Modeling and Algorithms for Optimizing Beam Steering Optical Crossconnects

by

Piyajit Phanaphat

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2003

Author.....
Department of Electrical Engineering and Computer Science
May 21, 2003

Certified by.....
Jungsang Kim
Lucent Bell Laboratories Technical Manager
VI-A Company Thesis Supervisor

Certified by.....
Carl J. Nuzman
Lucent Bell Laboratories Member of Technical Staff
VI-A Company Thesis Supervisor

Certified by.....
Vincent W.S. Chan
Professor of Electrical Engineering and Computer Science
M.I.T. Thesis Supervisor

Accepted by.....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Modeling and Algorithms for Optimizing Beam Steering Optical Crossconnects

by

Piyajit Phanaphat

Submitted to the Department of Electrical Engineering and Computer Science

May 21, 2003

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

Abstract

One of the most significant applications of Micro-Electromechanical Systems (MEMS) technology in optical communications today is in building large non-blocking optical crossconnects based on arrays of tiltable micro-mirrors. The complexity for these crossconnects to make all possible connections lies in the calibration or fine-tuning of the mirror tilt angles to optimize the transmissivity through each possible input/output pair. The result from the fine-tuning process that produces optimization at one point in time, however, does not guarantee optimization for future attempts. This thesis models the transmissivity as a function of control variables in the vicinity of an optimal point and uses this model to re-optimize the connections quickly when a connection is reestablished. The re-optimization algorithm achieves the goal of optimizing quickly by requiring that some prior knowledge about each connection is already known. Scalable methods for representing the per-connection transmissivity model are also studied. Experimental results of the algorithm performance on real crossconnect systems are reported, including connection setup in under 50 milliseconds.

VI-A Company Thesis Supervisor: Jungsang Kim
Title: Bell Laboratories Technical Manager

VI-A Company Thesis Supervisor: Carl J. Nuzman
Title: Bell Laboratories Member of Technical Staff

M.I.T Thesis Supervisor: Vincent W.S. Chan
Title: Joan and Irwin Jacobs Professor of Electrical Engineering and Computer Science

Acknowledgement

I would like to give my sincerest thank to my two supervisors at Bell Labs, Jungsang Kim and Carl Nuzman. Without their guidance, support, and encouragement, this thesis would not have been possible. I feel very fortunate and grateful to have had the opportunity to work with both of them.

I would like to acknowledge contributions by other members of Bell Labs, especially Alan Weiss for introducing the idea of hill shape studies for hillclimb applications. Others whose wisdom and support had made this project possible included Anatoli Olkhovets, Joe Krauss, Dong-jae Shin, and Chuck Lichtenwalner.

Researching and preparing for this thesis has been an extremely rewarding educational experience for me. In addition to my supervisors at Bell Labs, I would also like to thank my advisor at MIT, Professor Vincent Chan. His invaluable advices, constant support, and great kindness were deeply appreciated.

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1: Introduction

A typical optical communication network consists of transmission line systems that provide point-to-point links between two nodes, and switch elements at the nodes that direct traffic to its proper subsequent node. In today's optical networks, the major role of optics is to provide high-speed information pipes between the nodes. The switching function is performed in the electronic domain, after the optical signal is converted into electronic signal. Once the switching activity is performed, the electronic signal then has to be converted back into optical signal to be launched back into the fibers. The optical-electrical-optical conversion currently comprises the largest portion of network cost. As the traffic volume grows, it has been speculated that this type of switching function will be the bottleneck of the network. An alternate network architecture has been proposed, where the switching function is performed in the optical domain. In this approach, the unit of bandwidth to be switched is a wavelength in wavelength division multiplexed (WDM) transmission system, and the switch is capable of directing wavelengths of signal to their destinations. Such switch is referred to as an optical crossconnect (OXC). A schematic system diagram of an optical network is shown in Figure 1 below.

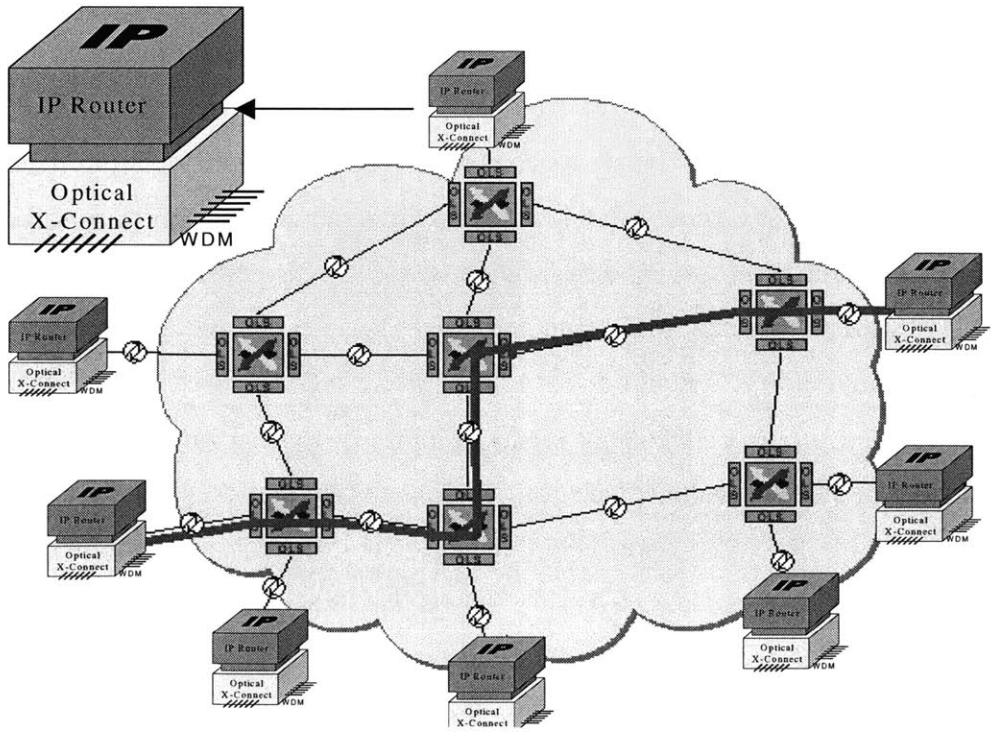


Figure 1: System diagram of an optical network

Because the OXC operates purely in the optical domain, it is transparent to data rate and signaling format. The OXC could switch optical wavelength division multiplexed signals as easily as switch single wavelength signals. With no optical-electrical-optical conversion, the optical switch provides lower cost and power reduction over current electronics solutions. These factors enable the OXC to support incremental and economical network expansion in a much more effective manner as traffic increases.

In a WDM transmission system, each fiber in the network will carry up to W wavelength channels. Suppose there are D fibers coming into the node. At the node each fiber would be demultiplexed so that each wavelength channel goes onto its own fiber, resulting in a total N = DW single-wavelength carrying fibers coming into the switch. Naturally, we could assume there are also N single-wavelength carry fibers coming out of the switch. We refer to N as the switch's port count. D is typically small, but N could potentially get large through the product of D and W. As traffic grows, a switch of large port counts will become increasingly in demand. When there are N single-wavelength fibers coming in and N single-wavelength fibers going out, there are N! input-output configurations. These N! configurations can be realized through combinations of N^2 optimal conditions of the switch. So an OXC at this node has to be able to make any of these N^2 possible connections.

One of the key building blocks that allow the concept of optical OXC to be realized is the advancement of the MEMS (Micro-Electro-Mechanical System) technology. Because of its inherent advantages such as batch fabrication technology, small size, integrability, and scalability, MEMS has shown great promise to become the dominant technology in optical crossconnect switches. Several architectural designs for the MEMS-based optical OXC switches have been proposed and evaluated in the research community, as we will discuss later. These MEMS-based designs typically use microscopic mirrors to reflect light signal coming in from an input fiber to a specified output fiber in order to make connection. The one design that stands out and proves to be the most feasible for a large port-count switch is the MEMS-based 3-dimensional (3-D) design, also known as the beam-steering design. The schematic and details of this design will be discussed in section 2.1. This 3-D design offers the great advantage of scalability and better reliability over other competing designs.

At the same time, the 3-D design also carries some inherent difficulties that need to be addressed. One such difficulty lies in the tuning of the mirrors to tilt to the precise position that would optimize the signal power going through and to maintain themselves at the optimal power position. In the manufacturing process of the switch, each of the mirrors has to be calibrated to find the precise control voltages that optimize signal power going through each of the connections involving this particular mirror. A record of all these calibration values is kept in the database for retrieval during future operations. But because changes in the environment and setup conditions could have some effect on the physical characteristic of the switch, the optimal control voltages may change over time. Mirrors thus need to be re-calibrated upon connection setup and periodically during use to make certain that they always stay close to the optimal position.

There are various scenarios in which a connection needs to be set up. One is initially setting up a primary path. Another is setting up a restoration path to recover from a failure elsewhere in the network. During operation, the time it takes to re-calibrate the mirrors or to re-optimize the connection dictates the connection's setup and restoration time. Faster restoration time allows the crossconnect to be able to support a wider variety of applications which have more stringent quality of service requirements. This factor leads to a motivation for us to find a method to re-optimize quickly. Various mathematical modeling and algorithms have been explored for this purpose. In the past, it has been difficult to find an algorithm to re-optimize connections that converges quickly enough. This thesis proposes, studies, and implements a method for accelerating the connection setup and restoration time. The study involves modeling perturbations around the optimal control voltages and making use of the model to come up with an accelerated re-optimization algorithm.

In Chapter 2, we will discuss the details of the 3-D design of the OXC switch, the source of problem that became the motivation of this thesis, and the approach we took to come up with a solution. Chapter 3 will talk about studies and mathematical modeling of the physical characteristics of the switch. In Chapter 4, we propose optimization algorithms that take advantage of the knowledge we have gained from the study on the nature of the switch from chapter 3. Chapter 5 will show the performance evaluation of

the chosen optimization algorithm on real optical OXC systems. Chapter 6 will be the conclusion of this thesis.

Chapter 2: Background

2.1 Description of the optical OXC switch

An optical connection inside a switch is a path that leads an optical signal from the source to the destination. In our case, the source is a set of input fibers and the destination is a set of output fibers. An optical OXC switch is responsible for making and maintaining multiple optical connections at a time. In NxN non-blocking optical crossconnect, an optical signal from any one of the N input fiber ports can be directed to any one of the N output fiber ports that is not being used. Such switching function can be achieved in a 3-D geometry, schematically shown in Figure 2.

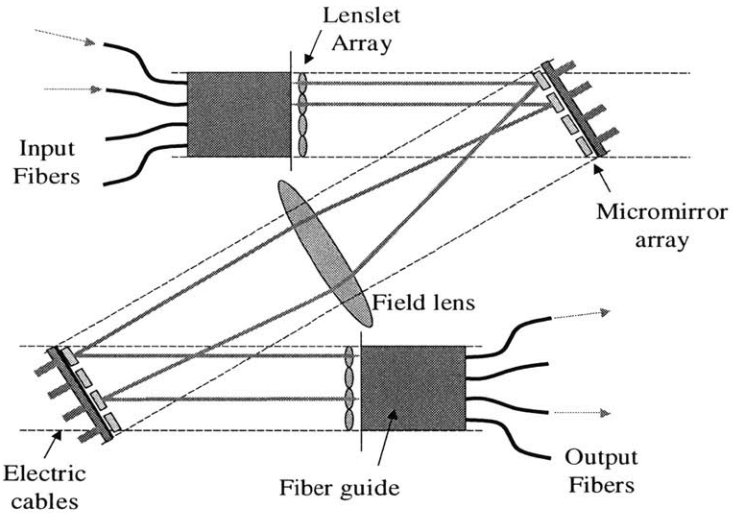


Figure 2: A top view schematic representation of a 3-dimensional optical crossconnect switch

The schematic shown above is the Optical Switch Module (OSM), which is the heart of the optical OXC switch. It is where the signals from the input fibers get routed to the appropriate output fibers according to their destination. The main components of the OSM consist of the input and output fibers, two micromirror arrays, two lenslet arrays, and one field lens. The OSM is capable of connecting signals from any input fiber to any output fiber. Each input fiber is aligned with its own input mirror, and each output fiber is aligned with its own output mirror. The input and output mirrors are fabricated as arrays

of micromirrors on two separate silicon chips. The signal path through the OSM can be traced as follows. The signal would come in from one of the input fibers. The refracted beam out of the fiber would be collimated as it goes through one of the lenses in the input lenslet array. The resulting collimated Gaussian beam is then reflected by the corresponding electronically controlled tilting mirror on the input chip which directs the beam to the spot on the field lens that would lead to the appropriate mirror on the output chip. The mirror on the output chip is set to tilt at the angle that would reflect the Gaussian beam to its corresponding output fiber. The beam is refocused as it passes through the lens in the output lenslet array and finally coupled into the output fiber.

Each micromirror is attached to two springs or flexures which enable it to tilt up and down when an appropriate force is applied, and these two springs are in turn attached to a gimbal which also has two springs attached under it to enable the gimbal and the mirror to tilt sideways. As a result, with the right combination of forces applied each micromirror is able to tilt with two degrees of freedom.

The electrostatic forces that are used to tilt the mirrors are generated by the voltages applied to the four electrodes that are positioned under each mirror and gimbal while the mirror and gimbal remain grounded at all times. The positions of the four electrodes and the springs relative to the mirror and gimbal for a typical mirror design are shown in Figure 3 below.

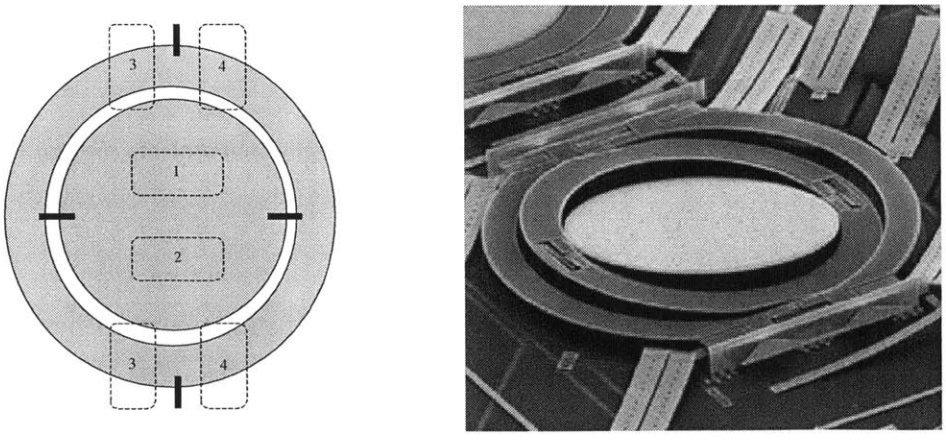
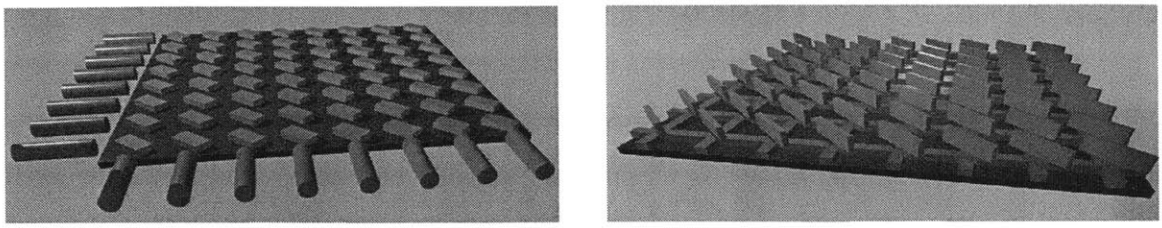


Figure 3: Micromirror and the four electrostatic electrodes

Voltage applied to electrode 3 and 4 (V_3 and V_4) moves the gimbal to the left or right. Voltage applied to electrode 1 and 2 (V_1 and V_2) moves the mirror up or down. Instead of applying voltages to two opposite electrodes simultaneously, we choose to operate in a more limited way – we apply a non-zero voltage to only one electrode and ground the other electrode (i.e. electrodes 1 and 2, or 3 and 4, are never on at the same time). The reasons behind this are that 1) this is sufficient to cover all degrees of freedom of the mirror tilt, and 2) applying high voltage to both electrodes could put too much stress on the spring. Having chosen this convention, we can describe the control variable of the mirror in each of the two dimensional directions by just one voltage value which the difference between the voltages applied to the pair of opposite electrodes in that direction. We define $V_c = V_4 - V_3$, and $V_r = V_1 - V_2$. As a result, each mirror is controlled by two voltage variables, V_c and V_r, and can be tilted to any angle within the range of ±5 degrees.

2.2 Advantage of the 3-D over 2-D switch architecture

The switch function of an all-optical OXC could be achieved by other switch architectures as well. What determine whether the design is practical are its reliability, scalability, and cost. Another prominent MEMS-based design of the switch is referred to as the 2-D switch architecture, or the crossbar design. In the 2-D architecture, the mirrors are arranged in a crossbar configuration as shown in Figure 4 below. Every possible connection has a mirror and a set of hardware dedicated to it, so for an NxN port switch, a total of N^2 mirrors are required for building a non-blocking switch. The mirrors are placed at the intersections of light paths between the input and output fiber ports. Each mirror has only two states, on position to reflect the light, and off position to let light pass.



off on

Figure 4: A 2-dimensional crossbar switching architecture

This binary or digital nature of the mirror greatly simplifies the control scheme of the switch. The control circuitry usually consists of simple transistor logic gates and some amplifiers to flip the mirrors up and down.

Regardless of its simplicity, however, the demand for so much hardware still prevents the 2-D architecture from achieving scalability. It is not practical to have N^2 mirrors for an NxN switch when N gets large. One alternative approach to increase the port count is to cascade multiple small switches in a multistage scheme. This cascaded approach, however, can get very complicated and typically requires a large number of small switches in order to afford all N^2 connections. The overall signal power loss of this cascaded switch unit is also accumulated. Another pitfall of this design is that when one mirror is defective, i.e. stuck in the on position, it can potentially be very damaging to the overall functionality of the switch. Not only will it ruin the usability of that particular connection, it could also affect all other connections involving either that input channel or the output channel.

The 3-D architecture as shown in Figure 2, on the other hand, uses a different approach to make connections. The advantage of this switch architecture is that the number of hardware components needed is linearly proportional to the port count of the crossconnect switch. Only 2N mirrors are required for an NxN switch. So, hardware-wise, this seems to be a very scalable approach. In addition, when there is a malfunction in any mirror, only the N connections involving that mirror will be affected.

In this 3-D design, the mirrors are analog and free to tilt to any angle within a small range. The complexity of the crossconnect switch to make all possible NxN connections thus lies in the analog fine-tuning of the mirror tilt angles so that the light

signal through the connection is optimized. Since each connection involves controlling both the input and output mirrors and each mirror has two control voltages, making a successful connection requires optimizing optical throughput by controlling four control voltages. The reliability and effectiveness of this complex analog control scheme is the challenge we need to overcome in order to make this switch design more practical for real world use.

2.3 Statement of the problem

During the switch manufacturing process, the precise values of the four voltages that maximize the amount of optical signal power through each connection are measured and stored in a database. This process of finding these voltage values is called the *training process*. Once the database is generated, one can retrieve the appropriate voltage values from the database and apply them to the electrodes in order to make any connection. Since the training process becomes increasingly challenging as the number of ports grows in the optical switch, it should be done only once in the factory with the hope that the resulting values stored in the database will be accurate and valid for all future uses.

In practice, there are many reasons why the voltage values that optimize the optical throughput of a connection during real operation of the switch may deviate from those measured at the time of training. If the voltage sources used during training process and the operation are different, there can be slight calibration errors between them. If the operating temperature of the switch is different from the temperature during training, the overall optical assembly shown in Figure 2 can expand or contract and the optimum voltages can change. Also, if the wavelength of the light used during operation is different from that used during training, the chromatic dispersion of optical components can slightly modify the optimum voltages. For all these reasons, there arise cases where one has to further optimize the connection after the voltage values from the database is applied, before the connection can be declared good. We need to come up with an accelerated method or algorithm to handle the case where the re-optimization is required.

If the re-optimization process is required, the number of steps needed for the re-optimization process determines the switching time of the optical crossconnect switch. This switching time, which is also known as the switch's restoration time, is a significant factor that determines the functionality of the switch in the network. The electronic switch used in today's network has a restoration time of <= 50 milliseconds. For the optical switch to find a real niche in the network and be well accepted as a replacement of the electronic switch, it is crucial for the optical switch to further minimize its connection setup time requirement. The current technique used in re-optimizing process of the optical switch requires over 100 ms most of the time, and it is clearly not good enough. Therefore, the focus of this new re-optimization method should be to reduce the number of measurement steps and time required for retraining to the least possible. Even if this comes at the expense of having to store more information, as long as the storage requirement does not prevent scalability of the algorithm usage, it is worthwhile for us to focus our best effort on reducing the number of measurement steps required.

2.4 Approach

In order to come up with effective control algorithm, it is important to have good knowledge about the loss characteristics of each connection in the vicinity of the optimum optical throughput spot. If one plots the transmissivity through a particular connection as a function of the four control voltages near the optimum point, the result is a four dimensional function we refer to as a hill, with the optimum throughput spot being the peak of the hill. Hill shapes describe how much loss is introduced when the applied voltages deviate from the optimum control voltage values. A good knowledge of the hill shapes is the key to developing efficient re-optimizing process, which reduces the switching time. Such efficient re-optimization process can also affect the demands on the electrical hardware, in this case meaning a good knowledge of the hill shape may also allow some relaxation of the specifications on the voltage drivers and other parts of the switch hardware.

The approach for this study is divided into three steps. First, we will make measurements on prototype devices, study hill shapes, and systematically look at the

statistics of the hill shapes across the switch. Second, we will apply the knowledge from studying the hill shape to come up with an accelerated re-optimization algorithm and some variations of it that are robust and practical for real world use. After we accomplish the first two steps, we will then move on to the third step, which is to implement the chosen algorithm on prototype systems.

2.5 Prototype systems

In addition to an OSM, a prototype system also has to include other components to control the mirrors and to detect the optical power going through the switch. The two main boards in the system are what we call the high voltage digital-to-analog converter (HVDAC) board, and the power monitoring board. The HVDAC board contains a digital to analog converter in order to convert the digital signal coming in from the main processor into an analog voltage value to control the mirror tilt. The interface between the HVDAC board and the OSM is thus in the electrical domain. The interface between the OSM and the power monitoring board is, on the other hand, in the optical domain. The power monitoring board has tap devices to read the optical power coming into the connection and going out from the connection in order to determine how much power is lost inside. Photo-detectors and amplifiers are used on the power monitoring board to convert detected photons into electric current and amplify the current to produce the power reading. This optical power reading will then go through an analog-to-digital converter (ADC) on the same board in order to be sent back as digital signal to the main processor. Because of these circuitry components outside the OSM, we are able to just supply the values of control variables and get the output readings on prototype systems. The system diagram of the switch is shown in Figure 5.

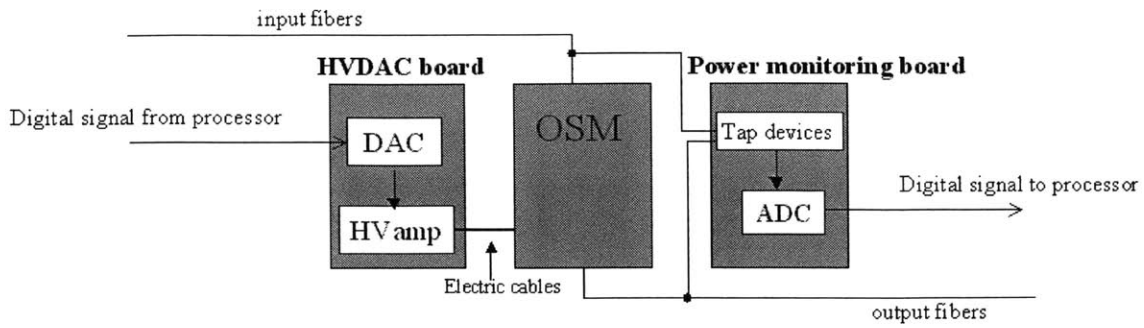


Figure 5: System diagram of the OXC

Prototype systems used in this study are of two different sizes. One is a switch of the size 1296x1296 while the other is of the size 64x64. The 1296 switch is roughly 20 times bigger in terms of the number of micromirrors on each mirror array, and in terms of number of connections, it is more than 400 times bigger. This means that there are 400 times more optimal positions and optimizations that need to be performed. The bigger size comes with many additional complications and issues. It is a good setting for us to test the scalability of the switch design and its control scheme. The arising issues associated with bigger switch fabrics and their effects on the performance of our algorithm will be discussed when we talk about performance evaluation on real systems in Chapter 5.

Aside from the size difference, the two prototype systems have different system architectures and maturities. The 1296 OSM line has been developed and gone through many manufacturing iterations. More than a dozen of the OSM's of this size have been produced, and many improvements have been made. The switch is therefore by nature more reliable and well understood. In addition, the test-bed and the hardware-controlling software have been well set up and implemented. Complete sets of data for several 1296 OSM's were available and this has helped us tremendously in terms of trying to understand the behavior of the switch. All the studies in this thesis that involve simulations will be based on the data from the 1296 OSM. Simulations based on these data is of a tremendous use to us since they give us the opportunity to do preliminary evaluation of our models and algorithms without requiring any real-time activity or any real-world complications. All the discussions in Chapters 3 and 4 are based on

preliminary evaluation done on the 1296 OSM's measurement based simulations. Once we have gained insights from the simplified setting of the simulations, we can go on to apply our algorithm on the prototype systems in real time.

The 64 OSM, however, was only going through its first iteration when the thesis work was done. The system level architecture of the OSM is much different from the 1296 version. The focus here is much more on the system-level performance rather than just the OSM as in the 1296 case. The underlying motivations for developing this particular 64 OSM module are the following. First is to take advantage of the technological improvement of the hardware components. The HVDACs, ADCs, and optical connections in the market can all perform better, have more resolution, or can be integrated into a smaller physical packaging than the ones available when the 1296 OSM's were built. Since the circuitry components are smaller and better integrated, we could reduce the space requirement of the system substantially. Second, we want to make improvements in terms of system architecture in order to make our design achieve more modularity. The new system architecture focuses on the communication between electronics and system controller. We want to introduce more intelligence into the system and eliminate all unnecessary dependence among different parts of the system. Third, since this 64 OSM module has a small manageable size, it is a perfect candidate for us to attempt many system level testings and optimizations.

In terms of evaluating the performance of our optimization algorithm, there is not much difference between the two types of switches. Although the system architecture and circuitry components might be different, the heart of the switch which is the OSM design remains the same in the two cases. We can judge the performance of our algorithm by looking at the number of measurement steps it requires before reaching the optimal value, and be blind to other system components outside the OSM. These other components of the system only come into the picture when we need to convert from the number of steps into the actual length of time the switch requires to restore a connection.

Chapter 3: Studying the Hill Shape

The approach for modeling and finding efficient re-optimization algorithms for this optical switch system consists of three steps. First, we make measurements on a prototype system, study hill shapes, and systematically look at the statistics via fitting a quadratic model. Second, we apply the knowledge gained from studying the hill shape to come up with an accelerated re-optimization algorithm and some variations of it that are robust and practical enough for real-world use. Third, we implement the algorithms on some prototype systems so we can evaluate their real-time performance. In this chapter, we will focus on the first step of studying the hill shapes. The other two steps will be discussed in the following chapters.

Before we go on, let's remind ourselves of the schematic of the switch again. The main components of the switching core consist of the input and output fiber arrays, two micromirror arrays, two lenslet arrays, and one field lens. We sometimes refer to each of the two micromirror arrays as input chip and output chip. In order to make a light pass from one input fiber to one output fiber, the input mirror needs to tilt so that it directs the light to the output mirror, and the output mirror has to tilt so that it directs light to the output fiber. Our problem focuses on the dynamic of the inner core of the switch, meaning that we assume all light signal from each input fiber is already properly collimated and perfectly aligned with an input mirror in the micromirror array and each output fiber is properly aligned in a similar fashion to its own output mirror. So the only controlling variables we have left that affect the light throughput are the voltages that tilt the mirrors. For each mirror, the controlling voltages consist of the horizontal component and the vertical component, V_c and V_r.

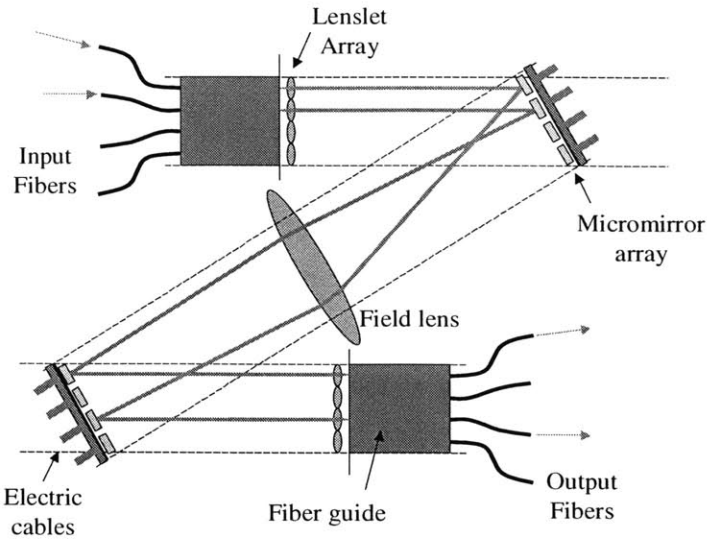


Figure 6: A schematic representation of optical crossconnect switch

The hill shape, in essence, is the transmissivity characteristic of each connection between any given input fiber and any output fiber. Every connection consists of four control variables, two being the horizontal and vertical control voltages of the input mirror, and other two being those of the output mirror. The optimum point of each connection is defined as the spot where the transmissivity or the optical throughput of the connection is maximized. If one plots the transmissivity of a connection as a function of the four control voltages near the optimum point, the result is a four dimensional hill, with the optimum throughput spot being the peak of the hill. Each and every connection has its own unique hill profile. Studying the hill shapes is crucial because a good knowledge of the hill shapes is the key to developing efficient re-optimization algorithm. Before we study the hill shapes, it is important to discuss the spaces or coordinate systems that we will look at our hills in. The first section of this chapter will discuss that. Studying the hill shapes in itself also involves two processes. First we study and extract parameters from each individual hill. Then we search for global patterns that might exist among the parameters of hills across the switch. These two processes are described in more detail in sections 3.2 and 3.3 of this chapter.

3.1 Voltage and Beam displacement spaces

This section will describe the idea of changing the working coordinates or spaces of our problem. There are many reasons why changing the problem's coordinate might be desirable. One is that a different coordinate might enable the problem to be described in a simpler way. Here we will focus on the two specific coordinates for our hills and the method of transforming back and forth between these two spaces. The two spaces referred to in this case are the voltage space and what we call the beam displacement space. In voltage space the independent variables are the voltage values that we apply to the electrodes to control the movement of the mirrors whereas in the beam displacement space the independent variables directly describe the reflected beam spot's position on a fixed plane in front of the mirror array. A good fixed plane to use can be the plane of the opposite (target) chip. We could use standard distance units such as millimeters to describe the beam displacement space. But a more convenient unit to use is mirror units, with one unit being equivalent to the distance from the center of one mirror to the center of its adjacent mirror.

The relationship between the two spaces of each connection is based directly on the voltage/beam displacement characteristic of the two mirrors involved in the connection. A few observations were made on the general nature of the mirrors. First, for small voltages, beam displacement is approximately proportional to the square of the applied voltage. However, as the voltage gets larger, a higher order terms are required to describe this relationship. Second, even though V_c seems to essentially control the horizontal movement of the beam while V_r does the vertical, there is still some cross coupling. This means that if we change the value of V_c, not only does the C or the column component of the beam displacement change, but also the R or the row component of the beam displacement is affected, although in a much smaller scale. Typical plots of squared voltage as a function of position in beam displacement space are shown in Figure 7. The voltage squared referred to in this case is not exactly the square of each voltage, but rather the product of the voltage and its absolute value, $V|V|$. The left plot shows the column voltage component, so the vertical axis of the graph is $V_c|V_c|$. The horizontal axes are the beam displacement position, whose unit has been centered and

normalized into mirror units and referred to by a row and column number. The right plot shows the analogous relationship of the vertical voltage component, $V_r|V_r|$.

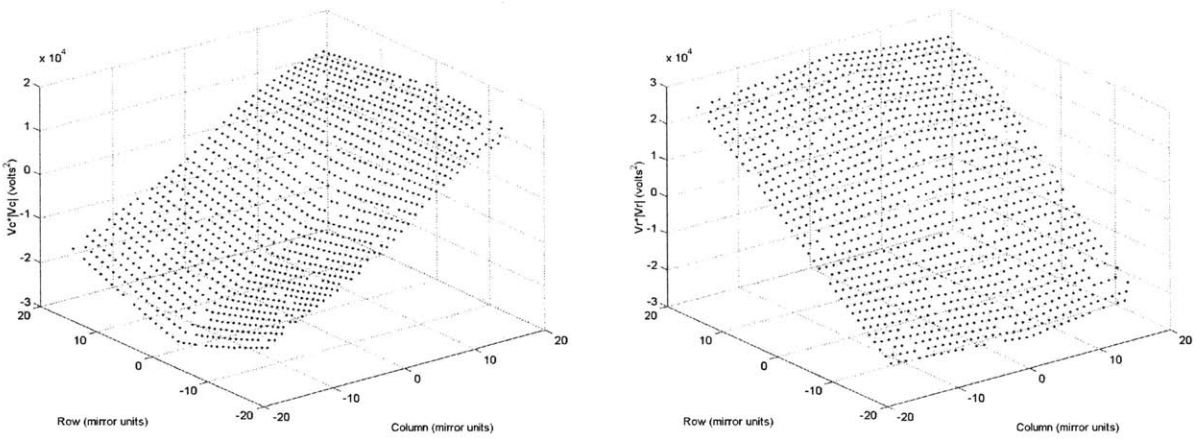


Figure 7: Voltage squared as a function of beam displacement position

The fact that the plots look relatively smooth suggests that we should be able to find a parsimonious model to capture these characteristics. One simple model that works reasonably well is the two dimensional quadratic functional form. The voltage squared can be approximated as a quadratic function of the horizontal and vertical components of the mirror's beam displacement. The exact form of the quadratic function used to model each mirror is:

$$V_c |V_c| \approx a_1C^2 + a_2R^2 + a_3CR + a_4C + a_5R + a_6 \qquad (3.1)$$

$$V_r |V_r| \approx b_1C^2 + b_2R^2 + b_3CR + b_4C + b_5R + b_6 \qquad (3.2)$$

In order for this model of voltage/beam displacement characteristic to be reasonably accurate, we model each of the four voltage space quadrants separately. The motivation here comes from the fact that 1) we can see in the plots above that there appear to be four separate regions, and 2) each quadrant involves activation of a different electrode pair. Modeling each of the four voltage quadrants separately means that each of the quadrants will have its own set of six quadratic coefficients to best represent the real characteristic of the mirror in that quadrant. Since we also model the two voltage components of each mirror, V_c and V_r, separately, we end up having a total of eight sets of quadratic coefficients, resulting in a total of 48 coefficients, to represent each mirror.

Now we go back to the issue of representing the hills. Each hill can be described in voltage and beam displacement spaces. Figure 8 below shows the physical origin of the two spaces of a chosen input mirror.

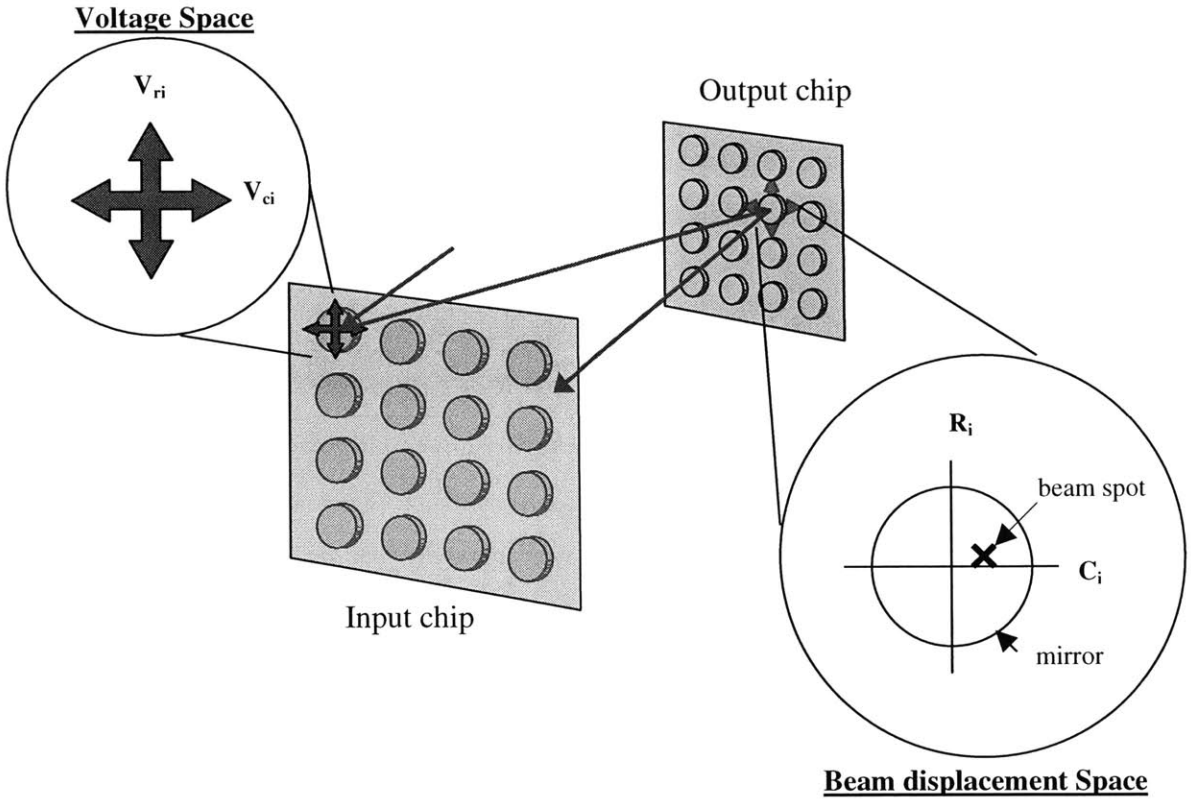


Figure 8: The voltage and beam displacement space of a chosen input mirror

In the voltage space, we plot the logarithm of optical power as a function of the four varying voltages, where two voltages control the input mirror and the other two control the output mirror. We denote the function and its four arguments as $P(V_{ci}, V_{ri}, V_{co}, V_{ro})$. V_{ci} is the voltage applied to control the movement of the input mirror in the column or the horizontal direction. V_{ri} is that of the input mirror in the row or the vertical direction, while V_{co} and V_{ro} are those of the output mirror. In the beam displacement space, as mentioned before, we plot the optical power as a function of the beam position reflected by the tilting mirrors onto the opposite chips. Similar to the voltage space, the beam displacement space is made up of four dimensions, each representing the vertical or horizontal component of the reflected beam spot from the input or output mirror. The four components in the beam displacement space are (C_i, R_i, C_o, R_o), where C_i is the

position in the column or horizontal direction on the output chip where the input mirror reflects the beam spot to, R_i is the that in the row or vertical direction, C_o is the position in the column direction on the input chip where the output mirror reflects the beam spot to, and R_o is that in the row direction.

Since making a connection involves controlling two mirrors, the relationship between the connection's two working spaces has to include the voltage/beam displacement characteristics of the two mirrors. For any connection, we can thus extend the relationship presented in Equations 3.1 and 3.2 as follows:

$$V_{ci}\,|V_{ci}| \sim a_1C_o^2 + a_2R_o^2 + a_3C_oR_o + a_4C_o + a_5R_o + a_6$$

$$V_{ri}\,|V_{ri}| \sim b_1C_o^2 + b_2R_o^2 + b_3C_oR_o + b_4C_o + b_5R_o + b_6$$

$$V_{co}\,|V_{co}| \sim c_1C_i^2 + c_2R_i^2 + c_3C_iR_i + c_4C_i + c_5R_i + c_6 \qquad (3.3)$$

$$V_{ro}\,|V_{ro}| \sim d_1C_i^2 + d_2R_i^2 + d_3C_iR_i + d_4C_i + d_5R_i + d_6 \qquad (3.4)$$

The subscripts i and o refer to the input mirror's and output mirror's values respectively. The first two equations refer to the voltage/beam displacement characteristics of the input mirror, while the latter two refer to those of the output mirror. Each mirror has its own unique relationships in the vertical and horizontal components and therefore any connection is made up of a total four voltage-beam displacement relationships. Each equation above models the voltage controlling the mirror in the vertical or horizontal direction on one chip as a function of the position of the reflected light beam spot on the opposite chip's plane referred to by the column and row number.

Although the hill profile of each connection can cover areas far beyond the position of its peak, the part of the hill that is relevant to us is only around of the hill's peak. Thus, instead of taking into considerations the four complete quadratic relationships as presented in Equations 3.1-3.4, we can simply approximate them around the local spot using linear slopes. Since a range of voltages that any one connection covers is significantly smaller than the whole voltage range a mirror can swing, using a linear approximation around the connection peak to model the voltage/beam displacement relationship of the whole connection seems to be a practical and sufficient approach. The range of any one hill is very small relative to the curvature of the whole

voltage/beam displacement function. This approach in effect turns the task of transforming back and forth between the two sets of coordinates into a simple linear transformation process. The procedure for making the transformation is described below.

We start with the four equations describing the voltage/beam displacement characteristics of the connection's two mirrors as shown in Equations 3.1-3.4 above. All connections would use these same common functional forms, but each connection would use the least square fitting technique on each of its two particular mirrors to obtain its own set of unique coefficients. The data points to be used in the least square fitting come from the database. The database contains all the voltage values required for making connections from any specific mirror on one chip to all the mirrors on the opposite chip. Therefore, for each mirror the database would provide us with the number of data points equal to the number of mirrors on the opposite chip. We however do not use all those available data points for the fit. The idea of looking at each of the four voltage quadrants separately was already mentioned earlier. We want the fitting coefficients to be customized to best represent the voltage/beam displacement characteristic around the peak of the connection, so we only use the points that are in the same voltage quadrant as the connection peak's voltage. There are around a fourth of the total number of mirrors on the chip in each quadrant.

Once we have obtained the fitting coefficients for all four equations, the next step is to approximate the voltage/beam displacement relationships locally with linear slopes. This is done by taking partial derivatives of the voltage with respect to the C and R variables of the beam displacement space and then substituting the actual values of the connection's C and R into the calculation in order to find the local slopes. Assuming that we are characterizing an input mirror A (that is making a connection with an output mirror B) whose position on the output chip is specified by column number X and row number Y, the mathematical form of the derivatives is as follows. We start with the column voltage component.

$$\text{Let} \quad W_c = V_c|V_c| = V_c^2 \text{sign}(V_c) \approx a_1C^2 + a_2R^2 + a_3CR + a_4C + a_5R + a_6,$$

$$V_c = -|W_c|^{-1/2} \text{ sign}(W_c) \qquad (3.5)$$

Then,

$$dV_c/dC = \tfrac{1}{2} |W_c|^{-1/2} (d|W_c|/dC) \operatorname{sign}(W_c)$$
$$= \tfrac{1}{2} |W_c|^{-1/2} (dW_c/dC)$$
$$= \tfrac{1}{2} (a_1C^2 + a_2R^2 + a_3CR + a_4C + a_5R + a_6)^{-1/2} (2a_1C + a_3R + a_4) \quad (3.6)$$

Similarly,

$$dV_c/dR = \tfrac{1}{2} (a_1C^2 + a_2R^2 + a_3CR + a_4C + a_5R + a_6)^{-1/2} (2a_2R + a_3C + a_5) \quad (3.7)$$

Therefore,

$$(dV_c/dC) \Big|_{C=X,\, R=Y} = \tfrac{1}{2} (a_1X^2 + a_2Y^2 + a_3XY + a_4X + a_5Y + a_6)^{-1/2} (2a_1X + a_3Y + a_4) \quad (3.8)$$

$$(dV_c/dR) \Big|_{C=X,\, R=Y} = \tfrac{1}{2} (a_1X^2 + a_2Y^2 + a_3XY + a_4X + a_5Y + a_6)^{-1/2} (2a_2Y + a_3X + a_5) \quad (3.9)$$

The relationships hold similarly for the V_r component of the same mirror:

$$W_r = V_r|V_r| = V_r^2 \operatorname{sign}(V_r) \approx b_1C^2 + b_2R^2 + b_3CR + b_4C + b_5R + b_6, \quad (3.10)$$

$$(dV_r/dC) \Big|_{C=X,\, R=Y} = \tfrac{1}{2} (b_1X^2 + b_2Y^2 + b_3XY + b_4X + b_5Y + b_6)^{-1/2} (2b_1X + b_3Y + b_4) \quad (3.11)$$

$$(dV_r/dR) \Big|_{C=X,\, R=Y} = \tfrac{1}{2} (b_1X^2 + b_2Y^2 + b_3XY + b_4X + b_5Y + b_6)^{-1/2} (2b_2Y + b_3X + b_5) \quad (3.12)$$

An output mirror B that is making connection with this input mirror A would obtain its own set of local slopes in a similar manner its own version of Equations 3.18, 3.19, 3.11, 3.12. So for any connection, we need a total of eight local dereivative values.

Let X_i, Y_i, X_o, Y_o be the physical position, denoted by the column and row numbers, of the input and output mirrors on the chips, and let V_{ci}^0, V_{ri}^0, V_{co}^0, V_{ro}^0 be the voltages that maximize the optical power of the connection. The connection's eight local derivative values can be arranged into a matrix form as followed:

$$J_i = \begin{bmatrix} \left.\frac{\partial V_{ci}}{\partial C}\right|_{C=X_o,R=Y_o} & \left.\frac{\partial V_{ci}}{\partial R}\right|_{C=X_o,R=Y_o} \\ \left.\frac{\partial V_{ri}}{\partial C}\right|_{C=X_o,R=Y_o} & \left.\frac{\partial V_{ri}}{\partial R}\right|_{C=X_o,R=Y_o} \end{bmatrix} \quad (3.13)$$

$$J_o = \begin{bmatrix} \left.\frac{\partial V_{co}}{\partial C}\right|_{C=X_O,R=Y_O} & \left.\frac{\partial V_{co}}{\partial R}\right|_{C=X_O,R=Y_O} \\ \left.\frac{\partial V_{ro}}{\partial C}\right|_{C=X_O,R=Y_O} & \left.\frac{\partial V_{ro}}{\partial R}\right|_{C=X_O,R=Y_O} \end{bmatrix} \tag{3.14}$$

The task of finding the local derivatives and the connection's J_i and J_o is quite tedious, since it had to go into the memory to retrieve the data points in the relevant quadrant in order to do a least square fitting on them. So it should be done only once when the connection is first attempted. Once the values of J_i and J_o are found, they should be stored locally so that they can be used to transform back and forth between the two spaces in all future attempts.

Now that J_i and J_o matrices have been obtained, they can be used to transform coordinates of the point around the peak of the connection from beam displacement space to voltage space, referred to as a forward transformation, as follows:

$$[V_{ci}, V_{ri}]^T = [V_{ci}^0, V_{ri}^0]^T + J_i [\Delta C_i, \Delta R_i]^T \tag{3.15}$$

where $\Delta C_i = C_i - X_o$ and $\Delta R_i = R_i - Y_o$

$$[V_{co}, V_{ro}]^T = [V_{co}^0, V_{ro}^0]^T + J_o[\Delta C_o, \Delta R_o]^T \tag{3.16}$$

where $\Delta C_o = C_o - X_i$ and $\Delta R_o = R_o - Y_i$

The backward transformation, going from voltage space to beam displacement space, is exactly the reversal:

$$[C_i, R_i]^T = [X_i, Y_i]^T + J_i^{-1}[V_{ci}-V_{ci}^0, V_{ri}-V_{ri}^0]^T \tag{3.17}$$

$$[C_o, R_o]^T = [X_o, Y_o]^T + J_o^{-1}[V_{co}-V_{co}^0, V_{ro}-V_{ro}^0]^T \tag{3.18}$$

In reality, since the hardware is controlled by the voltage values, the natural or default space of the hardware control is always the voltage space. The backward transformation needs to take place first in the case where we want to model our problem in beam displacement space. Once we are in the beam displacement space, we can model and do all the calculations required. But whenever we need to take new data points on the

hill of the hardware, we need to use the forward transformation to get back to the voltage space.

3.2 Measurements and Description of Individual Hills

The goal of this section is to gain knowledge about the individual hill shapes and explore whether they can be modeled with a simple common functional form. In other words, we want to see whether we can extract finite sets of parameters to represent the hills. A consequence of Taylor's theorem is that any smooth function can be well approximated by a quadratic function near any local maximum. In our case, the transmissivity in a connection can be approximated as a quadratic function of beam displacement when the beam alignment is close to optimal. The quadratic model holds over a wider range when transmissivity is expressed on a log scale (e.g. decibels). This may be explained partly by the fact that light beams have approximately Gaussian cross-sections, so that the hill inherits some Gaussian behavior. The connection's input and output power are read from the tap devices that detect the intensity of the beam going into the input mirror and that of the beam coming out from the output mirror. If the input power is known or at least known to be constant, the output power can be used as a proxy optimization function in place of transmissivity.

3.2.1 Typical hills

Samples of the data points in the vicinity of the hilltops from many connections were collected and studied. Since our hills are four dimensional, it is rather difficult to visualize them. One way of trying to do so is to fix the values of two dimensions and plot the transmissivity as a function of the other two dimensions. This way we get to see different two-dimensional slices of the hill depending on which other two dimensions are chosen to be fixed and the values at which they are fixed. Some views of a hill are shown in Figure 9.

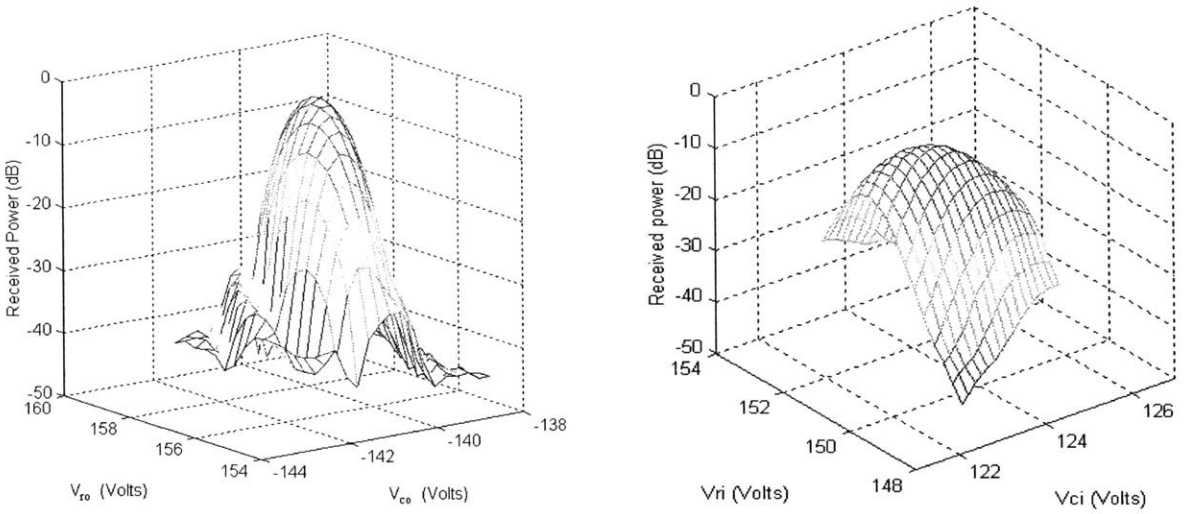


Figure 9: Mesh plots of the 2-dimensional hill slices

The left plot shows a slice of the hill when we fix the input mirror's voltages at their peak values and let the output mirror's voltages vary. The right plot shows the slice of the hill when we fix the output mirror's voltages and let the input mirror's voltages vary. The location of this connection's input mirror is around the edge of the input chip while the location of the output mirror is close to the center of the output chip. The fact that the left hill slice is steeper than the right hill slice shows that the connection' transmissivity is much more sensitive to changes in the output mirror's voltages than changes in the input mirror's voltages. This makes perfect sense since the close-to-the-edge position of the input mirror requires the output mirror to be tilted by high voltage values. The high voltage values are in a more sensitive region of the mirror's voltage/beam displacement characteristic plot, meaning that a small change in voltage in this region would result in a big change in beam displacement. A big change in beam displacement implies a big change in transmissivity. The same idea applies to the shallow hill slice case. The close-to-the-center position of the output mirror only requires the input mirror to be tilted by small voltage values. This means that we operate close to the origin of the voltage/beam displacement characteristic of the mirror so even a big change in voltage value will result in a small change in beam displacement. Therefore, the change in transmissivity is also small and the hill slice looks shallow. In addition to showing characteristics that reflect the mirrors' positions, these plots support the fact that a quadratic function should be able to model the hill quite well close to its peak.

The same two-dimensional hill slices shown above can be visualized in the form of contour plots as well. While mesh plots allow us to look at the hill from the side, contour plots give us the top view of the hill. Each contour line outlines the parts of the hill that have the same height. In our case, while the mesh plots let us see the smoothness of the hill surface more easily, the contour plots give us a better view of the overall picture of the hill, i.e. the overall size, shape, symmetry, and orientation of the hill. Figure 10 below show the contour plots from the exact same hill as the one in the mesh plots above. In addition to plotting the contours of the two dimensional slices in the voltage space, we also transform the hill data points into beam displacement space using the backward transformation process discussed in the previous section (section 3.1) and plot a set of two-dimensional slices of the same hill in beam displacement space.

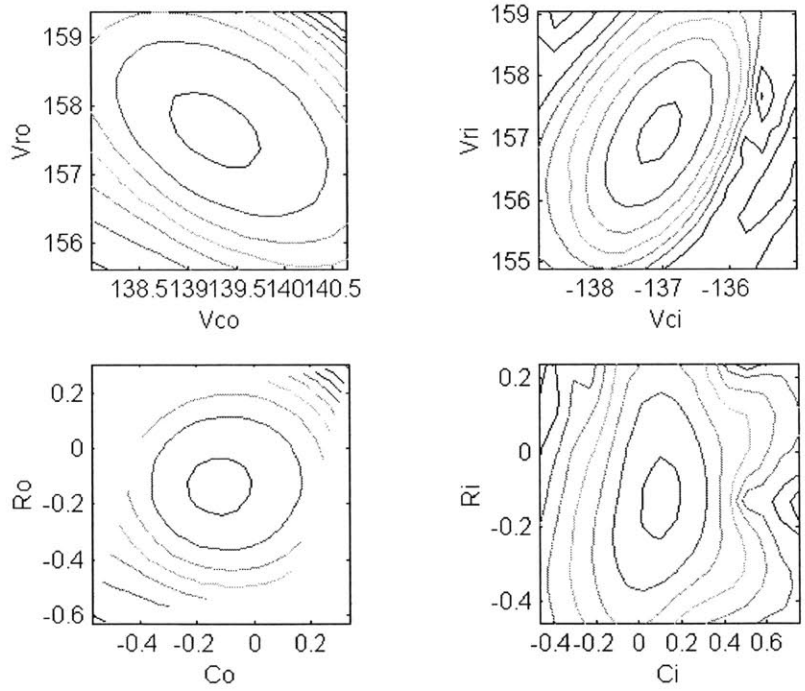


Figure 10: Contour plots of the same 2 dimensional hill slices. The top two are in voltage space and the bottom two are in beam displacement space

The top two graphs in Figure 10 are the contour plots in voltage space while the bottom two are those in beam displacement space. It is evident from the graphs that by transforming the hill into beam displacement space, we effectively reduce the coupling

between dimensions of the problem. The hill's major and minor axes in beam displacement space are much more closely aligned to the problem's dimensional axes. We give the credit to a good voltage/beam displacement relationship model that we used in transforming the problem from voltage to beam displacement space. Because the model approximates the actual relationship well, the coupling between axes is almost entirely accounted for in the space transformation process and therefore does not appear much at all in the beam displacement space.

Other advantages of transforming the problem into beam displacement space can be seen when we compare the contour plots from different connections. Contour plots of a different connection are shown in Figure 11. The transformation to beam displacement space seems to normalize the size and shape of the hills to some degree, making the hill profiles of different connections more uniform across the chips. In voltage space, the relationship between voltage/mirror's beam displacement is roughly quadratic. When we look at a particular connection that involves the two mirrors right in the middle of the chips, the required voltages around the peak of the hill are small. Since the quadratic curve is shallow at small values, a small increment in voltage step applied to that small starting voltage value produces a small increment in the beam displacement. But when we look at a connection involving the mirrors towards the edges of the chips, the voltages required to get to the peak are large. Since a small increment in voltage step applied to a large starting voltage value produces a bigger increment in the mirror's beam displacement, the hill profile of a connection close to the edges of the chips is by nature a lot steeper than that of connection in the middle of the chips. In effect, when we look at the variation of the hill profiles of the connections across the chips, we would start from very shallow hills in the middle, and as we move further out towards the edges of the chips the hill profiles get steeper and steeper. The existence of this prominent variation in the hill profiles across the chips affects us in two ways. First, this makes it harder for us to characterize and model the hill profiles as a whole, meaning that the variation introduces a uniqueness to each hill and a more complicated model is required in order to capture that. A simple model might no longer be good enough. Second, when the hill profiles vary a lot, with some hills being very steep while others being very shallow, it puts more pressure on the robustness requirement for the hill climbing algorithm that will

be discussed in the next chapter. When the hill is very steep, a small move can lead to a very big change in the function's value. In other words, a small error in the step size can in effect make us fall off the hill very easily. The tolerance for noise and error is very small in this case. Therefore, we prefer having the hill profiles being more uniform in size and shape to avoid the requirements for more complicated models and algorithms for the extreme cases.

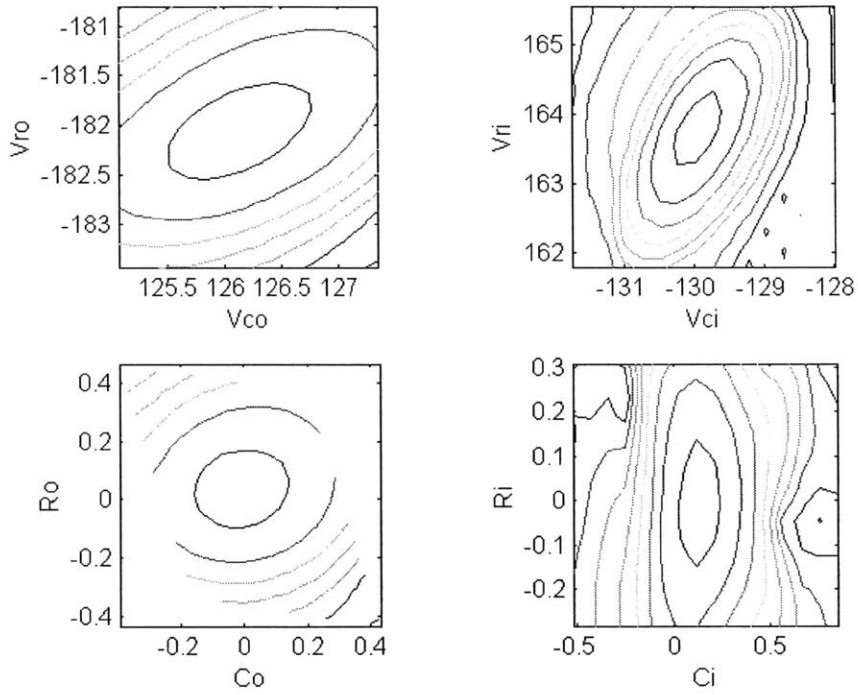


Figure 11: Contour plots of a different connection's 2 dimensional hill slices

In beam displacement space, we plot the relationship between the mirror's beam displacement and the connection's optical power directly without having to take into account the quadratic behavior of the mirror's voltage/beam displacement characteristic. Since the relationship between the mirrors' beam displacement and the connection's optical throughput is quite consistent across the chips, we achieve our goal of reducing the big variation among the hill profiles.

3.2.2 Modeling the hills with a quadratic function

The next step after studying the hills and somewhat normalizing them by transforming into beam displacement space is to find a simple model to capture

characteristics of the hills. A fitting of the hill tops with the four dimensional quadratic function is performed and evaluated. All the hills are fitted with the same quadratic functional form, but each hill will obtain its own unique set of parameters through the least square fitting technique. The parameters of the fitting thus capture characteristics of each individual hill. The procedure of how to form a least square fit problem will be described shortly. Since the hills are more uniform and normalized in beam displacement space, it makes sense to do the fitting in the beam displacement space as well so that the values of the fitting parameters from different hills across the chips are somewhat within the same range. Let x_1, y_1, x_2, y_2 be the values of the connection's four components in beam displacement space, and let T be the logarithm of the measured transmissivity when the voltages are set to the these values. The functional form of the hilltop's four dimensional quadratic fit is given by the equation:

$$T = a_1 + a_2x_1 + a_3y_1 + a_4x_2 + a_5y_2 + a_6x_1^2 + a_7y_1^2 + a_8x_2^2 + a_9y_2^2 + a_{10}x_1y_1 + a_{11}x_1x_2 + a_{12}x_1y_2 + a_{13}y_1x_2 + a_{14}y_1y_2 + a_{15}x_2y_2 \qquad (3.19)$$

To form a least square problem, assume we have a total of m sample points. Each sample point i would have its own T_i, x_{1i}, y_{1i}, x_{2i}, and y_{2i}. We would like to form a vector $\underline{T}$ of size m by 1, a matrix **M** of size m by 15, and a vector $\underline{a}$ of size 15 by 1. We put the value of T_i into the i^{th} row of vector $\underline{T}$. For the i^{th} row of matrix **M**, we put the values of 1, x_{1i}, y_{1i}, x_{2i}, y_{2i}, x_{1i}^2, y_{1i}^2, x_{2i}^2, y_{2i}^2, $x_{1i}y_{1i}$, $x_{1i}x_{2i}$, $x_{1i}y_{2i}$, $y_{1i}x_{2i}$, $y_{1i}y_{2i}$, $x_{2i}y_{2i}$ into the 15 columns. We do the same for all m samples. The goal here is find a solution set for $\underline{a}$ that minimizes $\|\mathbf{M}\underline{a} - \underline{T}\|^2$. This least square problem can be solved by standard linear algebra techniques.

From the equation above, a_1-a_{15} are the 15 quadratic fitting parameters. They can roughly be divided into two groups. a_1-a_5 are considered the linear terms while a_6-a_{15} are the quadratic terms. The above equation can be arranged and written in another form, which gives more insight to the physicality of the system. Let $A = (x_1, y_1, x_2, y_2)^T$ be a column vector of the four beam displacement space components. The quadratic terms are arranged into a real 4x4 symmetric matrix **Q**, while the linear terms can be written as a 4x1 vector $\underline{u}$ and a real number c. All these would yield another equivalent form of the equation above as:

$$T = \tfrac{1}{2} A^T \mathbf{Q} A + \underline{u}^T A + c \qquad (3.20)$$

where $\mathbf{Q} = \begin{bmatrix} 2a_6 & a_{10} & a_{11} & a_{12} \\ a_{10} & 2a_7 & a_{13} & a_{14} \\ a_{11} & a_{13} & 2a_8 & a_{15} \\ a_{12} & a_{14} & a_{15} & 2a_9 \end{bmatrix}$, $\underline{u} = \begin{bmatrix} a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$, and $c = a_1$

When written in this form, the analysis and interpretation of this mathematical system can be done and visualized more simply. First of all, the symmetric matrix **Q** captures all the quadratic component of the hill, and the fact that it is symmetric will allow us to do further analysis in various forms. The symmetry of **Q** has two implications: the eigenvalues will be real and the eigenvectors will be orthogonal. These two implications will be useful to us later on. Second, the peak position of the hill based on this quadratic model can be calculated easily. Let a 4x1 vector $\underline{p}$ be the predicted position of the peak based on this quadratic model of the hill. By taking partial derivatives of Equation 3.19 with respect to the four beam displacement variables and setting the partial derivatives equal to zero, we can obtain an expression for the predicted optimal point or the peak position in beam displacement space as:

$$\underline{p} = \mathbf{Q}^{-1}(-\underline{u}) \qquad (3.21)$$

It should be noted here that $\underline{p}$ represents a global maximum of L if and only if Q is a negative definite matrix. In our case, **Q** is clearly a negative definite matrix since $A^T\mathbf{Q}A < 0$ for every nonzero vector A. All diagonal terms of **Q** are negative. More discussion regarding the analysis of the **Q** matrix will be presented later on. With $\underline{p}$ being the position offset that corresponds to the peak position of the model and d being a scalar offset, the quadratic functional form can again be written in another equivalent form as:

$$T = \tfrac{1}{2} (A - \underline{p})^T \mathbf{Q} (A - \underline{p}) + d \qquad (3.22)$$

A decision as to how far down the hill top we should include in the model also needs to be made. There is a tradeoff involved here. Including more points down the hill

will enable us to handle the possibility of more drifting and shifting of the hill. However, if the accuracy of the quadratic model in representing the real hill decreases as we go down the hill, then it might not be wise to include the area further down the hill into the model. In our case, based on the assumption that the connections are quite stable, we decide to model the hills down to around 3-5 dB from the peak.

Since our hills are smooth around the top, the quadratic function discussed above should fit them quite well. Let $T_d(A)$ be the real data or the measured power as a function of A, and $T_m(A)$ be the power calculated from the quadratic model above. A way to quantify how well the quadratic model fits the data near the peak is to look at the degree to which $T_m(A) \approx T_d(A)$. Another figure of merit here is how close the peak of the model is to the true optimum position, i.e. the degree to which $\underline{p} \approx \text{argmax } T_d(A)$. We do the fitting on 25 hills across the chips which involved 5 chosen input mirrors and 5 chosen output mirrors. Sample points for the fitting of each hill are taken on a four dimensional grid in beam displacement space. The size of the grid for each hill is pre-determined so that the edge of the grid is roughly 3 dB from the peak. With 16^4 sample points from each hill all taken within 3dB from the peak of each particular hill, the mean of the overall loss difference between the actual hill and the fitted model, $\Sigma|T_m(A_i)-T_{d(Ai)}|/n$, is around 0.05 dB. The error in the peak position prediction or the distance from the predicted peak position to the actual peak position, $\|\underline{p} - \text{argmax } T_d(A)\|$, is around 0.03 mirror units. These small error numbers confirm our belief that the quadratic function does indeed fit our hills very well.

In practice, it becomes quite impractical to collect as many data points as 16^4 in order to have a good hill model. Since a large optical switch can have more than 1 million connections, we want to reduce the number of data points collected for each hill as much as possible. The tradeoff here is that with less number of points, the accuracy or how well the model fits the actual data will be not be as good, but the time it takes to collect the data and the amount of data to be processed is much less. The minimum number of data points required to do a four-dimensional quadratic fitting is equal to the total number of quadratic fitting coefficients which is 15 in this case. We want to have well over the minimum number in order to average out the effect of noise or any strange unexpected behavior at any particular point, so we decide on the total of 81 points. The positions of

these 81 points are again chosen to form a grid in the four-dimensional beam displacement space, and the size of the grid is again chosen so that the edge of the grid is no more than 3 dB down from the peak. We take one point right at the peak of the hill, and go 0.1 mirror unit to both sides of the peak in each dimension. This totals to 3 points in each dimension and with 4 dimensions we have in the end $3^4 = 81$ points. Instead of taking data points from only 25 hills across the chips, now we choose every third mirror on the 36x36 input and output mirror arrays. So for each of the 144 input mirrors, we make connections to each of the 144 output mirrors. The total number of possible connections involving theses chosen mirrors is 144x144 = 20736. For these 20736 chosen connections, we take 81 data points from each hill and obtain the quadratic fitting coefficients a_1-a_{15} for each of them. The distributions of the quadratic terms, a_6-a_{15}, from these different hills across the chips are shown in Figure12.

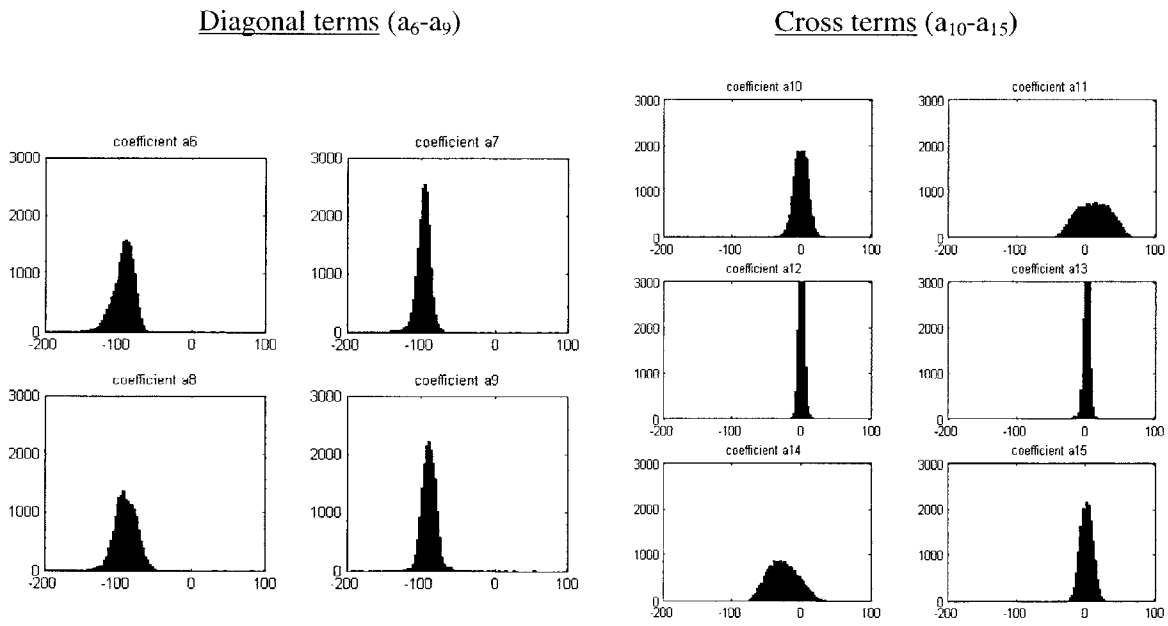


Figure 12: Distributions of the quadratic fitting coefficients in beam displacement space

We call the coefficients a_6-a_9 the diagonal terms since they are on the diagonal of the **Q** matrix and they are the coefficients of the purely squared terms. We call the rest of the hill's quadratic coefficients, a_{10}-a_{15}, the cross terms since they tell us about the degree

of coupling between any two dimensional variables. The diagonal terms are all negative and much larger in magnitude than the cross terms: the distributions of a_6-a_{10} are centered at around −100 while those of the cross terms are centered at either zero or a relatively small number. This is as expected because the diagonal numbers are directly proportional to the second derivatives of the hill with respect to each of the four dimensions. The negative second derivatives tell us about the curvature of the hill and our hills are strictly concave by nature. The diagonal terms' large magnitudes are as expected as well since the purely squared terms should be dominant compared to the effect of couplings between axes. Since the coupling effect is minimized in beam displacement space, all the cross terms are very small in magnitude. All except a_{11} and a_{14} are centered around zero. The coefficient a_{11} shows the coupling effect between input and output mirrors' horizontal movements, and its distribution in the plot above is centered at a small positive number. It is reasonable to have the horizontal movements of input and output mirrors being slightly positively correlated, since if we tilt an input mirror in a positive horizontal direction, in order to maintain the light beam spot going towards an output fiber at the same position we have to tilt the output mirror in a positive horizontal direction as well. The same argument with a slight twist goes with the coefficient a_{14}, which shows the coupling effect between the vertical movements of input and output mirrors. These two movements are slightly negatively correlated, because when we tilt an input mirror in a positive vertical direction, we have to tilt an output mirror in a negative vertical direction to compensate it. The theory behind tilting a mirror to compensate for the movement on another mirror on the opposite chip is the same for both vertical and horizontal directions. The only difference that makes one of them positively correlated while the other is negatively correlated is our convention of which way is considered a positive movement. For the horizontal direction, a mirror has to tilt so that the reflected light beam spot is moving to the right to be considered a positive direction. So when we take a top view of the two chips, if mirrors on both chips are tilting in a positive horizontal direction, the two mirror plates would be parallel. However, for the vertical direction, when we take a side view of the two chips, positive vertical movements from mirrors on the two chips would make the two mirror plates anti-parallel. An illustration of this is shown below in Figure 13.

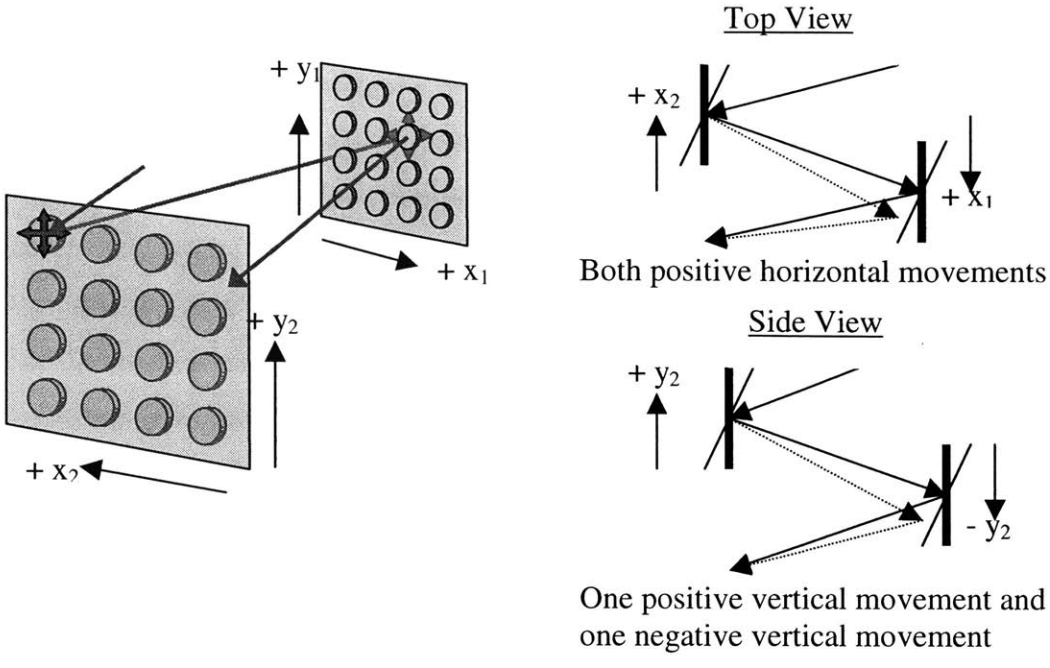


Figure 13: Conventions of vertical and horizontal movements

3.2.3 Relationship between quadratic model in beam displacement and voltage spaces

Even though we have decided to do the fitting in beam displacement space, it's important to note here that the quadratic fitting of the hills can be done in the voltage space as well. The task of performing the quadratic fitting in voltage space to obtain its corresponding **Q**, u, and p values can be achieved in two different ways. First, we can take the same set of sample data points that we used for the fitting in beam displacement space, but instead of having to transform the coordinates of the points from voltage into angle, we leave them in voltage space and use the least square fitting technique to obtain the corresponding fitting parameters a_1-a_{15} in voltage space directly. Once we have the new a_1-a_{15}, we follow the exact same calculations as in voltage space above to obtain the new **Q**, u, and p. The second option for obtaining **Q**, u, and p in voltage space is to derive them from their corresponding values in beam displacement space. Remember again that $A=(x_1,y_1,x_2,y_2)^T$ is a column vector consisting of the four corresponding components in the beam displacement space,. Let $V= (v_1,v_2,v_3,v_4)^T$ be a column vector consisting of the four controlling voltages, and $\mathbf{J} = \begin{bmatrix} Ji & 0 \\ 0 & Jo \end{bmatrix}$ be a 4x4 transformation matrix where J_i and

J_o are 2x2 voltage/beam displacement transformation matrices for the input and output mirror as mentioned in Chapter 3. Let us now denote the original set of $\mathbf{Q}$, $\underline{u}$, and $\underline{p}$ in beam displacement space as $\mathbf{Q}_a$, $\underline{u}_a$, and pa and the new set in voltage space as $\mathbf{Q}_v$, $\underline{u}_v$, and p_v. The quadratic fitting form in Equation 3.20 above can be written again as:

$$T = \tfrac{1}{2} A^T \mathbf{Q}_a A + \underline{u}_a^T A + c$$

Since the transformation relationship between V and A is $V = \mathbf{J}A$,

$$\begin{aligned} T &= \tfrac{1}{2} (J^{-1}V)^T \mathbf{Q}_a (J^{-1}V) + \underline{u}_a^T (J^{-1}V) + c \\ &= \tfrac{1}{2} V^T ((J^{-1})^T \mathbf{Q}_a J^{-1})V + (\underline{u}_a^T J^{-1})V + c \\ &= \tfrac{1}{2} V^T \mathbf{Q}_v V + \underline{u}_v^T V + c \end{aligned}$$

Therefore, $\mathbf{Q}_v = (J^{-1})^T \mathbf{Q}_a J^{-1}$.

3.3 Global patterns and statistics of hills across the switch

The set of model parameters a_1-a_{15} or $\mathbf{Q}$, $\underline{u}$, c for each hill shape extracted by the process in section 3.2.2 is the property of each individual connection. Therefore, the number of sets of parameters to be extracted and stored scale as N^2, which becomes increasingly costly as the number of ports N grows. It becomes impractical to measure and store the parameters of all the hills as N becomes large. For a practical solution, we need to exploit regularity among the connections and try to further parameterize the hill shape parameters. First, we collect data from dense set of connections, and extract the associated models parameters $\mathbf{Q}$, $\underline{u}$, c. The next step is to quantify the statistical variation of these parameters and see whether this variation could be reduced by predictive models. Regularities in the pattern of the extracted parameters are expected to exist, especially as functions of mirror locations. The goal of this stage is so that we would only need to measure a small subset of hills and use the extracted parameters of these measured hills to estimate the parameters of all unmeasured hills. Good hill shape estimate would then be derived for all connections, with only modest measurement and storage complexity.

Here we focus only on the variations of quadratic coefficients a_6-a_{15} composing the $\mathbf{Q}$ matrix. As later sections will discuss, the quadratic coefficients represent the hill shape or curvature, while the linear terms can be thought of as determining the hill location. Knowledge of the (relatively constant) hill shape will be used to enhance fast

algorithms for tracking small changes in the hill location. Three methods for modeling the hill shape coefficients as a function of connection parameters were explored and are described below.

3.3.1 Methods used to model the variation

Since regularities in the pattern of the extracted parameters might exist as a function of mirror location, the three methods attempted in this section are aimed to explore that possibility. Let i be an index of the quadratic coefficient of interest, j be an index of the connection's input mirror, k be that of the output mirror, Col and Row be column and row numbers of the mirror. All the quadratic coefficients a_i are in beam displacement space. The three methods, their forms, and the numbers of parameters required to represent each coefficient a_i are summarized in Table 1 below.

Method	**Form**	**Number of parameters needed to represent each coefficient a_i**
1) Mean	a_i of any connection j,k = mean of a_i from all connections	1 (the mean value)
2) Linear fitting	$a_i = c_1 + c_2Col_j + c_3Row_j + c_4Col_k + c_5Row_k$	5 (c_1-c_5)
3) Quadratic fitting	$a_i = d_1 + d_2Col_j + d_3Row_j + d_4Col_k + d_5Row_k + d_6Col_j^2 + d_7Row_j^2 + d_8Col_k^2 + d_9Row_k^2 + d_{10}Col_j*Row_j + d_{11}Col_j*Col_k + d_{12}Col_j*Row_k + d_{13}Row_j*Col_k + d_{14}Row_j*Row_k + d_{15}Col_k*Row_k$	15 (d_1-d_{15})

Table 1: Three methods used to model variation of quadratic coefficients across the chips

Assume there are m sample hills, each of the hill has its own set of quadratic fitting coefficients a_6-a_{15}. Method 1 just takes the mean of each coefficient a_i across all m hill samples. For methods 2 and 3, we use the least squares fitting technique determine the values of fitting parameters c_1-c_5 and d_1-d_{15}. The minimum numbers of connections required in the process of retrieving fitting coefficients are 5 for the linear fitting case, and 15 for the quadratic fitting case. In practice, we need far beyond the minimum

numbers to minimize noise and the set of connections used in the fitting also needs to be chosen to cover all areas of the chips.

In order to evaluate the performance for each method, we determined how close the prediction values based on that particular method are to the real values. The real values in this case are the actual quadratic coefficient values that we extracted by overfitting the four-dimensional quadratic model on 81 data points around each connection's peak spot. For this particular evaluation, we extracted the quadratic coefficient values for 144x144 connections and used these connections and their coefficients to further retrieve global pattern parameters c_1-c_5, d_1-d_{15}. With these global pattern parameters available, given the locations of the two involving mirrors we can figure out the predicted value of any particular connection. We call the difference between the real value and predicted value of the hill coefficients the residual. If we plot the distribution of this residual, a good modeling method should result in a distribution centered around zero with a very small standard deviation. The actual results from using the same 144x144 connections in the three methods are shown below.

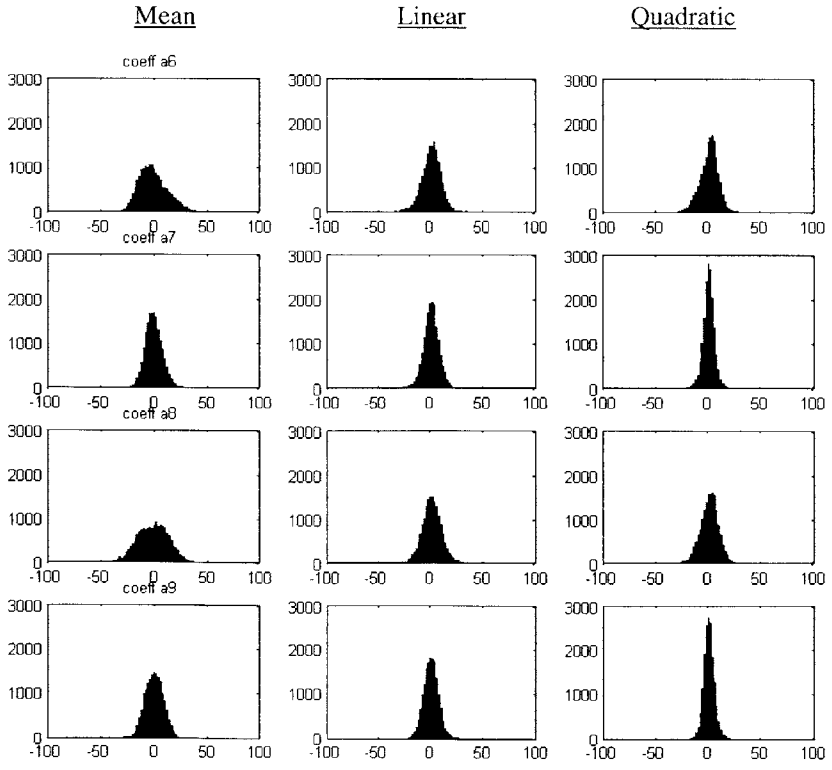


Figure 14: Distributions of the 4 diagonal coefficients' residual when modeled by the Mean, Linear fitting, and Quadratic fitting methods

The residual distributions shown above are of the four diagonal quadratic coefficients, a_6-a_9. These diagonal coefficients are of more importance to us than the cross-terms since they have far larger magnitude and hence are the dominant terms in our modeling. The distributions of the diagonal residuals shown above are centered around zero as expected. It turns out the variations in the diagonal coefficient values across the chips can indeed be modeled to a certain extent as a function of mirror location. As we move from the Mean method to the Linear fitting method and to the Quadratic fitting method, we essentially increase the degrees of freedom used in modeling. More degrees of freedom allow us to model the variations better, thus reducing the standard deviation of the residual distributions. Figure 15 shows that the same rationale applies to the cross-term coefficients as well.

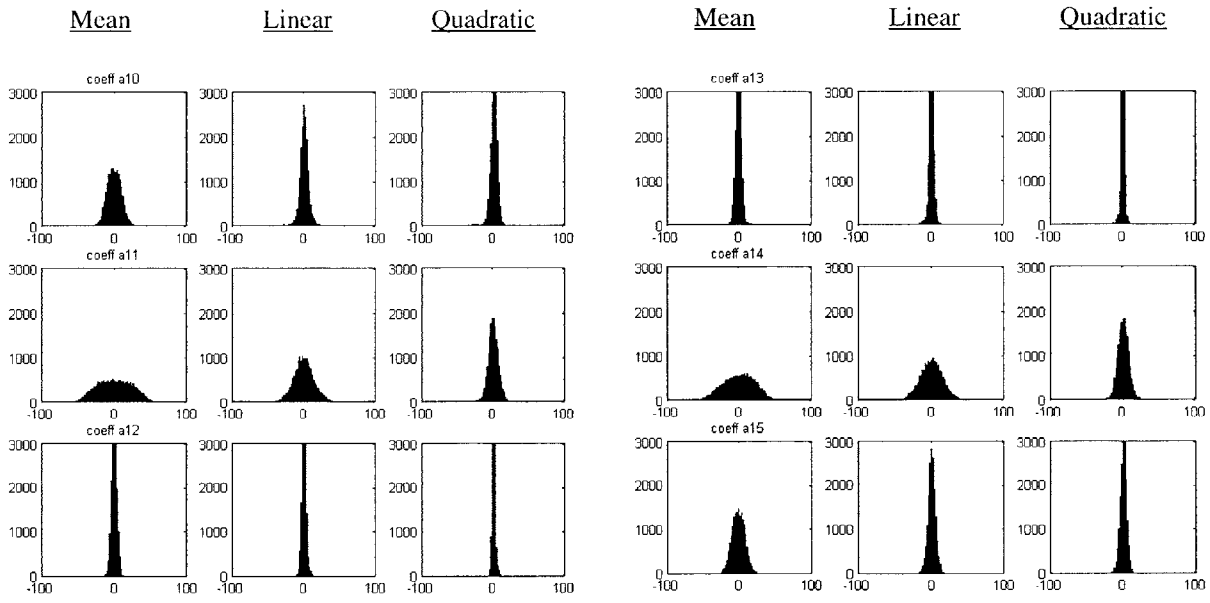


Figure 15: Distributions of the 6 cross-term coefficient residuals when modeled by the Mean, Linear fitting, and Quadratic fitting methods

Table 2 shows the standard deviations of these residuals shown in Figures 14 and 15. For all 10 coefficients, we see the decreasing trend of the residual value as we go from Mean to Linear fitting to Quadratic fitting methods.

Coefficient	Mean	Linear fitting	Quadratic fitting
a_6	17.2066	14.5852	13.8734
a_7	10.9976	10.5511	8.7243
a_8	18.5105	14.5207	13.6093
a_9	11.0035	10.2388	8.4491
a_{10}	12.1721	9.6184	8.6365
a_{11}	24.3166	15.5128	9.9822
a_{12}	4.2967	3.6012	2.8456
a_{13}	4.6210	3.9683	3.3298
a_{14}	22.6646	17.0868	10.8222
a_{15}	11.6901	9.3662	8.3605

Table 2: Standard deviation of the residual distributions seen in Figures 14 and 15

But a higher degree of freedom comes at a price; it requires more parameters to be stored. As shown earlier in Table 1, the number of parameters needed for making a prediction of each hill coefficient goes from 1 to 5 to 15 as we move from the Mean to Linear fitting to the Quadratic fitting method. In practice, for each connection or hill, we need to make a prediction for all of its 10 quadratic coefficients a_6-a_{15}. So the total number of parameters needed to be stored is 10 for the Mean method, 50 for Linear fitting, and 150 for Quadratic fitting. It is a tradeoff between performance and complexity in terms of measurement and storage. An even higher order functional form is expected to reduce the standard deviation of the residual distribution even further, but it will also increase the amount of coefficient-retrieving process and stored parameters substantially. For our application, the Quadratic fitting scheme is the method of choice since it provides adequate predictions without introducing too much complexity.

3.3.2 Benefits of global variation modeling

By choosing to use the Quadratic fitting method to model the global variation pattern, we essentially reduce the measurement and storage requirement from scaling with N^2 to a constant number of 150 parameters. We accomplished the goal of measuring

a small subset of hills and using the extracted parameters of these measured hills to estimate the parameters of all unmeasured hills. This is vital to the scalability of the algorithm usage, meaning it makes the algorithm practical for use for switches of all sizes. In addition, the process of global modeling could be a useful tool for outlier detection. That is, when any hill breaks the global patterns, it gives an opportunity for the system designer to look into why that particular connection behaves differently from others. Guided fault analysis can help to improve future implementations of the hardware.

Chapter 4: Applications of hill shape modeling

The goal for this chapter is to take advantage of hill shape models developed in the last chapter to come up with efficient algorithms to improve the switch's performance. Hill shape knowledge is particularly useful in the case of re-optimizing the connection, when we start from somewhere not too far from the hill peak and want to get to the hill peak as quickly as possible. The first section in this chapter will introduce an algorithm that could be used in this particular situation. The second section will briefly discuss other possible applications of hill shape knowledge.

4.1 Accelerated Re-optimization Algorithm

This section describes an accelerated re-optimization algorithm to be used in our switching system. The basic idea underlying this type of algorithm is that it assumes some partial knowledge of the hill it is climbing is already available prior to the start of the algorithm. As discussed in Chapter 3, 15 coefficients describing each hill may be estimated when the optical switch module is initially calibrated. When a connection is later re-established in the field, we assume that the 10 quadratic terms, a_6-a_{15}, can be trusted, while the remaining terms a_1-a_5 are subject to error due to unknown perturbations of the hill location. The reasoning behind this assumption is discussed below. Note that the hill coefficients used to accelerate re-optimization may be stored explicitly, or may be computed from the model discussed in Section 3.3.

4.1.1 Modeling hill shape perturbations

If the hill shape is completely constant over time and over various operating conditions, then there is no need for re-optimization, as the optimum control voltages determined during the initial training stage remain optimal. In practice however, the optimum voltages may change slightly under different operating conditions. It was mentioned earlier that when we model the hill with a quadratic functional form, quadratic coefficients naturally represent the general shape or the curvature of the hill while the linear coefficients represent the location of the hill in the four dimensional space. For our

algorithm, we made an assumption that despite a possibility of the hill's position shifting over time or an effect of any small perturbation that might happen on the system, the general shape of the hill should remain roughly the same. This is justified since the shape of the hill is mostly determined by the physical design and implementation of the Gaussian beam propagation between the input and the output fiber. The parameters for microlenses are chosen so that when the mirror angles are optimized, the light coming out of the input fiber is focused at the core of the output fiber. As either mirror tilts away from the optimum position, the focused light spot at the tip of the output fiber is displaced from the core, leading to non-optimal coupling and increased loss. The pattern of this process determines the shape of the hills, and they are mostly determined by the parameters of the microlenses and how the fibers are aligned with respect to them. Since these parameters are relatively constant over the lifetime of the switch we can safely assume that the quadratic coefficients stay roughly constant throughout future attempts to reestablish or maintain the connection. The linear terms, however, should be allowed to change freely in order to capture the possible movement of the position of quadratic response function as operating conditions of the switch change. Three possible scenarios which could alter the voltage conditions to achieve an optimum connection are the following.

First is the case where the whole voltage/beam displacement curve shifts in voltage. A possible cause for this is the charging and discharging behavior of the mirrors. It has been noted that some of the mirrors we have seen has a finite settling time (between the voltages are first applied and the mirror reaches a steady-state angle), due to charging of the dielectric material around the electrodes. Since it takes some time for the charges to accumulate on the dielectric material, the voltage required to achieve any specific angle tilt or beam displacement when the mirror plate is free of charge is likely to be different from that of when the mirror plate has already had a chance to accumulate charges and reached a steady state. In this case, the voltage/beam displacement curves from these two different instances would have the same general shape but they will be a voltage-shift copy of each other. We can see that the slopes of the two curves at the same height or beam displacement space value would still be the same. This means that although the voltage positions of the peak are different in the two cases, the incremental

change in voltage around the peak would result in the same incremental change in angle. The general shape of the hill should thus be the same for the two cases as well, while the location of the hill gets changed.

To set this scenario up mathematically, let's define the hill as a function of voltage as $T(v) = f(g(v))$. Here, $x = g(v)$ is the voltage/beam displacement curve, and $f(x)$ is the power as a function of displacement. In the voltage shift effect, $g(v)$ is replaced by $g_1(v) = g(v-dv)$. Then $T_1(v) = f(g_1(v)) = f(g(v-dv) = T(v-dv)$. So the hill T_1 is a shifted version of the original hill T.

Second case is when the operation temperature of the switch is modified. In our implementation, the change in output voltage from the high voltage digital-to-analog converters (HVDACs) as a function of temperature is negligible. However, the overall switch core might experience an expansion or contraction as the temperature changes, resulting in a different angle requirement to optimize a connection as a function of temperature. The mechanical design of the switch core was done to minimize this effect, but the angle requirement is so stringent that a considerable temperature change will result in shifting of the hills. We argue here that the change in the parameters affecting the hill shape (for example, focal length of the microlenses, or the distance between the microlens and the fiber) over a similar temperature range is much too small to modify the shape of the hill considerably.

Third case is the wavelength dependent effect. This happens when the light sources used at two different instances have different wavelengths. In this case, different wavelengths will result in different degree of light diffraction due to the dispersion of the field lens, and therefore the same voltage value applied to the mirrors might not result in the same light beam spot position. This would result in a vertical shift in the voltage/beam displacement characteristic curve. Again, assuming that the shift or the change in the scaling is small, the slope at any specific angle value should not change significantly. Therefore, we argue that in this case the shape of the hill should remain roughly the same as well.

This can be explained by the same mathematical terms. The function g as defined above will be replaced by $g_1(v) = g(v)+y$, where y is the beam position offset due to the vertical shift in the voltage/displacement relationship. If $g(v)$ is accurately modeled as

$g(v) = g(v_0) + J_0(v-v_0)$, for v near v_0, then $g_1(v) = g(v_0) + J_0(v+J^{-1}y-v_0)$. So $g_1(v) = g(v+dv)$ where $dv = J_0^{-1}y$. Now $T_1(v) = f(g_1(v)) = f(g(v+dv)) = T(v+dv)$. The hill T_1 is again just a shifted version of the original hill T.

The three cases above are presented to show that our assumption of the hills retaining their shapes regardless of any effect from drift and small perturbation should hold. Another way to think of this is that small perturbation of any kind can be roughly modeled as a linear shift of the hills. In our case, by keeping only the quadratic coefficients, we hope to capture the effect of perturbation by allowing the linear coefficients to change in value.

4.1.2 Description of the five-point algorithm

Now that we have justified the idea that the quadratic coefficients of the hill shape are relatively stable, we can go on to the details of the re-optimization algorithm itself. The goal here is use knowledge of the hill shape to be able to get to within some small distance from the peak as quickly as possible.

Before we go into the details, let's remind ourselves again of one of the three equivalent forms of the four-dimensional quadratic function that we used to model our hills:

$$T = \tfrac{1}{2} A^T \mathbf{Q} A + \underline{u}^T A + c \qquad (4.1)$$

where $\mathbf{Q} = \begin{bmatrix} 2a_6 & a_{10} & a_{11} & a_{12} \\ a_{10} & 2a_7 & a_{13} & a_{14} \\ a_{11} & a_{13} & 2a_8 & a_{15} \\ a_{12} & a_{14} & a_{15} & 2a_9 \end{bmatrix}$, $\underline{u} = \begin{bmatrix} a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$, and $c = a_1$

Note that $A = (x_1, y_1, x_2, y_2)^T$ is a column vector of the four beam displacement space components. When written in this form, the peak position $\underline{p}$ of the hill based on the quadratic model can be expressed as:

$$\underline{p} = \mathbf{Q}^{-1}(-\underline{u}) \qquad (4.2)$$

Since we imposed the requirement that all the quadratic terms or the **Q** matrix are already known, the only parameters left to be fitted are $\underline{u}$ and c. When Q is known, $\underline{u}$ and c are

unknown, we can write $\underline{u}^T A + c = T(A) - 1/2 A^T \mathbf{Q} A$. Because $\underline{u}$ is a vector consisting of four parameters, the total number of actual parameters to be fitted is five. We are required to take at least five additional measurements or steps before we could take advantage of the quadratic model to predict the peak position of the hill. That is, given the value of $\mathbf{Q}$ we need five additional measurements to obtain the value of $\underline{u}$ and c, and only after we have obtained $\underline{u}$ that we could find the peak position prediction $\underline{p}$. This is why we call this algorithm the five-point algorithm.

Given a set of five measurements, the question now is whether or not there is a unique set of parameters $\underline{u}$ and c for these five measurements. A unique set of parameters would exist if the five points were well distributed in the problem's four-dimensional space. In other words, we should try to avoid picking points that lie too close to any subspace of dimension less than 4. One general technique that ensures a valid set of four measurement positions is to choose the four points in a set of four orthogonal directions. Taking one start point, plus points coming from steps in four orthogonal directions, will ensure that the measurement matrix is non-singular.

The size of the measurement steps is also another important issue. If the measurement is noisy, we want to make sure that the step size is big enough to reduce the effect of noise. At the same time, we do not want the step to be so big that we risk falling off the hill too far when taking the measurement, to a point where our quadratic modeling is not accurate. Large power hits are also to be avoided if the connection being re-optimized is carrying live data. The eigenvectors of the $\mathbf{Q}$ matrix are a useful set of orthogonal vectors when it comes to choosing the step sizes. We could use knowledge about the curvature of the hill in each eigenvector direction to determine an appropriate step size for the measurement. This knowledge is embedded in the eigenvalue that corresponds to each particular eigenvector. If the magnitude of the eigenvalue is big, we know the hill is steep in that direction. We should then be cautious and use only a small step size. If eigenvalue is small in magnitude and the hill is shallow, then we can afford to take a larger step. So the step size should be inversely proportional to the magnitude of the eigenvalue. For our application, we decided to use $\varepsilon/\sqrt{\lambda}$ as our step size, with ε being any small constant and λ being the eigenvalue of that direction. In this case, we use $\varepsilon = 1$. The reason for choosing this step size is that if the eigenvalue does represent the

curvature of the hill in that direction well, walking in that eigenvector direction with this step size will move us up or down the hill by roughly 1 dB. And 1 dB seems conservative enough for our application. Now that we have considered the two important issues regarding the positions and the sizes of the steps, we are ready to describe the re-optimization algorithms.

In order to maximize the transmissivity T, we propose measuring the current power T_0 at the current (base) position A_0 with the current offsets $\underline{u}$ and c as unknowns. Then we take a step proportional to the inverse square root of an eigenvalue of **Q** in the direction of its corresponding eigenvector. We refer to this as position A_1. Now, see if the measured power at this position $T_1 > T_0$. If it is, then set A_1 as a new base, otherwise, leave the base at A_0. Then take steps in directions of other eigenvectors from the base point, with step sizes proportional to the inverse square root of the corresponding eigenvalues, measure power, reset the base if the new point is higher in measured power than the current base. This process would give us a set of measurements T_i, with i = 0,1,2,3,4, taken at positions A_i. These measurements are enough to uniquely determine the values of $\underline{u}$ and c. Once we know $\underline{u}$ and c, the peak position $\underline{p}$ can be easily predicted. We then take another measurement at $A = \underline{p}$ to confirm if this position indeed yields the maximum power or is within an acceptable range from the maximum spot. If measurement noise is significant, we can greatly reduce its effect by overfitting $\underline{u}$ and c with a few additional measurements.

If the start point A_0 is very far from the peak, we may find that our algorithm does not converge to the peak within 6 measurements. This is understandable, considering that the areas further away from the peak are not modeled well by a simple quadratic function. In a case like this, we have found that continuing the algorithm can lead to the peak quickly. We use the 6 measurements to predict the position of the seventh measurement point. If the seventh measurement is still not close enough to the peak, we use all 7 points to make another prediction. If we are still not there, we keep going in the same manner. As the algorithm continues, we oversample by keeping the seven highest power measurements in order to reduce the effect of noise. We keep the highest points because we believe the closer the measurements are to the peak, the better our model's prediction of the peak location. It is important to note that after step 7, if the algorithm is still

continued, we have to remember to step out in eigenvector directions once in a while to ensure that the set of 7 highest measurements we keep still cover the whole 4-dimensional space well.

Now we go back to the issue of picking the first 5 measurements. In addition to scaling the step size according to the eigenvalue magnitude, we could also choose to measure points in the four eigenvector directions in a particular order to help reduce the possibility of falling off the hill too far while taking measurements. As mentioned before, a measurement point that is too far from the peak is not as useful or effective when it comes to predicting the peak position. Because the base point keeps moving closer to the hill peak as the algorithm progresses, the risk associated with falling off the hill too far while taking measurements gets smaller as well. Since the most risky stage is at the very beginning, meaning the general slope of the hill at this point is likely to be the steepest, a slight error in terms of step size could result in a big fall. So we choose to go in the eigenvector direction with the smallest eigenvalue first. As the base point gets closer to the peak, the hill is less steep and we are less sensitive to error in step size. Hence, we go with the increasing order in the eigenvalue magnitude.

The algorithm is summarized again below.

1. Retrieve the quadratic coefficients a_6-a_{15}, or the **Q** matrix, from memory.
2. Retrieve the first position point from database. Take power measurement at this point. Set index i = 1. Set the base point x_b to this position.
3. Take measurement at $x_i=(x_b + \varepsilon/\sqrt{\lambda_i})$ in the direction of eigenvector i, with i = 1 being the direction with smallest eigenvalue and i = 4 being the direction with the biggest eigenvalue. λ_i is the eigenvalue of the eigenvector i. ε can be any small constant.
4. If the power L_i is greater than the base point power L_0, then set base point to this new position. Otherwise, leave the base point at the same position.
5. Set i to i+1. If i <= 4, then go to step 3. If i = 5, go to step 6.
6. Set index i = 0. Use the least square fitting technique on up to seven highest measurements to obtain the values of vector $\underline{u}$ and offset c.
7. Use the resulting $\underline{u}$ to predict the peak position according to equation 4.2

8. If the predicted peak position is within a reasonable distance d (d = 0.1 mirror units for example) from the base point, then measure the power at the predicted position. If the predicted peak position is not within a reasonable distance from the base point, we move in the predicted peak direction but scale the step size down so that we are not moving more than distance d from the base. Measure the power at this new position.
9. If the power at this predicted peak position is within 0.5 dB of the real peak power, stop. Otherwise, set i = i+1 and go to step 5.

4.1.3 Variations of the five-point algorithm

In addition to the standard form of the five-point algorithm, we also explored some variations of it. The motivation here is to see whether we can come up with a slightly different algorithm that uses the same principle of utilizing prior knowledge of hill shape to reduce the number of measurements required before getting to the peak, but also increases the accuracy of the peak position prediction. The two notable ones are what we call the bi-directional algorithm and the line search algorithm. They are described below.

a) **The bi-directional algorithm** – We mentioned before that since the simple quadratic function does not model the actual hill far from the peak well, we would like to have a measurement that is higher in power, or closer to the hill peak, than the base point for each eigenvector direction. An eigenvector only specifies a general line that our new measurement position should be on, but it does not specify whether we should go left or right on that line. Therefore, when we pick to go to the right on that eigenvector line, for example, we could be either climbing up or falling down the hill. This bi-directional algorithm specifies that if it turns out that the step we took results in us falling down instead of climbing up, we should then turn around to take another step of the same size in the opposite direction to make sure we get one measurement point higher up than the base point on that eigenvector line. If the first step we take in an eigenvector direction is, however, already higher than the base point, we can move on to the next

eigenvector direction in a normal manner as in the five-point algorithm without having to turn around to take another step in that same eigenvector direction. So the total number of measurements taken before we can make the first peak position prediction can range from 5 to 9, since there could be up to 2 measurements per eigenvector direction.

b) **The line search algorithm** – This is a slight improvement over the bi-directional algorithm. It has the same motivation of trying to make sure that we have a measurement that is higher up than the base point in each of the four eigenvector directions. But instead of just turning around and taking another step with that same standard step size, we could try to use the base point and that "wrong" measurement (the measurement in that same eigenvector direction that has lower power value than the base point) to sensibly predict the position of the highest point on that eigenvector line. In other words, the extra step taken in each direction would not just be a random step; rather, it would be at a position predicted to be the peak in that direction. The prediction is made based on the one-dimensional quadratic model in that eigenvector direction. Similar to the bi-directional algorithm, the total number of measurements taken before we can make the first global peak position prediction ranges from 5 to 9. But in this case the probability of the base point being close enough to the actual peak before we even make the first peak position prediction is higher than both the five-point algorithm and the bi-directional algorithm. This is because it involves more intelligence while picking the measurement positions. The base point should keep moving closer to the peak in at least every third step and we are thus more likely to be close enough to the peak before we have gathered all measurement points in all four eigenvector directions.

Comparison in terms of performance of the three hill climb algorithms is shown in Figure 16 below. The three algorithms were tested using measurement-based simulations. A dense set of sample points around the peak were taken from a set of connections, and we interpolated this set of sample points to obtain smooth functions

describing power as a function of displacement. Perturbations in hill position were modeled by using random starting point for the re-optimization algorithms. The simulated hill measurements did not include any simulated measurement noise.

There are two things involved in the testing here. First is how fast the algorithm could get to the peak. Second is how robust the algorithm is to error in the hill's quadratic coefficient values. There are two main sources of error. First comes from the fact that the set of coefficients we obtained through the least square fitting technique on a set of 81 sample points around the hill peak is not necessarily the best possible set to begin with. That is to say, with a limited number of measurements, the model does not necessarily give the best possible approximation of the real hill. There is a clear tradeoff between the number of sample points we are willing to take (in the factory) and the accuracy level of the obtained coefficient values. Second, we discussed in section 3.3 different methods to model the variation of the coefficient values across the chips. For very large port-count crossconnects, it is clearly impractical to extract coefficient values for all the connections and explicitly store all these values (which already contain some error to begin with as mentioned above) in memory so that they could be retrieved whenever a hillclimb algorithm is called. It makes more sense to store only a model that could be used to estimate the coefficient values for all the connections. The tradeoff here is between storage space savings and accuracy of the retrieved coefficient values. Therefore, if we choose to reduce the storage space requirement by storing only a model instead of all the actual coefficients, we have to pay the price of introducing error into the value of retrieved quadratic coefficients. Our hillclimb algorithm of choice should thus be able to tolerate this error introduced in the retrieved coefficient values decently well.

The histogram in Figure 16 shows the number of steps or measurements required for each algorithm to get to within 0.5 dB from the peak height given that the start point is randomly distributed within 3 dB from the peak. We could clearly see that the five-point algorithm always get to the peak within 6 steps. A big majority of the trials got to the peak in the 6^{th} step, and it is as expected because the 6^{th} step is the prediction step. The starting point was close enough to the peak that even a step in the wrong direction, meaning a sample point is further down the hill than the base point at the time, could still give us good enough information to predict the peak position. The bi-directional

algorithm was the one expected to improve the accuracy of the peak position prediction and to help us gain more useful measurement points. It did indeed improve the accuracy of the predicted peak position, but it comes at the expense of having to go through a lot more intermediate measurements before we could make the peak position prediction. So in the case where the start position is not too far from the actual peak position, we see that the bi-directional algorithm is not considered an improvement over the standard five-point algorithm. The line search algorithm is on the other hand an improvement in some aspects. Since the second measurement in any eigenvector direction is at a carefully chosen position which is thought to be the peak in that direction, the base point moves closer to the peak at a faster rate than the previous two algorithms. We can see that higher percentage of trials get to the peak before the 6^{th} step than in the five-point algorithm. But since the number of intermediate measurements can range from 5 to 9, the 6^{th} step is not always the prediction step and some trials thus require many more steps before the peak is reached.

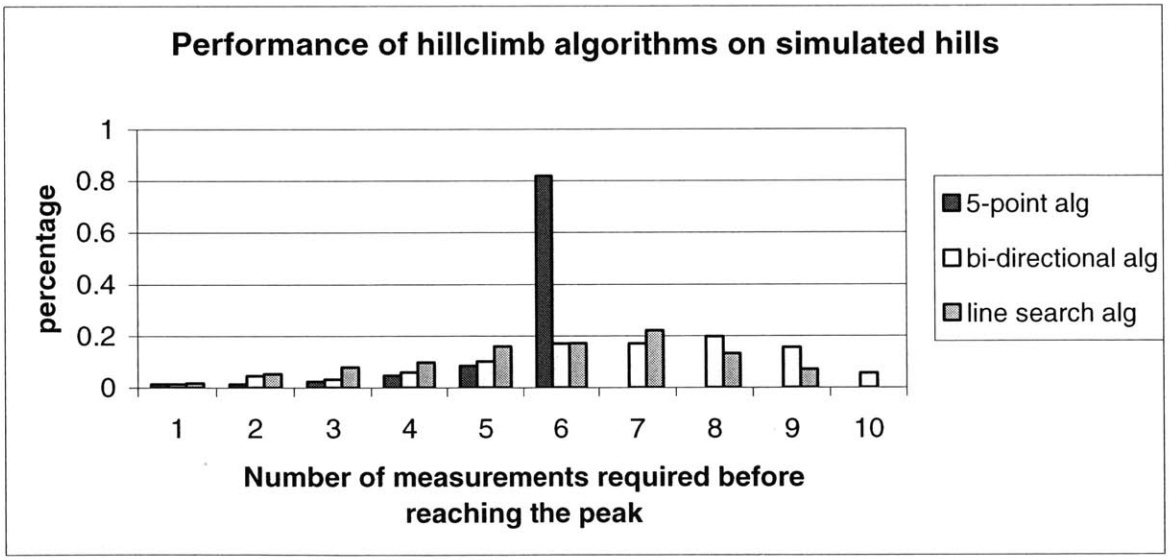


Figure 16: Performance of the three hillclimb algorithms on simulated hills

According to our results above, in the case where the start point is not too far off from the peak, the five-point algorithm seems to be the most efficient. Although a large majority of trials have to wait until step 6^{th} to reach the peak, no trial takes longer than 6 measurement to get to the peak of our stationary hills. The prediction step is the crucial

step and we should focus on getting to the prediction step as quickly as possible instead of hoping that the base point would move close enough to the peak during the measurement process. The five-point algorithm lets us get to the prediction step with the least number of intermediate steps required. We will therefore from now on focus on the five-point algorithm.

4.1.4 Further studies on the five-point algorithm

This section presents further studies done on the five-point algorithm on simulated hills. The purpose is so that we learn and gain as much insight about the algorithm performance as possible before implementing it on a real system.

We have mentioned before that the goal of this project is to come up with an algorithm that offers a great improvement over other previous techniques used in terms of the number of measurement steps required before reaching the peak. It might be useful to show an evidence of that by showing a performance record of one of the previous hillclimb techniques used for a comparison. Figure 17 below shows an algorithm performance of the Nelder-Mead simplex algorithm. This simplex algorithm is a different approach and it does not require or rely on any prior knowledge about the hill being climbed. The plot below has the start point randomly distributed within 3dB from the peak, and the algorithm stops climbing when it is within 0.5 dB from the peak.

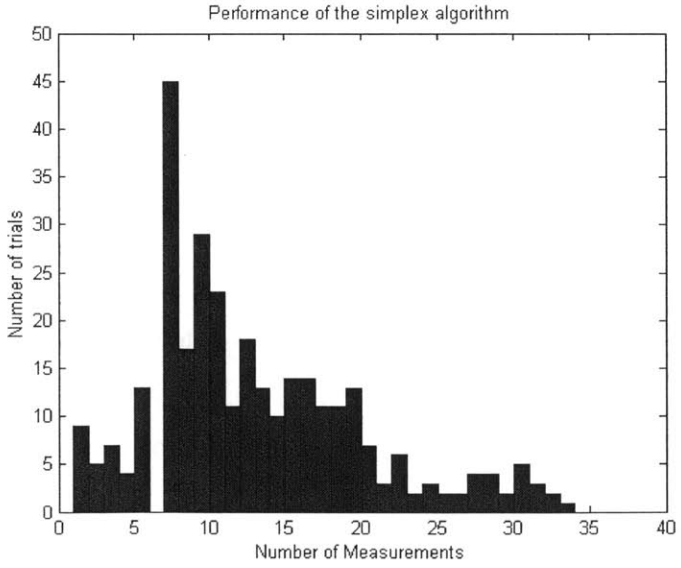


Figure 17: Performance of the Simplex algorithm

The advantages of using the simplex algorithm for our application are that it is robust and it does not require any additional data, advance preparation, or any storage space. In other words, it does not require any hill shape data. But the downfall of it is obvious. It requires many steps to reach the peak. Given start points that are within 3 dB from the peak, most of the trials took many more than 6 steps get within 0.5 dB from the peak. The worst-case scenario could take up to almost 35 steps

Now to give a comparison, a performance graph of the five-point algorithm is shown here in Figure 18. Here, the plot contains three sets of data. Each set has a different location bound for start points. Start points are randomly distributed within 3 dB from the recorded peak for the first set, within 4 dB for the second set, and within 5 dB for the third set. Contrary to the simplex method, all trials are able to get with 0.5 dB from the peak within 6 steps, even when the start point is a further down as 5 dB from the peak.

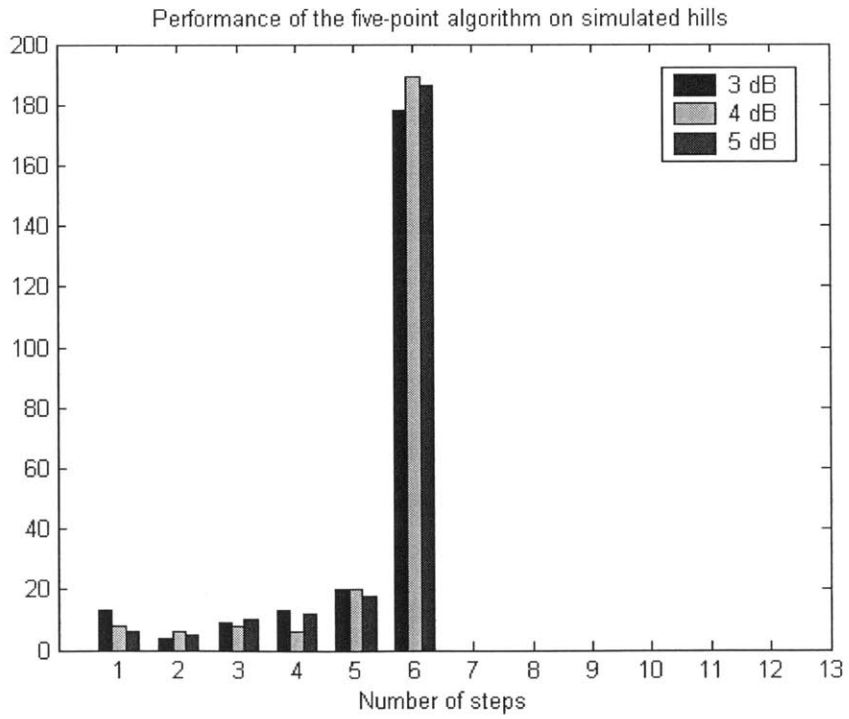


Figure 18: Performance of the five-point algorithm on simulated hills

The similarity among the 3 sets of data shown in Figure 18 above shows the robustness of the five-point algorithm against the variation of the start point position. In other words, the performance of the algorithm does not deteriorate much when the start point is further down the hill.

Another type of robustness that we want to evaluate is the algorithm's robustness against errors in the values of the retrieved hill coefficients. We want to see how accurate the given hill coefficient values have to be for the algorithm to give a good performance. So we added noise into the hill coefficient values and used these noise-added coefficient values together with the five-point algorithm to climb the hill. The results are shown in Figure 19. The noise added had a normal distribution with the standard deviation specified as a percentage of the coefficient magnitude. The standard deviation of the noise added is different for each of the four cases as noted in the figure. Again, there are 3 sets of data in each plot. Each set has a different location bound for start points. Start points are randomly distributed within 3 dB from the recorded peak for the first set, within 4 dB for the second set, and within 5 dB for the third set. It should also be noted that on the horizontal axes of these plots, the 13 bar represents trials that required 13 or more measurements.

It is clear from Figure 19 that the algorithm performance does deteriorate as the errors in the hill coefficient values used get bigger. This is as expected, because the algorithm relies on the information provided by the coefficient values to give a prediction of the peak position. The quality of the prediction does depend on the accuracy of the coefficient values. But again, the level of deterioration is not significant until we added the noise with the standard deviation value of 20%. The last plot on the bottom right is when we start to see some noticeable tail. So we can conclude here that the five-point algorithm is robust enough against small errors in the coefficient values used.

So far all the evaluations we have shown were done on simulated hills. Now that we have gained a lot of insights about our algorithm of choice, the five-point algorithm, the next step would be to implement the algorithm and evaluate its performance further on a real switch. This will be the focus of Chapter 5.

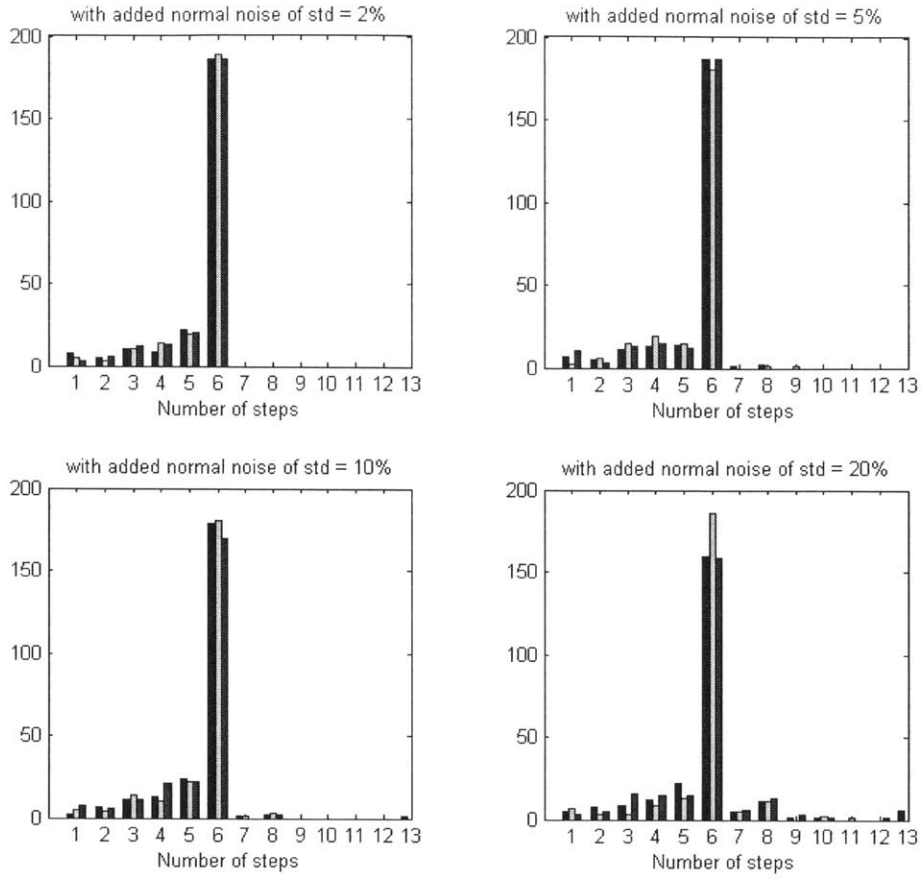


Figure 19: Performance of the five-point algorithm with added noise in the hill coefficient values

4.2 Other applications of the hill shape studies

It should be pointed out that there are other possible applications besides the hillclimb algorithm that could also benefit from the hill shape studies. One promising application is an ability for the switch to introduce variable loss into the optical signal. This way, the optical switch can perform power equalization as well as its switching function. Introducing variable loss into the signal can be done simply by sliding off the peak of the hill. There are certainly many ways and many paths to slide down the hill. The question is how can we slide down the specified height in the most efficient manner and without ever letting the overall signal loss go beyond the intended loss. By understanding hill shape, we should be able to find a path to slide down that is the most conservative. It is important to note that one intended loss value does not specify just one

specific point on the hill. Rather, it specifies one whole contour which consists of all the points that have that same specific height. So we can pick to operate at any point on that contour. A connection operating on the side of a hill will typically be more sensitive to perturbations than a connection sitting at the peak. Using the knowledge gain from the hill shape studies, we should be able to find the most stable operating point for any height specification. Although this variable attenuation application was not explored in this project, we believe it has a great potential and could be the focus of further studies.

Chapter 5: Performance evaluation on real systems

An important aspect of this project is quantitative evaluation of the performance of the optimization algorithm on real systems. Conditions on real systems are far from ideal. The first section of this chapter will discuss some of the systems' constraints and non-ideal conditions, as it is crucial for us to realize and understand what effects they will have on the performance of our algorithm. The following section will then present the actual results and analysis of the algorithm performance.

5.1 Real system's constraints and non-ideal conditions

In previous sections so far we have done all our preliminary testing on simulated hills. The testing gave us an opportunity to gain insights about the general nature of our hillclimb problem without having to worry about the added complexities of the real system. Once we have gained those insights and made decisions accordingly, it is time to move on to the real system. The algorithm is not useful if it could only work well on simulated stationary hills. As mentioned, our switching systems have many non-ideal characteristics or conditions that could potentially affect the performance of the hillclimb algorithm. In this section, we will discuss two major conditions; the mirror settling time and the discretization noise.

5.1.1 Mirror Settling time

Mirror settling time depends largely on physical properties of the mirror and the spring or flexures that connect it to the chip. We consider the mirror and the springs connecting it to the chip as a mechanical harmonic oscillator system. Most of our systems have shown the characteristics of underdamping. It takes several oscillation periods for the system to settle to its final state. But the resonance frequency of our system is also shown to vary as a function of the mirror's tilt angle. This phenomenon is known as the electrostatic softening of the springs. The resonance frequency decreases as the tilt angle increases. So the connections involving the mirrors at the edges would naturally take longer to settle than those involving the mirrors in the middle of the chips.

Past experimental data has suggested that the time it takes for the mirror to settle or to reach its steady state should roughly be a factor of 2-3 from the resonance frequency. Since the system controller in the past did not have the speed to control the mirror so precisely in terms of time, only the "order-of-magnitude" estimation was thought to be sufficient. We used this rough factor of 2-3 from the estimated resonance frequency to determine the length of time we should wait for the mirrors to settle in general. However, as the system controller becomes more advance and could handle more precision, it has become quite clear that the factor of 2-3 that we assumed and the order-of-magnitude estimation might no longer be good enough. We have experienced that the hill shape could look significantly different if we do not allow enough time for it to reach its final state before taking the reading. While the mirrors in the middle might be able to comfortably reach its steady state within a specified waiting time, the edge mirrors might require a lot longer. An example is shown in Figure 20 below. The connection in the figure above involves two mirrors close to the edge of the switch fabric of size 1296x1296. Again, for each plot we fix the value of two variables and vary the other two. So each plot is a two dimensional slice of the real hill. The two surfaces on each plot represent the hill slice profiles taken after different waiting times. One was taken after 10 ms of waiting time, while the other was after 20 ms.

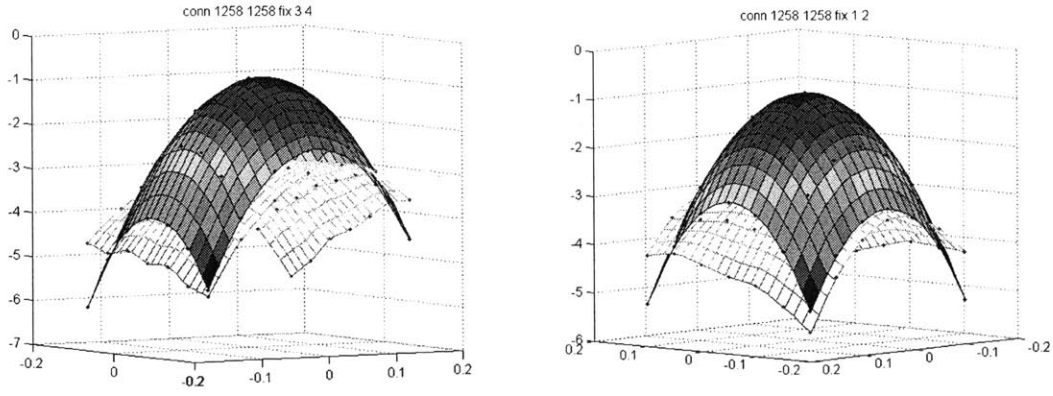


Figure 20: Hill slices of an edge connection taken at two different instants

We can see clearly from the plots that the shapes or curvatures of the hill from the two instances are vastly different. Since the curvatures are different, the hill's quadratic

coefficients from these two different instances are bound to be very different as well. The fact that these two sets of coefficients are not the same will violate our hillclimb algorithm's assumption. The problem of history dependence is introduced here: the mirror position depends on where it has been and is no longer a simple function of current voltages. For this reason, we have to make sure that we leave enough time for the mirrors to settle both when we collect the hill shape data at the manufacturing site and when we climb the hill at the customer's site. Otherwise we might be using a set of hill shape parameters that have little to do with the hill being climbed.

As mentioned earlier, this issue of mirror's long settling time is more likely to affect the connections involving edge mirrors more than those involving center mirrors. Figure 21 shows the two hill slices each taken after 10ms and 20 ms wait time of the connection involving two mirrors close to the middle of the chips. We can clearly see that the two surfaces are almost on top of each other. They look much more similar to each other than those of the connection shown in the previous figure. In this case, the quadratic coefficients from the two instances are very similar and the assumption made by our hillclimb algorithm about the quadratic terms retaining their values over time is well justified here.

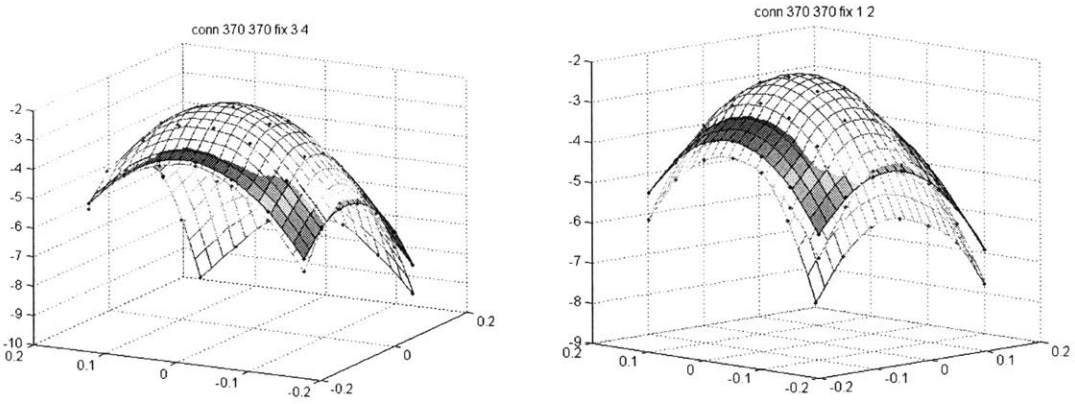


Figure 21: hill slices of a middle connection taken at two different instants

5.1.2 Quantization Noise or Finiteness of granularity of steps

Although the mirror's tilting and the actual optical power of the switch are on the analog scale, the processing of information and computations are done in the digital realm. The signal sent to the HVDAC which generates the exact voltage to control the

mirror's tilt angle is a digital signal. So is the output of the ADC that gives us the value of the detected output power. Therefore, quantization noise is bound to exist. The HVDAC and the ADC introduce noise into the system in slightly different ways. This is shown in Figure 22 below.

Here we show the hill profile as a one-dimensional hill, with the horizontal axis specifying the voltage and the vertical axis specifying the optical power of that hill. The noise from HVDAC would create some uncertainty in the applied voltage. In other words, it is considered a horizontal noise on the figure. The corresponding uncertainty in power depends on the slope of the hill. The ADC on the other hand introduces uncertainty in the value of the optical power, so we can view it as a vertical noise.

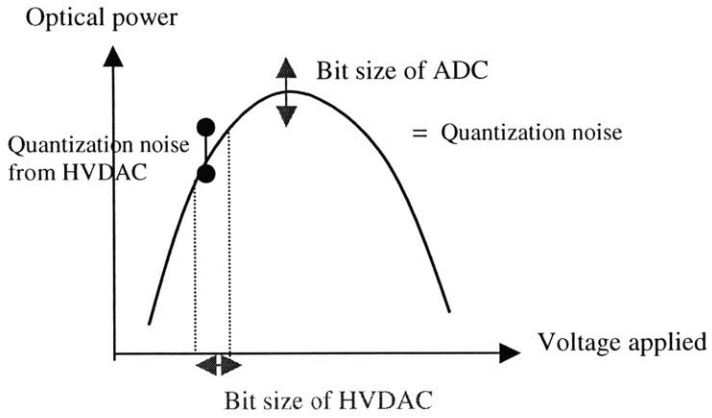


Figure 22: Vertical and Horizontal noise

The question now is whether all these noises are significant enough to affect the performance of our algorithm. One way to determine this is to compare the size of the steps we take in the hillclimb algorithm to the size of one digital bit. If the former is much larger than the latter, we can assume that the effect of quantization noise is not very significant. In our case, one digital bit corresponds to roughly 25 mV. The specification for the worst-case sensitivity for the mirror angle vs. voltage of our mirrors is 0.3 degrees/volt. 0.3 degrees roughly correspond to 1 mirror unit (center-to-center spacing) in beam displacement space. The mirror design typically achieves 0.1-0.2 degrees/volt, even in the worst case of process variation. Therefore, the requirement in the current mirror design is enough to avoid the susceptibility to quantization noise. For the ADC side, since

we are using the log (dB) scale while the amplifiers that follow the photodiodes have linear gain, the quantization noise from the ADC should be insignificant as we operate near the top of the dynamic range of the ADC. So we conclude here that both the vertical and horizontal noises do not have significant effect on our hill-climbing procedure.

5.2 Real system's performance and analysis of the performance

In this section, we now describe the performance evaluation of our algorithm on real systems. We ran our algorithm on two types of real systems- the 1296x1296 switch model and the 64x64 switch model. Aside from the size difference, these two systems have different characteristics that allow us to test different aspects of the hillclimb algorithm on. For example, the 1296x1296 switch is very large, so it gives us a perfect setting for testing the idea of using a model to estimate the values of the hill coefficients instead of storing and retrieving the actual hill coefficient values for each and every hill. However, at the same time the tilting range of each mirror has to be reasonably big since it has to be able to reach the edge of the opposite chip. While at the high tilting position, we run into the risk of having problems with long mirror settling time and greater sensitivity to error of the model. In other words, we get to see traces of problems associated with scalability of our modeling and algorithm approach. The 64x64 switch is, on the other hand, of a very manageable size, so each mirror does not have to tilt much in order to reach its mirror pair on the edge of the opposite chip. We have an advantage of not having to worry about the problem associated with big tilting angle. But the idea of using interpolated hill coefficients instead of actual ones is not that as helpful or relevant here.

We will discuss the results of our algorithm performance on each of these two systems separately in the two sections below.

5.2.1 Performance of the algorithm on the 1296x1296 switch

Before we get to the stage of performance evaluation, major efforts had to be devoted to the implementation of the algorithm. The algorithm was implemented, using the system controller software as an interface between the algorithm itself and the

hardware of the switch. The default waiting time used by the system controller for the mirror to settle to its steady state was 10 ms. As mentioned before, since the switch has a large size, it is a good setup for us to see whether the hillclimb algorithm is robust enough to tolerate errors in the retrieved quadratic coefficient values introduced by the estimation model.

The implementation of this hillclimb scheme can be separated into two parts. First is the part that needs to be executed at the manufacturing site of the switch. Details of this have been discussed earlier in section 3.3. We take a dense set of 81 data points around the hill peak of 12^4 connections of the switch. Recall that the switch of size 1296x1296 consists of two MEMS chips, each having 1296 micromirrors on it, arranged in the 36x36 array pattern. The 12^4 connections are picked so that the 12^2 input mirrors that are involved form a grid that contains every third mirror on the input chip, and the 12^2 output mirrors involved form the grid in the same nature on the output chip. The pattern of the mentioned grid in shown in Figure 23 below. This way we can be certain that the connections we use as a basis to construct the coefficient estimate model cover the whole chips and represent all parts of the chips well.

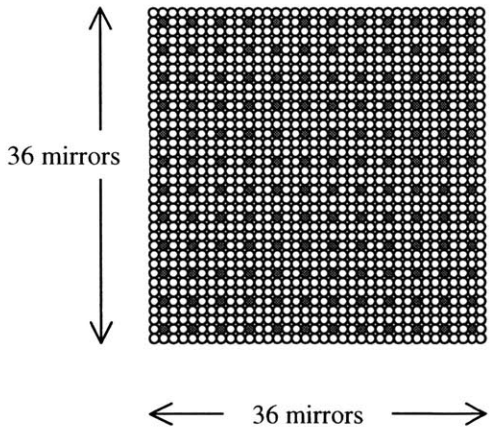


Figure 23: Grid pattern of mirrors used in extracting the hill quadratic coefficients

We used 81 points in each hill to each hill to extract the hill coefficients for that particular hill. After this step, we now have 12^4 sets of extracted hill coefficients. For each set, using the argument discussed in section 4.1.1, we decide to keep only 10 quadratic terms out of 15 total terms. We model each of these 10 terms as a quadratic function of the connection's input and output mirror locations and use the least square technique to extract the value of this model's 15 parameters. Since there are 15

parameters required to reconstruct the estimation model for each of the 10 hill coefficients kept, the total number of parameters we need to store is 15x10 = 150. We store these 150 estimation model parameters in memory so that they could be readily available when any connection is to be restored in the field. At the manufacturing site, we also need to find the position of the hillpeak or the optimal values of the hill's four control variables for all 1296x1296 connections and store these values in the database. We also store the optical power of each connection at its peak in the database.

The second part of the hillclimb implementation is to be executed in the field during operation. The implemented algorithm for the 1296x1296 switch proceeds in the following steps.

1. Retrieve the estimation model that is used to estimate the quadratic hill coefficient values from memory.
2. Find out which mirror pair is involved in the connection. Use the location of the mirror pair to obtain the estimate of the quadratic hill coefficients, a_6-a_{15}, for this particular connection.
3. Use the database and the location of the mirror pair to obtain the $\mathbf{J}$ and $\mathbf{J}^{-1}$ matrices to be used in the voltage-beam displacement and beam-displacement-voltage transformations. Store the values of $\mathbf{J}$ and $\mathbf{J}^{-1}$ in local memory so that they are readily available for use whenever space transformation is needed for this connection.
4. Retrieve the old optimal values of this connection's four control variables from database and apply them to the connection's two mirrors.
5. Apply the 5-point hill climb algorithm (using the steps presented in section 4.1.2).

Because the wavelengths, temperatures, and other factors in the laboratory were fairly constant, the hills held a constant optimal position. To simulate changes in hill position, we added random offsets to the optimal voltages retrieved in step 4. In this way we were able to measure the robustness of our algorithm while increasing the size of the offsets in a controlled manner.

With the implementation following the guideline above, we are now ready to evaluate the performance of the algorithm. We break the evaluation process into two cases: one is on the connections close to the middle of the chips or in the middle zone, the other is on the connections close to the edge of the chips or in the edge zone.

a) Performance on connections in the middle zone

The performance of our hillclimb algorithm on connections in the middle zone is shown in Figure 21 below. We ran the algorithm over 5,000 trails on more than 100 connections in this zone. We plot the distribution of the number of steps or measurements the algorithm requires in order to reach the peak on these connections. The plot contains three sets of data. Each set has a different location bound for start points. Start points are randomly distributed within 3 dB from the recorded peak for the first set, within 4 dB for the second set, and within 5 dB for the third set. For performance evaluation purpose, what we refer to as reaching the peak is not reaching the exact value of the recoded peak. We stop the algorithm and declare a connection good whenever we are within 0.5 dB from the recorded peak. This decision is justified because our goal is to use this accelerated hillclimb algorithm to get us to within an acceptably small range from the peak as fast as possible during the setup or restoration of the connection. Once we are within an acceptable range, we can as much more time as we want to climb up to the absolute peak.

The plot shows a significant jump in the number of trials that reach within 0.5 dB distance from the peak at the 6^{th} step. This is as expected since the 6^{th} step is the algorithm's peak position prediction step. More than 80% of the trials get to within 0.5 dB from the peak within 6 steps. The majority of those that require more than 6 steps get to the peak within 7 steps. This is clearly a major improvement over the previous method used, i.e. the simplex hillclimb method, which almost always took a lot more than 6 steps for all trials regardless of how close the start point was to the peak.

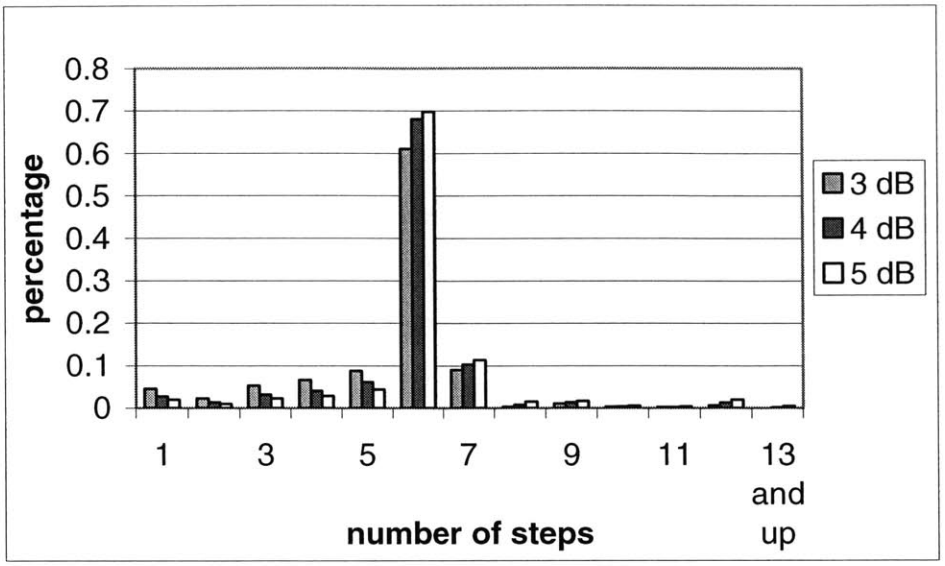


Figure 24: Performance of hillclimb algorithm on connections in the middle zone

Comparing the three data sets on the plot, we can see that cases with start points closer to the peak require fewer number of steps to reach the peak in general. This argument is supported by the fact that the distribution of the first set is more front-loaded than the second set, and that of the second set is more front-loaded than the third set. Regardless of that, however, the plot clearly shows that our hillclimb algorithm is robust enough to handle start points 5 dB down from the peak without sacrificing much of its performance merit. Although we do not expect the old optimal values stored in database to ever be as far down as 5 dB from the peak, the result of this testing gives us additional confidence on the robustness of our algorithm.

Possible reasons for the tail of the distribution after step 6 are the following. First, it might be the result from the errors introduced in the quadratic coefficients that we derived from the estimation model or that certain hills may not be well modeled by any quadratic. A second possible source might be the fact that the default waiting time of 10ms might not be enough for some of these connections' mirrors to settle to their final state before measurements were taken. These possible sources of error are believed to have a more much pronounced effect in the edge zone, as we will see next.

b) Performance on connections in the edge zone

As mentioned before, the edge zone is different from the middle zone in a way that it has additional issues related to longer mirror settling time and greater sensitivity to the errors since hills at the edge are much steeper than hills in the middle in general. The result of the hill performance on connections in the edge zone is shown in Figure 25. The default waiting time was also 10 ms in this case.

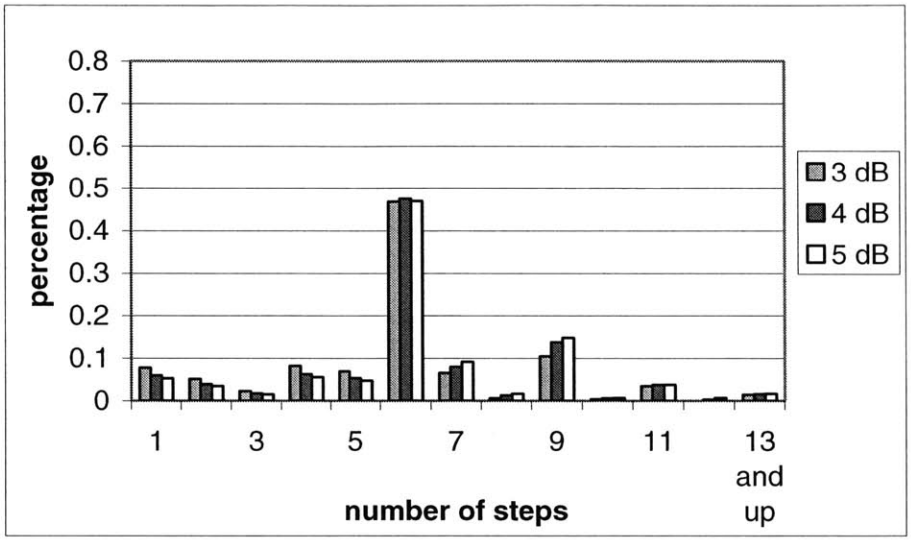


Figure 25: Performance of hillclimb algorithm on connections in the edge zone with the wait time of 10 ms

The plot shows the performance that is much worse than that of the middle zone case. Many more connections require more than 6 steps to reach the peak. There are a number of possible sources that might have led to this. We explored each possible source one by one to see which effect is responsible for deteriorating the performance of the algorithm on the edge connections.

The first source might be from the fact that there is simply too much error introduced in the estimated hill coefficients that we use. In other words, the predicted values of the coefficients might be too far off from the actual values that our hillclimb algorithm cannot handle. To investigate further into this, we ran the algorithm on the same set of edge connections but used the actual extracted hill coefficients, not the

estimates extrapolated from the model according to the mirror locations. The results came out looking very much like the plot in Figure 25 above. There is a slight improvement, but the improvement is not significant enough to conclude that it is the error introduced in the hill quadratic coefficients that was the main source of problem here. This agrees with what we learned earlier from the simulations shown in Figure 19 that the five-point algorithm should be able to handle some small errors in the hill coefficient values decently well.

We then moved on to explore the second possible source, which was that the default wait time we use might be too short to allow the mirrors to settle properly. For the same set of edge connections, we collected the hill shape data and extracted the hill coefficients for each hill using the least square fitting technique again, but this time we changed the default wait time from 10 ms to 20 ms. By this, we allow 10 ms longer for the mirrors to settle into their steady state. Since we only extracted a new set of hill coefficients for a small number of edge hills, we did not have enough information to construct a new model of hill coefficient estimate as a function of mirror location. We thus used the actual hill coefficient values to run our hillclimb algorithm on these edge connections and the result is shown in Figure 26.

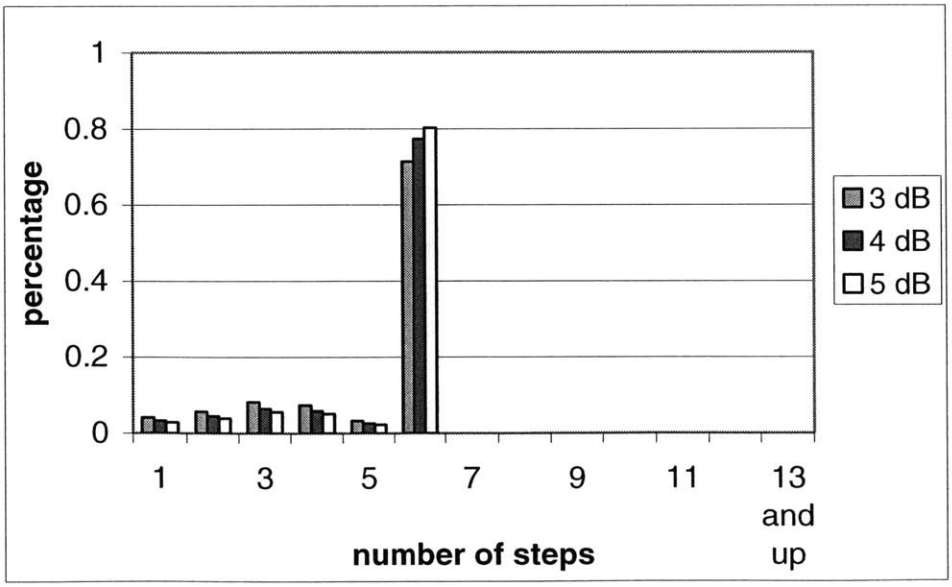


Figure 26: Performance of hillclimb algorithm on connections in the edge zone with the wait time of 20 ms (using actual hill coefficient values)

The plot above shows an enormous improvement over the result in Figure 24. We were able to get to within 0.5 dB from the peak within 6 steps for all trials. This is the best result we could ever hope for. Because of the vast improvement we see here, we could safely claim that it is the fact that the wait time was not long enough for the mirrors to reach their final state that deteriorated the performance of our hillclimb algorithm earlier in Figure 25. In that case, the extracted hill coefficients were not correct, since the hills were not at their steady state when the measurements were made, and the hills we ran our hillclimb algorithm on were also not at their steady state when we climbed. So we essentially used the hill coefficient values that were off to climb the ghost hills that did not actually represent our real hills. In contrast, what we did in Figure 26 above is we used the good hill coefficient values to climb the right hills. It is thus no surprise that the performance of the hillclimb algorithm improved very significantly.

Other than the two possible sources discussed above, we believe that the accuracy of the database that we used to do the coordinate or space transformation from voltage space to beam displacement space (as discussed in section 3.1) also affects the performance of the hillclimb algorithm, especially on the edge connections. Since edge connections have steeper hill profiles by nature, even small errors could result in a significant difference in the hill's height. So the edge connections are more sensitive to any errors we might have in the database and to rounding errors when we convert back and forth between exact voltage values applied to the hardware and the beam displacement space values used in the algorithm. Because of the steep hill profiles, the edge connections are more sensitive to all types of noises as well. Due to all these sensitivity to errors, a poorer performance of the hillclimb algorithm on edge connections is justified.

5.2.2 Performance of the algorithm on the 64x64 switch

The testing of the algorithm on the 64x64 switch was geared more towards system optimization, meaning that not only were we trying to demonstrate that this new hillclimb algorithm could reach the peak with the minimal number of intermediate steps or measurements required, we were also hoping to show the actual length of time the system takes in order to restore any connection. The results in this section should be of more

interest to people in the network level, since it shows the system performance of the switch. The conceptual part of the algorithm is the same as discussed in section 5.2.1 above, so there is no need to go into details of that again. The only important underlying difference is that this 64x64 switch has a much smaller size, so the mirrors never have to tilt beyond small angles. We thus do not have any complications related to edge connections as we had in the 1296x1296 switch. In a way, we could treat all connections in this 64x64 switch as connections in the middle zone of the 1296x1296 switch.

Since the switch is small, the idea of having a model to estimate the values of the hill coefficients instead of extracting and storing all actual hill coefficients is not that relevant or useful here. The number of connections is much less, and we could afford to extract and store all actual hill coefficients without much difficulty. So we used the actual hill coefficients for all our evaluation here.

In terms of testing the performance of our algorithm, we also did not intentionally move the start point further down the hill. We used the old optimal values stored in the database as our start points. As a result, this testing condition is very close to what we would get in a real switch operation.

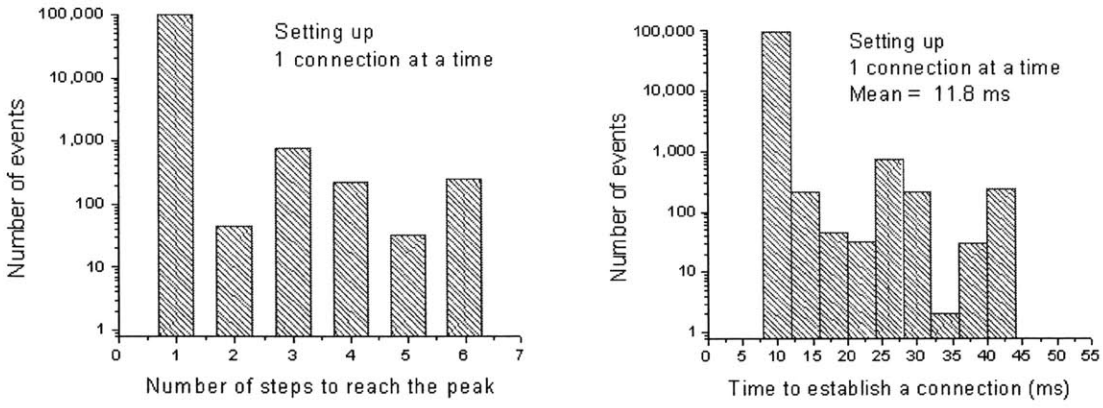


Figure 27: Result of algorithm and system performance of the 64x64 switch

The histogram on the left in Figure 27 shows the algorithm performance on the switch while the one on the right shows the system performance of the switch. Note that the vertical axes of the two plots are on a log scale. First we look at the left plot, we see that almost all connections make it to 0.5 dB from the peak in the first step. This means

that the hill has stayed relatively stable, no major drift has occurred, and the old optimal values remain very close to the hill peak. For the rest of the connections, the number of steps or measurements required to reach the peak ranges from 2 to 6. This means that our attempt to move the base point gradually up the hill during the measurement steps also paid off. The 6^{th} step, which was the prediction step, was also very effective here, since there were no trials that required more steps beyond this prediction step. We give credit to stability of hill profile, accuracy of hill coefficients used, and the fact that start points were not that far off the peak for contributing to good performance of the hillclimb algorithm.

The plot on the right of Figure 27 shows the system performance of the switch in terms of how long it takes for a connection to be restored. This histogram shows the distribution of the amount of time the switch takes to get the optical throughput of a connection to be maximized. As we can see, the restoration time specification of the system is dictated mainly by the algorithm performance, meaning that if the algorithm requires more steps, the system will most likely take more time to restore the connection as a result. There is a strong correlation between the number of steps and the total time required. But the system performance is also affected by system components other than the algorithm performance. For example, aside from the required wait time for the mirrors to settle to their final state, added delay could also come from communications within the system and the operating system of the switch. The default wait time set for this particular experiment was ~7 ms. In the plot above, we see that all connections were stored within 45 ms, and the mean was around 11.8 ms. But since then we learned through some additional experiments that the optimal wait time for the mirror to settle the first step was ~9 ms, whereas for subsequent steps 5 ms was sufficient. With the new wait time assignment of 9 ms for the first step and 5 ms for the rest, we could potentially reduce the overall restoration time of the distribution tail and further improve the system performance.

Chapter 6: Conclusion

This thesis consists of a study, implementation, and performance evaluation of a modeling and algorithm scheme that can be used to accelerate re-optimization in the MEMS-based 3D optical crossconnect design. The scheme models the transmissivity characteristic of each connection as a four-dimensional quadratic function. A set of fifteen coefficients is extracted to represent each connection. Each connection is considered a successful or active connection only when the transmissivity of the optical signal going through is close to the maximum achievable value. Significant effort is required to generate a database containing optimum control voltages for every connection of the switch. However, these optimum voltages are not guaranteed to remain the same over time. Changes in the environment and setup conditions could potentially perturb the optimal condition of a connection. When this is the case, an accelerated re-optimization process is needed to ensure that the connection is restored to become active again as quickly as possible.

The proposed modeling and algorithm scheme allows the voltage/beam displacement curve to be modeled as locally linear. The perturbations are modeled as shifts in either the ordinate or the abcissa of this function. This leads to the assumption that the transmissivity characteristic function of each connection retains its general shape over time, while its location with respect to the values of the four control variables is allowed to change. Ten out of fifteen coefficients from the four-dimensional quadratic model are kept to represent the retained hill shape of the each connection. These ten coefficients need not be explicitly kept for each connection. We may instead describe each of these coefficients as a quadratic function of the mirror locations. This parameterization substantially reduces the memory required, making this modeling and optimization approach practical even for a large switch with ~1000 ports.

By taking advantage of the stored coefficients, the proposed optimization algorithm was able to get back to the maximum spot with a minimal number of measurement steps. Almost all connections were able to reach within 0.5 dB from the optimum transmissivity level within 6 measurement steps, when the start point is within an acceptable distance (~5 dB) from the optimum level. For the 64x64 OSM prototype,

all connections tested reached the optimal region within 6 steps. With some other system-level optimization efforts, the switch showed the average restoration time of less than 12 milliseconds and the worst-case restoration time of less than 45 milliseconds.

For the 1296x1296 OSM prototype, while the majority of the connections tested were also able to reach the optimal region within 6 steps, there was a visible tail in the distribution. We concluded that these poorer performances could come from combinations of 1) not having allowed enough time for mirrors to reach their steady state before taking measurements, 2) errors introduced by the coefficient parameterization process, and 3) higher sensitivity to any noise and inaccuracy in the database of the mirrors when tilting to high angles. Among these three, we believed the first reason to be the primary source of poorer performances by the algorithm. In general, when the optics range of a switch is large, the edge connections are more prone to experience the effects of all these three factors. As a result, the algorithm is more likely to show better performance on the middle connections than the edge connections. However, given a proper operating condition and enough accuracy in the database, the algorithm proves to be as effective in optimizing the edge connections.

In this thesis we focused on the application of accelerated re-optimization, and the result of our study has shown a great promise. But in addition to this, there are other applications that could greatly benefit from further research in the hill shape studies as well. Hill shapes could be useful for other control problems, such as introducing variable loss into the optical signal within the switch for dynamic gain equalization. Hill shape models could also be used for fault detection, to learn about the sources of variation in the transmissivity function, in order to reduce insertion losses and improve uniformity among connections within the switch. Guided fault analysis can be of great help in the attempt to improve future implementations of the hardware.

Bibliography

D. J. Bishop, C. R. Giles, and S. R. Das, "The Rise of Optical Switching," *Scientific American*, January 2001, p.88-94.

D. J. Bishop, V. A. Aksyuk, C. A. Bolle, C. R. Giles, F. Pardo, J. A. Walker, "MEMS/MOEMS for lightwave networks:can little machines make it big?", *Proc. of SPIE*, Vol. 4178, 2000, p.2-5

T.W. Yeow, K.L. E. Law, and A. Goldenberg, "MEMS Optical Switches," *IEEE Communications Magazine*, November 2001, p.158-163.

V. A. Aksyuk, F. Pardo, D. Carr, H. B. Chan, M.E. Simon, A. Gasparyan, H. Shea, V. Lifton, C. Bolle, S. Arney, R. Frahm, M. Paczkowski, M. Haueis, R. Ryf, D. Neilson, J. Kim, R. Giles, and D. Bishop, "Beam-Steering Micromirrors for Large Optical Crossconnects", to appear in *Journal of Lightwave Technology*

V.A.Aksyuk, S.Arney, N.R.Basavanhally, D.J.Bishop, C.A.Bolle, C.C.Chang, R.Frahm, A.Gasparyan, J.V.Gates, R.George, C.R.Giles, J.Kim, P.R.Kolodner, T.M.Lee, D.T.Neilson, C.Nijander, C.J.Nuzman, M.Paczkowski, A.R.Papazian, F.Pardo, D.A.Ramsey, R.Ryf, R.E.Scotti, H.Shea, M.E.Simon, "238 x 238 Micromechanical Optical Cross Connect", *IEEE-Photonics Technology Letters*, Vol 15, No.4, April 2003, p: 587-589

Ben Noble, James W. Daniel, "Applied Linear Algebra", third edition,1988, Prentice Hall, Englewood Cliffs, New Jersey

William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, "Numerical Recipes in C++: The Art of Scientific Computing", Second edition, 2002, Cambridge University Press, Cambridge, United Kingdom