Progress in the Active Development of Large Optics for Astronomy

A Thesis Submitted for the Degree

of

Doctor of Philosophy of the University of London

by

David Jon Rees

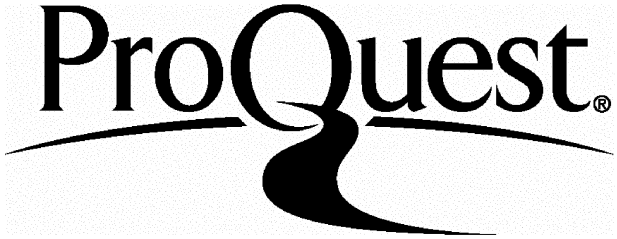
Optical Science Laboratory

Department of Physics and Astronomy

University College

University of London

1994



This Thesis is dedicated
to Debbie,
my inspiration in life.

Abstract

An international consortium consisting of the United States, United Kingdom, Canada, Brazil, Chile and Argentina are planning to revolutionize the field of astronomy by building the two GEMINI 8m astronomical telescopes. These are designed to provide unprecedented image quality, which should significantly increase our knowledge and understanding of the structure and dynamics of the universe. To provide such superb image quality, the specifications of almost every aspect of the telescope are tighter than for any other ever built.

The work described in this thesis is part of the ongoing research currently being undertaken in the Optical Science Laboratory into the production of large, highly aspheric optical surfaces. Meeting the design specifications for the GEMINI secondary mirrors is believed to be impossible using conventional craft techniques, but it is expected to be a tractable problem when utilizing the Active Lap technique described herein. The goal of the project is to demonstrate this technique by producing a $\frac{1}{3}$ scale model of these mirrors.

Broadly speaking, the Active Lap uses closed loop control of a system comprising of arrays of load cells and force actuators to control the ablation of the mirror in real time. This is a significant step forward in the field, and aims to propel conventional craft techniques which date back to the time of Sir Isaac Newton into the 21st century!

A major contribution of the author to the Active Lap research project was the data acquisition and control software, which was designed to be ergonomic and make efficient use of cpu time. Other significant contributions involved calibration methods, system testing, and development of the closed loop control algorithms. In particular the novel idea of utilizing artificial neural networks to replace these algorithms is discussed.

Finally, the performance of the Active Lap is evaluated, and suggestions are made for both the strategy for its future use, and the investigations required for its future developments.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Astronomical Background

Astronomy has strong claims to be the oldest of all scientific disciplines, with direct references to astronomical events being found in the mythology and religious beliefs of every civilization. Events such as eclipses, comets or meteor showers etc. have long been thought to bear influence on mankind's destiny[1], and other prehistoric monuments such as Stonehenge and the Pyramids have been erected with remarkable, and certainly intensional alignment with the heavens. Indeed the very timescales which we now use to govern our increasingly ordered lives (ie. days, months and years) have their origins back in antiquity, as we know from calendars dating back to the Assyrians and the Chaldeans. [1]

A natural curiousity has always been the driving force behind mankind's quest to unravel the mysteries of the Universe, in addition to which, the study of Astronomy helps us to understand and appreciate our environment and to glimpse something of our origins and evolution. It allows us to explore a time span of 15 billion years into the past, and to conjecture a series of events that will occur long after mankind has ceased to exist. It is a mark of mankind's increasing maturity that we are able to use the pursuit of abstract knowledge to define, and come to terms with, our somewhat insignificant rôle in the universe.

[1] Well known examples of this would include the appearance of Halley's Comet in 1066 being seen as a fatal omen for King Harold, and the Star of Bethlehem heralding the birth of Christ.

Modern Astronomy can be said to have been born when Galileo first turned an optical telescope skywards in the early 17th century [1] [16], and observations which for millennia had been taken using the naked eye alone were then superceded. With only a few nights observation Galileo probably contributed more new knowledge than all his predecessors put together. Thus began a trend that has continued ever since: the introduction of new and improving observational techniques going hand in hand with the advancing of theoretical knowledge.

A long list of technical advances since Galileo's time have led us to the state of the art new generation of astronomical telescopes which are currently being built around the world. These have adopted a two pronged approach to improving on previous technologies: firstly to increase the aperture size of the telescope, which simply increases its light gathering power and allows us to explore deeper into the Universe at greater resolution; and secondly the advent of space telescopes which are placed above the Earth's atmosphere, eliminating atmospheric seeing effects and giving access to wavelengths which are absorbed by the atmosphere.

1.1.1 The GEMINI Project

The work described in this thesis is in connection with the construction of ground based, optical telescopes with extremely large apertures: specifically, the GEMINI project. This comprises of an international consortium of the USA (50%), UK (25%), Canada (15%) and Brazil/Chile/Argentina (10%), which is building two 8m telescopes. In order to provide full sky coverage, one telescope is situated in each hemisphere at a site with exquisite seeing: namely at Mauna Kea in Hawaii and Cerro Pachon in Chile.

The larger the telescope aperture, the greater its light gathering power. This enables it to observe fainter objects than a smaller telescope, and also increases the resolution it can achieve. Therefore it is imperative to attempt to construct the largest telescope compatible with current technology (and within the available budget!). At the present time an 8m telescope satisfies these criteria, being able to take advantage of the best performance the atmosphere can deliver [27], balanced with the cost, risk, feasibility of transportation, fabrication of instrumentation and main optical components, etc. [16] [26] that the construction of such an instrument will

incur.

Key examples of the Astronomical program the two GEMINI telescopes are expected to undertake can be summarized as follows [16] [28], in order of increasing scale.

- What is the nature of the disks around nearby stars?
- What is the origin of our planetary system and are there others?
- How do stars form and what conditions lead to protostellar collapse?
- What is the internal structure of stars?
- What is the chemical enrichment history of the Galaxy and the Universe?
- How did galaxies form and evolve in the Universe?
- What is the structure and history of the Universe as a whole?

In order to be able answer these questions satisfactorily, the images the telescope produces need to be of the absolutely highest quality. Any degradation of the image quality could be catastrophic and leave the afore mentioned questions unanswered. Therefore the technical specifications of every aspect of the telescope, from the site seeing to the mirror properties to the stability of the instrumentation, are tighter than for any other ever built.

The item of particular interest for this thesis is the secondary mirror, the specifications of which will be extremely challenging to meet. In order to reduce the size and cost of the telescope support structure the primary mirror will be very fast at f/1.8.This means that the secondary mirror will necessarily be very highly aspheric. There will eventually be two interchangeable secondary mirrors: namely f/6 and f/16, to be used as follows [28]

- The f/6 secondary will be used for wide field optical/UV observations. It should provide 45 arcmin field of view, with a primary spectral range of 0.3–1.0μm, but usable up to 2.2μm. It should also provide image quality of 0.25 arcsec.

- The f/16 secondary will provide high angular resolution operation, allowing 3.5 arcmin of view and a primary spectral range of 0.4–30μm extendable to 0.35–1000μm. Near diffraction limited imaging of <0.1 arcsec at >2.2μm will be achieved using active optics and tilt-tip correction.

At first light, the f/16 is the only secondary mirror that will be provided, with the f/6 following afterwards.

There is as yet no formal specification for the secondary mirror, but in order to reach the image quality figures detailed above, it is likely that it will require its surface figure to be accurate to within at least $\frac{1}{10}$ of the wavelength of the light from HeNe laser. [7] In particular the f/6 will be 2.5m in diameter and over 1000 waves aspheric. (ie. It will be a hyperbolic, convex mirror, with the outside edge of the mirror departing from the closest sphere by over 1000 waves.) Since conventional techniques naturally produce spherical mirrors, we have the classic, extremely difficult problem of controlling a large parameter (asphericy $> 1000\lambda$) within very fine tolerances ($\pm\frac{1}{10}\lambda$).

Producing large, highly aspheric mirrors with such a surface finish is believed to be impossible using conventional craft techniques. The Active Lap is the Optical Science Laboratory's attempt to solve this problem.

1.2 Technological Background

Polishing a mirror is arguably the most critical stage in its fabrication. The aim is to correctly *figure* the surface (ie. create the correct surface profile) whilst removing defects, to produce a perfectly smooth surface. The figure defines the image quality the mirror is able to produce, and as detailed above, will probably need to be accurate to within $\frac{1}{10}\lambda$. However, a further complication is that even within this tolerance, the mirror could still be unacceptable if the surface has high spatial frequency ripples, as these will result in the stray light performance being unacceptable.

1.2.1 Conventional Polishing

The conventional polishing techniques described in this section are *craft* techniques, where a successful outcome is governed by the accumulated skill and intuition of the optician doing the polishing. These techniques have remained essentially unchanged in principle since the first telescopes were made in the 17th century. Applying modern computer-controlled technology can only improve their success rate and efficiency.

Conventional polishing uses a rigid tool (the lap) made of either wood, metal or glass onto the bottom surface of which are stuck pitch facets. [14] [33] Prior to polishing the lap is allowed to rest on the mirror, so that the pitch can flow and make intimate contact with the glass. This process is known as "pressing the lap". Once the lap is pressed an abrasive slurry (e.g. cerium oxide and water) is poured between the mirror and the lap, and polishing can begin.

The mirror is rotated on a turn-table, and the lap is moved in a zig-zag motion across it. This is done either by hand or by two mechanical arms, depending on the size of the lap. The typical stroke of the lap is about $\frac{1}{3}$ of the diameter of the work piece. A smoother and smoother surface can be achieved by using finer and finer grains of carborundum for glass on glass, then cerium oxide on pitch, for progressive polishing runs. The figure can to some extent be controlled by altering either the length and geometry of the stroke or the distribution of pitch facets on the lap.

After a polishing run, the mirror is tested, using various methods such as the knife edge test or optical interferometry, which will show whether there are any figuring errors or zonal defects in the mirror surface. These are then corrected during subsequent polishing runs using either the full size lap, or local figuring using smaller laps. The full size lap has a superior ability to remove material and smooth the surface, but due to its rigidity is unable to ablate localized zonal spots. These can be removed using the small laps, but tend to produce high frequency spatial ripples in the mirror surface.

Aspherising a mirror is normally carried out after a best-fit sphere is produced, by changing the stroke and pitch distribution used during polishing. It is also possible to make small aspherics by using a lap embodying a compliant (e.g. rubber) layer,

to adapt to the aspheric profile of the mirror. Both these approaches are successful, if slow processes for small and medium size optics with asphericies up to a few tens of waves, but cannot cope with the large optics and asphericy $> 1000\lambda$, which are probably required for the GEMINI secondary mirrors.

Consider the position once the lap has pressed on a near perfect convex hyperbolic secondary mirror. The profile of the pitch will now have exactly the same hyperbolic shape as the mirror. For large optics, as soon as the lap is displaced, its rigidity is such that it will cause a mismatch between the two hyperbolas and polishing will occur preferentially at the high pressure areas indicated in figure 4.3. Thus the edge and centre of the mirror will automatically be ablated, and the surface will revert to a spherical profile. The only alternative to this is to attempt aspherising using small laps. However at the edge of the mirror, the radius of curvature is changing so rapidly that there would still be areas of non contact within the areas of local figuring, in addition to which there are the problems of producing high spatial frequency ripples.

1.2.2 Other Modern Methods of Polishing

The classical methods detailed above are the techniques for producing aspheric mirrors that the Active Lap attempts to supercede. There are however other research groups around the world have come up with their own ideas for improving on the conventional polishing techniques. These are briefly outlines in the following sections and attention is drawn to the problems inherent in each approach. A detailed survey can be found in [16].

1.2.2.1 Polishing with a Stressed Lap

The Steward Observatory at the University of Arizona is developing polishing with a stressed lap. A $\frac{1}{3}$ sub-diameter lap is populated with 12 moment generating actuators. During polishing, these apply a bending moment to the lap, continuously distorting its shape in order to attempt to make it fit the local shape of the work piece. The shape of the lap is described by only the first three Zernike polynomial terms, and the mirror profile is controlled by manually selecting the path of the lap

over the work piece. (See section 5.2 for a brief description of Zernike polynomials.)

Calibration of the stressed lap is done off-line in a static condition. It is not clear whether applying the results to the dynamic situation during polishing is indeed theoretically valid [16], although it appears to work well in practice.

Using only the lowest three Zernike polynomials, combined with a somewhat dubious offline calibration procedure, the stressed lap is not able to conform to to work piece to within optical precision, with a best reported performance of about 3μm RMS. In addition to this they have reported significant hysteresis effects. This misfit will cause different ablation conditions on the mirror surface during polishing.

Two further problems with the stressed lap are firstly that using a sub-diameter lap will introduce high spatial frequency ripples on the mirror surface; and secondly, it hasn't eliminated the need for the optician to know his craft, and choose the correct pitch distribution, path of the lap (i.e. stoke), etc.

1.2.2.2 Computer Controlled Optical Surfacing Processes (CCOS)

Two institutions, namely Litton Itek Optical Systems and the Contraves Goerz Corporation, are independently developing CCOS polishing. A lap, typically $\frac{1}{10}$ to $\frac{1}{5}$ the size of the work piece, is driven in a orbital motion whilst scanning the work piece. The lap surface conforms to the mirror surface using a multi axis CNC machine for the lap movement, and ablation is controlled by dwelling for different lengths of time at different parts of the mirror.

A major problem with this technique is that the physical processes that control polishing are not well understood, and there are no firm theoretical or empirical foundations for using dwelling time as the primary parameter for controlling the ablation. Also, this technique removes glass at a very slow rate, which means that in practice it requires precision grinding prior to polishing if the process is to be completed within a reasonable time span. Once again, the use of a sub-diameter tool will introduce high spacial frequency ripples.

As indicated in [16] it is interesting to not that Itek has not been using this technique to produce the Keck telescope segments, but instead used the stressed mirror technique mentioned below. They did however use it to correct the segments after

cutting when they had warped, but as mentioned above it did introduce high spatial frequency ripples.

1.2.2.3 Stressed Mirror Polishing

The University of California at Berkeley has developed a stressed mirror technique to produce the Keck telescope segments, which was also used by Litton Itek and the Tinsley Laboratory to figure them.

A 1.8m hexagonal segment is produced by stressing a circular work piece by a pre-calculated amount. This is then polishing using conventional processes to a spherical shape and cut into a hexagon. The idea is that the hexagonal, spherical mirror will "spring back" into the correct aspheric shape once the stress given to the mirror is removed.

The problem with this technique is that it was found that excess stress relief when cutting caused the surface to warp, and the hexagonal mirror could not be refigured. Intensive research culminated in the production of a *warping harness* which deforms the surface to the figure required. This provided a one off solution applicable to the Keck telescope, but the stressed mirror polishing technique itself has not yet been demonstrated to be a successful process.

1.2.2.4 Linear Membrane Tool Processes

Linear membrane tool process are being developed at two separate institutions, Zeiss and the University of Turku. These use Preston's Law as the underlying principle of their polishing processes. (See section 2.1) As opposed to the processes described above, both these institutions realize that it is the polishing pressure, not the tool shape, that is primary variation for ablation control.

The polishing tool consists of a thin membrane which has a set of force actuators on its rear side. The membrane oscillates along a radius of the mirror, and is designed to be flexible enough to take up the shape of the mirror surface. The actuators modulate the pressure in real time in order to control the ablation rate during polishing.

There are several problems with this technique. Firstly, the validity of Preston's

Law has yet to be proved, despite several attempts by different institutions. This, combined with the fact that they do not incorporate a way of actually measuring the pressure on the mirror means that there is no way of really knowing what is happening to the mirror. Finally, as with the stressed lap polishing, calibration is performed off-line, in a static situation that has little direct relevance to the operation during polishing.

1.2.2.5 Pressured Rod Polishing

The French firm REOSC uses a pressured rod polisher, of which there seems to be a lack of detailed information, probably due to industrial secrecy. It is thought that they use a full size flexible lap with pressured rollers on its back. This is used to polish the mirror in its cell, which is actively distorted on its support system to control the ablation.

This process would have the advantages of avoiding high spatial frequency ripples, and if the support system has arrays load cells/actuators could measure, and therefore better control, the polishing pressure in real time. However, it seems likely that as with the stressed mirror polishing mentioned above, the mirror may suffer warping when the excessive stress is released after polishing.

1.2.2.6 Ion-Ablation Process

Eastman Kodak has produced a completely different approach to mirror figuring, by bombarding the work piece with a high energy ion beam, which strips glass from the mirror on the atomic scale.

The ion beam is much smaller than the workpiece, and is moved stepwise across the mirror. It avoids problems caused by the overlapping of the beam on different passes across the mirror by having a Gaussian footprint. The whole process works extremely well, with the only drawbacks being its relatively slow speed and high cost.

1.2.2.7 Summary

The current emphasis of the developments mentioned above is into the production of 8-10m diameter concave primary mirrors, with the production of large, highly aspheric, convex secondary mirrors being somewhat neglected world wide. This, combined with the technical drawbacks all the above processes possess, is the inspiration for the Optical Science Laboratory to develop the Active Lap. Although, this is being developed very much with the production of highly aspheric secondaries in mind, the technology should be equally applicable to the production of virtually any other figure.

1.3 Aim of the Active Lap Project

The aim of the Active Lap project as a whole is to develop a new computerised method for the production of large aspheric astronomical optics, and in particular to demonstrate the application of this method to making the GEMINI secondary mirrors.

To this end, the project entails making a $\frac{1}{3}$ scale model of the f/7 2.5m hyperbolic GEMINI secondary mirrors. (The initial specification was for an f/7 mirror, which has subsequently been changed to f/6. However the pre-polishing work had already been undertaken on the f/7 scale model and since it was considered that this would still be a viable demonstration of the technique, work continued on this.)

1.4 Summary of the Thesis

This thesis details the authors contribution to the Active Lap project. This is an ongoing research project at the Optical Science Laboratory, and as such this thesis does not encompass the entire R&D process.

- Chapter 1 gives the scientific and technological background, and outlines the scope of the work described in this thesis.

- Chapter 2 describes the design of the Active Lap, with overviews of the mechanical and electrical systems.
- Chapter 3 describes the workings of the data acquisition and control software which the author has written, including the user interface.
- Chapter 4 details the complete systems testing and calibration that was undertaken by the author.
- Chapter 5 describes the author's research into the software interpolation algorithm.
- Chapter 6 details the author's investigations into the real-time update routines, including experiments undertaken and the application of neural network technology.
- Chapter 7 concludes the thesis an evaluation of the performance of the Active Lap. It also suggests a strategy for its future use, and details further investigations that are required for its continued development.
- Appendix A gives details of the derivations of the non-trivial mathematical formulæ that are quoted in the thesis.
- Appendix B provides as background to chapter 6, an introduction to neural network technology.

Chapter 2

Design of the Active Lap

N.B The process of designing and building the Active Lap occurred before the author joined the project, and is therefore not within the scope of this thesis. This chapter describes the design of the Active Lap as essential background to the author's contribution to the project. Extensive details of the design process can be found in [16].

2.1 The Principle of Active Polishing

Active polishing means that during polishing the lap is continuously adjusted to control the ablation of the workpiece, in a predictable manner. Necessarily, this requires the lap to be able to measure, and react to, the relevant polishing parameters in real-time.

The physical nature of glass polishing is still poorly understood, but is thought to be a combination of several processes including mechanical wear, plastic flow and chemical attack. The best quantitative measure of the ablation is given by Preston's Law [4], which states that if A is the instantaneous ablation rate, P is the polishing pressure and V is the relative velocity, then

$$A = kPV^{\alpha} \tag{2.1}$$

where k and α are constants.

This, however, is an approximation, which has empirically been found to give inaccurate results, with the values of the constants in particular being ill defined [16].

Active polishing, therefore, only relies on the broad concept of Preston's Law that the ablation depends on the polishing pressure, the relative velocity between the lap and workpiece, and the integrated polishing time.

Active polishing aims to measure, in real-time, the pressure distribution across the mirror and the relative velocity distribution between the lap and the mirror. It will use this information to calculate the instantaneous ablation rate via an *ablation algorithm.* A feedback loop then modulates the pressure distribution to provide a means of controlling the instantaneous, and therefore the integrated, ablation.

At the end of a polishing run, the integrated ablation is compared to the actual ablation found via optical testing. The differences can then be used to update the ablation algorithm. The optical test will also provide an error map of the current surface profile of the mirror. Knowing this error map, the lap is able to maintain a required pressure distribution during subsequent polishing. This pressure distribution will be so defined that it will result in an integrated ablation that will remove these errors.

Active polishing will close the loop between the testing and the polishing of a workpiece. It aims to put the figuring of large optics onto firm scientific grounds, and remove the need for the conventional *craft* techniques that rely on the skill and judgement of the optician.

This will mean that the production of large highly asperic optics, which is thought to be an intractable problem using conventional techniques, will now be possible. Also, simpler optics will be able to be made more quickly, and will be correspondingly cheaper.

2.2 The Active Lap

The Active Lap is a full sized lap. Full sized laps have several advantages over sub diameter laps.

- They have a well proven natural smoothing action, contrary to a sub-diameter tool's tendency to produce high spatial frequency ripples.

- They have much faster ablation rates for a given pressure than sub-diameter laps, simply because the area in contact with the glass is larger.
- Much lower pressure can be used, which greatly reduces the possibility of producing print-through from the mirror support, or from a mirror's honeycone substrate.

During polishing, the aspheric profile of the mirror is mainly accommodated by the natural flexibility of the lap, which uses small, but significant strokes.

The lap is populated with two dimensional arrays of custom designed load cells [16] and linear stepper motor actuators. The load cells measure the force applied to the mirror. This is converted into pressure and interpolated to form the pressure distribution over the whole mirror. The actuators are able to increase, or decrease, the force applied the mirror, and therefore are able to modulate the pressure distribution.

The design of the 85cm prototype Active Lap is shown in figure 2.1.

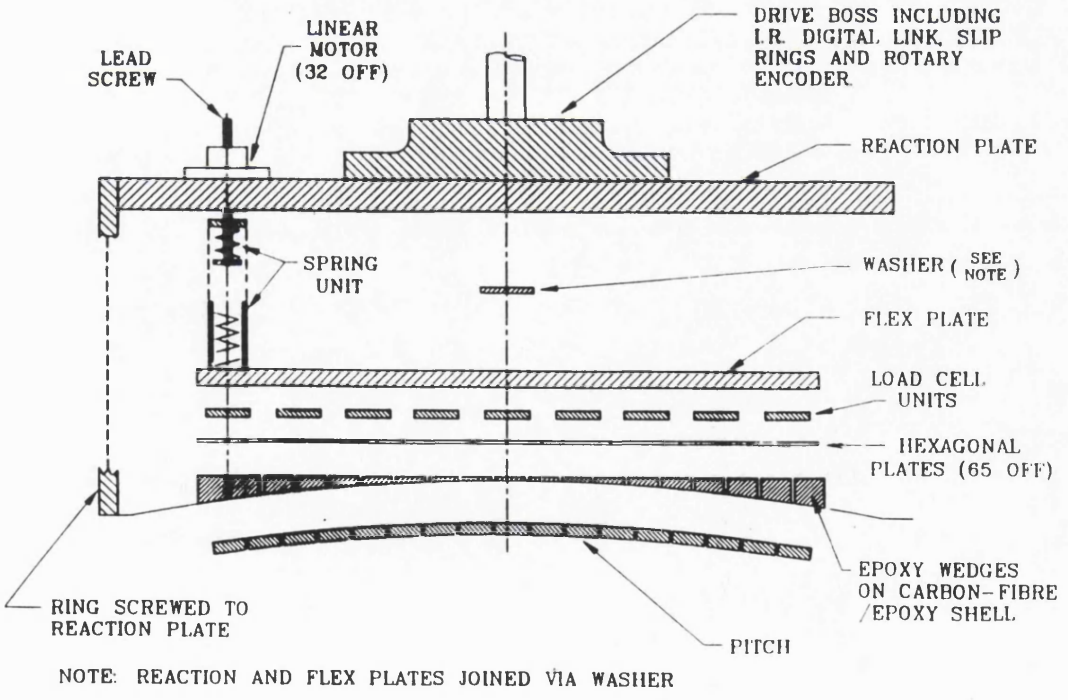


Figure 2.1: Exploded View of the Active Lap.

At the heart of the lap are two metal discs. The reaction plate is stiff and inflexible, and provides a reference against which the actuators can push. 32 actuators are mounted on the reaction plate, and are able influence the flex plate via a spring unit. The spring unit converts the linear motion of the actuator's lead screw into a force applied to the flex plate. The actuators are positioned in three concentric circles, each containing 8, 8 and 16 actuators respectively. It is a key point to note that the actuators do not bend the flex plate, but modulate the pressure on the workpiece.

Mounted on top of each actuator is a small microswitch. This is positioned so that when the actuator lead screw is moved as far as is allowable upwards, the switch is just closed.

The actuators are only able to move a certain distance in either direction before they stall. This distance, d, is known. To position a microswitch, its associated actuator is set to its null position, where it is exerting no net force onto the flex plate. The actuator lead screw is then moved upwards by the distance d, and the microswitch positioned so that it it just closed. [10] The actuator positions are then always measured in relation to the microswitch. e.g. When the actuators are reset, they are moved to the aforementioned null position by driving the lead screw upwards until it reaches the microswitch, then downwards the number of pulses equivalent to the distance d.

The reaction and flex plates are joined via a central washer, which acts as a spacer to keep the plates separate, and eliminates lateral movements of the plates relative to each other.

The flex plate is considerably thinner (6mm) than the reaction plate (15mm). It smooths the force functions exerted by each of the actuators, to a continuous function. Simultaneously, it has been carefully FE modelled to be able to filter out high spatial frequencies that may introduce ripples into the mirror surface. [16] The flex plate is also the medium that introduces cross-talk not only between the actuators and the load cells, but between the actuators themselves. This gives rise to the *coupling coefficients* that quantify the degree of cross-talk, which are mentioned in detail later in this thesis.

61 load cell attachments are mounted in a hexagonal pattern directly onto the flex plate. Of these, 22 are populated with strain gauges, with the remaining 39 left unpopulated, being referred to as "dummy" load cells. Each load cell is covered by a thin hexagonal plate. The hexagons are linked by silicon cement. This effectively seals the load cells away from any possible contamination, whilst the silicon cement being highly compliant, ensures that one load cell will not piston the next, and disturb it's response.

Beneath the hexagons is a grid of small epoxy wedges that match the hexagon's flat geometry to the curve of the carbon fibre reinforced epoxy shell. This shell is cast using the generated mirror profile as a mould, and is used as the mount for the pitch facets that do the polishing. As is consistent with normal polishing practices, the pattern of the pitch facets are off centred with respect to the lap, and therefore the load cell and the actuator geometries. This is a standard practice that helps to avoid the formation of concentric rings on the mirror surface.

The edge of the carbon fibre shell is attached to a metal ring that is screwed to the reaction plate. This serves two purposes. Firstly it means that lateral forces generated by friction are transmitted to the reaction plate, and are therefore *not* transmitted to the load cells. This is important because the load cell's design precludes their being subject to lateral forces. [16] Secondly, the ring completely encases the whole of the delicate part of the lap, which means it should be safer from accidental damage or contamination.

The drive boss on the top of the lap is attached to two polishing arms, the other end of which are attached to two off-centred cams. These rotate at slightly different speeds, driving the lap in a pseudo random path. A load cell is positioned in each of the polishing arms, which can be combined to measure the total force with which the lap is driven. This means that the total frictional force can be calculated. This is used as another parameter in the ablation algorithm, and gives the user a measure of the quantity, and the condition, of the polishing slurry that is present. As polishing progresses, the slurry becomes drier as water both evaporates and is driven out from under the lap. As this happens, the total frictional force increases, and can become a useful warning, as the lap can literally stick to the mirror if the slurry dries out too much.

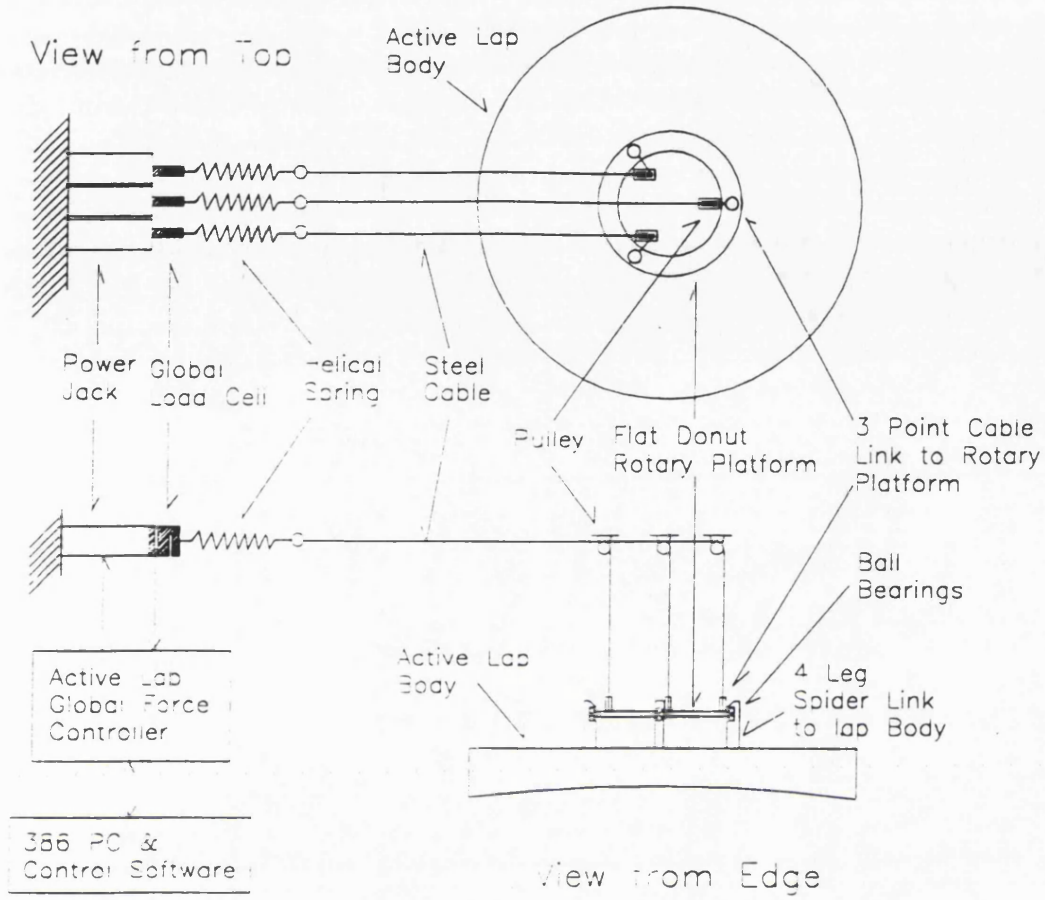


Figure 2.2: Global Force Actuators.

Because the lap continually rotates when polishing, the the power for the on-lap electronics comes in via slip rings in the central boss. There is also a bi-directional infra-red communications link that provides a serial data connection between the lap and the outside world.

The positions of the lap and the mirror are measured using four encoders. Two of these, which are mounted on the support structure around the polishing machine, have wires wrapped around their spindles, and then connected to the central boss. When the lap moves it pulls the wires. This then turns the encoders, which are able to measure how far the lap has moved. Two other rotary encoders measure the rotations of the lap and the mirror. These are positioned on the lap's central boss, and under the polishing machine on the mirror's turntable drive, respectively.

As shown in figure 2.2 there is a ring on the upper side of the lap that is constrained so that it does not rotate with the lap. To this are attached three *global force actuators.* These provide further control over the pressure distribution on the mirror. They can pull on the ring to vary the absolute overall polishing pressure, or to tilt the pressure distribution, effectively rocking the lap.

The lap, the global force actuators, and the encoders, are all controlled by the host 66MHz 486 IBM-PC. As well as providing the user interface, graphical maps, and data storage, this communicates via the infra-red link with an *i*960 micro computer that is mounted on the top of the lap. This controls the reading of the load cells and the actuator movements.

Above the epoxy wedges, the whole lap is plane parallel. This means that it is readily adaptable to be able to work on other mirror profiles. A new carbon fibre shell has to be cast to the correct shape, and the correct epoxy wedges milled, but once these are in place the rest of the lap remains as it is.

One problem with this design of the Active Lap is that the physical thicknesses of the hexagons, and of the load cells, are all slightly different. This is an inevitable consequence of their manufacture. However, this means that if, for example, one load cell/hexagon pair was slightly withdrawn compared to the adjacent ones, it will not be in contact with the epoxy wedges. This loss of internal contact will result in this load cell never recording a force. i.e. The lap will not be correctly reading the forces exerted on the mirror.

To attempt to solve this an *epoxy bag* was placed between the epoxy wedges and the hexagons. This consisted of two sheets of plastic that were sealed around their edges, and filled with epoxy glue. The amount of glue was carefully measured so that it would not spill out of the bag and damage the insides of the lap.

When the lap was reassembled, the idea was that the epoxy glue would flow whilst wet, and take up any small spaces between the wedges and the hexagons. This would mean that when it set, it would provide total internal contact. As described in section 4.2.1.1, this proved to be unsuccessful, and was replaced when the lap's design was modified.

A photograph of the Active Lap sitting on the mirror can be seen in figure 2.3.

2.3 Modifications to the Initial Design

As detailed in section 4.2.1.1 it was found that the lap as described above was not working satisfactorily. Forces on the pitch were not being propagated through to the load cells properly. It seemed likely that the epoxy bag was not successful in providing total internal contact between the hexagons and the epoxy wedges. It was also thought that the lap was not compliant enough to conform to the mismatch between the lap and the mirror, when the lap was offset.

The lap, therefore, needed to be modified to rectify these faults. The top half of the lap from the hexagons upwards was retained in its entirety, but everything below the hexagons was replaced.

The new lap is shown in figure 2.4.

There is a rubber sheet, approximately 5mm thick, directly below the hexagons. But as opposed to the epoxy bag which simply rested inside the lap, this rubber sheet is not only glued to the hexagons, it is also glued to the flat carbon fibre membrane which is directly beneath it. This means that internal contact is guaranteed. In

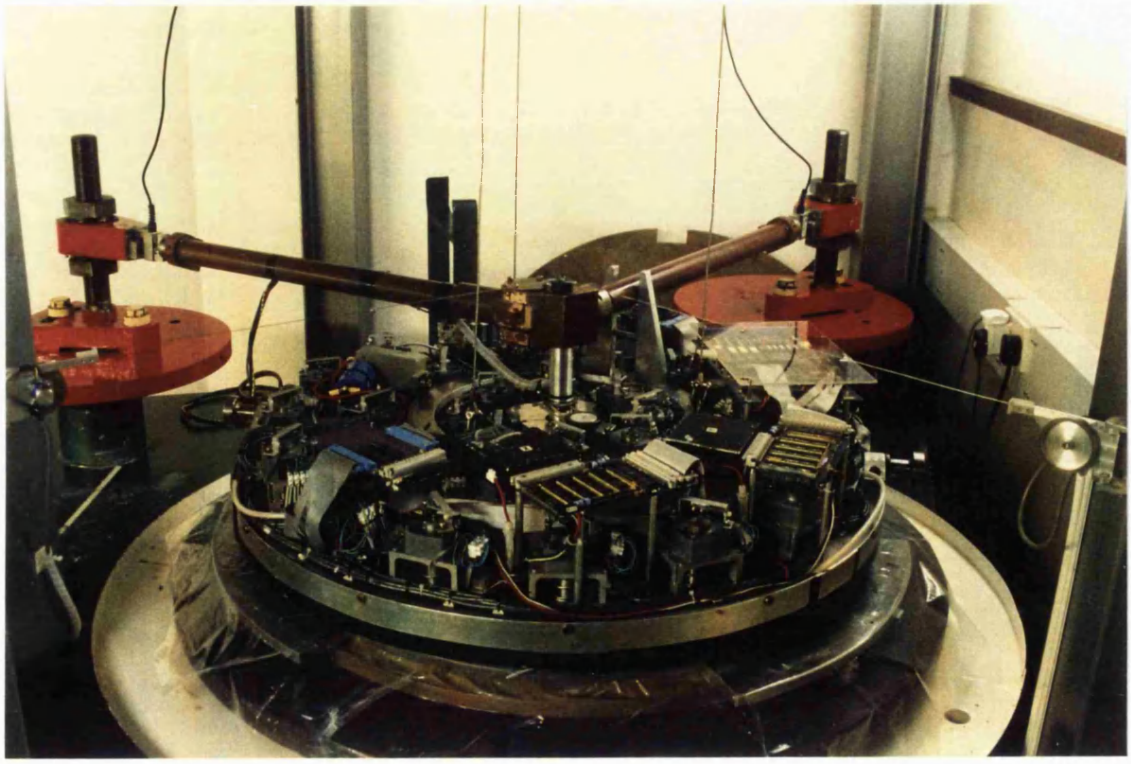


Figure 2.3: Photograph of the Active Lap.

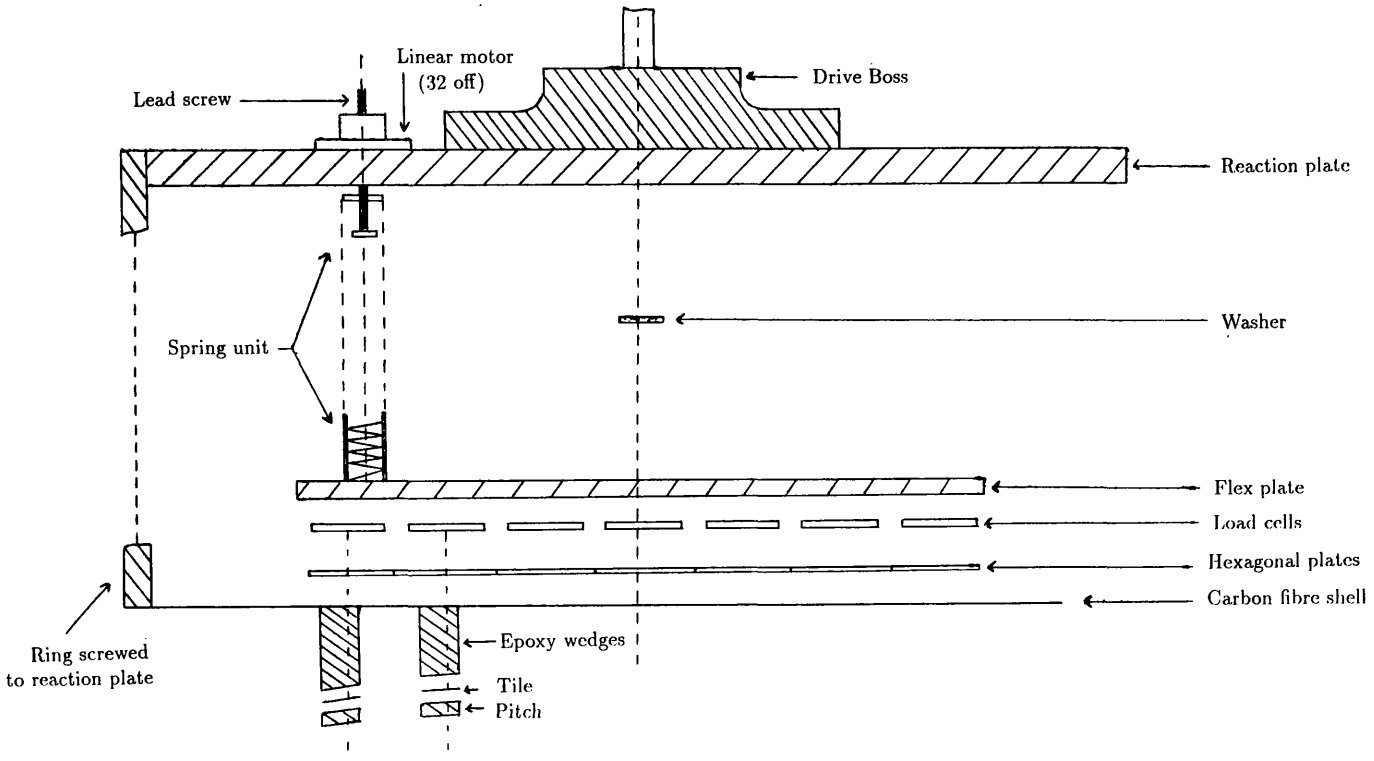


Figure 2.4: Exploded View of the Modified Active Lap.

addition, the rubber sheet is very much more compliant than the epoxy bag.

The carbon fibre membrane is much thinner than before, which means it will bend easier. Also, its flat geometry will be much more compliant than previously, since its former hyperbolic shape would necessarily have introduced a certain amount of rigidity into its structure.

Below the carbon fibre membrane are epoxy wedges, which take up the spacing between the flat membrane and the hyperbolic mirror. Onto the epoxy wedges are stuck pieces of tile, then the pitch facets. The tile is there simply because the pitch would not adhere to the epoxy wedges.

Another major difference with the new lap, is the distribution of the epoxy wedges and pitch facets. Before, these were not in vertical alignment with each other, or with the load cells. This means that the load cells would not be measuring the forces applied to the mirror directly. Instead, the forces would be redistributed as they were propagated upwards.

For the new lap the epoxy wedges/pitch facets were carefully arranged so that each

one was vertically beneath a load cell. This means that the load cells should have a much more direct measurement of the forces that are actually applied to the mirror.

However, this introduced a further complication. The load cells are arranged in a hexagonal pattern, with one of them being exactly at the centre of the lap. If the epoxy wedges/pitch facets were placed directly below each load cell, they would obviously be positioned on a regular grid with a pitch facet at the centre of the lap. However, this is a well known recipe for polishing concentric rings onto the mirror surface. The solution to this was to have the area of the pitch facet half that of the load cells. The epoxy wedge/pitch facets could then be offset from the centre of the lap by half a load cell (approx. 8cm), and still be completely within a load cell's footprint, thereby maintaining the direct one-to-one relationship. This offsetting caused three of the pitch facets to be repositioned so they would be partly overhanging the edge of the mirror when the lap was being pressed. This is clearly impractical, so these were removed and three load cells deactivated, leaving a total of 19 active load cells.

As demonstrated in section 4.2.1.1, this modified design for the Active Lap proved to be much better than the original. Forces from the mirror are being correctly propagated to the load cells, and the lap is compliant enough to detect the mismatch between the off-centred lap and the mirror correctly.

Chapter 3

Description of Active Lap Data Aquisition and Control Software

This chapter describes the software that has been written to run on the IBM compatible PC. The purposes of this software are threefold. [17]

- To acquire and store data from several sources, include data from the *i*960 processor that controls the electronics on the lap.
- To relate this data to the user in a suitable manner, to allow its' interpretation in terms of the actual polishing of the mirror.
- To provide an ergonomic interface between the user and the lap, to facilitate easy control of all the different software and hardware functions.

A schematic of the overall structure of the Active Lap data aquisition and control software is shown in figures 3.1 and 3.2.

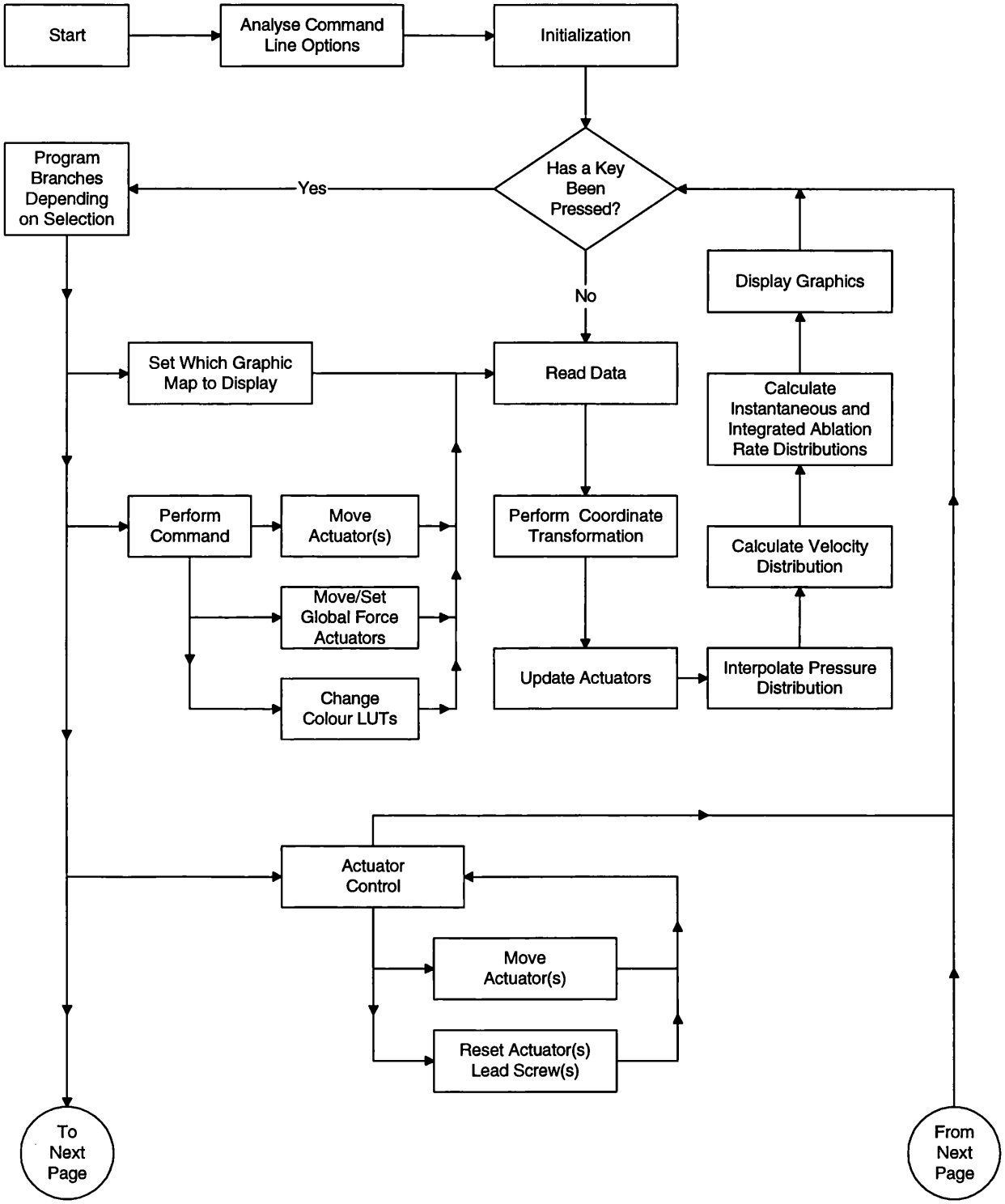


Figure 3.1: Schematic Diagram of Active Lap Software (A).

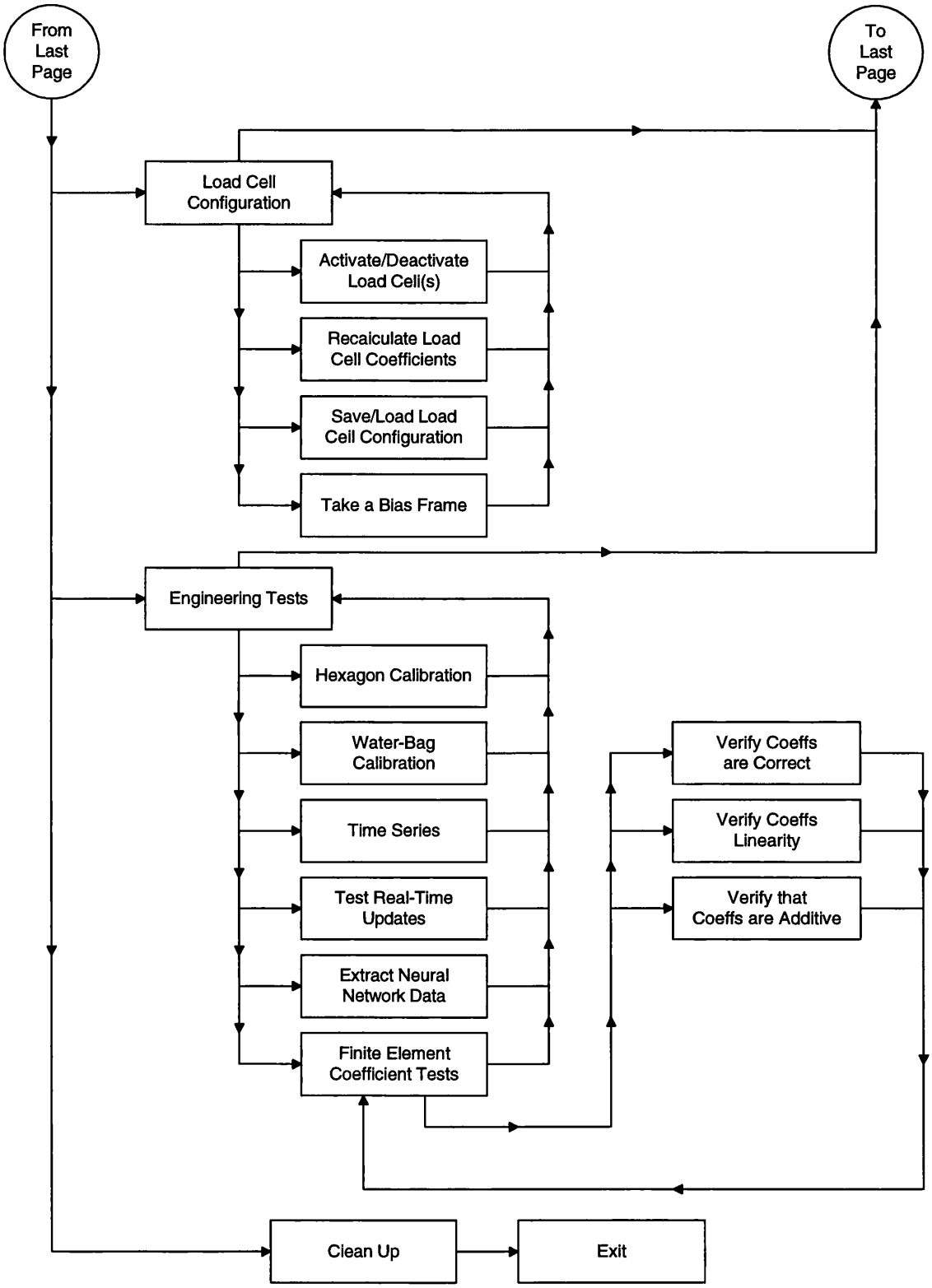


Figure 3.2: Schematic Diagram of Active Lap Software (B).

3.1 Notes on Programming Language Use and Optimizations

It was envisaged at the beginning of the project that the speed requirements of the software would stretch the PC to its limits. Therefore the programming language chosen needed to produce the most efficient code possible with respect to the cpu time used, whilst retaining its' portability and being easily maintainable.

Several alternative languages were considered, including BASIC, FORTRAN, pascal, C, C++ and assembly language.

The last of these was rejected because although it would produce the most efficient code, its complete lack of portability and difficult maintenance made it unsuitable for our purposes.

Of the others it was decided that C was the appropriate choice. This was because C is a relatively low level, general purpose language. It combines the advantages of control structure and portability found in the other high level languages, with the ability to deal with 'machine level' objects such as characters, addresses, pointers and bitfields. These can be manipulated by the arithmetical and logical operators implemented by the cpu hardware, making C a relatively efficient language to use. [11] [15] [24]

The other languages also have their strong points, e.g. FORTRAN's better facilities (i.e. simpler syntax) for manipulating multi-dimensional arrays and dynamic declaration of array size [24], but the most important underlying consideration was to produce the fastest executing code. C++ was rejected because its run time overheads would use up valuable cpu time, for which its object orientated facilities were judged be insufficient compensation.

In order to preserve these advantages of programming in C, the software was written to conform to the Starlink C Programming Standard. [24] This consists of a number of suggestions made with maintainability, portability and efficiency in mind.

Despite all the above considerations, it was found that the software was nearly always taking longer than required. (i.e. The control loop was running significantly slower than 10Hz.) Therefore, during periods of software development, there had to

be a continual process of optimizing the code with respect to the cpu time used.

In an effort to maximize the efficiency of the code, three principles in particular were implemented wherever possible.

- When appropriate, calculations were performed using pre-computed look-up-tables (LUTs). It involves much less cpu time to access the values in a memory location than to perform the calculations in real-time.

- When the above was not possible, calculations were performed using integer arithmetic, since this executes considerably quicker than floating point calculations. Loss of precision was minimized by scaling up all the terms in the calculation.

- The interpolation algorithm was manipulated so as to be expressed in the form of a simple series. e.g. If the interpolated pressure, P, at an arbitrary point i is a function of n load cell readings, L, then this function was rewritten to give

$$P_i = \sum_{j=1}^{n} C_{ij} L_j \tag{3.1}$$

  where C_{ij}, the interpolation coefficients, usually had a considerably more complex algebraic form than the original function. These coefficients were then only calculated once, during the software initialization, and stored in memory. This meant that during each control cycle the interpolation required for each point involved only the simple calculation shown in equation 3.1.

It was still found, however, that the most cpu intensive routines, namely interpolating the pressure between the load cells and writing the graphical display to the computer screen, were not executing as quickly as required. This was because although C is efficient relative to other high level languages, compilers are not yet sufficiently intelligent to produce the most efficient code possible. A significant decrease in execution time of these routines was achieved by rewriting and manually optimizing them in assembly language. This meant the loss of portability, but the increased speed of execution was a more important consideration. Also, since assembly language is much more difficult to maintain than C, a considerable effort was made to simplify and extensively comment the code.

3.2 User Interface

3.2.1 DOS vs Windows

The first consideration when designing the user interface was whether to run the software under DOS or under Windows. The latter would mean that advantage could be taken of the Graphical User Interface (GUI) available when using the Windows environment. This would mean that the user could use a standard, well-written interface without having to learn a new system. It would also have saved the programmer time in constructing a new interface from scratch. Unfortunately Windows imposes overheads on the system and will cause the software to run slower. [35] Whether this would be significant or not is almost impossible to predetermine, so it was therefore decided to run the software under DOS, although it may be appropriate to review this decision at a later date.

Note that the other advanced features of Windows such as multi-tasking and transferring or linking data between different applications were decided to be unimportant for our purposes.

3.2.2 Key based system vs GUI

It was a major requirement that the control software provided a well designed, ergonomic interface for the user. Two possibilities were considered.

- A key based system, where different functions are executed by pressing different keys on the computer keyboard. (e.g. Pressing <F1> displays the graphical pressure map, whilst <F7> accesses the Actuator Control menu.)
- A GUI where different functions are executed by selecting the appropriate item from a graphical menu.

Of these, it was decided that implementing either could meet our requirements, so the former was put into effect simply because it was considered that this would take less programming effort. This operates by using the function keys (<F1-10>) to select the Main Menu options, the numeric keys to select different parameters within the Main Menu, and the alphabet keys to enter user commands.

This proved initially to be very successful, while the number of functions available to the user was fairly small, but has became somewhat unwieldy as the number of functions has grown. Therefore it would be appropriate in the future to put the latter into effect, whereby the user can select options from different menus as required.

This selection should be carried out either by clicking the mouse on the desired option, or by pressing the key corresponding to a highlighted letter in the menu option. (e.g. The *P*ressure map menu option would have a highlighted letter *P*, so could be selected either by clicking the mouse on the option, or by pressing the <P> key on the keyboard.)

It it thought that this GUI design will be much more successful. It should combine much easier access to all the different functions available with the simplicity of using the mouse as an input device, whilst retaining the earlier option of making selections via keypresses.

3.3 Mode Selection

In order to allow the user either to record data about the current polishing run, or to review data from a previous polishing run, the software is designed to run with two separate modes of operation, namely *real-time data acquisition mode* and *demo mode.* The mode required was selected by applying the appropriate command line option when running the software.

- In *real-time data acquisition mode* the software reads in all data from the hardware, and then saves it to the hard disk for subsequent analysis. The graphical displays and calculations are therefore being done on data from the current polishing run.
- In *demo mode* the software reads in data stored on the hard disk during a previous polishing run. Hence the software will be showing a demonstration of work carried out previously.

3.4 Graphical Displays

3.4.1 General Properties

A suitable method for relating data to the user had to be devised, of which the simplest method was to display a table of numbers. This was useful to have on occasion, particularly when the lap was undergoing engineering tests, but for more general use was very difficult to interpret in terms of what was actually happening to the mirror.

It was therefore decided to have graphical representations of the data. This approach had proved to be extremely successful in previous OSL projects, particularly for the control system for the UCL Échelle Spectrograph (UCLES) which was designed and built by the OSL for the Anglo-Australian Observatory. [22]

There were two distinct types of data which were displayed in different ways.

- Values of a parameter that is distributed over the whole mirror (e.g. pressure, velocity) are represented by a graphics map showing the mirror as a circle on the screen which is filled with pixels of different colours. Each colour represents a different range of numerical values of the parameter at that point on the mirror. (e.g. red might represent a large value of the data, green an intermediate value and blue a low value.)

  The relationship between the colour levels and the data values are given by the colour look-up-tables (LUTs), described in section 3.4.1.2.

- Values of a single parameter (e.g. a load cell reading the force exerted by a global force actuator) are displayed as 'thermometer bars', where as the numerical value of the parameter increases, the level of the thermometer increases, and vice versa.

Several different displays were available for the user, the general properties common to all of which are detailed below.

3.4.1.1 Screen Layout

In order to provide easy access to many different pieces of information at once, the layout of the computer screen was carefully designed so as to avoid becoming cluttered and confusing. Photographic examples of the computer screen can be seen in chapter 4.

There were six separate components to each screen, namely

- Title – This showed which display was active and the appropriate units for the colour LUT.
- Map – This was the graphical display of the parameter currently selected. (e.g. pressure map, velocity map, etc.)
- Colour LUT – This displayed the range of numerical values corresponding to each colour level on the display map. The units were displayed in the title.
- Global Force Actuators – This showed the force read by the load cells on the global force actuators. The values read, along with their sum (i.e. the total force the lap is lifted by) were displayed as 'thermometer bars'.
- Total Frictional Force – This was very similar to the global force actuator display, but showed the force read by the load cells in the polishing arms, also displayed as 'thermometer bars'.
- Information – This either showed the mean time per frame (see section 3.9.3), or, when the user entered a command, this echoed the keyboard and gave error messages when applicable.

3.4.1.2 Colour LUTs

With the PC VGA display that was being used for this project, a maximum of 256 different colours levels are available to be displayed on the screen at any one time. Of these, four have to be permanently assigned to single colours, while the others ware free to be assigned with different colours when the LUT is altered.

These four levels are assigned to the following colours

- Black – Screen background colour.
- White – Colour for title and information.
- Red – Used for 'thermometer' displays and error messages.
- Yellow – Used for 'thermometer' display.

This leaves 252 available levels, which are divided equally between each of the four basic types of displays (i.e. pressure, velocity, instantaneous and integrated ablation rates), giving 63 colour levels for each display.

It would have been possible to allow each display to use all 252 levels, but this would have been more complex to accomplish, thereby increasing the cpu time used. In addition, it was found empirically that there was very little difference discernable to the user between 252 and 63 colour levels.

The 63 levels are initially set to cover 63 ranges from the minimum to the maximum possible value of the parameter. It is, however, desirable for the user to be able to adjust these ranges so as to be able to increase (or decrease) the contrast at a particular place. It was therefore possible to change the lower and upper numerical values of the LUT to accomplish this.

Three different LUTs are available for use. This was so that the main user, who has colour-impaired vision, can choose which one best suits his needs in different situations.

The three LUTs available are

- RGB LUT – This is a three coloured LUT, where high values are shown in red, with decreasing values graduating through green until low values are displayed in blue.
- Rainbow LUT – This is a multicoloured LUT with seven different gradations of colour covering the complete range of values.
- Grey-Scale LUT – This is a single colour LUT, where high values are displayed in brilliant white and decreasing values going through shades of grey until the lowest value is displayed in black.

3.4.1.3 Pixel size

The size of the pixels used for all the graphical displays are determined by the size of the pixels required for the continuous pressure map. This is because one of the goals of the software is to successfully predict the integrated ablation rate at each pixel. This is a function of the pressure and the relative velocity between the lap and the mirror at each pixel.

The velocity map is calculated from the relative positions of the lap and mirror, hence it can be calculated on an extremely fine pixel grid, limited only by the precision of the computer.

Therefore, since the spatial sampling frequency of the pressure distribution is orders of magnitudes larger than this (the distance between the centres of the load cells is 17.3 cm), it is this parameter that will determine the size of pixels used for all other displays. It would have been possible to have different pixel sizes for different displays, but this would have introduced an unnecessary increase to the amount of cpu time used.

Two factors affect the size of the pixels used for the pressure map. These are

- The physical spacing of the active load cells.
- The ease the user has interpreting the display.

Empirically, it was found that the pressure distribution was difficult for the user to interpret when the pixel size was large. Equally, if the pixel size is considerably smaller than the spacing between the load cells, then the interpolated pressure will not be physical meaningful.

A compromise was found by displaying the graphical maps as 30×35 pixels on the computer screen, which corresponds to a physical pixel size of 2.37×2.76cm on the mirror. The size of the pixels was different in the X and Y directions because in 256 colour mode the VGA screen pixels are rectangular, with an aspect ratio of 6:7. Therefore, in order to achieve a circular display on the screen, we require what is effectively an elliptical distribution of pixels.

It would have been possible to perform calculations on a circular, rather than an elliptical grid of pixels, but this would then have to be compensated for every time

the pressure map was displayed, which would use up valuable cpu time.

Although the physical size of the pixels is much smaller than the spacing between the load cells it is thought to still have some physical meaning. This is due to the fact that the Active Lap was designed so that the flex plate filters out high spacial frequencies, resulting in the fact that the pressure between the load cells will vary fairly uniformly. Empirically, it was discovered that a sampling rate less frequent than 30×35 pixels made it very difficult for the user to interpret what was happening.

3.4.2 Details of each Graphical Display

The following section describes the salient details for each of the graphical displays available to the user.

3.4.2.1 Continuous Pressure Map

This display shows the continuous pressure distribution at each point where the lap is on the mirror, in mirror coordinates.

The force exerted on the flex plate at each pixel is interpolated from the load cell readings as detailed in chapter 5. The pressure to be displayed at each pixel is then found by dividing the interpolated force by the area of a pixel, and displayed using the coordinate transformations detailed in appendix A.

3.4.2.2 Relative Velocity Map

This shows the relative velocity between the lap and the mirror, in mirror coordinates.

For each pixel in the screen display of the mirror, this is found by performing the following calculation.

At time t_1, the section of the lap immediately above point P on the mirror is at coordinates (L_x^1, L_y^1) in the rest frame of the mirror. The last time data was acquired, at time t_0, this section of the lap was known to be at coordinates (L_x^0, L_y^0) in the rest frame of the mirror.

The distance this section has moved since the last frame is then found by comparing its present position with its position last frame, and *assuming* it has travelled in a straight line. (This assumption is likely to be valid since when acquiring data at 10Hz, the average distance moved by any section of the lap is of the order of only 1cm.)

If the time interval between the frames is Δt, where $\Delta t = t_1 - t_0$, then the relative velocity between the lap and the mirror at point P on the mirror, at time t_1 is given by

$$V_{t_1}^P = \frac{\left\{(L_x^1 - L_x^0)^2 + \left(L_y^1 - L_y^0\right)^2\right\}^{\frac{1}{2}}}{\Delta t} \tag{3.2}$$

Hence, the relative velocity displayed at any point on the mirror, is actually the average velocity of the section of the lap immediately above that point, during the last time interval.

However, since the values of $(L_x^1 - L_x^0)$ and $(L_y^1 - L_y^0)$ will be small, it is very likely that the errors in L_x^0, L_y^0, etc. will be dominant, giving enormous errors in the velocity values calculated. Therefore, after calculating $V_{t_1}^P$ as above, it is then modified to include the overall trend of the previous few calculations. This is achieved by performing a least squares fit to a straight line, and using this line to give the velocity at t_1.

3.4.2.3 Ablation Rates

For each pixel in the screen display of the mirror, the instantaneous and integrated ablation rates are calculated from the pressure and velocity, and displayed. The exact form of the algorithm to be used is discussed in [16], and will be the focus of research to be carried out in the very near future.

3.4.2.4 Composite Display

This is a composite display showing all four of the above maps on one screen. This does not display the global force actuator or the total frictional force readings, as there is not enough room to have these on the screen *and* have four graphical displays of a useable size.

3.4.2.5 Spot Pressure Map

Another pressure map is available which shows the pressure read by the load cells as 'spots' of pressure directly below each load cell. No interpolation is performed, with zero pressure being displayed between the spots.

3.5 Actuator Control

The user needs to be able to control the actuators in one of two ways. Firstly he needs to be able to manually move an actuator or actuators, and secondly he needs to be able to reset the actuator lead screws when required.

3.5.1 Manually Moving the Actuators

Two methods of providing the user with direct control of the actuators on the lap have been implemented. Firstly by having a separate menu option from where the user could select the appropriate actions to be taken, and secondly by entering a command whilst viewing one of the graphical displays. The latter was added so that the user could easily see the effect on e.g. the pressure map, of moving a set of actuators. When this was initially implemented as a menu system, the user could not see any changes to the pressure map as the actuators were being moved.

To allow the user to crudely, but simply, control the pressure distribution across the mirror it is possible to move the actuators in three different configurations. In each case the user specifies which configuration is required and the number of pulses to send to each selected actuator.

The three different configurations are

- Single actuator – Move one individual actuator.
- Opposite pair – Move a pair of radially opposite actuators.
- Ring – Move a ring of actuators. This is possible because the actuators are positioned in three concentric rings.

3.5.2 Actuator Resetting

This option allows the user to either reset all the actuator lead screws, or a single specified actuator lead screw, to its zeroed position. After the resetting, it displays the number of pulses the actuator(s) had actually moved from their zero position, along with the net number of pulses they had been sent. This gave the user information about how many pulses (if any) the actuator had lost.

3.6 Global Force Actuator Control

Once again, in order to allow the user to have a clear view of its effect on the mirror whilst polishing, control of the global force actuators is achieved via a user command entered when viewing a graphical display, exactly the same as the actuator control above. The global force actuators can be set to two different modes of operation.

3.6.1 Static mode

To effectively reduce the weight of the lap, and hence allow the polishing of the mirror to proceed at a slower rate, the global force actuators are set in static mode, where each global force actuator keeps lifting the lap with a constant force. This force is set, and can be modified, by the user.

3.6.2 Dynamic mode

The dynamic mode is an attempt to try to counteract the natural turning down of the edge of the mirror when part of the lap is overhanging the mirror.

Again, by entering a command whilst viewing one of the graphical displays, the user sets the total force to be lifted by the global force actuators when the lap in central on the mirror. When the lap is displaced from this position, the global force actuators automatically lift up the overhanging edge of the lap by an amount F, where $F = \alpha W_O$. W_O is the weight of the lap which is overhanging the mirror, and α is a parameter than can be altered by the user to either accentuate or reduce this effect. By default $\alpha = 1$.

If the surface area of the lap overhanging the mirror is A_O, the total surface area of the mirror is A_T and the total weight of the lap is W_T, then F is given by

$$F = \alpha \left\{ \frac{A_O}{A_T} \right\} W_T \tag{3.3}$$

If the centre of the lap is displaced from the centre of the mirror by a distance D, and the radius of the lap is r, then A_O can be found from the following equation, (See Appendix A for its derivation.)

$$A_O = r^2 \left\{ \pi - 2 \arccos\left(\frac{D}{2r}\right) \right\} + D \left\{ r^2 - \frac{D^2}{4} \right\}^{\frac{1}{2}} \tag{3.4}$$

3.7 Load Cell Configuration

A separate menu option to change the load cell configuration was available to the user. It was considered whether this should done via a command entered whilst viewing a graphical display but this was rejected since altering the load cell configuration would render the previous data useless.

Four options were available to the user in the load cell configuration menu.

- An individual load cell could be activated or deactivated. An active load cell would be used in the pressure map interpolation and for calculating the real-time actuator updates, whilst a deactivated load cell would simply have its data discarded. This option was specified to allow the user to experiment with different configurations, or to discard data from a faulty load cell.

- Once the configuration of load cells had been altered as above, the interpolation coefficients require recalculation. The second menu option accomplishes this.

- The third option is to either save the current configuration and interpolation coefficients to a file, or to read in a previously saved configuration and coefficients.

- Take a bias frame. A bias frame averages a specified number of frames and uses the information to calculate the offsets required for the load cell calibration. (See chapter 5.) Before polishing it is therefore important firstly to take a bias

frame, and secondly to do it under the correct conditions. Namely, ensure all the actuator lead screws have been reset, the lap is positioned centrally on the mirror and the lap has been in place long enough to be pressed.

3.8 Engineering Tests

This menu selection gives the user access to a number of routines that are used for testing and calibration of the lap. These are

- Hexagon Calibration. This allows the user to calibrate the load cells by placing weights directly onto the hexagons. (See chapter 5.)
- "Water-bag Calibration. This allows the user to calibrate the load cells using the "water-bag". (See chapter 5.)
- Time Series option. This takes and stores a specified number of frames, with a specified time interval between each frame. This was useful for recording the pressing of the lap.
- Test the real-time updates. This slowly performs the real-time updates, whilst giving the user information about the internal workings of the algorithms being used.
- Extract neural network training data. This extracts from a saved data file the pertinent information required to train the neural network models used for the real–time updates, converting and formatting the data as appropriate.
- Finite element coeff tests. This option allows the user to control the experiments performed to verify that the finite element analysis coefficients can be used for the real–time updates. (See chapter 6.)

3.9 Data Manipulation

Within the Active Lap software, data is acquired and is manipulated from a number of different sources. The different ways this is implemented is described below.

3.9.1 Data Transfer between Lap and PC

Data transfer between the lap and the PC occurred via an RS232 serial link running at 38400 baud. The transfer is accomplished simply by loading the appropriate array with the desired variables, and calling one of several functions to perform different tasks.

e.g. Receiving the 88 bridge readings on the PC from the load cells on the lap is performed by calling the function `READ_LC()`. The data is then available in an array called `darray[]`. This consists of some discardable junk data which was sometimes present in the communications data buffer, followed by a '*' which signifies that the next 88 characters are the data values. [13]

3.9.2 Using the Analogue to Digital Conversion Cards

In order to acquire values of analogue voltages from the encoders, the global force actuator load cells, and the polishing arm load cells, the control software was required to read data from an Amplicon Liveline PC27 data acquisition card, which is connected to the PCs ISA expansion bus. This provides 16 channels of 12 bit high speed analogue to digital conversion.

The voltages were read from the ADC card using standard programming techniques, and were then converted into physical units using the appropriate calibration data, as detailed in chapter 5.

In order to control the force setting of the global force actuators, the appropriate voltage is required to be sent via an Amplicon Liveline PC24 card. This is a four channel, 12 bit digital to analogue voltage output converter card, which again is connected to the PCs ISA expansion bus.

The forces required were converted into voltages using the appropriate calibration data as detailed in chapter 5, and were then sent to the DAC card once again using standard programming techniques.

3.9.3 Timing of frames

It was important to have an accurate measure of the time between successive frames, since this is used when calculating both the relative velocity between the lap and the mirror, and the instantaneous ablation rate. Since the Amplicon Liveline PC27 analogue to digital conversion card also incorporates three 16 bit, 4MHz counter/timers, it was a straightforward matter to use these to measure the time between frames.

At 4MHz it takes 0.016 seconds to count 2^{16} pulses, which is the maximum number possible on one 16 bit counter. If two counters are cascaded, then it would take about 18 minutes to count the 2^{32} pulses. This is likely to be far longer than it it will ever be necessary to time, but it was considered to be advisable to additionally cascade the third counter. It would take an enormous $2\frac{1}{4}$ years to count the 2^{48} pulses at 4MHz!

During each frame, the counters have to be stopped, read, initialized and restarted. This must take a finite amount of time, which will result in a systematic error in the readings. In order to allow for this error, a large number of frames (namely $10,000$) were timed both by the counters and with an accurate stopwatch. This was repeated several times, and it was found that the counter always read between 16.72 ms and 16.98 ms less per frame than timed on the stopwatch. Therefore, since the calculations that use the time between successive frames are performed using integer arithmetic (see section 3.1) the required resolution for the timing is ± 1 ms. Hence, when reading the time between successive frames an offset of 17 ms is always added.

3.9.4 Saving of data for Later Analysis

It was necessary for all data acquired during a polishing run to be saved for later analyses. (e.g. to be used to predict the polishing parameter for a subsequent polishing run, or to determine the ablation rate algorithm, etc.) Hence, every frame, the load cell readings, global force actuator readings, polishing arm load cell readings, temperature probe readings etc. were all written to a data file.

It is also possible to add comments to the data file, either by the software during a polishing run, or manually after a polishing run.

During a single polishing run these data files could become very large (up to 20MB). In order to reduce this size, and also to minimize the amount of time taken to write the data each frame, the data was written in the most compact form possible. This obviously involved reducing the number of white spaces as much as possible, but also, since most of the data values are 8 bit numbers, they can be saved as a single character, rather than 3 digits + one white space to separate them.

When running in *demo* mode, all data is read from these saved data files rather than from the hardware.

If the first character of the input line shows this line to be a comment, then the rest of the line is discarded, otherwise the characters on that line are inputted to the software as data values. If the end-of-file marker is encountered, then the software triggers a warning bell, and starts reading data from the beginning of the file.

Chapter 4

Calibration and Testing of the Active Lap

4.1 Calibration of the Active Lap

Various subsections of the Active Lap required calibration. These were the units that outputted a voltage proportional to the parameter they were measuring. Calibration provided a direct relationship between the voltage and the physical units of the measured parameter.

4.1.1 Load Cells

In order to calibrate the load cells, two methods were used. Firstly by placing various weights directly onto the hexagonal plates that cover the load cells and secondly by utilizing a "water-bag" when the lap is assembled.

4.1.1.1 Calibration Directly on the Hexagons

This was carried out by placing different known weights directly onto the hexagonal plates that cover the load cells, when the lap was upside-down before final assembly. For each load cell, the weights were successively increased from the minimum weight to the maximum, then decreased back to the minimum. This was done in order to account for any hysteresis effects. The weights used were in the range 0–4kg, which

corresponds to the range over which the load cells are required to operate during polishing. [16] This was then repeated twice more, and a graph of output from load cells in ADU against weight in grammes was plotted for each load cell. A weighted least squares fit to a straight line then gave the results shown in table 4.1

Load Cell	Gradient	Offset	Load Cell	Gradient	Offset
1	0.026773	51.71623	12	0.025938	68.29318
2	0.028506	60.29636	13	0.026460	49.82160
3	0.024833	48.27906	14	0.027517	64.86839
4	0.029337	60.27720	15	0.027486	55.11242
5	0.026530	68.88402	16	0.026887	61.85991
6	0.029561	67.89999	17	0.025906	53.59303
7	0.025930	55.63203	18	0.027941	63.46406
8	0.025755	55.53122	19	0.030240	53.87537
9	0.028950	47.54465	20	0.025328	67.93731
10	0.026926	77.72169	21	0.026858	75.54615
11	0.029165	63.61860	22	0.026631	64.25896

Table 4.1: Load Cell Calibration Data.

It was found, however, that the offsets varied over time, although the gradients remained constant. This was probably due to the gradual warming of the lap. Heat is produced not only from the electronics but a great deal is generated from the friction between the lap and the mirror.

To counter this, a *bias frame* was taken before a polishing run, when the lap had pressed and the electronics had been allowed to warm up, and again after the polishing run. Since the lap has been pressed, equal forces must be being applied to the mirror by each pitch facet. It can therefore be assumed that all the load cells should be measuring an equal force. The offsets were then adjusted by the bias levels so that each load cell was measuring the same fraction of the total weight of the lap. The weight lifted by the global force actuators also had to be taken into account.

If W_L is the total weight of the lap, the global actuators are lifting with a combined force of F_{GFA} and there are 61 dummy and populated load cells, then the bias levels

are adjusted so that each load cell is reading

$$\frac{W_L - F_{GFA}}{61}$$

4.1.1.2 Calibration with a Water-Bag

In order to try to calibrate the lap in its *natural* state, ie. when all the load cells are experiencing the same force, a "water-bag" was constructed. This consisted of two circular polythene sheets, slightly larger than the lap, separated and sealed by a thin annular piece of wood, and filled with water. The idea was that when the lap was allowed to rest on the water-bag which in turn was on the mirror, because the water-bag would ensure hydrostatic equilibrium, all the load cells would read the same force. By lifting up with the global force actuators by different amounts, the load cells could be calibrated.

Since calibration by placing weights directly onto the hexagonal plates shows that the load cell responses are very linear, it was expected that the same would be seen here. However, there were some anomalous results obtained with the water-bag technique.

- Some of the load cells had highly non-linear responses, and exhibited significant hysteresis.
- Some of the responses, particularly from load cells at the edge of the lap, were much less sensitive, or were completely unresponsive, when compared with the previous calibration. There were also large variations in response from unit to unit.
- In particular, an error analysis showed that load cells 1 and 2, being at one edge of the lap, had very low responses with large errors. The errors decreased towards load cells 21 and 22, which are at the opposite end of the lap.

These anomalies result from two problems with the water-bag technique.

- Due to errors in the global force actuator servo mechanisms, there are always small differences when they are all set to the same force. This results in the lap tilting. This effect is small, and is usually negligible. However, when using

the water-bag it means that one side of the lap (in this case where load cells 21 and 22 are situated) is tilted down and has a more consistent support, whilst the other side of the lap (numbers 1 and 2) is tilted upwards and has an irregular support, which introduces large errors.

- The polythene sheets have there own spring nature, which is evident at the edge of the water-bag. This means that the load cells at the edge of the lap have a mechanical support which is not present for the load cells at the centre. This obviously means that the water-bag calibrations can never be reliable since it is taken with different conditions for different load cells.

4.1.1.3 Summary

Of the two calibration techniques tried, the water-bag calibration is suspect, so necessarily the calibration must be carried out by placing weights directly onto the hexagons. Using the latter technique is not entirely satisfactory since it means that the load cells are calibrated in a artificial situation that is not readily applicable to actual polishing. (ie. The lap is upside down, and the epoxy wedges/carbon-fibre shell/pitch facets are missing.) However, it is not absolutely necessary for the calibration to be completely accurate. This is because any errors in the calibration will be propagated through to the integrated ablation map. The accuracy of this will be measured by comparing it to the optical tests of the mirror, and any discrepancies will cause modifications to the ablation rate algorithm. Hence, errors in the calibration will result in terms in the ablation rate algorithm not being strictly physically correct, but this is not vital since the algorithm will be correct for this Active Lap.

4.1.2 Global Force Actuators

To calibrate the global force actuators, the load cells they contained were removed from the apparatus. They were then hung vertically with known weights suspended from them. The voltage generated by the load cells were then measured when different weights were attached. The weights were added by starting with the smallest one, and going upwards. When the largest weight had been attached, the weights were then decremented down to the smallest one again, to investigate whether any

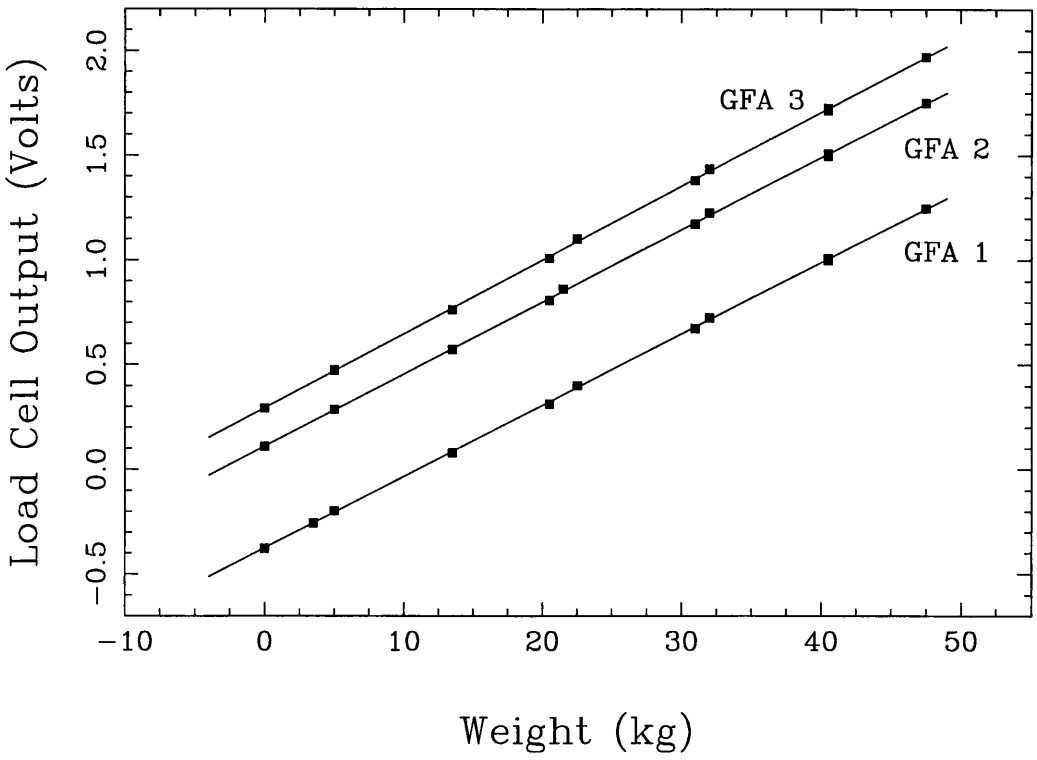


Figure 4.1: Global Force Actuator Calibration Curves.

hysteresis was present. The process was then repeated three times.

Figure 4.1 shows the calibration curves obtained from these experiments. A weighted least squares fit to a straight line then gave the results shown in table 4.2.

These results are therefore used to find the forces in kg which correspond to the voltages measured for each global force actuator load cell. These forces are then converted into Newtons in the usual way.

Global Force Actuator	Gradient	Offset
X	$3.408(\pm 0.004) \times 10^{-2}$	$-3.75(\pm 0.01) \times 10^{-1}$
Y	$3.446(\pm 0.005) \times 10^{-2}$	$1.11(\pm 0.02) \times 10^{-1}$
Z	$3.527(\pm 0.004) \times 10^{-2}$	$2.94(\pm 0.01) \times 10^{-1}$

Table 4.2: Global Force Actuator Calibration Data.

4.1.3 Encoders

4.1.3.1 Position Encoders

The calibration of the two position encoders is required in order to convert from a change in the ADU measured to a physical distance travelled.

To do this, each encoder was moved an amount measured by a ruler, and the corresponding change in ADU recorded. This was repeated several times. Figure 4.2 shows the calibration curves obtained. A weighted least squares fit to a straight line gave the results shown in table 4.3.

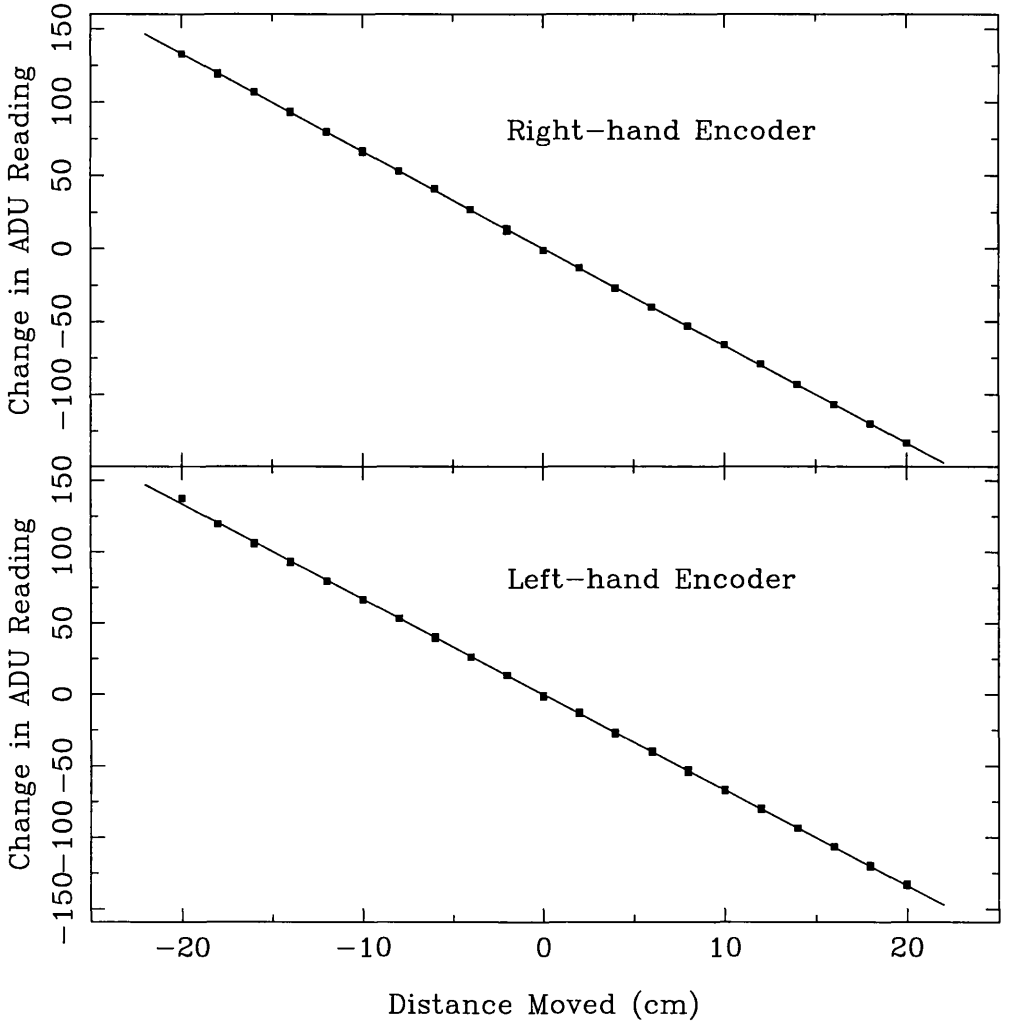


Figure 4.2: Position Encoder Calibration Curves.

Encoder	Gradient
Left	-6.683117
Right	-6.656167

Table 4.3: Position Encoder Calibration Data.

Each time the lap was removed then replaced onto the mirror (normally done so the figure of the mirror could be tested) the position encoders would be set to a slightly different position. To allow for this, each time the software was run it recorded the encoder readings when the lap was centralised on the mirror, and used this as a zero point.

4.1.3.2 Rotary Encoders

The two rotary encoders do not require calibration as such. The Active Lap control software uses LUTs for calculations involving these encoder readings. These LUTs require the readings in the range 0–255 ADU to correspond to angles of 0–359°. The voltages (and hence the ADU values) of the rotary encoders when they complete one full turn and go from 359° to 0° are set to achieve this, using trimming potentiometers.

4.2 Testing of the Active Lap

Before polishing the mirror, two subsections of the lap's behaviour needed to be tested to ensure they were operating correctly: namely the load cells and the actuators.

4.2.1 Testing the Load Cells

Three methods were used to test the behaviour of the load cells in particular, and the integrity of the lap in general. These both examined the pressure distribution with various conditions of static support, which are in fact time slices of the real-time pressure distribution during polishing.

4.2.1.1 Offsetting the Lap

The first method was to offset the lap from the centre of the mirror, and examine the pressure map. The form of the map should correspond to the mismatch between the mirror and lap hyperbolic profiles, and should result in a series of high and low pressure areas as shown in figure 4.3. Low pressure should be seen at areas with a relatively large mismatch, and high pressure areas where the lap and mirror remain in good contact.

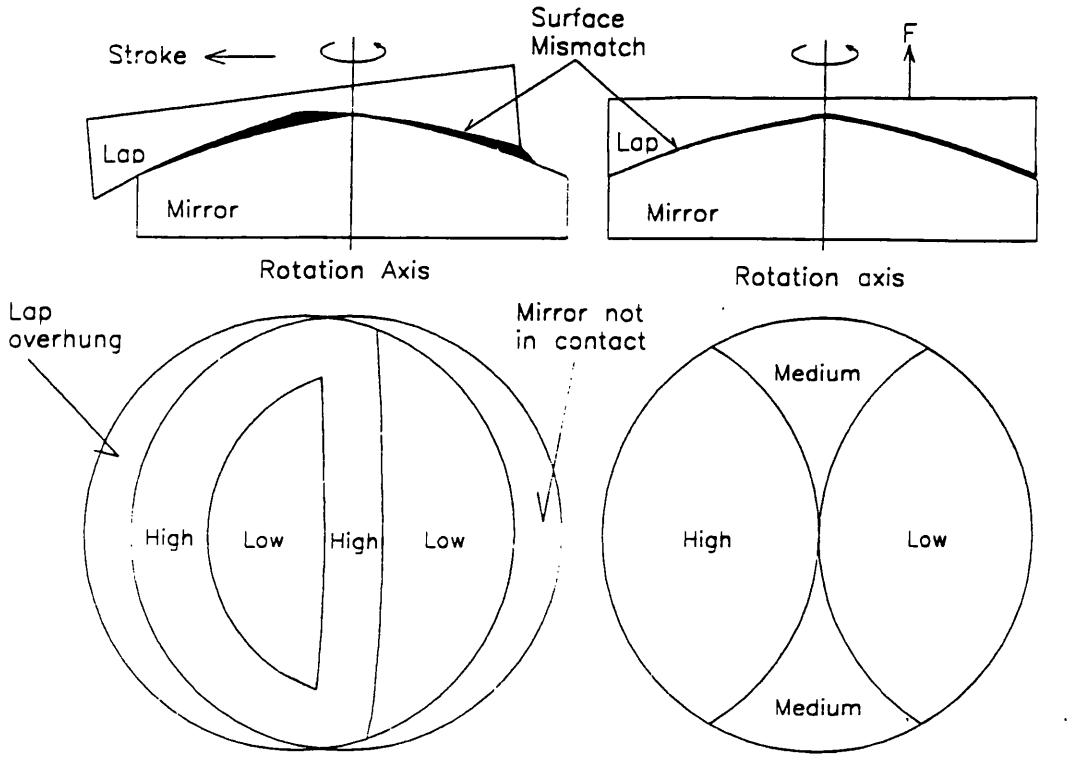


Figure 4.3: Expected Pressure Maps.

This experiment uncovered a problem with the lap. There was very poor agreement between the observed and the expected pressure distributions. Since the earlier calibration experiments had shown that the load cells did respond correctly when changing forces were applied to them, the source of this problem had to be that the forces were not being propagated through the lap to the load cells correctly. There were two possible causes of this, both of which probably contributed to the problem.

- The epoxy bag in the centre of the lap was not in complete contact with the hexagonal plates. This could arise in different ways:

- The epoxy bag was underfilled (or overfilled) when it was made, or perhaps given insufficient time to set.
- When the lap is displaced the different support conditions cause small movements inside the lap. These could easily result in the epoxy bag losing contact with the load cells.
- When the lap is lifted by the global force actuators the reaction plate bends, and internal contact is lost.
- Continual handling, vibrations, etc. cause the bag to move slightly, again causing internal contact to be lost.

- The lap was not compliant enough conform to the mismatch between it and the mirror, and would therefore loose contact. The rigidity of the lap is thought to have two causes:
  - When it is set, the epoxy bag is completely inflexible.
  - Due to its geometry, the hyperbolic shape of the carbon fibre shell gives it rigidity. In particular, it may be unable to bend in the appropriate manner to conform to the mismatch.

This lead to the redesigning and rebuilding of part of the lap to increase its compliancy and aid the propagation of the forces experienced by the pitch facets to the load cells. (See section 2.3.)

In order to quantitively evaluate whether the new design behaved better than the old one the following experiment was carried out.

Two error measurements are defined: the RMS load cell error as the root-mean-square difference between each load cell reading and the mean of the load cell readings; and the P-V error as the difference between the maximum and minimum load cell readings. When the lap has been pressed, these should both be zero (+ noise) and as the lap is displaced they should both increase.

The lap was displaced from the centre of the mirror in discrete steps, and the above errors calculated at each position. This was then repeated starting from a large displacement to allow for any hysteresis effects. Once the lap is displaced, the positions of the load cells relative to the mirror are no longer symmetrical. To allow

for this, four reading were taken at each displacement for different rotations of the lap, and the readings averaged.

The results from this experiment are shown in figure 4.4. From this it can clearly be seen that the modified lap is far more responsive than the original. The almost perfect horizontal spread of the original lap's points indicates that the load cell's response as the lap is displaced is negligible. For the modified lap the errors clearly increase with displacement, as is should if the forces are being propagated to the load cells.

It should also be noted that on these graphs the errors at zero displacement are not zero. This is due to two factors. Firstly because there is a small amount of noise on the load cell reading, but more importantly because each point is an average of four readings at different rotations of the lap. Only for the one rotation where the lap was pressed do we expect the errors to be zero (+ noise). As soon as the lap is moved, the aspherical profile of the mirror is so severe that the mismatch is not negligible. The same reason accounts for the reading at zero displacement to be anomalously low, which is particularly noticeable for the original lap.

Qualitatively it is now possible to compare the observed pressure map, shown in figure 4.5 with the predicted pressure map in figure 4.3. It can be seen that allowing for the course spatial sampling of the load cells, the observed pressure map corresponds well with the predicted one. This was repeated for different amounts of offset and for different rotations of the lap, with the agreement being good in all cases.

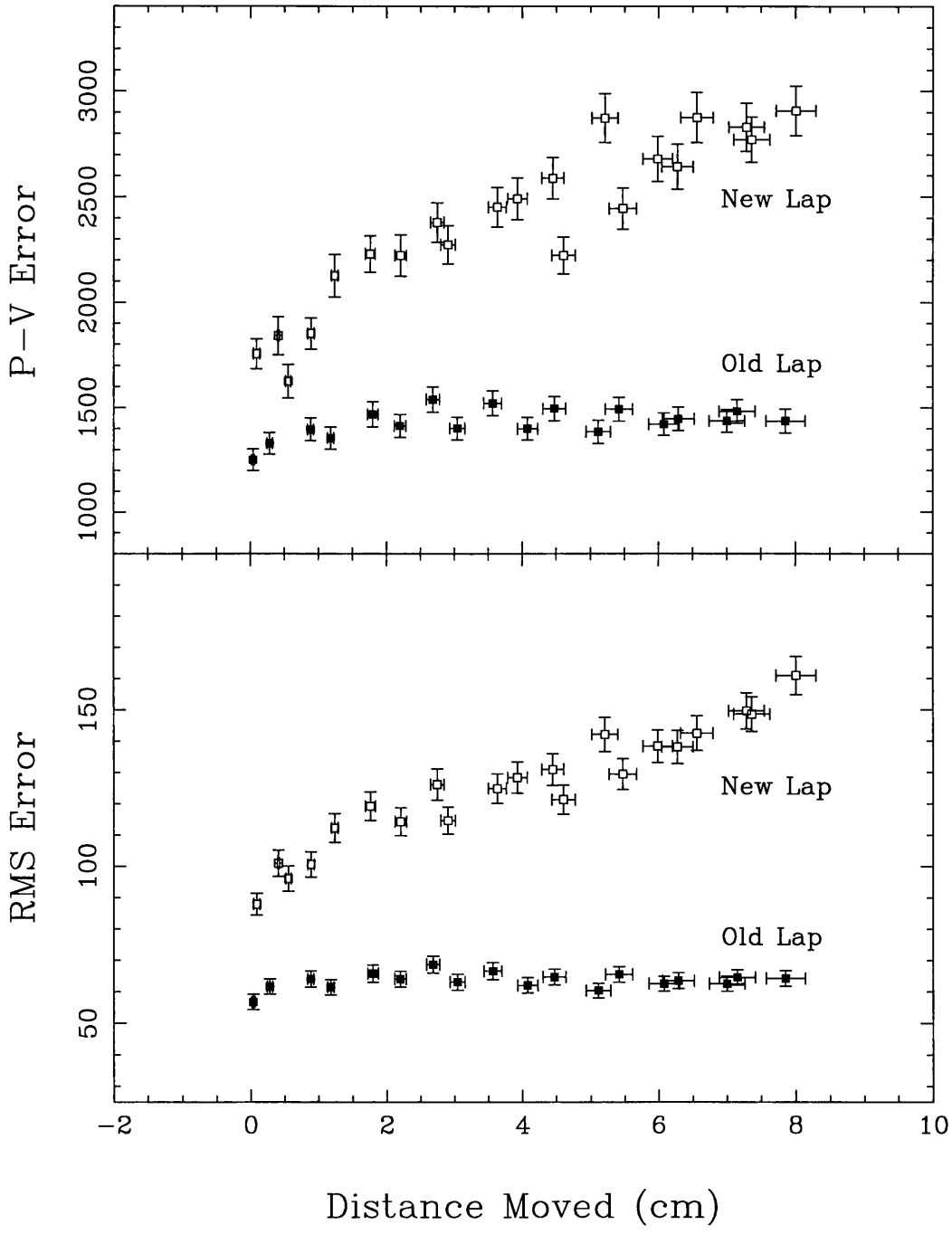


Figure 4.4: Lap Displacement Experiment.

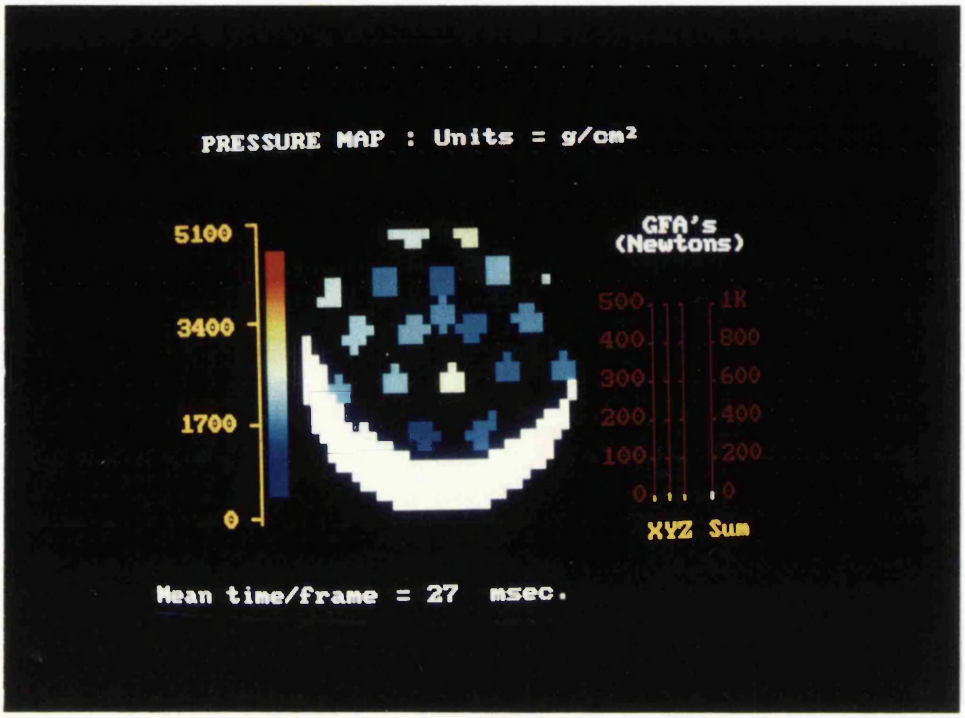


Figure 4.5: Observed Pressure Map when the Lap is Offset.

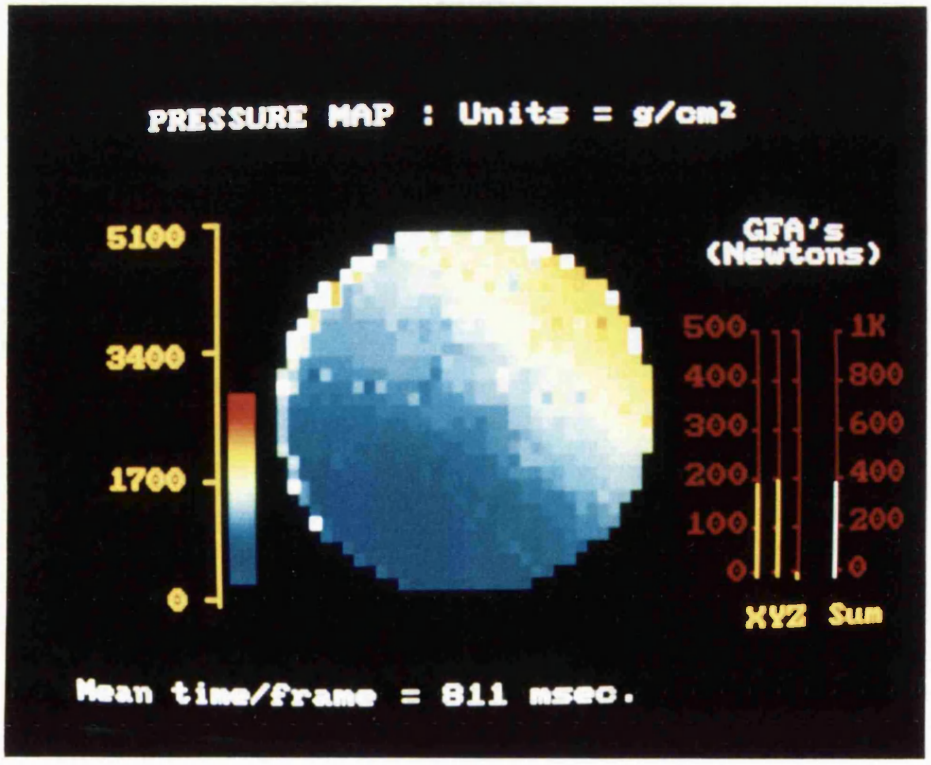


Figure 4.6: Observed Pressure Map when the Lap is Tilted.

4.2.1.2 Tilting the Lap

The second experiment was to tilt the lap by lifting with one of the global force actuators more than the other two. This should result in a constant tilt in the pressure distribution. This can clearly be seen working correctly in figure 4.6.

4.2.1.3 Controlling a Central High Pressure Area

Figure 4.7 shows that it is possible to produce an area of high pressure at the centre of the mirror: a central "hot spot" of pressure. Since the hyperbolic shape of the mirror is turned up with respect to the closest fit sphere, it is necessary to polish mainly at the centre of the mirror. A proposed way of achieving this is the use the lap in the so called *semi-active mode*. tilting the lap.

In this semi-active mode, the actuators are positioned to produce this central hot spot, which is moved around the mirror by the global force actuators as polishing progresses. In effect, this means rocking the lap with the global force actuators, and provides a way of using the lap for local figuring of the mirror. This is also beneficial

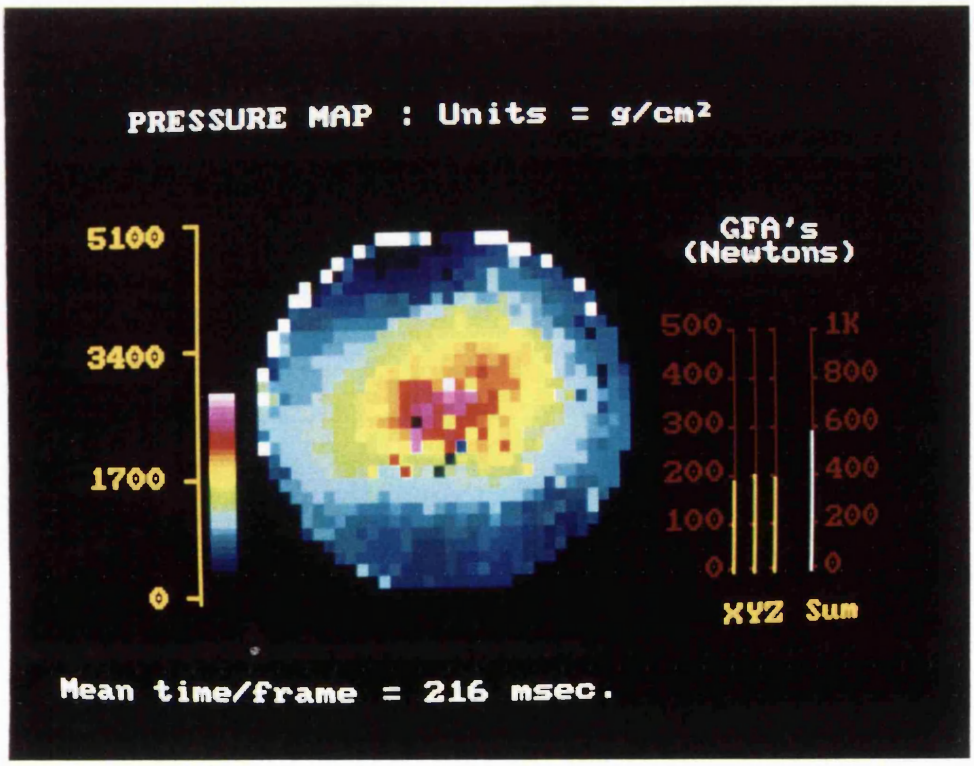


Figure 4.7: Observed Pressure Map with a Central "Hot-Spot".

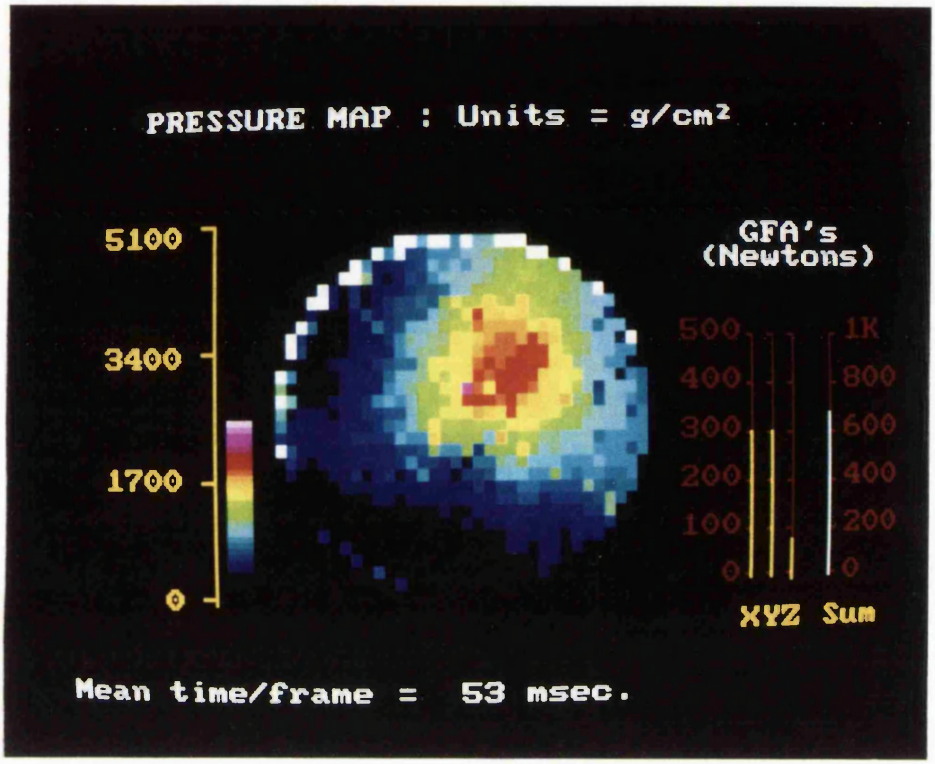


Figure 4.8: Controlling the Position of the "Hot-Spot".

because it means that the lap can be usefully employed whilst the real-time actuator updates are being developed.

A demonstration that it is possible to control the position of the hot spot is shown in figure 4.8, where it has been offset to one side.

Moving the hot spot whilst polishing can be achieved using the dynamic global force actuator routines, detailed in section 3.6.2. Using a low value of α, < 1, would mean that the global force actuators are lifting the overhang by a small amount, which would keep the hot spot close to the edge where the lap is overhanging the mirror. This would result in it preferentially ablating a concentric ring close to the edge of the mirror. Similarly, a high value of *alpha*, > 1, would preferentially ablate a concentric ring close to the centre of the mirror.

At the time of writing, preliminary results of polishing using this semi-active mode seem to indicate that it works extremely well. [36] As detailed in chapter 7, it is planned to primarily use this technique for the next few months, to produce the GEMINI prototype secondary.

4.2.1.4 Summary

These three experiments have not only shown us that with the modified lap the load cells are working as they should, in the sense that the correct pressure maps are being constructed, but also have other implications about the behaviour of the lap as a whole.

- The spatial sampling of the pressure distribution as dictated by the spacing of the load cells is fine enough to detect the features of the actual pressure distribution.
- The lap is now flexible enough to remain in contact with the mirror when it is offset, otherwise zero pressure would be read at the areas of greatest mismatch.
- The lap is stiff enough to still detect the mismatch as areas of high and low pressure. ie. The flexibility of the lap is quite acceptable.
- The lap is able to operate in its semi-active mode.

4.2.2 Testing the Actuators

The testing of the actuators also involved two separate experiments.

4.2.2.1 Wiring/Software Tests

The first was simply to ensure that when the control software attempted to control an individual, or a subset of the actuators, that the correct actuators responded in the correct way. This uncovered minor software bugs and faulty wiring, which were easily corrected. An example of using the actuators to change the pressure map as input by the user of the control software is shown in figure 4.7. This shows the resultant pressure map when the ring of actuators at the edge of the lap are all moved upwards. Because the flex plate and the reaction plate inside the lap are joined via the central washer, this results in a *hot-spot* of pressure forming at the centre of the lap.

4.2.2.2 Pulse Loss Experiments

One potential problem with stepper motor actuators is that pulses sent to an actuator may be lost. ie. If, say, 30 pulses were sent to an actuator, then the actuators may only move 29, or fewer, steps. This could cause two problems if it were found to be a significant effect.

- The actuators would simply not respond as required, so the feedback-loop would not work correctly.
- If the pulses were not lost in equal amounts when driving the actuators up or down, then there might be a tendency for an actuator lead screw to be forced beyond its safe operating range, possibly causing mechanical damage, e.g. to the microswitches.

To find out if the actuators were losing pulses, the following experiment was carried out. In turn, each actuator was sent 1000 sets of pulses, with each set containing a random number of between ± 30 pulses. By measuring the number of pulses the actuator was from the microswitch before and after the experiment, the net number of pulses moved could easily be calculated. This was repeated several times, and graphs of the numbers of pulses moved against the number of pulses sent were plotted for each actuator.

Typical graphs are shown in figure 4.9. From these it can be seen that some of the actuators are losing pulses. The points that do not lie on the solid $y = x$ line, lie below the line. Since the positive direction for pulses is defined as downwards, this means that when pulses are lost, the lead screws are higher than expected.

This can be understood as follows. When moving the actuators down, they are pushing against the mirror, an immovable object. Hence there is a very large force opposing their motion. When the actuators are being moved upwards, there is no such immovable object and hence a much reduced opposing force. It is the presence of this large opposing force that is thought to cause the actuator motors to occasionally fail to move their lead screws, and hence a pulse is "lost".

Whether or not these lost pulses will present a significant problem must now be addressed. It was considered that since even for the worst actuators, so few pulses are

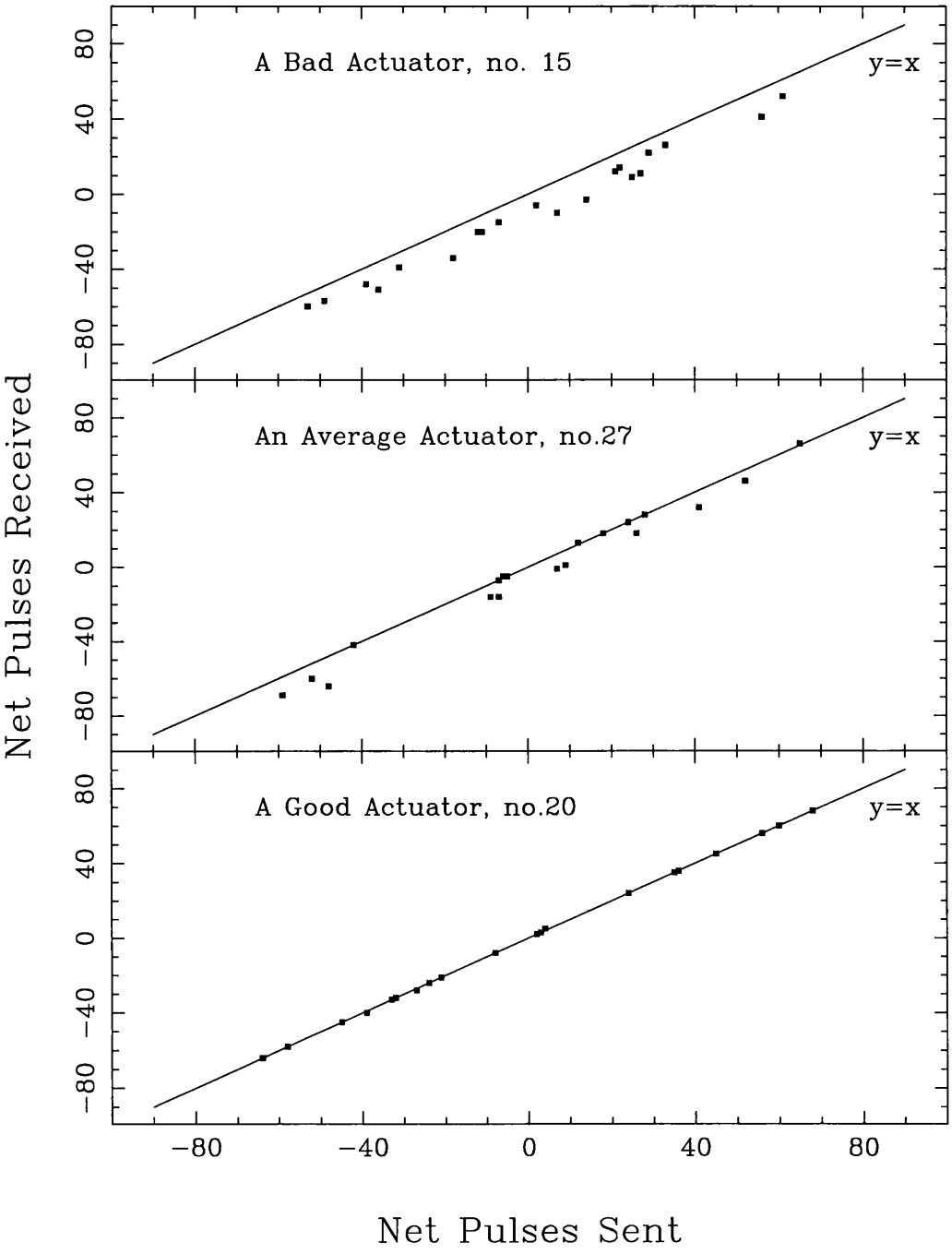


Figure 4.9: Actuator Pulse Loss Experiment.

being lost (the points on the graphs are only losing about 10 pulses per 1000 updates) this would not cause significant problems. However, as a precaution against driving

the actuators hard up the microswitches and possibly causing mechanical damage, polishing is briefly suspended after every 1000 updates and all the actuators reset. This takes approximately 15 seconds every 5 minutes, and so is not an arduous task. Having said this, it is not really satisfactory to simply work around this problem, so Nixon [10] is researching an alternative to the current actuator drive electronics that will provide greater force, and should reduce the number of pulses lost.

Chapter 5

The Interpolation Algorithm

5.1 Introduction

The purpose of the interpolation algorithm is to use the 22 load cell readings to produce a continuous map of the pressure distribution over the whole mirror. This is necessary for two reasons. Firstly to give the user information about what is going on during the polishing run, and secondly, more importantly, to facilitate the calculation of the instantaneous and integrated ablation rates at each point on the mirror. (See chapter 3.)

Ideally, the interpolation algorithm should represent a physically meaningful way of calculating the pressure between the load cells. Unfortunately this has proven difficult to achieve, and will be the focus of some further research in the near future.

Several different approaches have been considered, as detailed in the following sections.

5.2 Zernike Polynomials

It is a common practice in optics to express surface properties, wave-front errors etc. in terms of *Zernike polynomials.* [6] e.g. The surface profile of a mirror can be expressed as a series of these Zernike polynomials, which is useful because each term in the series directly corresponds to a geometrical feature such as the different orders

of astigmatism, coma and spherical aberration. [6] [12] Additionally, the Zernike polynomials are mathematically *orthogonal.* This means that when one polynomial is changed (e.g. a mirror is polished to remove one order of astigmatism) all the other polynomials are not affected.

However, with regards to the pressure distribution, Zernike polynomials are not readily applicable. This is because the dominant feature of the pressure map will be caused by the mismatch between the lap and the mirror when the lap is displaced, which will have a completely different mathematical form than the Zernike polynomials. Although the mismatch could be described by using sufficient high order polynomials, the individual polynomials no longer have any physical meaning, which is the property that makes them attractive to use.

An additional disincentive for using Zernike polynomials is that the calculations will involve sin and cos functions raised to high powers, which will necessarily involve a significant amount of cpu time.

5.3 Using the FE Analysis

A second way of finding a physically meaningful interpolation algorithm is to use the Finite Element analysis, detailed in [16].

There were two ideas about how this could be used.

- The load cell–load cell coupling coefficients could be used to create a three dimensional distribution that shows the relative influence each load cell would have over an arbitrary point on the lap.

  The relative influences from each load cell could then be used to calculate the interpolation coefficients.

- Once polishing has begun, several sets of representative actuator positions / load cell readings could be fed into the FE models. From the models it would then be possible to find the pressure readings at the 39 "dummy" load cell positions. These could then be used as test values for different algorithms to be developed. It would then have been established that the final algorithm found would work correctly for all the sets of representative data.

The major problem with this approach would be the difficulty in determining the correct set of representative data to be used for developing the algorithm.

However, before either of these approaches could be tried, it had to be determined whether or not the FE models of the lap are valid. This has been tested extensively, and as detailed in chapter 6 it was deduced that the models are not valid representations of the Active Lap. Therefore both these approaches have had to be abandoned.

5.4 Surface Fit

The third class of algorithms that was looked at were those that perform a *surface fit.* These fit a three dimensional surface to the load cell data values. This would have been straight forward if the load cells had been situated on a regular rectilinear grid. However the irregular, hexagonal grid that the load cells are mounted on made the fitting of a surface to the points significantly more difficult. [31]

To attempt to fit to the irregular grid several different techniques were tried from the standard NAG library of numerical routines. [25] The three NAG routines that seemed applicable to our data were to generate the surface fit by interpolation using either the Renka and Cline method, a modified Shepard method, or to perform a weighted least-squares bicubic spline fit. (These correspond to NAG library codes E01SAF, E01SEF and E02DAF respectively.)

In each case the interpolation routine had to perform satisfactorily with several different characteristic data sets. (i.e. They have to correctly interpolate a "flat" data set where all the load cells read the same, as well as data sets which correspond to the maximum mismatch between the hyperbolæ, etc.)

It was found in all three cases that the interpolated surfaces were never acceptable for both the data sets mentioned above. e.g. Instead of producing a smooth surface between the data points, an algorithm that successfully interpolates a "flat" data set would produce extremely large deviations between the points of the maximum mismatch data set, etc. Therefore these three methods had to be abandoned.

When considering the Active Lap project as a whole, it was decided that the detailed

investigation that was required to find an acceptable interpolation algorithm had a lower priority than other key tasks that required attention, particularly work needed to implement the real–time updates. It was therefore appropriate to temporarily abandon the idea of finding a physically meaningful interpolation algorithm, and to implement an algorithm that simply presents the user with an easily understandable graphics display. To accomplish this, the following scheme was put into practice.

5.5 Nearest Neighbours

For each pixel in the screen display of the pressure map, the nearest n load cell values are used to interpolate the pressure.[1] If the distance from the pixel to the load cell is r, and each load cell reads L, then the force at an arbitrary point i is given by

$$F_i = \sum_{j=1}^{n} \frac{N}{r_j} L_j \tag{5.1}$$

where N is a normalization constant given by

$$N = \sum_{j=1}^{n} \frac{1}{r_j} \tag{5.2}$$

The pressure at each pixel is then calculated by dividing the force by the area of a pixel.

This proved to be a fairly good graphical representation of the data, but there were often polygonal features on the pressure map, which clearly were not correct. A modification was then made which smoothed the data by averaging each pixel with the eight others that surround it. The interpolated pressure map was now found to corresponds well with the "spot" pressure map.

Using equation 5.1 for the interpolation algorithm had the added advantage that the pressure is calculated as a simple series involving coefficients that can be pre-calculated and values of L. As detailed in section 3.1 this is useful for minimizing the amount of cpu time used. When the smoothing function was implemented, the algorithm was simply rewritten to retain this series characteristic.

[1] Somewhat arbitrarily $n = 6$. This is trade off between using as many load cells as possible, and keeping the cpu time used to a minimum.

5.6 Summary

It has proven difficult to find a physically meaningful algorithm that will successfully interpolate the pressure between the load cells in every case required. Considering the project as a whole, it has been decided that further investigation in this area should be temporarily postponed, and an interim solution adopted.

A possible reason for the difficulties found is that as detailed in section 3.4.1.3 we are trying to interpolate to pixels of size 2.37×2.76cm from the load cells that are 17.3cm apart. With hindsight it appears that the spacing between the load cells has been designed to be larger than is optimum, an issue that is under consideration as a possible future modification.

Chapter 6

Closing the Feedback Loop : The Real Time Updates

6.1 Introduction

Closing the feedback loop between the arrays of load cells and actuators was a vital part of the Active Lap project, as detailed in chapter 2.

The mechanical design of the lap produces a complex influence function between the actuators and the array of load cells. It was thought that this could be described by a matrix of *coupling coefficients*, which define the influence each actuator has on the array of load cells. [16]

This chapter firstly describes the algorithmic technique that was to be used to perform the real time updates using both theoretical (from FE models of the lap [16]) and experimental coupling coefficients. However, the validation experiments carried out showed that the algorithmic technique cannot be successfully applied with the current Active Lap hardware.

Therefore, an alternative approach is detailed, utilizing an artificial neural network. This has shown encouraging signs that it may become a viable technique, and results of experiments to close the feedback loop using this method are presented.

6.2 The Algorithmic Approach

The coupling coefficient between an actuator and a load cell is the change in force measured by the load cell when the actuator applies a force of 1 Newton. It has been postulated that the total force update on the load cell when all the actuators apply forces can then be approximated by a simple linear sum, resulting in the following system of equations [16]

$$L_i = \sum_{j=1}^{32} C_{ij} A_j \qquad \text{for } i = 1, 2, \ldots, 22 \tag{6.1}$$

where A_j is the force applied by actuator j, L_i is the corresponding change in force measured by load cell i, and C_{ij} is the coupling coefficient between load cell i and actuator j. [16]

Equations 6.1 can then be used to calculate the actuator updates. When the pressure distribution is sampled by the 22 load cells at their current positions, by comparing the readings to some pre-defined target pressure distribution, the change in force each load cell requires to conform to this target pressure distribution can be found. i.e. in equation 6.1, all the L_i values, along with the pre-calculated C_{ij} coefficients are known, and the actuator updates needed to achieve the target pressure distribution can be found by simultaneously solving the equations.

However, this is not trivial. Since there are 22 equations and 32 unknowns, it is not possible to find a unique solution[1], but it is possible to find the family of solutions that all satisfy equations 6.1 using the numerical technique *Singular Value Decomposition* (SVD). [31] Each of the sets of actuator updates that SVD gives us can produce the desired pressure distribution. The particular solution to be used, would then be the one with the smallest mean actuator update, since this would achieve the desired pressure distribution in the quickest time, as well as being the least likely to exceed the maximum update that the actuators can achieve per frame.

[1] In general, for M equations and N unknowns, where $M < N$, there will usually be an $N - M$ dimensional family of solutions [31]

6.2.1 Experimental Verification

The above method is based on applying the theoretical FE models of the Active Lap. Before it was put into practice, a number of experiments were necessary to ensure that the lap behaves as predicted, and that this algorithmic approach on the whole is valid. These experiments were

- Find the empirical coupling coefficients with the lap centrally on the mirror, to ensure that they agree with the theoretical ones within experimental error.
- Find the empirical coefficients with the lap at its maximum displacement from the centre of the mirror, to ensure that they remain constant.
- Ensure that the empirical coupling coefficients are repeatable at different times.
- Ensure that the load cells respond linearly to different actuator forces, which is implicitly assumed in equations 6.1.
- Ensure that the total force update on the load cells when all the actuators apply forces can be approximated by a simple linear sum.

6.2.1.1 Finding the Empirical Coupling Coefficients

With the lap positioned centrally on the mirror, and allowed to press, actuator number 2 was moved by 50 pulses and the change in all the load cell readings monitored. Actuator 2 was chosen since it is in the central ring of actuators, and not at the extremes of the inside or the edge of the lap.[2] This was repeated 16 times to reduce the experimental error and the coupling coefficients between actuator 2 and each of the load cells found by normalizing the results to an update of 1 Newton. A graph of experimental verses theoretical FE coefficients was then plotted and is shown in figure 6.1.

If the experimental and theoretical coefficients were identical, then the points should lie on the straight line $y = x$, shown dashed in figure 6.1. Possible reasons why this

[2]Had the algorithmic approach proved to be valid when using only actuator number 2, then the experiments would have been repeated using other actuators.

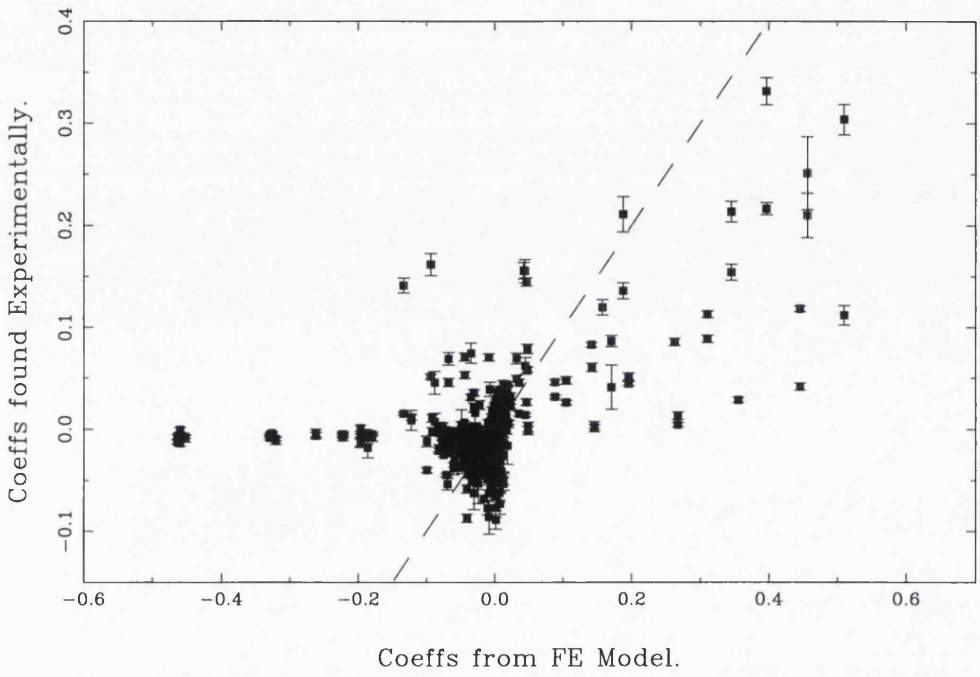


Figure 6.1: Experimental verification of FE Coupling Coeffs.

is not the case include

- The FE models of the lap are not valid. Due to software constraints the models do not include the carbon fibre skin, the epoxy wedges and the epoxy bag. The effects of these may not be negligible. The spring constant of the carbon fibre skin and the inability of the epoxy bag to provide perfect internal contact may both be significant.
- As detailed in section 4.1.1.3, the calibration of the load cells may not be completely reliable.

Despite this, the algorithmic approach to the real time update could still be valid if the experimental coefficients are used rather than the theoretical FE coefficients.

6.2.1.2 Ensuring the Coeffs Remain Constant with Lap Displacement

The above experiments were repeated exactly, but with the centre of the lap displaced 10 cm from the centre of the mirror, this being the likely extreme of the

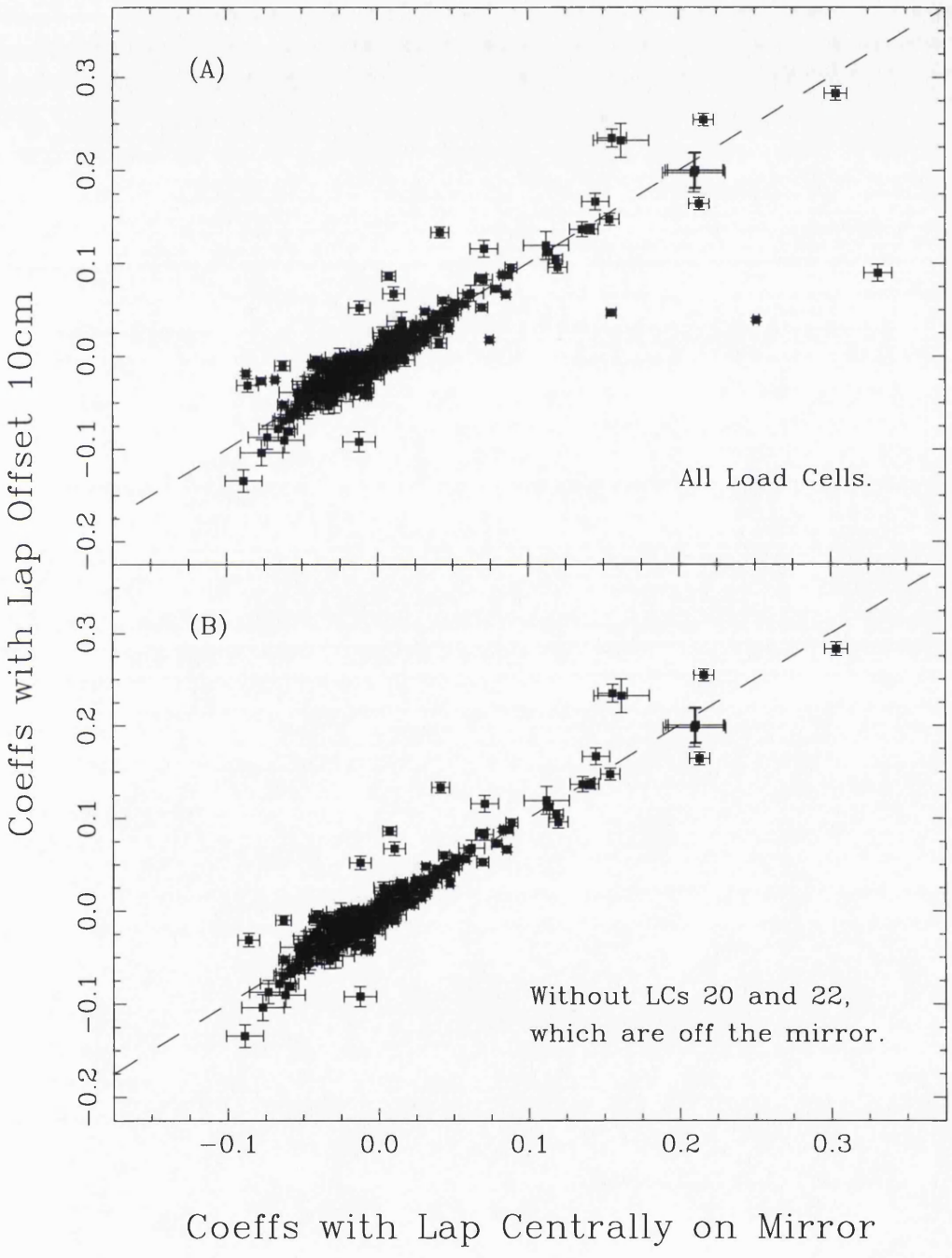


Figure 6.2: Ensuring the Experimental Coupling Coeffs Remain Constant with Lap Displacement.

stroke to be used during polishing. A graph of the coefficients found with that lap displaced 10 cm verses the coefficients found with the lap centrally on the mirror was plotted, and is shown in figure 6.2 (A). As before, if the coefficients remain constant, then the points should lie on the straight line $y = x$, again shown dashed.

Clearly most of the points lie very close to the line $y = x$, but there are a few a significant distance away. Figure 6.2 (B) shows the same graph, but without the coupling coefficients for load cells 20 and 22. When the lap was displaced, these two load cells were no longer on the mirror, and therefore have a different support condition. It would therefore be unreasonable to expect their response to the actuators to remain constant, which is born out by the absence of the "rogue points" found in figure 6.2 (A). Indeed, the fact that approximately 98% of the points in figure 6.2 (B) lie on, or extremely close to the line $y = x$ confirms that the response of the load cells that remain on the mirror is independent of the lap's displacement within experimental error.

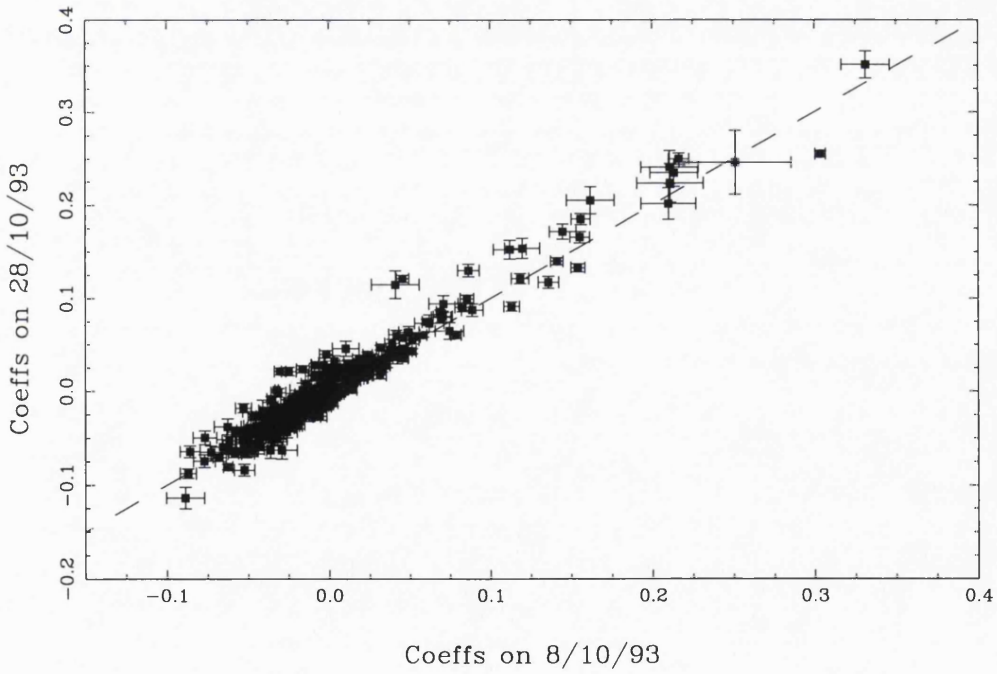


Figure 6.3: Ensuring the Experimental Coupling Coeffs Remain Constant with Time.

6.2.1.3 Ensuring the Coeffs Remain Constant with Time

To ensure that the experimental coefficients were repeatable at different times, the experiment detailed in section 6.2.1.1 was repeated after a 20 day interval. The coefficients found on each occasion were plotted against each other, which is shown in figure 6.3. Since the points clearly lie on the line $y = x$ (shown dashed) it is concluded that within experimental error, the coefficients are indeed repeatable over time.

6.2.1.4 Ensuring the Load Cells Responses are Linear

Equations 6.1 contain the implicit assumption that each of the load cells responds linearly to different actuator forces. This is equivalent to saying that for different actuator force updates, the coupling coefficients should remain constant. To verify this, actuator number 2 was moved by different amounts, and by normalizing to 1 Newton, the coupling coefficients for each load cell calculated each time. Therefore, if the load cells response is linear, a graph of coupling coefficient verses actuator

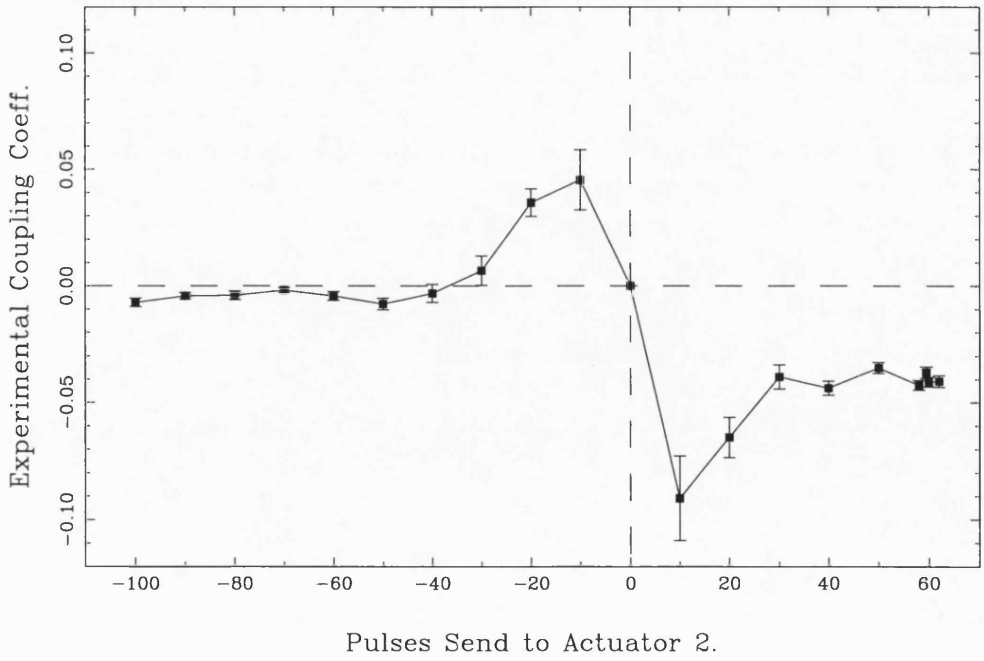


Figure 6.4: The Response of Load Cell 3 to Different Forces from Actuator 2.

update should produce a straight horizontal line $y = C_{i2}$, where C_{i2} is the coupling coefficient between load cell i and actuator 2.

It was found that the load cell response was anything but linear. A typical response curve is shown in figure 6.4. This shows that when actuator 2's lead screw is moved upwards (i.e. the actuator is sent a negative number of pulses) the load cell 3 has very little response, but when it is moved downwards the load cell's response is significantly greater. (i.e. The *magnitude* of the load cell's response is larger.)

A possible explanation for this is that when the lead screw is moved down, it is pushing against an immovable object (i.e. the mirror). Therefore nearly all of the kinetic energy of the lead screw is transferred into potential energy in the actuator spring, since the flex plate is unable to move. (A small amount will go into the bending of the reaction plate, but this is likely to be negligible.) Thus, nearly all the energy will be used to increase the forces read by the load cells. However, when the lead screw is moved upwards the flex plate is free to move, so some of the kinetic energy of the lead screw is used in bending the flex plate, and less is stored in the actuator spring. Hence the response of the load cells will not be as great.

Although this non-linearity is a complication, in itself it does not invalidate the algorithmic approach, if an appropriate function is fitted load cell response. Thus equation 6.1 becomes

$$L_i = \sum_{j=1}^{32} f_{ij}\left(A_j\right) \qquad \text{for } i = 1, 2, \ldots, 22 \tag{6.2}$$

where $f_{ij}\left(A_j\right)$ are the functions fitted to the load cell response curves.

Hence, the problem is now to solve a system on non-linear equations. This is significantly more difficult than solving a system of linear equations, but not intractable.

Firstly, the appropriate function needed to be selected. Since the solution of non-linear equations is an iterative technique involving many calculations, it seemed advisable to choose the simplest function possible that would adequately approximate the load cell response. Therefore, the function chosen was a simple "step function", where

$$f_{ij}\left(A_j\right) = \begin{cases} C_{i,j}^{-} & \text{if pulses sent to actuator} < 0 \\ C_{i,j}^{+} & \text{if pulses sent to actuator} \geq 0 \end{cases} \tag{6.3}$$

where $C_{i,j}^{-}$ and $C_{i,j}^{+}$ are constants, being the mean response either side of zero pulses.

Having found this functional form, equations 6.2 can be solved using a modified Newton-Raphson method. [31] i.e. Rewrite equations 6.2 as

$$f_i(A_1, A_2, \ldots, A_{22}) = 0 \qquad \text{for } i = 1, 2, \ldots, 22 \tag{6.4}$$

If we let $\mathbf{\Lambda}$ denote the entire vector of values A_j, then in the neighborhood of $\mathbf{\Lambda}$, each of the functions f_i can be expanded as a Taylor series

$$f_i(\mathbf{\Lambda} + \delta\mathbf{\Lambda}) = f_i(\mathbf{\Lambda}) + \sum_{j=1}^{32} \frac{\delta f_i}{\delta A_j} + O(\delta\mathbf{\Lambda}^2) \tag{6.5}$$

By neglecting terms of order $\delta\mathbf{\Lambda}^2$ and higher, a set of linear equations for the corrections $\delta\mathbf{\Lambda}$ are obtained, which move each function closer to the solution, namely

$$\sum_{j=1}^{32} \alpha_{ij}\delta A_j = \beta_i \qquad \text{for } i = 1, 2, \ldots, 22 \tag{6.6}$$

where

$$\alpha_{ij} \equiv \frac{\delta f_i}{\delta A_j} \qquad \text{and} \qquad \beta_i \equiv -f_i \tag{6.7}$$

Matrix equation 6.6 can then be solved using SVD as detailed in section 6.2. The corrections are then added to the solution vector,

$$A_j^{new} = A_j^{old} + \delta A_j \qquad \text{for } j = 1, 2, \ldots, 32 \tag{6.8}$$

and the process is iterated to converge.

This method will only converge if we are in the neighborhood of a solution. i.e. it is necessary to have a good initial *guess*. Since working with an update frequency of 10 Hz corresponds to a linear displacement of only 1 cm on the mirror, the solution vector will change very little between two successive frames. Hence, it is possible to use the solution vector from the previous frame as a good initial guess for the solution vector for the current frame. It also means that it is very likely that only one or two iterations will be required to solve the equations, which will clearly minimize the CPU time used.

6.2.1.5 Ensuring the Load Cell Responses are Additive

It was necessary to ensure that the load cell responses are additive. i.e. that combining load cell response functions from each actuator can be approximated by a simple linear sum, as given by equations 6.2.

To accomplish this, different random numbers of pulses were sent to all the actuators simultaneously and the responses of each load cell measured. This was done with 4 different sets of random updates, with each set repeated 16 times.

A graph of the load cell responses measured, versus the responses predicted by equations 6.2 (calculated using the experimental coupling coefficients in equation 6.3) was plotted, and is shown in figure 6.5. If the experimental load cells responses agree

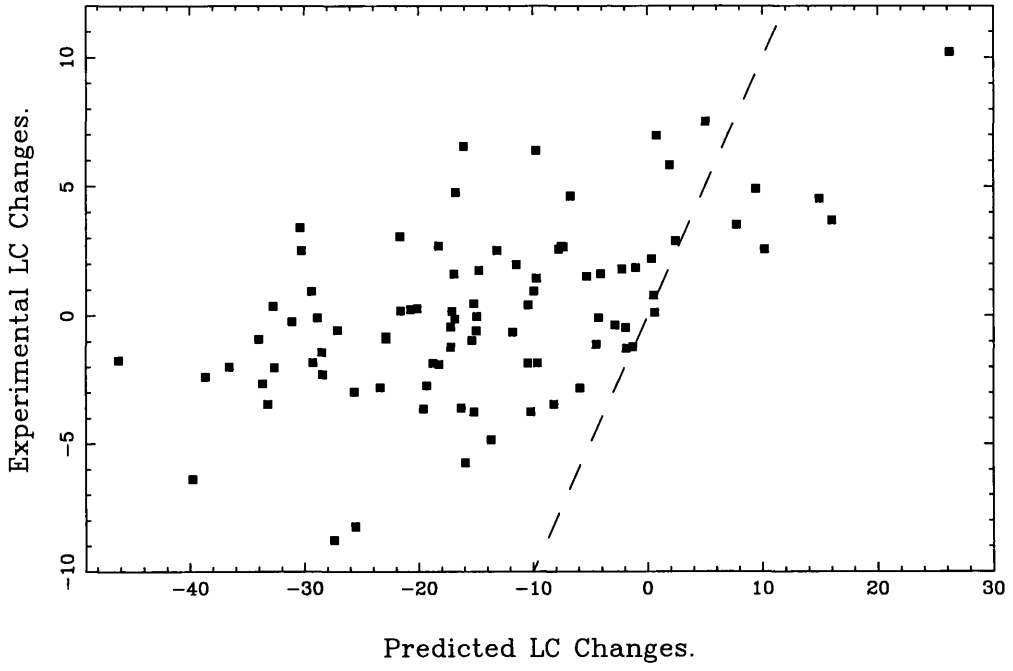


Figure 6.5: Additive experiments with Random Updates.

with the predicted ones, then the points in figure 6.5 should lie on the line $y = x$, which is shown dashed. Clearly this is not the case. It was thought, however, that figure 6.5 may not be directly applicable to *polishing* with the Active Lap. When polishing, it seems unlikely that the actuators will be sent arrays of random updates. Instead they are likely to require updating in groups. As an example of this, the above experiment was repeated, but with the following number of pulses sent to the actuators.

Outer ring	-30 pulses
Middle ring	Zero pulses
Inner ring	30 pulses

This simulates the fist order effect that the lap has to perform, which is to turn up the edge of the mirror with respect to the nearest sphere. The results of this are plotted in figure 6.6, with the line $y = x$ shown dashed. The fact that in both figure 6.5 and 6.6 the points do not lie on the line $y = x$ may be explained as follows:–

1. It is not valid to assume that the load cell response functions are linearly additive: there are cross terms between different actuators which are not negligible. i.e. if a force F is applied to an arbitrary actuator A, then another arbitrary actuator B will experience a force equal to $K_{AB}F$, where K_{AB} is the actuator–actuator coupling coupling coefficient between actuators A and B.

   Physically, this means that when actuator A is moved, since the reaction and flex plates are coupled as detailed in chapter 2, the plates either separate or get closer together. This will either extend or compress actuator B's spring respectively. This is equivalent to a force being applied to actuator B directly.

2. Fitting a simple step function to the load cell response curves is inadequate,

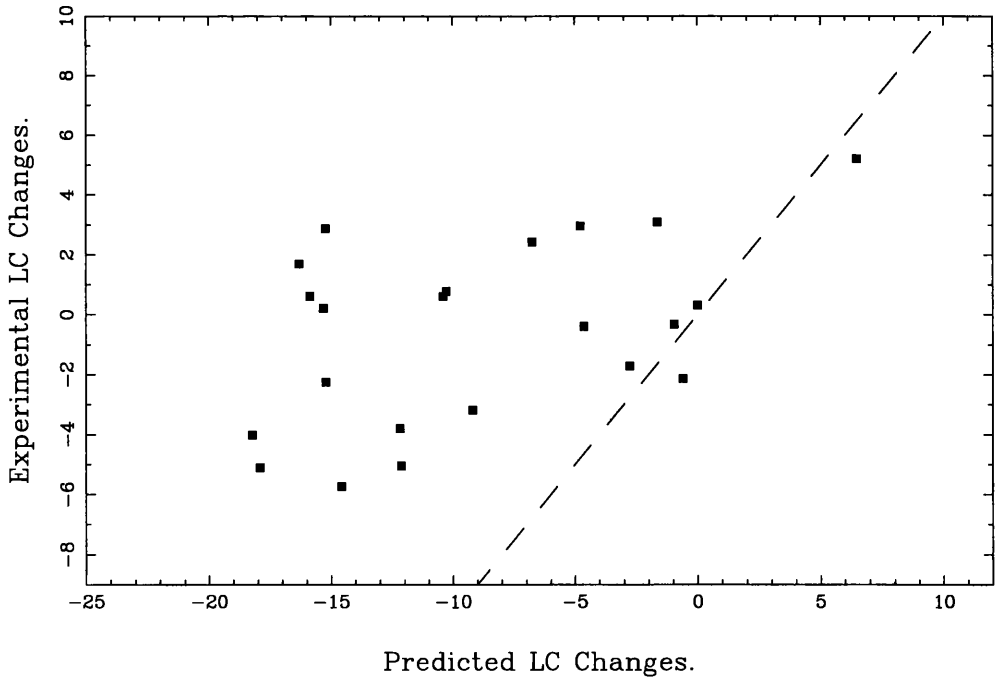


Figure 6.6: Additive experiments with Simulation Updates.

and produced significant errors into equations 6.2.

Theoretically it is possible to overcome both of these problems, but in practice is thought to be impossible.

1. If the actuator–actuator coupling is not negligible, then equations 6.2 must be rewritten as

$$L_i = \sum_j f_{ij}\,(A_j) + \sum_{j,k} f_{ijk}\,(A_j A_k) + \sum_{j,k,l} f_{ijkl}\,(A_j A_k A_l) + \ldots \qquad (6.9)$$
$$\text{for } i = 1, 2, \ldots, 22$$

   Simple probability theory then gives the number of different terms in each equation to be

$$\sum_{i=1}^{32} {}^{32}\mathbf{C}_i \simeq 4.3 \times 10^9 \qquad (6.10)$$

   where $\mathbf{C}$ is the standard "combination" symbol, given by

$${}^{n}\mathbf{C}_r = \frac{n!}{r!(n-r)!} \qquad (6.11)$$

   It appears likely that most of these terms will be negligible, but even if this is the case it does not seem possible to design an experiment that will enable these actuator–actuator coupling coefficients to be determined without modification of the Active Lap hardware[3].

   There is the possibility of using theoretical actuator–actuator coefficients found from the FE models. These are likely to be more reliable than the load cell–actuator ones, since being situated above the flex plate they are less dependent on the support condition. However, this does not make them reliable enough to use without experimental verification.

2. If fitting a simple step function to the load cell response curves is inadequate, then the next logical step is to try fitting a polynomial of degree n, where n could possibly be as high as 5. Although time consuming, this would be possible using standard numerical techniques off–line. However, the technique for solving the system of non-linear equations detailed in section 6.2.1.4 would now

[3] e.g. by inserting a device to measure the extension of each actuator spring.

appear to require an enormous amount of CPU time for its many calculations. When solving matrix equation 6.6, each iteration will require the calculation of the 32×22 partial derivatives, α_{ij}, each one involving raising to powers up to $n - 1$. Given the fact that SVD also involves complicated calculations and that more than one iteration may be necessary, it seems almost certain that this technique will take far longer than to perform than the target 0.1 seconds per frame.

Hence, for the above reasons, (i.e. the need for, and inability to find, the actuator–actuator coupling coefficients, and the likely excess of CPU time required) it was decided that the algorithmic approach for calculating the real time updates was not the appropriate one. Although it may be advisable to reapraise this some time in the future, the algorithmic approach was therefore abandoned.

6.3 Using Neural Networks

N.B. Throughout the remainder of this chapter references will be made to standard neural network terminology and nomenclature. For readers not familiar with this, Appendix B: An Introduction to Neural Networks, provides all the necessary background knowledge.

Neural networks are very good for modelling highly non-linear systems, such as the problem of performing the real-time updates. Indeed expert advice [3] confirmed that neural networks should be ideal to solve the type of problem encountered here.

In this case, a neural network will be used to learn the relationship between an array of desired changes to the load cell readings, and the actuator updates required to make those changes. Mathematically speaking, it will be used to learn the mapping between the two parameter domains.

The software used to run the neural network simulation, was the *Stuttgart Neural Network Simulator*, version 3. 2. This is a sophistication piece of application software written by the Institute for Parallel and Distributed High Performance Systems at the University of Stuttgart. It was run under X Windows on the STARLINK Alpha workstations. This was used because it was known to be reliable software [23] that is available free of charge for non commercial use.

6.3.1 Choosing Network Parameters

There are many different possible types of neural network models, each of which is applicable to different situations. Also, within each type of model there are many different parameters that have to be set. At the current stage of our understanding of neural networks, the setting of all these parameters is still very much down to the skill of the user, rather than being based on rigorous scientific methods.

The emphasis of this work is obviously to use neural network technology as a tool, rather than to research the behaviour of the different models. Therefore, whenever possible, the simplest "standard" options were chosen. As detailed later in the chapter, preliminary results seem to indicate that these have been good choices, but fine tuning may well need to be done in the future.

The type of network used and the salient parameter choices are listed below. The choice of most of these was confirmed by referring to [3].

- *Network Architecture* – Choosing the type of network architecture to use was a relatively simple task. Although there are numerous different types of network architecture, a feed-forward network was considered eminently suitable for our purposes. [3] It has successfully been applied many times to model the mapping between two independent domains. [32] Indeed, it has been proved that a feed-forward network with hidden layers is theoretically able to learn the mapping between any two independent parameter domains. [9]

- *Training method* – It was decided to start off by using supervised learning. This was purely for simplicity so that adjustments to the network could be made until a working model was found. Once a working model was up and running, then it would be appropriate to continue training the network using unsupervised learning. This would allow the model to continue to adjust to changes in the polishing environment,

- *Training algorithm* – Once again, purely for simplicity, standard back-propagation would be used for training the network, at least for the initial supervised phase.

- *Number of Layers* – To start with, a three layer model would be used, with one input, one hidden, and one output layer. If this turned out to be unsuccessful

then a second hidden layer would be added.

- *Activation Function* – Two activation functions were used. Firstly the standard logistic function, and secondly, since it has good generalization properties, the tanh function. In order to use these, the input and output data has to be normalised to the ranges 0.1-0.9 and ± 0.9 respectively. (These ranges are basically 0-1 and ± 1, but since the functions change very little at the edge of their ranges, these parts are excluded to avoid losing information. [37])

- *Number of Inputs* – Two different models were trained, both of which had the 19 desired changes to the active load cell readings, the x and y displacements of the lap, and ϕ, the rotation of the lap, giving a total of 22 inputs. In order to see whether for each actuator the update was dependent on the position of the lead screw, one of the models also had the 32 lead screw positions as inputs. (Total=54.)

- *Number of Outputs* – The output from the neural network are the number of pulses that are needed to be sent to the 32 actuators to achieve the desired load cell changes.

- *Number of Hidden Units* – For an network of this size (25 inputs and 32 outputs) the usual number of hidden units that has achieved a successful mapping is of the order of ≤ 10. [3] [32] Therefore, four different models were tried with 4, 6, 8 and 10 hidden units.

6.3.2 Acquiring *Realistic* Training Data

In general, neural networks are very good at interpolating between the training data that they learn from, but not-surprisingly are not so good at extrapolating. [32] [9] Therefore, to achieve good performance when polishing, the network should be training using *realistic* training data. ie. data which is directly applicable to a polishing run. Mathematically speaking, there needs to be good coverage of the two parameter domains in which the neural network is required to have good generalization properties. This means that the training data should contain the type of changes needed to the load cell readings, and the corresponding actuator updates, that will

occur during actual polishing.

There is a problem, however, in obtaining such data. Since there is no working update algorithm prior to the neural network being successfully trained, there are no *realistic* updates from previous polishing runs available. In addition to this, there is no model of the required updates that would allow some form of simulated training data to be generated.

The solution to finding realistic training data was as follows. For the first polishing run, the actuators were sent random updates, and their effect on the load cell readings monitored. These sets of data were then used to train a neural network. This training data would not have good coverage in the two parts of the parameter domains where good generalization is required, but would be a useful first approximation.

For the second polishing run, the network trained on the random updates would be used to try to control the pressure map. Since this uses only a first approximation neural network, this is not expected to adequately control the pressure, but will be better than the random updates. However, the data obtained will allow the training of a second network. This training data should have better coverage of the desired parameter domains, and should therefore produce an neural network with better generalization properties. This in turn should mean that the network is better able to control the pressure map.

This process of iteration should continue until an acceptable network is produced.

6.3.3 Results

To collect the first set of training data, the mirror was polished whilst the actuators were being sent random numbers of pulses. The range of these updates was ± 30 pulses. For each set of updates the resultant changes to the pressure read by the load cells was also recorded. 16000 sets of training data were stored in all. This may seem a large number, but when comparing to the total sizes of the two parameter domains is relatively small small. For the simplest model that does not include the actuator positions, the two parameter domains have the following sizes.

- Load cell update domain: Each of the 19 active load cells, the x and y displacements, and ϕ, all have a total of 256 possible values. This means there are $256^{22} = 9.6 \times 10^{52}$ different combinations.

- Actuator domain: For each actuator, there are 61 possible updates, which means there are $61^{32} = 1.4 \times 10^{57}$ different combinations.

Although $10^{57} \gg 16000$, nearly all the possible combinations will not be applicable to actual polishing. The dominant effects during polishing are the mismatch between the lap and the mirror hyperbolas and the "digging in" of the leading edge. As polishing occurs, this will mean that there are repeatable patterns in the load cell data, which the actuators will respond to. These repeatable patterns will only cover a subset of the total parameter domains. Although it is not known *a priori* how small this subset will be, it is consistent with current neural network practice to use a training data set of the size. If anything, this training set is larger than the current norm. [37]

Once the network is trained, its performance needs to be evaluated. With a training set of 16000 it is appropriate to train the neural network with 14000 samples, and withhold 2000 as test data with which to evaluate the network. [37] The evaluation procedure was to run the neural network with the load cell updates from the 2000 training samples as inputs, and record the output from the network. These outputs were then compared to the training data's actuator updates, with the mean difference between the two giving an error measure of the networks performance. ie. If for each actuator a, in training data set s, the network output is o_a^s pulses, the training data target output is t_a^s pulses, and there are n training patterns, then the error is given by

$$\frac{\sum_{s=1}^{n} \sum_{a=1}^{32} o_a^s - t_i^s}{n \times 32}$$

In the ideal case, the neural network would be in perfect agreement with the test data. This would result in an error of zero. However, an error of zero would also result if all the $(o_a^s - t_i^s)$ were distributed symmetrically about zero. This will probably be the case, as the errors are expected to be random. Therefore, it will be more representative if the spread of the errors were considered. In the ideal case the spread around zero will be very small, but more realistically, the errors are expected

Learning Parameters	Act. Fn.	Number of units			Error	
		Input	Hidden	Output	Mean	SD
0.2 - 0.1 - 0.05	logistic	22	4	32	-0.2	13.1
	logistic	22	6	32	-0.3	12.9
	logistic	22	8	32	-0.2	12.7
	logistic	22	10	32	-0.5	12.6
0.2 - 0.1 - 0.05	tanh	22	4	32	-0.7	14.6
	tanh	22	6	32	-1.1	14.9
	tanh	22	8	32	-1.4	15.0
	tanh	22	10	32	-1.7	14.9
0.2 - 0.1 - 0.05	logistic	54	4	32	-0.1	13.1
	logistic	54	6	32	-0.2	12.8
	logistic	54	8	32	-0.4	12.6
	logistic	54	10	32	-0.4	12.5
0.2 - 0.1 - 0.05	tanh	54	4	32	-1.3	15.4
	tanh	54	6	32	-0.7	16.3
	tanh	54	8	32	-2.5	17.8
	tanh	54	10	32	-1.8	19.4

Table 6.1: Neural Network Training Results - A.

to be spread widely. The standard deviation of the errors will therefore be used, since this gives a quantitive measure of this spreading.

The results from training the various models are shown in tables 6.1 and 6.2.

Two different sets of training parameters were tried.

- Learning rate = 0.2, Momentum = 0.1, Flat Spot Elimination = 0.05
- Learning rate = 0.05, Momentum = 0.01, Flat Spot Elimination = 0.05

The first set are standard values, that are typical of those used successfully elsewhere. [3] [32] The second set of much lower values were tried because with a relatively large training data set of 14000 items, it was thought that these may train the network more successfully. [3]

Learning Parameters	Act. Fn.	Number of units			Error	
		Input	Hidden	Output	Mean	SD
0.05 - 0.01 - 0.05	logistic	22	4	32	-0.1	13.0
	logistic	22	6	32	-0.2	12.8
	logistic	22	8	32	-0.2	12.7
	logistic	22	10	32	-0.3	12.6
0.05 - 0.01 - 0.05	tanh	22	4	32	-0.3	13.2
	tanh	22	6	32	-0.4	13.1
	tanh	22	8	32	-0.5	13.0
	tanh	22	10	32	-0.6	12.9
0.05 - 0.01 - 0.05	logistic	54	4	32	0.0	13.0
	logistic	54	6	32	-0.1	12.7
	logistic	54	8	32	-0.3	12.5
	logistic	54	10	32	-0.2	12.4
0.05 - 0.01 - 0.05	tanh	54	4	32	-0.4	13.3
	tanh	54	6	32	-0.4	13.1
	tanh	54	8	32	-0.6	13.0
	tanh	54	10	32	-0.8	12.9

Table 6.2: Neural Network Training Results - B.

From these tables it can be seen that most of the errors have a small negative mean, and an SD of around 13 pulses. It is not known why they all have small negative means, instead of being randomly distributed around zero. One of the problems with using neural networks is that they behave very much as a black box. Very little is known about what is happening inside the network, which is the focus of a large amount of neural network research at the current time. An error of 13 pulses, although being quite large at 40% of the entire ± 60 pulse range, is probably acceptable as a first approximation. It means that when large changes are required to the pressure distribution, the updates should behave acceptably. ie. When a large number of pulses are required (say +30 pulses), a moderate to large number are sent (one SD is at 17 pulses).

In choosing which model would be used as the first attempt to polish using a neural network, clearly the models that use a logistic activation function are better the those that use tanh. The model with the lowest error is that with 54 inputs (ie. 19 active load cells, x, y, ϕ and 32 actuator positions) and 10 hidden units. This model, when trained with learning parameters 0.05 - 0.01 - 0.05, has an error with SD 12.4. However, the model without the 32 actuator position has a SD of 12.6. The difference between these is so small, that it was decided to use the latter. Since this has far fewer input units, the amount of cpu time required to run this model will be far less.

Before polishing, it was useful to see whether using the neural network for real-time updates works with the lap stationary. When the lap had pressed, the uniform pressure distribution was disturbed by moving the three rings of actuators. The updates were then configured to try to restore the uniform pressure distribution. Two representative graphs, indicating how the updates reduced the RMS and P-V errors defined in chapter 4, are shown in figure 6.7. Table 6.3 shows the actuator setting for the two experiments. These results were very encouraging. With only a first approximation network, the updates were able to significantly reduce the errors, with a single iteration. However, for actual polishing it is likely the output from the neural network will need to reduce these error far more than these graphs indicate the current, first attempt, is able to achieve.

A second polishing run under the exact same conditions as the first was then carried out. This time, however, the neural network was used to perform the real-time updates. Purely to demonstrate whether it was able to control the pressure distribution, the updates were configured so as to attempt to keep the pressure distribution constant over the mirror. As before, how well this is achieved is measured by the

	Actuator Positions (no. of pulses)		
	Inner Ring	Middle Ring	Outer Ring
Expt 1	-30	-60	-90
Expt 2	+30	0	-30

Table 6.3: Actuator Setting for Static Neural Network Expts.

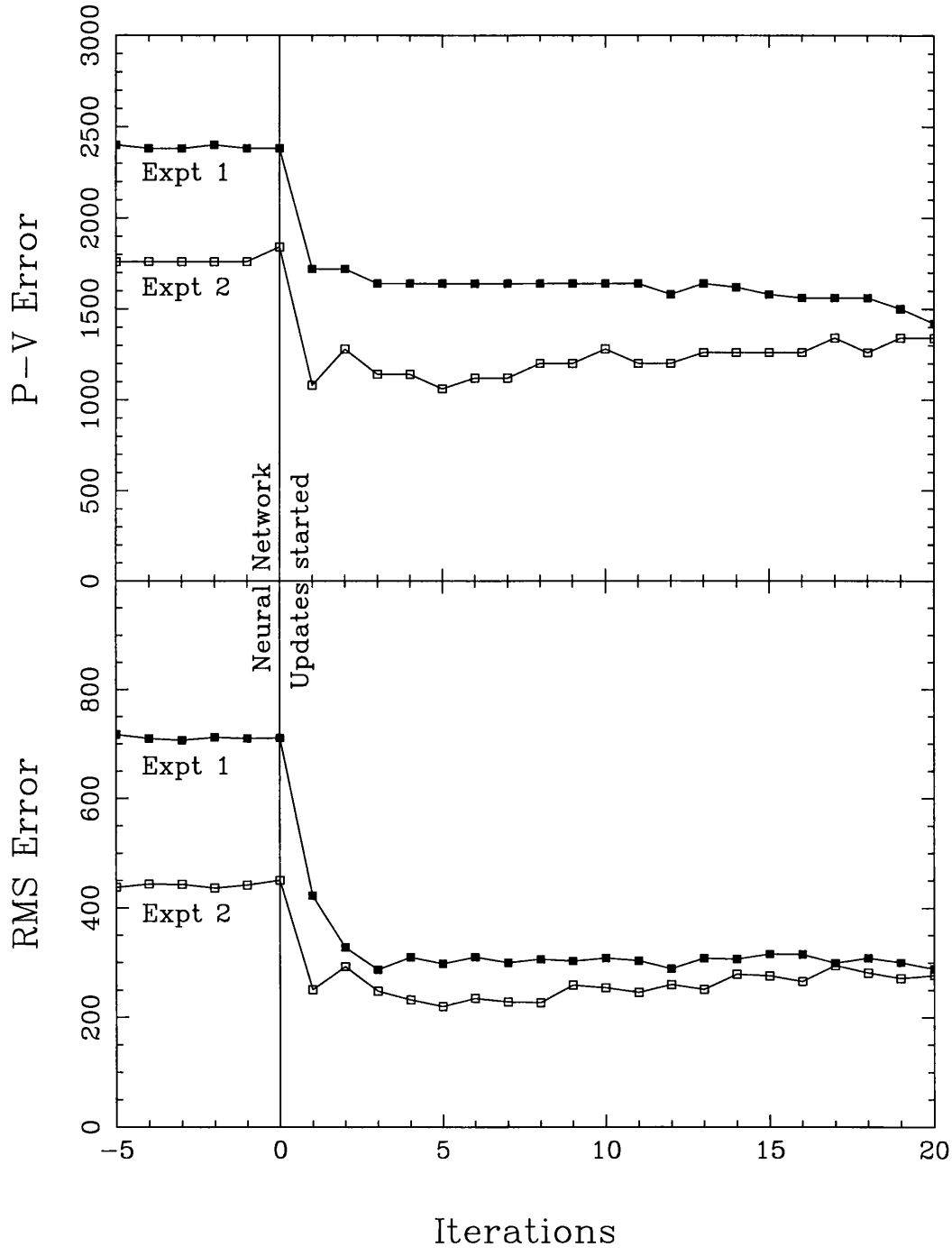


Figure 6.7: Neural Network Expts with a Static Lap.

Update algorithm.	RMS Error	P-V Error
None	529.0 ± 2.0	1773 ± 7
Random	725.5 ± 3.4	2295 ± 13
Neural Network	665.3 ± 4.9	2324 ± 19

Table 6.4: Results from Real-time Update Experiments.

RMS and P-V load cell readings.

The results from this experiment are shown in table 6.4

Therefore, this first neural network has resulted in an 8.3% improvement on the RMS error, when compared to the random updates. Within experimental error, there is no change in the P-V error.

This should tentatively be described as an encouraging result. Although still worse than no updates, it is a significant improvement on the random updates, and augurs well that subsequent networks should behave as expected.

Unfortunately, within the time scale of this thesis, there has not been sufficient time to continue the process, and train another neural network using the data from this second polishing run. The updates for the second polishing run were able to control the pressure distribution better than the first. The training data resulting from it will therefore give better coverage of the desired parts of the parameter domain. This will mean that the next neural network that is trained on this better data should be a further improvement.

Another longer term refinement to the experiment is also needed. The Active Lap is required to control the ablation of the mirror. Firstly it needs to be demonstrated that the neural network is able to adequately maintain a flat pressure distribution. When it can do this, the experiment should be modified so that it makes use of the ablation algorithm. It will attempt to maintain a pressure distribution that will result in a flat integrated ablation distribution. If this can be achieved, then the final step will be to maintain a pressure distribution that will result in the integrated ablation being such that it removes the errors in the mirror profile.

As detailed in the next chapter, continuing investigations into using neural networks to control the Active Lap will be the focus of further research in the near future.

Chapter 7

Conclusion

Significant progress in polishing large optics using the Active Lap has been demonstrated in this thesis.

- The data acquisition and control software has been written and successfully tested.
- The Active Lap hardware systems have been calibrated and tested, with the load cells, lap actuator and global force actuator systems all working well.
- After modifications, the lap as a whole has been shown to behave as expected in the static situation.

The most important aspect of the lap which needs further developments is performing the real-time feedback loop. Implementing this using an algorithmic approach has been shown to be impractical, and although the application of neural network technology has shown encouraging signs of being a viable technique, it still needs further investigation.

However, as mentioned in chapter 4, the lap currently works extremely well in its semi-active mode. In this it pulls up where the lap overhangs the mirror, which eliminates the turning down of the edge of the mirror, as is the case with passive polishing. Furthermore, it has been demonstrated that the lap can be configured so that there is an area of high pressure at the centre of the mirror. The mirror could then be figured by moving the position of this "hot spot" with the global force

actuators. This should prove to be an invaluable aspect of the semi-active mode. It would facilitate local figuring of subsections of the mirror, without introducing the high spatial frequency ripples conventional sub-diameter tools produce.

7.1 Future Developments

7.1.1 The Immediate Future

An exciting development concerning the Active Lap project is that the Optical Science Laboratory has recently signed a formal collaboration with the National Physical Laboratory and Optical Surfaces Ltd. to respond jointly to the tender to produce the GEMINI secondary mirrors. The preliminary enquiry document has been received from GEMINI and a response from the consortium returned. It is anticipated that the tender responses will be required in February-March 1995. Now that OSL is also in receipt of a new industrially related research grant, there is finance to continue development of the Active Lap, and we are optimistic that we shall be in a strong position to respond to the GEMINI opportunity.

Clearly time is short, and therefore future developments will be carefully targeted at completing the prototype secondary. The strategy which it is intended to adopt is as follows:

1. Use the lap in its semi-active mode for local (zonal) figuring of the mirror. The lap will be configured to produce a convex pressure distribution with respect to the mirror, with a central "hot spot". Local figuring will then be achieved by using the dynamic global force actuator update routines to pull up where the lap overhangs the mirror, by different multiples of the weight of the overhang. These multiples are for now manually selected by the user.

2. Use the lap with a uniform pressure distribution, a very short stroke and lifting up the overhang. This will be used for smoothing the mirror.

7.1.2 Longer Term Developments

There are several aspects of the project that require further investigation before the lap is able to work as a fully active tool. These will be undertaken as longer term, background tasks, secondary to the strategy outlined above.

The following sections detail some of these developments.

1. As mentioned above, semi-active local figuring of the mirror should prove to be an extremely important aspect of the polishing process: indeed it has already been demonstrated that the lap is able to move a "hot spot" of pressure around the surface of the mirror.

   Although perfectly usable in its current form, where the user manually selects the multiple of the overhang to use, it would be advantageous if this were automated. Using data from previous polishing runs, a table of the changes to the mirror profile for different multiples could easily be assembled. This could then be used in conjunction with the current mirror profile, to develop software algorithms that select the appropriate multiple, as well as specifying appropriate polishing times etc.

2. Despite showing encouraging signs of being a useable technique, the neural network approach to performing the real-time updates is at the present time not good enough to be used for polishing a mirror.

   The reason for this is thought to be as follows: The training data which the network uses is not sufficiently representative of the two parameter domains between which the network maps. However, it is likely that further iterations of the process of acquiring increasingly more realistic training data, and maturing the neural network, will provide a useable tool.

3. As detailed in chapter 5, finding a physically meaningful interpolation algorithm has proved difficult, probably because the spacing between the load cells is larger than optimum.

   Further investigations into the form of the algorithm need to be undertaken, as does the possibility of decreasing the spaceing between the load cells. A possible solution to the latter would be to populate the dummy load cells in

the current lap, which would immediately decrease the spacing by a factor of 2. The major problem with this is that at the current time it would be prohibitively expensive. It would also take the lap out of use for longer than the current project schedule will allow.

Looking further ahead, once the lap is able to maintain a predefined pressure distribution on the mirror during polishing, there are three final areas that need investigation.

- Although it has not been within the scope of this thesis, that the optical testing of the $\frac{1}{3}$ scale model of the f/7 2.5m hyperbolic GEMINI secondary mirror that the Active Lap is working on has proved problematical, and is the focus of current research at the OSL. [18]
- Differences between the optical tests after subsequent polishing runs need to be analysed to find the correct form of the interpolation algorithm.
- Error maps from the optical tests need to be analysed, and fed into the Active Lap software as target pressure maps. This would be the final stage of closing the polishing/testing loop.

7.1.3 Concluding Remarks

I have been extremely fortunate that the work detailed in this thesis has been part of a fascinating and challenging research project, and that I have been involved in an international collaboration of the highest academic stature.

Almost inevitably with a project of such complexity there have been unforeseen technical difficulties that have had to be overcome. Although these may have delayed the working of a fully *active* lap, we now have a useable polishing tool that for the first time ever measures, and uses, the polishing parameters in real-time. This should prove to be is a significant and lasting contribution to the field of optics production.

Although primarily focussed towards producing Astronomical mirrors, it should also be noted that there are other scientific groups that require large optics. e.g. The current experiments into producing power via nuclear fusion use lasers that require

optics of up to approximately 35cm. If these were to be scaled up to power station size, then they would require 1-2m optics with very good stray light performances. The Active Lap may be an idea tool for making these.

However, in the immediate future we look towards producing Astronomical mirrors. With the GEMINI tender rapidly approaching, and others on the horizon, the Active Lap should have an extremely bright future.

Appendix A

Mathematical Derivations

Several mathematical formulas have been quoted in the text of this thesis. The derivations of the non-trivial ones are detailed below.

A.1 Coordinate Transformations

In order to produce graphical displays that the user is able to interpret, two coordinate transformations are required.

- Transformation of the values read by the two position encoders to x and y displacements.
- Transformation from the pressure distribution sampled in the rest frame of the lap, to display the rest frame of the mirror on the computer monitor.

A.1.1 Coordinate Transformation of the Position Encoder Readings

In order to calculate the coordinate transformation required to map the position encoders to x and y displacements, the following calculations are required.

Referring to figure A.1, the two position encoders are situated at points **A** and **B**, with the centre of the lap at **C**. The distance between the encoders, D, has been measured to be 1244mm, and the encoders measure the lengths a and b.

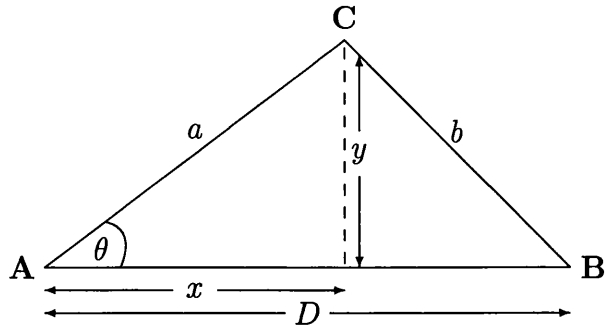


Figure A.1: Coordinate transformation for position encoders.

Since $x = a\cos\theta$ and using the Cosine Rule $b^2 = a^2 + D^2 - 2aD\cos\theta$, we obtain

$$x = \frac{a^2 - b^2 + D^2}{2D}$$

then, applying Pythagoras' Theorem

$$y = (a^2 - x^2)^{\frac{1}{2}}$$

These displacements are required in order to perform the transformation from lap to mirror coordinates detailed below and in finding the total force applied to the lap by the polishing arms.

A.1.2 Coordinate Transformation from Lap to Mirror Coordinates

A coordinate transformation from the rest frame of the lap to the rest frame of the mirror is important so that the user can clearly see exactly what is happening to the mirror. Since the load cells are physically situated in the lap, the pressure distribution is measured in the rest frame of the lap. If the pressure map, or any other of the graphical displays, is shown in anything other than the rest frame of the mirror, it would be extremely difficult for the user to determine the effects that polishing was having *on the mirror.*

A further complication also arrises from the fact that a circular display is required on the computer monitor. As detailed in section 3.4.1.3, the display circle is essentially an ellipse a with major axis of 35 units, minor axis of 30 units.

The problem is therefore to transform a data value at an arbitrary point in a circular distribution in the rest frame of the lap, to a point in an elliptical distribution in the rest frame of the mirror.

In order to accomplish this, three translations are required. These are due to the independent motions in the lap/mirror system. With respect to the laboratory rest frame, these are

- Rotation of the mirror.
- Linear translation of the lap.
- Rotation of the lap.

Incorporated into the rotations is a forth translation, which converts from a circular to an elliptical data distribution.

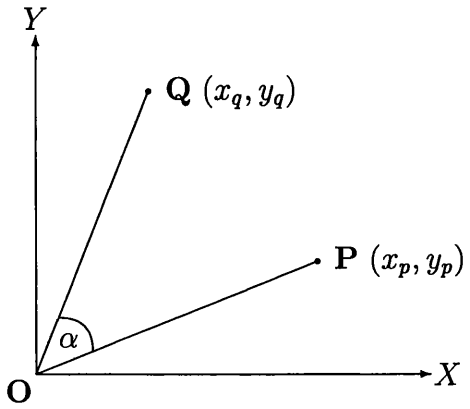


Figure A.2: Rotational coordinate transformation.

Referring to figure A.2, a general rotation of α degrees from **P** to **Q** is accomplished by the following transformations [30]

$$x_q = x_p \cos\alpha - y_p \sin\alpha \tag{A.1}$$

$$y_q = x_p \sin\alpha + y_p \cos\alpha \tag{A.2}$$

The coordinate system of the displayed graphical map is shown in figure A.3. To translate from the rest frame of the lap to the circular display on the computer screen, we need to know where the point (a, b) on the screen is situated in the rest frame of the lap.

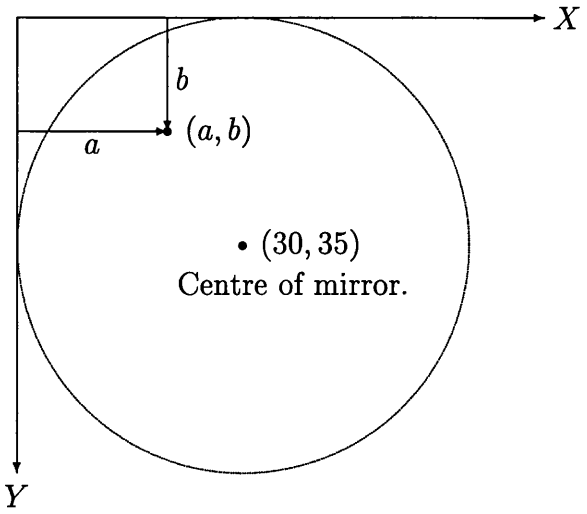


Figure A.3: Coordinate system of displayed graphics map.

In order to perform one rotational coordinate transformation of the displayed graphics map, the following procedure must be carried out.

1. A linear transformation so that the centre of the displayed circle is at the origin. This results in the point (a, b) being translated to $(a-30, b-35)$. This will be written as

$$\begin{aligned} a &\rightarrow (a-30) \\ b &\rightarrow (b-35) \end{aligned}$$

2. A scale reduction in the X and Y directions to translate the ellipse into a unit circle.

$$\begin{aligned} a &\rightarrow (a-30)/30 \\ b &\rightarrow (b-35)/35 \end{aligned}$$

3. A rotation by ϵ degrees using equations A.1 and A.2.

$$\begin{aligned} a &\rightarrow \{(b-35)/35\}\sin\epsilon + \{(a-30)/30\}\cos\epsilon \\ b &\rightarrow \{(b-35)/35\}\cos\epsilon - \{(a-30)/30\}\sin\epsilon \end{aligned}$$

4. An increase of the scale to convert back to the ellipse.

$$a \rightarrow \{30(b-35)/35\}\sin\epsilon + (a-30)\cos\epsilon$$
$$b \rightarrow (b-35)\cos\epsilon - \{35(a-30)/30\}\sin\epsilon$$

5. A linear translation to restore the centre of the circle to its initial position.

$$a \rightarrow \{30(b-35)/35\}\sin\epsilon + (a-30)\cos\epsilon + 30$$
$$b \rightarrow (b-35)\cos\epsilon - \{35(a-30)/30\}\sin\epsilon + 35$$

To perform the complete coordinate transformation, the three independent motions of the lap/mirror system must be allow for as mentioned above.

If the rotation of the mirror is θ, the rotation of the lap is ϕ, and the translations of the centre of the lap with respect to the centre of the mirror are x and y as detailed in section A.1.1, then the following equations will map a point (L_x, L_y) on the lap to a point (M_x^S, M_y^S) on the screen display of the mirror. (Note the signs of the translations to allow for the different orientations of the coordinate frames.)

$$M_x^S = \{30(T_y-35)/35\}\sin\theta + (T_x-30)\cos\theta + 30 \quad \text{(A.3)}$$
$$M_y^S = (T_y-35)\cos\theta - \{35(T_x-30)/30\}\sin\theta + 35 \quad \text{(A.4)}$$

where T_x and T_y are temporary variables given by

$$T_x = \{30(L_y-35)/35\}\sin\phi + (L_x-30)\cos\phi + 30 + y$$
$$T_y = (L_y-35)\cos\phi - \{35(L_x-30)/30\}\sin\phi + 35 - x$$

Since the values of θ and ϕ are read from the ADC card (see section 3.9.2) they are not continuous variables. Therefore these equations can be simplified so as to allow the use of pre-computed LUTs in order to reduce the cpu time used. These simplified forms are

$$M_x^S = (L_x-30)K_x^0 + (L_x-35)K_x^1 + J_x \quad \text{(A.5)}$$
$$M_y^S = (L_y-30)K_y^0 + (L_y-35)K_y^1 + J_y \quad \text{(A.6)}$$

where

$$J_x = xK_x^3 + yK_x^4 + 30$$
$$J_y = xK_y^3 + yK_y^4 + 35$$

The coefficients K_i^j are functions of θ and ϕ only. They can therefore be calculated during the software initialization for all values of θ and ϕ and stored as LUTs. K_i^j are found from the following equations.

$$\begin{aligned} K_x^0 &= \cos(\theta+\phi) \\ K_x^1 &= 30\sin(\theta+\phi)/35 \\ K_x^3 &= -30\sin\theta/35 \\ K_x^4 &= \cos\theta \end{aligned}$$

$$\begin{aligned} K_y^0 &= -35\sin(\theta+\phi)/30 \\ K_y^1 &= \cos(\theta+\phi) \\ K_y^3 &= -\cos\theta \\ K_y^4 &= -35\sin\theta/30 \end{aligned}$$

Since J_i depends on the x and y translations detailed in section A.1.1 as well as θ and ϕ, if they were to be stored as LUTs, they would have to be stored as 4 dimensional LUTs. There are 256 possible readings for each of the four parameters, so this LUT would have 256^4 ($= 4.3 \times 10^9$) elements. Therefore, since even storing one byte per element would make this too large to be feasible (4096 MBytes) the J_i values have to be calculated in real-time. However, the cpu time used to do this can be minimized by performing the calculation once per frame and storing the values in memory. Then, when performing the coordinate transformation, this value can be accessed, rather than calculated, for each pixel in the display.

A.2 Total force in Polishing Arms

Figure A.4 shows a schematic of the polishing machine, indicating the relative positions of the forces in the left and right polishing arms, which are $\mathbf{F_l}$ and $\mathbf{F_r}$ respectively. Additionally, S_l and S_r are the strokes of the left and right polishing arms, D_l and D_r are the lengths of the arms (measured as 1060mm and 1080 mm respectively) and L is the distance between the centres of the two polishing arm turn-tables (measured as 1500mm). O_L is the centre of the lap.

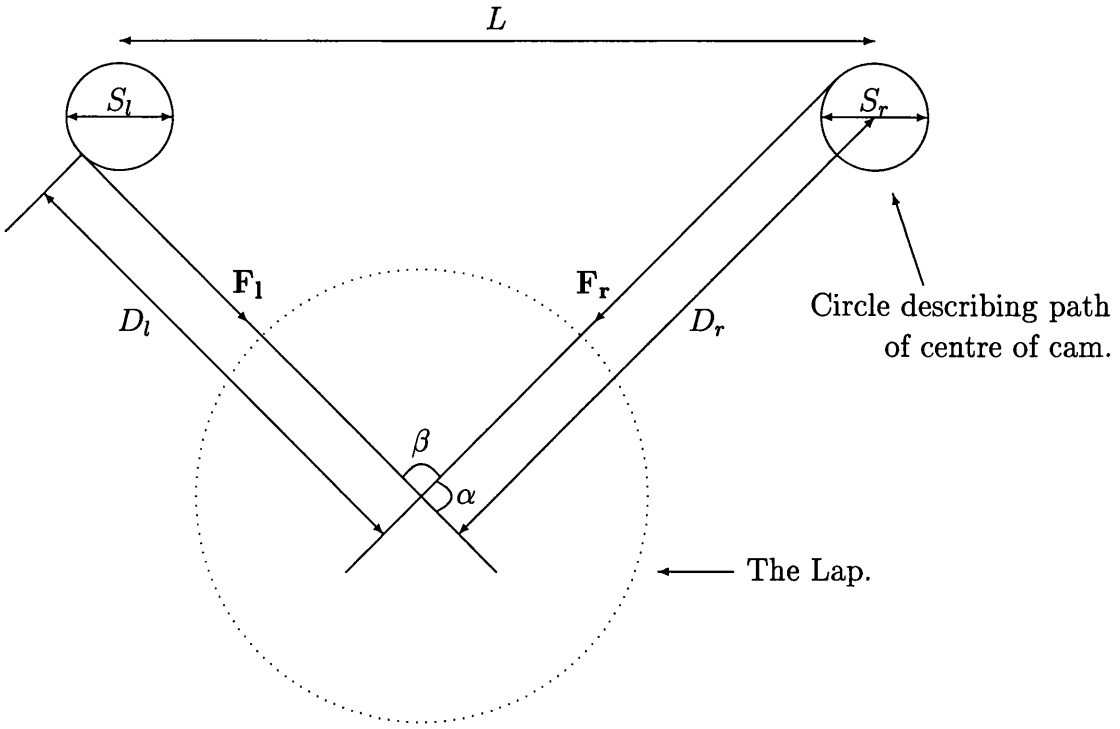


Figure A.4: Schematic Diagram of Polishing Machine.

The total force applied to the lap by the polishing arms is the magnitude of the resultant force, $|\mathbf{R}|$. Referring to figure A.5 and using the Cosine Rule, we obtain

$$|\mathbf{R}| = \left\{ |\mathbf{F_l}|^2 + |\mathbf{F_r}|^2 - 2|\mathbf{F_l}||\mathbf{F_r}| \cos\alpha \right\}^{\frac{1}{2}} \tag{A.7}$$

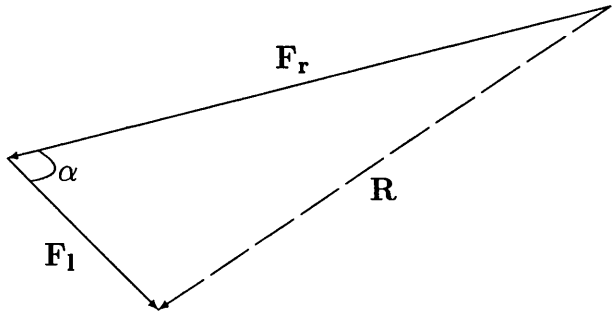


Figure A.5: Resultant of Forces Applied to Polishing Arms.

$|\mathbf{F_l}|$ and $|\mathbf{F_r}|$ are measured by the load cells in the polishing arms, it is therefore necessary to calculate α. This can be achieved using the following method.

1. From the two position encoders, the coordinates of the centre of the lap can be calculated (see section A.1.1). Note that the centre of the lap is where the two polishing arms intersect.

2. Knowing the position of one end of the polishing arms from above, and the length of the polishing arms, two possible points on the circular path of the turn-tables can be found, where the other end of the arms are situated. These are indicated are points P and Q in figure A.6.

3. The turn-tables only rotate clockwise. Therefore whether the arms are currently at point P or Q can be determined by whether the last time $P = Q$ occurred at the maximum or minimum position.

4. Now that the coordinates of both ends of the arms are known, the angle between the arms, β, and hence $\alpha(= 180 - \beta)$ can be calculated.

A.2.1 Finding the Coordinates of the Ends of the Polishing Arms

Figure A.6 shows the intersections of the circles described by both the motion of the left hand turn-table, and the possible positions of the left hand polishing arm. The point (x_t, y_t) is the coordinate of the centre of the lap, found by calculating the x and y translations of the lap as shown in section A.1.1. (x_l, y_l) is the coordinate of the centre of the left hand turn-table.

The equations of the circles are

$$(x - x_l)^2 + (y - y_l)^2 = \left(\frac{S_l}{2}\right)^2 \tag{A.8}$$

$$(x - x_t)^2 + (y - y_t)^2 = {D_l}^2 \tag{A.9}$$

Solving these equations for y, it can be shown that

$$y = \frac{\left(\frac{{S_l}^2}{4} - {D_l}^2\right) + ({x_t}^2 - {x_l}^2) + ({y_t}^2 - {y_l}^2) - 2(x_t - x_l)\,x}{2(y_t - y_l)}$$

This can then be written as

$$y = Jx + K \tag{A.10}$$

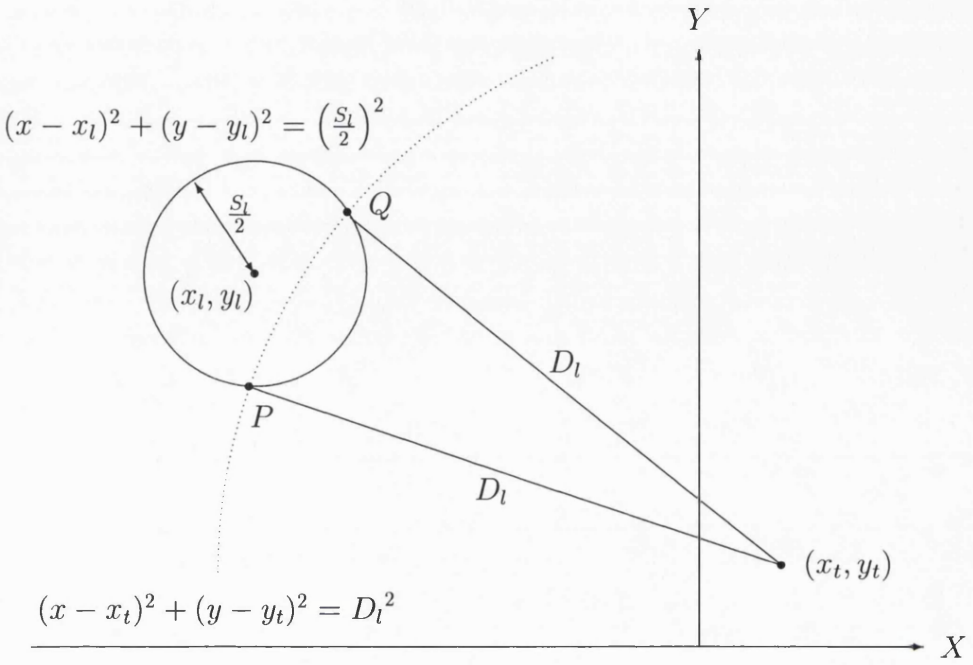


Figure A.6: Coordinates of Ends of Polishing arms.

where J and K are given by

$$J = -\frac{(x_t - x_l)}{(y_t - y_l)}$$

$$K = \frac{\left(\frac{S_l{}^2}{4} - D_l{}^2\right) + (x_t{}^2 - x_l{}^2) + (y_t{}^2 - y_l{}^2)}{2\,(y_t - y_l)}$$

Substituting back into equation A.8, we obtain

$$x^2\left\{1 + J^2\right\} + x\left\{2J\,(K - y_l) - 2x_l\right\} + \left\{x_l{}^2 + (K - y_l)^2 - \frac{S_l{}^2}{4}\right\} = 0 \qquad \text{(A.11)}$$

Thus the x coordinates of points P and Q can be found using the standard formula for solving quadratic equations, and their corresponding y values found using equation A.10.

The same equations are used to find the points corresponding to P and Q on the right hand turn-table, but using x_r, y_r, S_r and D_r.

A.2.2 Finding the Angle Between the Polishing Arms

Once it has been determined whether the polishing arm is currently at point P or Q, as enumerated above, the following procedure is required to find the angle between the polishing arms.

Referring to figure A.7, points (p_l, q_l) and (p_r, q_r) are the coordinates of the ends of the polishing arms attached to the left and right turn-tables, and (p_t, q_t) is the coordinate of the centre of the lap.

L and **R** are vectors from the centre of the lap along the polishing arms. If **i** and **j** are unit vectors along the x and y axis respectively, then **L** and **R** are given by

$$\mathbf{L} = (p_l - p_t)\,\mathbf{i} + (q_l - q_t)\,\mathbf{j} \tag{A.12}$$

$$\mathbf{R} = (p_r - p_t)\,\mathbf{i} + (q_r - q_t)\,\mathbf{j} \tag{A.13}$$

From these we obtain

$$\mathbf{L} \cdot \mathbf{R} = (p_l - p_t)(p_r - p_t) + (q_l - q_t)(q_r - q_t) \tag{A.14}$$

Now, if for two arbitrary indices A and B we define $P_{AB} = (p_A - p_B)$ and $Q_{AB} = (q_A - q_B)$, then using $\mathbf{L} \cdot \mathbf{R} = |\mathbf{L}||\mathbf{R}| \cos\beta$, where $|\mathbf{L}| = D_l$ and $|\mathbf{R}| = D_r$, it can be shown that

$$\beta = \arccos\left[\frac{P_{LT}P_{RT} + Q_{LT}Q_{RT}}{D_l \cdot D_r}\right] \tag{A.15}$$

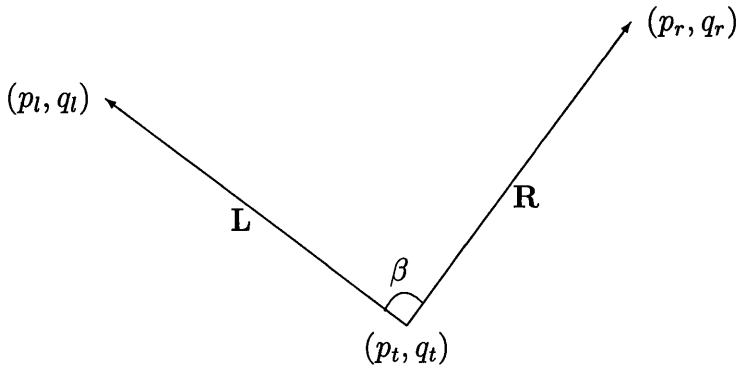


Figure A.7: Find the Angle Between Two Vectors.

A.3 Area of Lap Overhanging Mirror

Figure A.8 shows the centre of the lap, O_L, displaced from the centre of the mirror, O_M by a distance D, with the radius of both the lap and the mirror being r. Now,

$$\cos\left(\frac{\theta}{2}\right) = \left(\frac{D}{2r}\right)$$

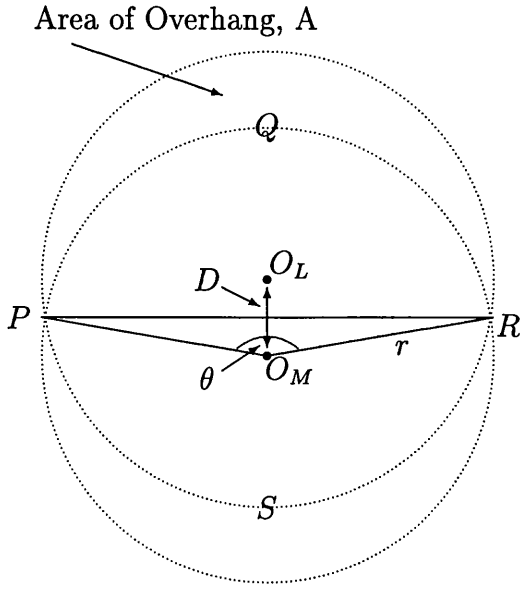


Figure A.8: Area of Lap Overhanging Mirror.

therefore

$$\theta = 2 \arccos\left(\frac{D}{2r}\right) \tag{A.16}$$

If the area of the sector $PQRO_M$ is $A(PQRO_M)$ and using equation A.16, we obtain

$$A(PQRO_M) = \left(\frac{\theta}{2\pi}\right)\pi r^2 \tag{A.17}$$

$$= r^2 \arccos\left(\frac{D}{2r}\right) \tag{A.18}$$

Using the same notation, the area of the triangle PRO_M is given by

$$A(PRO_M) = \frac{1}{2}\left\{2\left[r^2 - \left(\frac{D}{2}\right)^2\right]^{\frac{1}{2}} \cdot \frac{D}{2}\right\} \tag{A.19}$$

$$= \left[r^2 - \frac{D^2}{4}\right]^{\frac{1}{2}} \cdot \frac{D}{2} \tag{A.20}$$

The area of intersection of the two circles, $A(PQRS)$, can be found from

$$A(PQRS) = [A(PQRO_M) + A(PSRO_L)] - [A(PRO_M) + A(PRO_L)]$$

Since $A(PQRO_M) = A(PSQO_L)$ and $A(PRO_M) = A(PRP_L)$ this reduces to

$$A(PQRS) = 2\,[A(PQRO_M) - A(PRO_M)]$$

Therefore, using equations A.18 and A.20 the area of the lap overhanging the mirror, A, is given by

$$\begin{aligned} A &= \pi r^2 - A(PQRS) \\ &= \pi r^2 - 2\left\{ r^2 \arccos\left(\frac{D}{2r}\right) - \left[r^2 - \frac{D^2}{4} \right]^{\frac{1}{2}} \cdot \frac{D}{2} \right\} \\ &= r^2 \left\{ \pi - 2\arccos\left(\frac{D}{2r}\right) \right\} + D\left(r^2 - \frac{D^2}{4} \right)^{\frac{1}{2}} \end{aligned}$$

Appendix B

An Introduction to Neural Networks

B.1 Introduction

Computing using neural networks is a relatively new, but expanding discipline. Since its origins in the 1940's it has largely been overshadowed by conventional computer techniques, but received an upsurge in popularity in the late 1980's due to new developments in the field, and enormous advances in computer hardware technology. [34]

Historically, neural networks attempted to model the biological structure of the human brain and the way it processes information. However, most current neural network architectures do not try to closely imitate their biological models, but can instead be regarded simply as a class of parallel algorithms. [38]

B.2 General Properties of Neural Networks

There are several properties of neural networks that makes them extremely attractive to use. [29]

- They are very good for modelling highly non-linear systems, which can be extremely difficult, or impossible to model using more conventional methods.

- They have the ability to learn from examples, rather than being explicitly programmed. They can therefore learn from experience in order to improve their performance and adapt their behaviour to new and changing environments.
- They are good at using past knowledge to formulate the best response to a new situation and to uncertain or noisy data. (A property known as *Generalization.*)
- They exhibit *graceful degradation* in the event of a fault in part of the network. ie. the system is able to continue to function when parts are damaged, rather than the failure of one component causing the failure of the whole machine.

Inevitably there are also some drawbacks to using neural networks.

- There are many different parameters in a neural network model that need to be assigned. (e.g. number of units, learning rates, activation functions, etc.[1]) At the present stage of our understanding of neural networks, the values that these parameters should take for a particular application are assigned by trial and error. This makes it a long process to determine the best model.
- For complicated problems, the size of a network can become enormous. This results in the network learning phase taking a very long time, often several days, which only exasperates the item above.

Despite these drawbacks, neural network models have been used very successfully across a wide variety of applications, particularly those which have until now been either impossible or commercially impractical for conventional computers to address.

B.3 Elements of a Neural Network

There are a number of different elements which comprise a neural network model, of which the basic aspects are shown in figure B.1.

There is a set of processing units, indicated by circles in the diagram. At each point in time, t, each unit u has an activation value $a(t)$. This activation is a function,

[1] See later in this appendix for explanation of these terms.

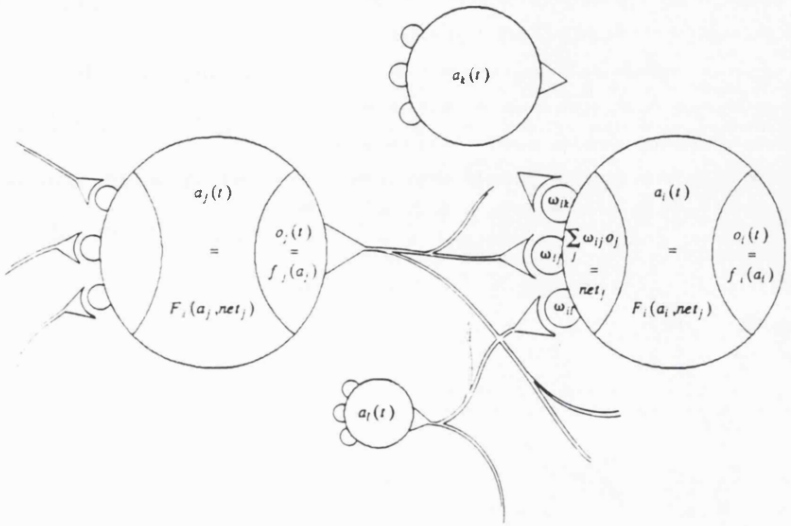


Figure B.1: The Basic Elements of a Neural Network.

F_i, of the previous activation, $a(t-1)$, and the net input to the unit, *net*. ie.

$$a_j(t) = F_i\{a_j(t-1), net_j\}$$

where a subscript attached to a parameter denotes the unit to which the parameter is associated.

This activation is passed through an output function f to produce an output value $o(t)$.

$$o_j(t) = f_j(a_j)$$

The output value from a unit passes through a set of uni-directional connections (represented by lines in the diagram) to other units. With each connection is an associated strength, or *weight*, designated w_{ij}, which determines the amount of effect the first unit has on the second.

All the inputs to a unit are then combined to form the net input,

$$net_j = \sum_j w_{ij} o_j$$

The net input is then used with the current activation of the unit to calculate the new activation value, $a_i(t+1)$, and the process continues. [32]

These systems are not static. They are dynamic in the sense that the weights

associated with the interconnections, and hence the *pattern* of connections can evolve as the network learns.

Unlike conventional computer systems, whose behaviour can be traced step-by-step with a particular piece of information being located at, e.g., a particular memory location, a neural network behaves as a "black box". It is not possible to tell how the network achieves it's results, or where a particular piece of knowledge is located. Information is distributed throughout the network, being implicitly encoded in the numerical weights between the units.

The different elements of the models are detailed in the following sections.[2]

B.3.1 Units : The Building Blocks of a Neural Network

A neural network consists of a number of elementary *units*, each of which attempts to simulate the behaviour of a single neuron in the human brain. Each unit is a simple mathematical processor, which receives one or more inputs and produces an output. Each input has an associated *weight* which determines the "intensity" of the input. The process the unit performs is to use the inputs and weights in some predefined algorithm to calculate the *activation* of the unit. The output of a unit is then calculated using another predefined function of the activation.

Within the network architecture it is useful to classify three types of units: *input, output* and *hidden.* Data is fed into the network through the input units and is sent out via the output units. Hidden units are those whose only inputs and outputs are within the network: they are therefore "hidden" from the outside world. See figure B.2.

B.3.2 The Activation of a Unit

If there are N units in a neural network model, then at any time t the state of the system is represented by a vector of N real numbers $\mathbf{a}(t)$. Each element of the vector stands for the activation of one of the units, with the activation of unit u_i at time t

[2]Except where explicitly noted, information throughout this appendix has come from a number of sources, namely [3] [9] [29] [32] [34] [38].

being designated as $a_i(t)$.

Different models make different assumptions as to how the activation is calculated, which in turn affects the properties of the network. The general form that the activation function, f_{act}, takes is

$$a_i(t+1) = f_{act}\{net_i, a_i(t), \theta_i\} \tag{B.1}$$

where $a_i(t+1)$ is the activation of unit u_i at time $t+1$, net_i is the net input to the unit and θ_i is a bias term associated with the unit.

The most commonly used activation function is the *logistic* function, which redistributes the net input to a unit into the range]0,1[. This defines the activation at time $t+1$ as

$$a(t+1) = \frac{1}{1+e^{(-net_i-\theta_i)}} \tag{B.2}$$

Another commonly used activation function is the *tanh* function, which is given by

$$a(t+1) = tanh(net_i) \tag{B.3}$$

This is useful because it is well known that the network converges quicker during learning and produces a model that has very good generalization properties. [3]

There are numerous other activation functions, some of which can be found in [38].

B.3.3 The Output of a Unit

The output of each unit is calculated as a function of the units activation, namely

$$o_i = f_{out}\{a_i(t)\} \tag{B.4}$$

where o_i is the output from unit u_i.

The exact form of the output function is less critical than the activation function, often simply taking the form of the identity function, $f_{out}(x) = x$. More common, however, is f_{out} being some kind of threshold function, so that a unit only produced an output if it's activation exceeds a certain value.

B.3.4 The Pattern of Connectivity

The basic units are connected to one another, with the outputs from some units fed to the inputs of others. It is this pattern of connectivity (known as the *network topology*) that constitutes what the system knows and determines how it will respond to any arbitrary input. The number of units in a network can range from tens to thousands.

The most popular topology in use today is shown in figure B.2. This consists of three layers of units, with each unit in a particular layer being connected to every unit in adjacent layers. This is known as a *feed–forward* network. Data flows into the network through the input layer, passes through one or more hidden layers and finally flows out of the network through the output layer, i.e. the data is simply fed forward from the input to the output layers. Since the output of any unit can be fed to the input of any other, it is obvious that many other topologies are possible. Some networks allow for backwards as well as forwards connections, or even for units to send their outputs back to themselves. The topology of the network determines its behaviour, and the application for which it is suited.

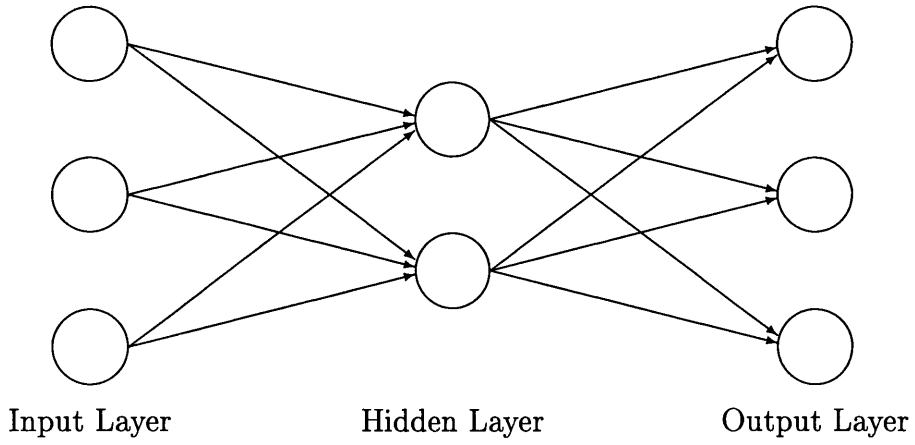


Figure B.2: Schematic Diagram of a feed Forward Network.

B.3.5 The Rule of Propagation

This describes how the output values and connection strengths are combined to produce a *net input* to a unit. The form of this is usually straightforward: if unit u_j receives inputs from n units, and the connection strength between unit u_i and u_j is w_{ij}, then net input to unit u_j is given by

$$net_j = \sum_{i=1}^{n} w_{ij} o_i \tag{B.5}$$

When complex network topologies are used, then more complicated rules of propagation are required. [32]

B.4 Learning in Neural Networks

There are two basic classes of learning algorithms that are used in neural networks: unsupervised and supervised learning.

B.4.1 Unsupervised Learning

In unsupervised learning, the network is allowed to experiment and discover for itself the required response. Allowing the network to determine its own set of rules like this is particularly suitable for cases where no model answers are available.

The network is presented with a set of input conditions, which initially produces a random output. Whether this output is judged to be good or bad is determined by a predefined merit function. If it produces a good output then the network connections are adjusted to increase the response, or vice versa for a bad output. When this process is repeated a number of times, acceptable performance may be achieved.

The advantage of this type of learning over supervised learning is that once the network is put to work, it can be allowed to continue to learn, and hence is able to adjust to changes in its environment.

B.4.2 Supervised Learning

In supervised learning, the network is presented with a number of input conditions along with the corresponding outputs that it is required to reproduce. By examining the difference between the network's outputs and these required responses, error values are calculated. The connections within the network are then adjusted so as to minimize the errors.

Largely for historical reasons, supervised learning is more often used than unsupervised. There are many different algorithms used, but the most common one is called *backpropagation.*

B.4.2.1 Backpropagation

Backpropagation is particularly well suited for training a feed–forward network. (See section B.3.4.) It is popular for three reasons

- It is relatively easy to develop a neural network using this method.
- It is readily applicable to a wide range of applications, particularly for classification and prediction.
- Most neural network development tools support this algorithm or one of its many variations.

The backpropagation weight update rule is known as the *generalised delta rule.* The algorithm used for changing the connection weights in the network following the presentation of an input/output vector pair is given by

$$\Delta w_{ij} = \eta \delta_j o_i \tag{B.6}$$

$$\delta_j = \begin{cases} f'_{act}(net_j)(t_j - o_j) & \text{if unit } j \text{ is an output unit} \\ f'_{act}(net_j) \sum_k \delta_k w_{jk} & \text{if unit } j \text{ is a hidden unit} \end{cases} \tag{B.7}$$

where Δw_{ij} is the update to be applied to the weight w_{ij}, η is the learning rate, δ_j is the error of unit u_j (i.e. the difference between the real output o_j and the target output t_j), o_i is the output of the preceding unit u_i, w_{ij} is the connection weight between u_i and u_j, and w_{jk} is the weight between u_j and the u_k, a successor to the current unit u_j. [38]

Learning using backpropagation is a *gradient descent* technique. It corresponds to performing steepest descent on the multi-dimensional error surface, where height is equal to the error measure. In this system η, the learning rate, is a constant of proportionality. [32] True gradient descent would require infinitesimally small steps to be taken, which would mean an infinitesimally small value for η. However, for practical purposes, η is chosen to be as large as is possible without causing oscillation, usually being in the range [0.1,0.9]. A common way to avoid oscillations is to modify the generalised delta rule to include a *momentum* term α, giving

$$\Delta w_{ij}(t+1) = \eta \delta_j o_i + \alpha \Delta w_{ij}(t) \tag{B.8}$$

This provides a kind of momentum in weight space that effectively filters out high frequency variations in the error surface, and therefore allows the effective weight steps to be larger. Typically, $\alpha \approx 0.9$.

A further refinement is to add a *flat spot elimination* term. This is a constant value that added to the derivative of the activation function to enable the network to pass flat spots in the error surface.

There is one major problem with backpropagation that is common to all gradient descent techniques, the problem of local minima. The problem is that of getting stuck in a "hollow" in the error surface. Since the error is at a minimum, albeit a local one, it will no longer learn. The aim of learning is to end up at the global minimum, i.e. the smallest value possible on the whole error surface. However, since it is possible to change the initial conditions of the network, i.e. the initial connection weights, it is possible to commence learning at many different positions on the error surface. Therefore, it is invariably possible to obtain a set of working parameters. However, the process of trial and error for finding these can be very time consuming. [9]

B.5 Summary

The capability of applying neural network to complex problems is extensive, particularly where conventional computing techniques are especially difficult or impossible to use, such as for highly non-linear systems. The most popular applications at

present are in classification and prediction, both of which make use of a neural network's ability to identify patterns contained within the data. The real power of a neural network lies in it's ability to learn from examples or from its environment, disassociating it from the computer programmers inabilities to analyse and solve these complex problems in sufficient depth.

Acknowledgements

I would firstly like to thank David Walker for supervising this project. The many hours of help and advise he has given me over the last four years have been invaluable, and are deeply appreciated.

I would also like to thank my colleges with whom I have had the pleasure of working on the Active Lap project on a daily basis. They have all been more than helpful. They are Dave Brooks, Sug-Whan Kim, Heshmat Jamshidi and Gil Nixon.

There are many other people from OSL and the Physics and Astronomy Department as a whole who I would like to mention, all of whom have helped me at some time or another. They are Adrian, Alan, Andy, Andy, Barry, Barry, Brian, Bruce, Dave, Ee-Eul, Francisco, Han, Keith, Lee, Mark, Martin, Mike, Nigel, Paolo, Phil, Ron, Tim, Young-Soo and Prof. Duff. I am indebted to SERC for my original studentship, and both the Physics and Astronomy Department and the Graduate School for continuing the funding.

On a personal note, I would like to thank my wife Debbie. Without her continual love and support, none of this would be possible. I owe her everything.

Finally, I would like to thank my parents for the years of help and encouragement they have given me throughout my education.

Bibliography

[1] Jean Audouze
The Cambridge Atlas of Astronomy, 2nd edition.
Chapter 2, *Astronomy Today*

[2] S. Apiki and J. Udell
Smoothing out C
Byte Magazine, Feb. 1989

[3] Ugur Bilge, Dept. of Computer Science, UCL.
Private Communication, 1994.

[4] K. Beckstette, M. Kűchel and E. Heynacher
Large Mirror Figuring and Testing
Astrophysics and Space Science, Volume 160, pp. 207-214.

[5] Philip R. Bevington
Data Reduction and Error Analysis for the Physical Sciences
McGraw-Hill, 1969

[6] Max Born & Emil Wolf
Principles of Optics
4th edition, 1970.

[7] David Brooks, OSL Optical Technician
Private Communication, 1994.

[8] R. F. Ferraro
Programmers guide to the EGA and VGA Cards
2nd ed. 1990

[9] Francisco A. Camargo
Learning Algorithms in Neural Networks.
Technical Report CUCS-062-90,
Computer Science Department, Columbia University, 1990.

[10] Gil Nixon, Optical Science Laboratory, UCL.
University of London Ph.D. Thesis, 1995. In preparation.

[11] Samuel P. Harbison and Guy L. Steele Jr.
C - A Reference Manual 2nd edition, pp.1-2
Prentice Hall Software Series, 1987.

[12] M. Iye et al.
Active Optics Experiments with a 62cm Thin Mirror.
SPIE Proceedings, Volume 1236, pp 929–938

[13] Heshmat Jamshidi, Optical Science Laboratory, UCL.
Private Communication, 1991.

[14] D. F. Horne
Optical Production Technology
1972

[15] Brian W. Kernighan and Dennis M. Ritchie
The C Programming Language
2nd edition, pp.ix–xi
Prentice Hall Software Series, 1988.

[16] S.-W. Kim
Active Production of Large Aspheric Optics for Astronomy
University of London Ph.D. Thesis, 1993.

[17] S.-W. Kim
Specifications of the Active Lap Control Software
OSL Internal Document, 1991.

[18] Y.-S. Kim
University of London Ph.D. Thesis, 1995. In preparation.

[19] D. Kleppner and R. J. Kolenkow
An Introduction to Mechanics
International Student ed. 1981

[20] Bart Kosko
Neural Networks & Fuzzy Systems:
A Dynamical Systems Approach to Machine Intelligence
Prentice Hall, 1992

[21] ed. P. G. J. Lisboa
Neural Networks – Current Applications
Chapman & Hall, 1992

[22] William Lupton
ECHWIND– *UCL Échelle Spectrograph Observation Planning*
STARLINK User Notes 53.2, 1989.

[23] David Mills
Private Communication, 1993.

[24] J. Murray
Starlink C Programming Standard
STARLINK General Paper 4.1, 1991.

[25] NAG FORTRAN LIBRARY – Mark 14, Volume 3.

[26] G. Oertel
The NOAO 8m Telescopes – I. Executive Summary and
Scientific Programs
Proposal to the National Science Foundation.

[27] G. Oertel
The NOAO 8m Telescopes – II. Technical Description
Proposal to the National Science Foundation.

[28] Patrick S. Osmer
Gemini Science Requirements
Version 1.1, November 1992.

[29] R.P. Palmer
Introduction to Artificial Neural Networks
ERA Technology Ltd., November 1993

[30] S.L. Parsonson
Pure Mathematics
Volume 1, 1977, p.242

[31] W.H. Press et al.
Numerical Recipes : the Art of Scientific Computing.
1991.

[32] D.E. Rumelhart, J.R. McClelland and the PDP Research Group
Parallel Distributed Processing: Explorations in the Microstructure of Cognition.
Volume 1: Foundations, Chapter 2.

[33] J. Texereau
How to Make a Telescope Interscience Publishers.

[34] N.R. Tubb
Embedded System Engineering
Volume 1, No. 5. October 1993, pp41–43

[35] *What Personal Computer*
Issue 44, March 1993, p.50

[36] David Walker
Private Communication. 1994.

[37] Sholom M. Weiss and Casimir A. Kulikowski
Computer Systems that Learn. 1991.

[38] Andreas Zell et al.
Stuttgart Neural Network Simulator User Manual
Version 3.2