5.687	8.612	10.780
51	50.098	-208.735	750.026	3.852	8.955	5.834	11.361
52	50.089	-140.568	750.029	1.611	5.798	4.801	7.698
53	50	-140.499	750.035	2.2	11.679	-15.795	19.767
54	50	-70.111	750.044	1.14	7.561	5.686	9.529
55	50.009	0.061	750.06	-0.029	6.799	14.11	15.663
56	50.009	70.72	750.063	-0.929	6.33	10.057	11.920
57	50.014	140.997	750.064	-1.874	-19.507	-22.774	30.045
58	50.014	95.856	600.31	-1.664	4.404	-0.38	4.723
59	50.02	-177.907	600.318	2.99	12.227	-7.878	14.849
60	50.02	-140.557	600.318	2.75	10.137	-8.118	13.275
61	50.02	-69.993	600.318	1.43	6.393	5.062	8.279
62	50.02	0.859	600.318	-0.21	5.641	7.762	9.598

Number	Robot x	Robot y	Robot z	Error X	Error Y	Error Z	Error Dist
63	50.021	34.556	600.32	-0.681	6.644	13.4	14.972
64	50.021	70.27	600.32	-0.721	6.24	11.99	13.536
65	50.019	35.145	500.037	-0.719	4.935	11.253	12.309
66	50.018	78.991	500.039	-1.538	6.619	13.911	15.482
67	50.017	-149.362	500.044	1.803	10.602	-8.554	13.741
68	50.018	-100.015	500.043	1.262	6.445	-1.133	6.664
69	50.018	-50.668	500.042	1.062	4.858	5.348	7.303
70	50.018	0.026	500.014	-0.488	5.114	14.126	15.031
71	50.019	49.95	500.04	-1.119	4.86	8.69	10.019
72	50.02	35.56	400.55	-1	4.41	8.68	9.787
73	50.017	67.328	400.552	-1.807	5.692	12.898	14.213
74	50.014	-121.626	400.556	1.106	8.046	-4.026	9.065
75	50.014	-100.697	400.556	1.296	5.577	0.714	5.770
76	50.014	-50.772	400.555	0.476	4.372	2.085	4.867
77	50.014	0.879	400.55	-0.784	3.801	10.65	11.335
78	50.015	50.996	400.554	-1.165	4.464	7.916	9.162
79	50.086	0.056	350.038	-0.746	3.414	8.602	9.285
80	50.085	55.725	350.04	-1.505	3.645	7.52	8.491
81	50.086	-107.094	350.042	1.784	5.064	3.248	6.275
82	50.085	-65.871	350.043	0.675	3.771	2.257	4.446
83	50.086	-33.421	350.042	0.104	3.171	4.158	5.230
84	50.086	33.208	350.041	-1.076	3.362	6.319	7.238
85	50.087	0	300.043	-0.707	2.6	5.677	6.284
86	50.088	47.262	300.044	-1.948	3.618	10.386	11.169
87	50.087	-92.989	300.047	1.133	4.899	1.063	5.139
88	50.088	-65.531	300.047	0.562	3.081	2.373	3.929
89	50.088	-32.888	300.046	-0.178	2.458	5.334	5.876
90	50.088	33.166	300.045	-1.348	2.944	6.445	7.213
91	50.106	28.653	250.082	-0.526	0.737	0.268	0.944
92	50.107	-78.45	250.083	1.143	3.03	0.357	3.258
93	50.107	-50.606	250.084	0.733	1.546	-0.924	1.945
94	50.107	-25.24	250.085	0.403	1.36	-1.325	1.941
95	50.106	0.092	250.087	0.174	1.068	-0.287	1.119
96	50.107	15.167	250.087	0.063	0.723	0.173	0.746
97	50.104	15.177	200.424	-0.664	0.233	0.626	0.942
98	50.104	23.753	200.425	-0.764	0.317	0.325	0.889
99	50.11	-64.664	200.425	0.45	2.684	-0.245	2.732
100	50.111	-40.147	200.425	0.139	0.797	0.185	0.830
101	50.112	-20.75	200.425	0.088	0.81	-0.465	0.938
102	50.113	0.179	200.425	-0.183	0.501	-0.015	0.534
103	50.156	0.175	150.006	-0.866	0.105	0.854	1.221
104	50.158	37.172	150.006	-1.198	-0.412	0.494	1.360
105	50.164	-50.088	150.007	0.416	1.418	1.283	1.957
106	50.165	-25.289	150.007	-0.505	0.259	1.003	1.152

Number	Robot x	Robot y	Robot z	Error X	Error Y	Error Z	Error Dist
107	50.167	25.14	150.008	-1.047	-0.38	0.882	1.421
108	20.078	0.158	350.1	0.942	0.582	-2.61	2.835
109	20.069	88.936	350.112	-0.819	0.014	-1.502	1.711
110	20.07	-105.804	350.116	2.01	5.264	-6.766	8.805
111	20.069	-65.174	350.118	1.711	2.204	-4.708	5.473
112	20.069	-33.928	350.119	1.261	1.348	-3.619	4.063
113	20.069	30.013	350.118	-0.059	0.027	-1.798	1.799
114	20.069	60.545	350.117	-0.339	-0.085	-2.007	2.037
115	20.071	0.458	300.456	0.459	0.212	0.504	0.714
116	20.072	73.277	300.461	-1.032	-0.127	-1.401	1.745
117	20.072	-91.534	300.464	2.078	3.424	-2.654	4.805
118	20.073	-60.22	300.465	1.467	1.37	-2.375	3.110
119	20.075	-30.236	300.465	0.905	0.446	-0.285	1.048
120	20.077	25.391	300.466	-0.027	0.119	0.404	0.422
121	20.078	50.612	300.466	-0.848	0.148	1.414	1.655
122	20.115	50.744	300.467	-0.782	0.236	1.533	1.737
123	20.116	50.75	250.008	-1.453	-0.31	0.992	1.786
124	20.112	63.007	250.009	-1.639	-0.247	2.411	2.926
125	20.108	-76.99	250.009	2.165	3.24	-6.579	7.646
126	20.107	-50.126	250.01	1.356	0.506	-3.76	4.029
127	20.108	-25.142	250.025	0.755	-0.028	-3.315	3.400
128	20.109	0.015	250.025	0.004	-0.385	-0.515	0.643
129	20.109	20.128	250.026	-0.376	-0.478	-0.096	0.616
130	20.11	40.267	250.026	-0.967	-0.487	-0.446	1.171
131	20.113	20.235	200.58	-0.53	-0.735	2.51	2.669
132	20.042	43.066	200.582	-1.349	-0.126	3.318	3.584
133	20.195	-62.84	200.588	1.488	1.22	-2.078	2.832
134	20.128	-40.133	200.588	1.045	-0.407	0.222	1.143
135	20.12	-20.058	200.589	0.713	-0.882	0.491	1.236
136	20.112	0.097	200.589	0.211	-1.067	1.531	1.878
137	20.103	20.977	200.589	-0.65	-0.727	2.831	2.994
138	80.021	0.084	300.009	0.002	0.716	2.761	2.852
139	80.041	-5.114	204.471	-1.451	0.774	-0.201	1.657
140	20.032	-5.078	213.056	-0.692	-0.772	-1.446	1.779
141	20.134	-5.078	213.702	0.339	0.603	0.588	0.908
142	80.064	0.111	198.681	-0.301	0.722	4.599	4.665
143	80.064	0.113	200.09	-0.311	0.76	4.81	4.880
144	80.046	53.267	210.121	-0.723	0.476	2.969	3.093
145	80.059	-65.033	210.121	0.804	2.226	2.019	3.111
146	80.06	-40.456	210.12	0.433	-4.121	0.67	4.198
147	80.06	-20.87	210.12	0.243	0.013	1.33	1.352
148	80.061	0.06	210.12	0.072	-0.237	1	1.030
149	80.062	25.022	221.06	-0.409	-0.289	-10.02	10.033
150	80.062	25.143	250.057	-0.079	-0.06	1.213	1.217

Number	Robot x	Robot y	Robot z	Error X	Error Y	Error Z	Error Dist
151	80.063	66.891	250.057	-0.83	-0.878	1.683	2.072
152	80.063	-64.621	250.06	0.93	0.834	0.89	1.534
153	80.063	-40.042	250.06	0.62	0.505	0.39	0.890
154	80.063	-20.265	250.059	0.36	0.078	0.991	1.057
155	80.063	0.09	250.058	0.1	0.113	1.512	1.520
156	80.063	25.245	250.058	-0.22	-0.122	0.752	0.793
157	80.063	50.015	250.057	-0.71	-0.382	1.933	2.094
158	80.062	50.013	300.176	-0.549	-0.38	2.104	2.207
159	80.061	80.459	300.178	-0.598	-1.186	0.472	1.410
160	80.06	-79.745	300.181	0.693	1.498	-2.501	2.997
161	80.06	-50.749	300.181	0.453	0.452	1.349	1.493
162	80.061	-25.787	300.18	0.262	0.07	0.95	0.988
163	80.061	0.331	300.179	-0.128	-0.298	-0.249	0.409
164	80.061	25.487	300.179	-0.428	-0.384	0.531	0.783
165	80.04	25.552	300.12	-0.53	-0.442	-2.61	2.700
166	80.047	25.549	350.014	-0.737	-0.369	-1.484	1.698
167	80.046	93.645	350.014	-1.046	0.815	5.186	5.353
168	80.045	-104.088	350.019	0.785	2.308	-8.469	8.813
169	80.045	-65.877	350.018	0.375	0.697	-6.838	6.884
170	80.045	-33.042	350.018	0.175	-0.668	-2.518	2.611
171	80.045	0.081	350.017	-0.565	-0.711	-4.517	4.607
172	80.045	33.108	350.017	-0.535	-0.258	-0.577	0.828
173	80.046	65.943	350.016	-0.876	0.157	0.604	1.076
174	80.057	65.944	400.097	-0.937	0.126	3.523	3.648
175	80.057	107.191	400.097	-1.177	-0.041	2.753	2.994
176	80.076	-117.219	400.113	0.684	2.749	-10.953	11.313
177	80.076	-80.736	400.113	0.534	1.206	-8.983	9.079
178	80.077	-40.22	400.112	0.313	-0.48	-3.202	3.253
179	80.076	0.084	400.111	0.194	-1.194	-0.511	1.313
180	80.076	33.88	400.11	-0.196	-0.93	0.93	1.330
181	80.077	33.885	500.059	-0.267	-0.675	1.911	2.044
182	80.075	134.126	500.06	-1.575	0.644	6.74	6.951
183	80.076	-139.17	500.06	1.024	3.68	-15.89	16.343
184	80.076	-100.959	500.059	0.924	1.219	-11.079	11.184
185	80.076	-50.075	500.058	0.504	0.315	-8.078	8.100
186	80.076	0.042	500.057	-0.126	-1.422	-2.457	2.842
187	80.076	50.929	500.056	-0.576	-1.309	0.064	1.432
188	80.077	100.853	500.055	-1.567	-0.323	3.365	3.726
189	80.077	100.858	600.049	-1.817	-1.638	1.631	2.940
190	80.077	160.009	600.05	-2.257	3.771	17.39	17.937
191	80.081	-134.645	600.06	2.189	0.485	-6.58	6.951
192	80.08	-100.466	600.06	2.07	-0.484	-4.35	4.842
193	80.08	-50.158	600.061	1.2	-1.372	-3.141	3.632
194	80.079	0.919	600.062	0.281	-1.949	-2.942	3.540

Number	Robot x	Robot y	Robot z	Error X	Error Y	Error Z	Error Dist
195	80.079	50.652	600.063	-0.079	-1.052	8.487	8.552
196	80.079	100.575	600.063	-0.439	0.585	15.807	15.824
197	80.08	-163.666	750.879	2.07	0.876	-12.769	12.965
198	80.081	197.358	750.879	-1.711	4.692	23.631	24.153
199	80.081	130.714	750.879	-0.661	1.676	27.561	27.620
200	80.082	60.064	750.878	0.638	-2.204	15.842	16.007
201	80.082	0.548	750.878	0.698	-2.708	-4.068	4.937
202	80.082	-50.328	750.878	1.778	-2.492	0.932	3.200
203	80.082	-100.273	750.877	1.568	-0.597	-10.957	11.085

E.3 Single Wiimote test results on the tool-changing unit

Table E-3: Single Wii Z Error (mm)

				Ave	1.48
				Max	10.27
				Min	-6.75
Reading	Z on Axis	Wii Z	Wii Z Difference	Wii Z Offset	Wii Z Error
1	125.5	363.38	237.88	228	9.88
2	83.5	314.56	231.06	228	3.06
3	93	323.2	230.2	228	2.2
4	103.5	332.43	228.93	228	0.93
5	114	345.51	231.51	228	3.51
6	123.5	357.83	234.33	228	6.33
7	126.5	359.64	233.14	228	5.14
8	123.5	356.01	232.51	228	4.51
9	113	345.51	232.51	228	4.51
10	103	335.6	232.6	228	4.6
11	93	320.28	227.28	228	-0.72
12	83.5	315.97	232.47	228	4.47
13	83.5	313.16	229.66	228	1.66
14	83.5	318.85	235.35	228	7.35
15	83.5	317.41	233.91	228	5.91
16	83.5	318.84	235.34	228	7.34
17	83.5	313.16	229.66	228	1.66
18	83.5	311.77	228.27	228	0.27
19	83.5	310.38	226.88	228	-1.12
20	83.5	304.99	221.49	228	-6.51
21	83.5	318.84	235.34	228	7.34
22	83.5	318.85	235.35	228	7.35

Reading	Z on Axis	Wii Z	Wii Z Difference	Wii Z Offset	Wii Z Error
23	83.5	321.77	238.27	228	10.27
24	83.5	315.98	232.48	228	4.48
25	83.5	309.02	225.52	228	-2.48
26	84	309.01	225.01	228	-2.99
27	105	332.44	227.44	228	-0.56
28	105	335.62	230.62	228	2.62
29	105	326.25	221.25	228	-6.75
30	105	327.78	222.78	228	-5.22
31	105	334.02	229.02	228	1.02
32	105	332.44	227.44	228	-0.56
33	124	354.23	230.23	228	2.23
34	124	354.17	230.17	228	2.17
35	124	350.63	226.63	228	-1.37
36	124	352.42	228.42	228	0.42
37	124	352.39	228.39	228	0.39
38	124	350.68	226.68	228	-1.32
39	124	348.94	224.94	228	-3.06
40	124	345.46	221.46	228	-6.54
41	124	348.95	224.95	228	-3.05
42	124	350.7	226.7	228	-1.3
43	124	350.63	226.63	228	-1.37
44	124	350.63	226.63	228	-1.37

Table E-4: Single Wiimote and Festo Error comparison (mm)

Reading	Z on Axis	Festo Z	Festo Offset	Ave	-1 mm
				Max	8.09 mm
				Min	-10.01 mm
					Difference Between
					Wii Error and Festo
					Error
1	125.5	294	168.5	0	7.38
2	83.5	251.75	168.5	-0.25	0.81
3	93	261.1	168.5	-0.4	0.1
4	103.5	271.17	168.5	-0.83	-0.74
5	114	281.22	168.5	-1.28	2.29
6	123.5	291.18	168.5	-0.82	4.65
7	126.5	294.12	168.5	-0.88	3.52
8	123.5	290.96	168.5	-1.04	3.05
9	113	281.12	168.5	-0.38	2.39
10	103	270.95	168.5	-0.55	2.65
11	93	261.28	168.5	-0.22	-3

Reading	Z on Axis	Festo Z	Festo Offset	Festo Error	Difference Between Wii Error and Festo Error
12	83.5	251.68	168.5	-0.32	2.29
13	83.5	251.68	168.5	-0.32	-0.52
14	83.5	251.68	168.5	-0.32	5.17
15	83.5	251.68	168.5	-0.32	3.73
16	83.5	251.68	168.5	-0.32	5.16
17	83.5	251.68	168.5	-0.32	-0.52
18	83.5	251.68	168.5	-0.32	-1.91
19	83.5	251.68	168.5	-0.32	-3.3
20	83.5	251.68	168.5	-0.32	-8.69
21	83.5	251.68	168.5	-0.32	5.16
22	83.5	251.68	168.5	-0.32	5.17
23	83.5	251.68	168.5	-0.32	8.09
24	83.5	251.68	168.5	-0.32	2.3
25	83.5	251.68	168.5	-0.32	-4.66
26	84	254.31	168.5	1.81	-7.3
27	105	274.26	168.5	0.76	-3.82
28	105	274.26	168.5	0.76	-0.64
29	105	274.26	168.5	0.76	-10.01
30	105	274.26	168.5	0.76	-8.48
31	105	274.26	168.5	0.76	-2.24
32	105	274.73	168.5	1.23	-4.29
33	124	292.8	168.5	0.3	-0.57
34	124	292.8	168.5	0.3	-0.63
35	124	292.8	168.5	0.3	-4.17
36	124	292.8	168.5	0.3	-2.38
37	124	292.8	168.5	0.3	-2.41
38	124	292.8	168.5	0.3	-4.12
39	124	292.8	168.5	0.3	-5.86
40	124	292.8	168.5	0.3	-9.34
41	124	292.8	168.5	0.3	-5.85
42	124	292.8	168.5	0.3	-4.1
43	124	292.8	168.5	0.3	-4.17
44	124	292.8	168.5	0.3	-4.17

Figure E-3 shows the graphical distribution of the Wii error compared to the Festo error in the Z-direction.

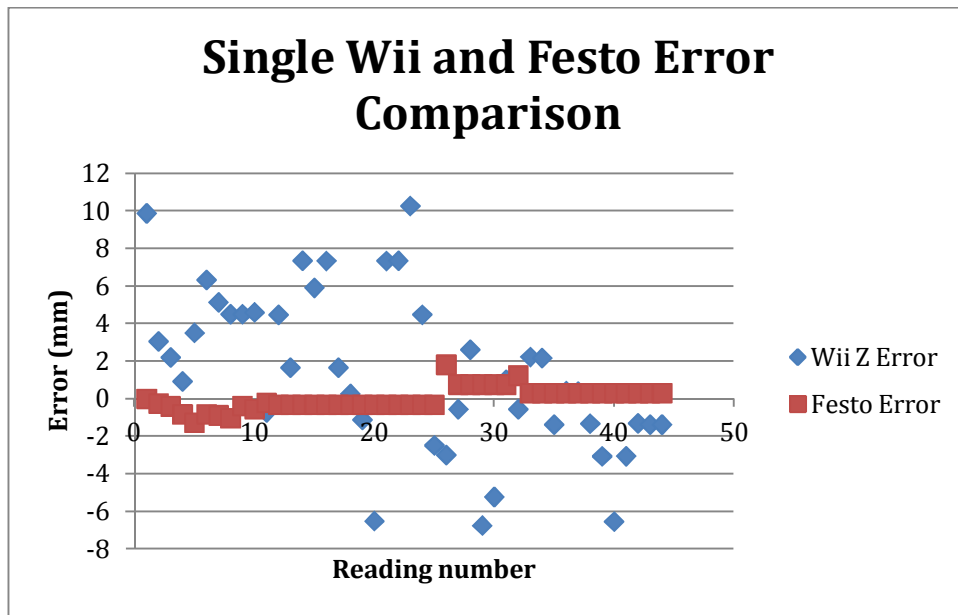


Figure E-3: Comparison between Festo and Single Wii Errors

Table E-5: Single Wiimote X and Y Errors (mm)

					Ave	0.33	0.18
					Max	2.5	2.42
					Min	-1.13	-1.72
Reading	Z on Axis	Wii Y	Festo Y	Diff Y	Session Y Offset	Wii Y Error	Wii X Error
1	125.5	38.43	19.81	18.62	18	0.62	2.42
2	83.5	38.71	19.55	19.16	18	1.16	0.19
3	93	38.37	19.83	18.54	18	0.54	0.49
4	103.5	37.79	19.46	18.33	18	0.33	0.93
5	114	37.78	20.04	17.74	18	-0.26	1.56
6	123.5	37.97	19.95	18.02	18	0.02	2.15
7	126.5	37.77	19.78	17.99	18	-0.01	2.36
8	123.5	37.65	19.73	17.92	18	-0.08	2.06
9	113	37.78	19.8	17.98	18	-0.02	1.56
10	103	37.91	19.8	18.11	18	0.11	0.85
11	93	37.56	19.65	17.91	18	-0.09	0.69
12	83.5	38.54	19.63	18.91	18	0.91	0.14
13	83.5	35.94	19.48	16.46	15	1.46	-0.32
14	83.5	26.96	11.42	15.54	15	0.54	-0.51
15	83.5	22.18	6.75	15.43	15	0.43	-0.12
16	83.5	15.33	0	15.33	15	0.33	-0.84
17	83.5	35.94	19.48	16.46	15	1.46	-0.32
18	83.5	40.85	24.89	15.96	15	0.96	-0.5

Reading	Z on Axis	Wii Y	Festo Y	Diff Y	Session Y Offset	Wii Y Error	Wii X Error
19	83.5	46.86	30.99	15.87	15	0.87	-0.45
20	83.5	52.96	37.57	15.39	15	0.39	-0.06
21	83.5	61.7	44.2	17.5	15	2.5	0.73
22	83.5	16.02	1.7	14.32	15	-0.68	-0.17
23	83.5	25.83	10.2	15.63	15	0.63	0.31
24	83.5	36.49	22.2	14.29	15	-0.71	0.26
25	83.5	45.87	30.7	15.17	15	0.17	0.36
26	84	41.83	28.01	13.82	12	1.82	-1.72
27	105	41.39	28.83	12.56	12	0.56	-0.24
28	105	27.78	15.37	12.41	12	0.41	-0.86
29	105	56.05	43.5	12.55	12	0.55	0.18
30	105	62.46	50.06	12.4	12	0.4	-0.1
31	105	14.87	3.38	11.49	12	-0.51	-1.35
32	105	28.49	16.68	11.81	12	-0.19	-0.88
33	124	28.43	16.68	11.75	12	-0.25	-0.65
34	124	29.19	17.62	11.57	11	0.57	-0.52
35	124	37.46	26.74	10.72	11	-0.28	-0.42
36	124	37.78	26.74	11.04	11	0.04	-0.22
37	124	47.45	35.92	11.53	11	0.53	0.15
38	124	57.3	45.6	11.7	11	0.7	0.57
39	124	57.14	45.6	11.54	11	0.54	0.49
40	124	56.07	45.6	10.47	11	-0.53	0.09
41	124	69.46	58.46	11	11	0	0.62
42	124	69.94	58.46	11.48	11	0.48	0.7
43	124	28.14	17.88	10.26	11	-0.74	-0.67
44	124	13.36	3.49	9.87	11	-1.13	-1.16

E.4 Single Wiimote Repeatability

The following three tables detail the repeatability readings recorded for each axis of the Wiimote. Festo Y is the height recorded by the Festo vertical arm when it touched the tip of the spindle. This was the most accurate way of measuring the height of the spindle. The tables display the average, minimum and maximum of each error as well as for the difference between the errors (the repeatability). All results are in mm.

Table E-6: Single Wiimote Z repeatability

	Ave	1.660	1.568	-0.092
	Max	7.35	10.27	4.21
	Min	-3.05	-6.54	-5.22
Position on Machine axis (mm)	Festo Y	Error Z	Error Z (2)	Difference in the Z Errors
83.5	11.42	7.35	10.27	2.92
83.5	19.48	1.66	1.66	0
83.5	19.55	3.06	4.47	1.41
83.5	24.89	0.27	4.48	4.21
83.5	30.99	-1.12	-2.48	-1.36
93	19.83	2.2	-0.72	-2.92
103	19.46	0.93	4.6	3.67
105	15.37	2.62	-0.56	-3.18
113	20.04	3.51	4.51	1
123.5	19.95	6.33	4.51	-1.82
124	17.62	2.17	-1.37	-3.54
124	26.74	-1.37	0.42	1.79
124	45.6	-1.32	-6.54	-5.22
124	58.46	-3.05	-1.3	1.75

Table E-7: Single Wiimote X repeatability

	Ave	0.209	0.352	0.143
	Max	2.15	2.06	0.82
	Min	-0.86	-0.88	-0.48
Position on Machine axis (mm)	Festo Y	Error X	Error X (2)	Difference in X Errors
83.5	11.42	-0.51	0.31	0.82
83.5	19.48	-0.32	-0.32	0
83.5	19.55	0.19	0.14	-0.05
83.5	24.89	-0.5	0.26	0.76
83.5	30.99	-0.45	0.36	0.81
93	19.83	0.49	0.69	0.2
103	19.46	0.93	0.85	-0.08
105	15.37	-0.86	-0.88	-0.02
113	20.04	1.56	1.56	0
123.5	19.95	2.15	2.06	-0.09
124	17.62	-0.52	-0.67	-0.15
124	26.74	-0.42	-0.22	0.2
124	45.6	0.57	0.09	-0.48
124	58.46	0.62	0.7	0.08

Table E-8: Single Wiimote Y repeatability

	Ave	0.501	0.103	-0.399
	Max	1.46	1.46	0.48
	Min	-0.28	-0.74	-1.67
Position on Machine axis (mm)	Festo Y	Error Y	Error Y (2)	Difference in Y Errors
83.5	11.42	0.54	0.63	0.09
83.5	19.48	1.46	1.46	0
83.5	19.55	1.16	0.91	-0.25
83.5	24.89	0.96	-0.71	-1.67
83.5	30.99	0.87	0.17	-0.7
93	19.83	0.54	-0.09	-0.63
103	19.46	0.33	0.11	-0.22
105	15.37	0.41	-0.19	-0.6
113	20.04	-0.26	-0.02	0.24
123.5	19.95	0.02	-0.08	-0.1
124	17.62	0.57	-0.74	-1.31
124	26.74	-0.28	0.04	0.32
124	45.6	0.7	-0.53	-1.23
124	58.46	0	0.48	0.48

F

Quotation and Datasheet

F.1 National Instruments Smart Camera Quotation



University of KwaZulu-Natal
 James Collins
 King George V avenue
 Durban KwaZulu Natal 4041
 South Africa

T/A NI Solutions Ltd.
VAT# 4010193144

Date: 10/19/2010
 Phone: 27-31-2601227
 Fax:
 Email: 891115639@ukzn.ac.za
 RFQ/Ref :

Quotation No. NISQ 8213

FOR QUESTIONS CONCERNING THIS QUOTE, PLEASE CONTACT National Instruments South Africa at 011 805 8197
 Please reference your quote number when you place an order or make a payment.

QTY.	Product PN	Description	List Price	Disc %	Amount
1	-----	NI Smart Camera - Includes NI Vision Builder for Automated Inspection Software	.00	10.00	.00
1	780146-01	Camera, NI 1722 Smart Camera	18,230.00	10.00	16,407.00
1	-----	Power Supplies and I/O Accessories	.00	10.00	.00
1	781093-01	NI PS-15 Power Supply, 24 VDC, 5 A, 100-120/200-240 VAC Input	1,910.00	10.00	1,719.00
1	-----	Mounting Accessories	.00	10.00	.00
1	780239-01	Mounting Kit, NI-17XX Series Smart Camera Tripod Adapter Plate	540.00	10.00	486.00
1	-----	Low-Cost Camera Link	.00	10.00	.00
1	779210-02	NI PCI-1426 with NI-IMAQ, 32MB	8,650.00	10.00	7,785.00
1	-----	Cable	.00	10.00	.00
1	187676-02	Camera Link Cable 2 m	1,360.00	10.00	1,224.00
1	-----	Optional Accessories	.00	10.00	.00
1	777574-01	LabVIEW Machine Vision and Image Processing Training Kit Includes course manual and course solutions disk. Apply 100% of price to a course within 90 days	2,540.00	10.00	2,286.00
SUBTOTAL (ZAR):					29,907.00
VAT (ZAR):					4,186.98
TOTAL (ZAR):					34,093.98

F.2 Vishay TSAL6400 Infrared LED Datasheet

The following pages so the datasheet for the IR LED used in the positioning system.



TSAL6400

Vishay Semiconductors

High Power Infrared Emitting Diode, 940 nm, GaAlAs/GaAs



FEATURES

- Package type: leaded
- Package form: T-1 $\frac{1}{2}$
- Dimensions (in mm): $\varnothing 5$
- Peak wavelength: $\lambda_p = 940$ nm
- High reliability
- High radiant power
- High radiant intensity
- Angle of half intensity: $\varphi = \pm 25^\circ$
- Low forward voltage
- Suitable for high pulse current operation
- Good spectral matching with Si photodetectors
- Compliant to RoHS directive 2002/95/EC and in accordance to WEEE 2002/96/EC
- Halogen-free according to IEC 61249-2-21 definition



RoHS
COMPLIANT
HALOGEN
FREE

DESCRIPTION

TSAL6400 is an infrared, 940 nm emitting diode in GaAlAs/GaAs technology with high radiant power molded in a blue-gray plastic package.

APPLICATIONS

- Infrared remote control units with high power requirements
- Free air transmission systems
- Infrared source for optical counters and card readers

PRODUCT SUMMARY				
COMPONENT	I_o (mW/sr)	φ (deg)	λ_p (nm)	t_r (ns)
TSAL6400	40	± 25	940	800

Note

Test conditions see table "Basic Characteristics"

ORDERING INFORMATION			
ORDERING CODE	PACKAGING	REMARKS	PACKAGE FORM
TSAL6400	Bulk	MOQ: 4000 pcs, 4000 pcs/bulk	T-1 $\frac{1}{2}$

Note

MOQ: minimum order quantity

ABSOLUTE MAXIMUM RATINGS				
PARAMETER	TEST CONDITION	SYMBOL	VALUE	UNIT
Reverse voltage		V_R	5	V
Forward current		I_F	100	mA
Peak forward current	$t_p/T = 0.5, t_p = 100 \mu s$	I_{FM}	200	mA
Surge forward current	$t_p = 100 \mu s$	I_{FSM}	1.5	A
Power dissipation		P_V	160	mW
Junction temperature		T_J	100	$^\circ C$
Operating temperature range		T_{amb}	- 40 to + 85	$^\circ C$
Storage temperature range		T_{stg}	- 40 to + 100	$^\circ C$
Soldering temperature	$t \leq 5$ s, 2 mm from case	T_{sd}	260	$^\circ C$
Thermal resistance junction/ambient	J-STD-051, leads 7 mm soldered on PCB	R_{thJA}	230	K/W

Note

$T_{amb} = 25$ $^\circ C$, unless otherwise specified

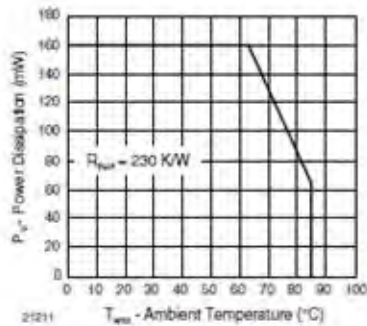
TSAL6400Vishay Semiconductors High Power Infrared Emitting Diode,
940 nm, GaAlAs/GaAs

Fig. 1 - Power Dissipation Limit vs. Ambient Temperature

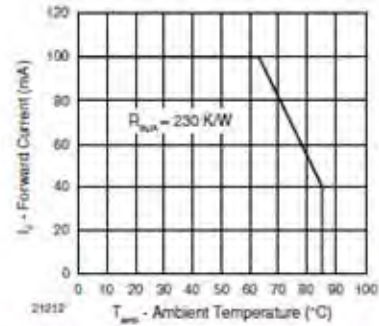


Fig. 2 - Forward Current Limit vs. Ambient Temperature

BASIC CHARACTERISTICS						
PARAMETER	TEST CONDITION	SYMBOL	MIN.	TYP.	MAX.	UNIT
Forward voltage	$I_F = 100 \text{ mA}$, $t_p = 20 \text{ ms}$	V_F		1.35	1.6	V
	$I_F = 1 \text{ A}$, $t_p = 100 \mu\text{s}$	V_F		2.6	3	V
Temperature coefficient of V_F	$I_F = 1 \text{ mA}$	TK V_F		-1.8		mV/K
Reverse current	$V_R = 5 \text{ V}$	I_R			10	μA
Junction capacitance	$V_R = 0 \text{ V}$, $f = 1 \text{ MHz}$, $E = 0$	C_j		25		pF
Radiant intensity	$I_F = 100 \text{ mA}$, $t_p = 20 \text{ ms}$	I_a	25	40	125	mW/sr
	$I_F = 1 \text{ A}$, $t_p = 100 \mu\text{s}$	I_a	220	310		mW/sr
Radiant power	$I_F = 100 \text{ mA}$, $t_p = 20 \text{ ms}$	ϕ_a		35		mW
Temperature coefficient of ϕ_a	$I_F = 20 \text{ mA}$	TK ϕ_a		-0.6		%/K
Angle of half intensity		φ		± 25		deg
Peak wavelength	$I_F = 100 \text{ mA}$	λ_p		940		nm
Spectral bandwidth	$I_F = 100 \text{ mA}$	$\Delta\lambda$		50		nm
Temperature coefficient of λ_p	$I_F = 100 \text{ mA}$	TK λ_p		0.2		nm/K
Rise time	$I_F = 100 \text{ mA}$	t_r		800		ns
Fall time	$I_F = 100 \text{ mA}$	t_f		800		ns
Virtual source diameter	Method: 63 % encircled energy	d		2.2		mm

Note $T_{amb} = 25^\circ\text{C}$, unless otherwise specified.



TSAL6400

High Power Infrared Emitting Diode, Vishay Semiconductors
940 nm, GaAlAs/GaAs

BASIC CHARACTERISTICS

$T_{amb} = 25\text{ }^{\circ}\text{C}$, unless otherwise specified

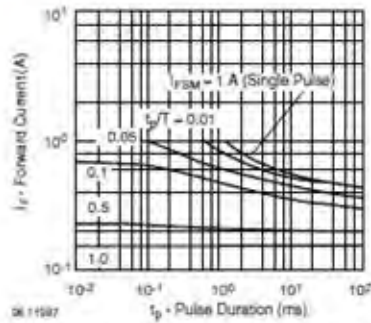


Fig. 3 - Pulse Forward Current vs. Pulse Duration

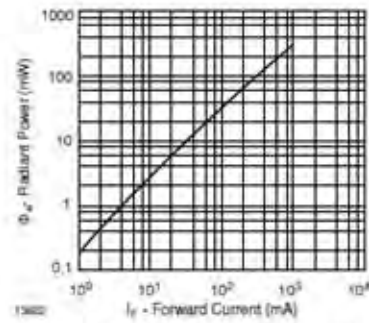


Fig. 6 - Radiant Power vs. Forward Current

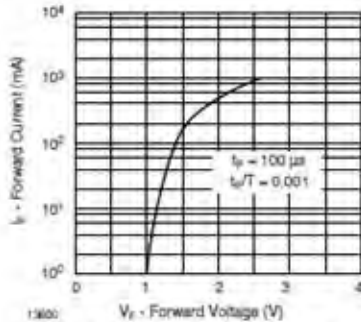


Fig. 4 - Forward Current vs. Forward Voltage

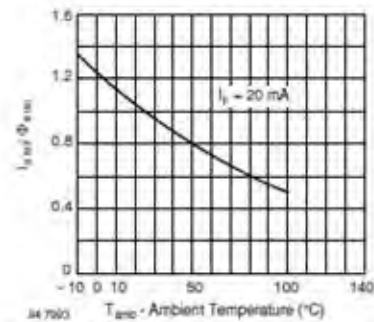


Fig. 7 - Relative Radiant Intensity/Power vs. Ambient Temperature

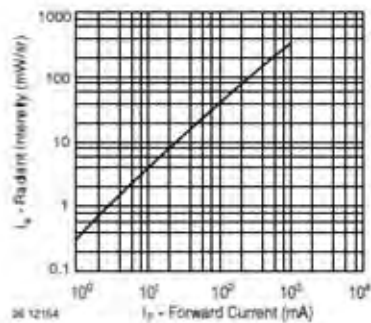


Fig. 5 - Radiant Intensity vs. Forward Current

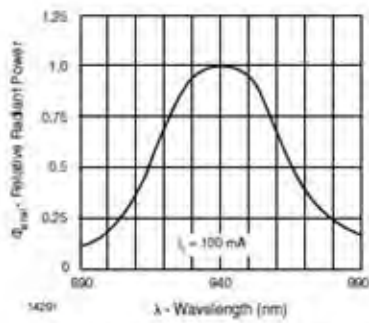


Fig. 8 - Relative Radiant Power vs. Wavelength

TSAL6400

Vishay Semiconductors High Power Infrared Emitting Diode,
940 nm, GaAlAs/GaAs

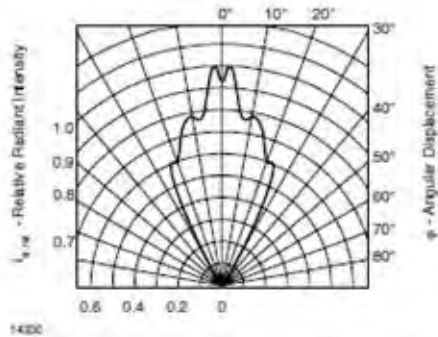
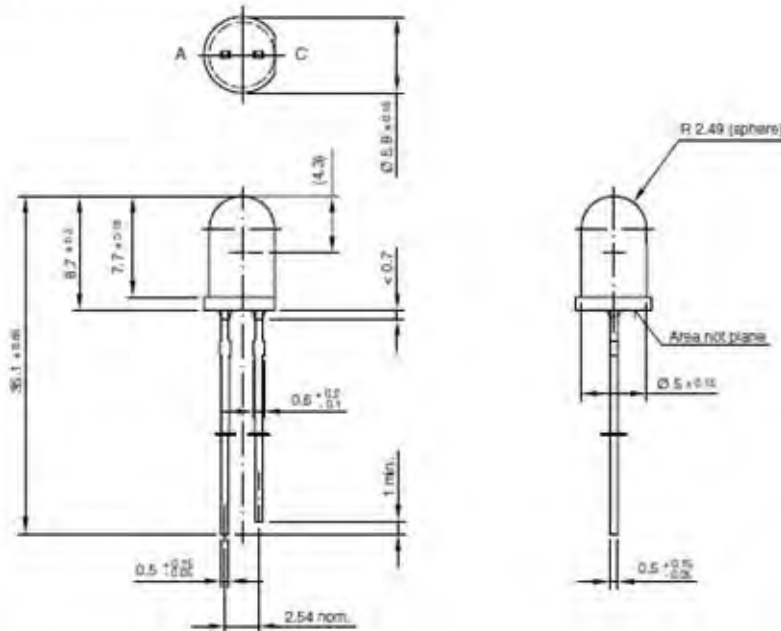


Fig. 9 - Relative Radiant Intensity vs. Angular Displacement

PACKAGE DIMENSIONS in millimeters



Drawing-No.: 6.544-5259.07-4

Issue: 4; 19.05.09

14340