THE TRUE-TIME-DELAY (TTD) LASER BEAM
STEERING SYSTEM DESIGN BASED ON FOURIER
CELL

M.S. Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of
Science in the Graduate School of The Ohio State University

By

Yifan Liu, B.S

*****

The Ohio State University

2009

Thesis Committee:

Professor Betty Lise Anderson, Advisor

Approved by

Professor Ronald Reano

Advisor

Electrical and Computer Engineering

Graduate Program
In this thesis, an innovative design of a true time delay laser beam steering system based on a multiple-bounce optical system and piston-style microelectromechanical systems (MEMS) micromirror arrays is suggested. Using true-time delay for steering optical beams instead of phase-shifting eliminates dispersion (beam squint) and reduces side lobes significantly. We examine theoretically two approaches, the White cell and the Fourier cell. A simulation program in Matlab is first verified by theoretic analysis and mathematic derivation. First it is used to examine a design based on White cell, and prove that while this design will steer beams as required, it suffers from laser beam quality degradation due to imperfect reproduction of the phase front on multiple bounces. Another design based on the Fourier cell is suggested, and validated in simulation. A detailed design is developed, and the performance of this design based on the simulation results is presented.

**Keywords:** Fourier cell, White cell, true time delay, beam steering, simulation
ACKNOWLEDGMENTS

Here I would like to thank everyone that helps me during my research program. First and most important, I would like to thank my advisor Prof. Betty Lise Anderson. It is her enthusiastic efforts that help me to overcome the difficulties throughout the program. Then I would like to thank Joseph Porembski for his experiment that supports my analysis greatly. Last I would say thank you to my parents who give me all their help.
VITA

April 23, 1985……………………Born – Xi’an, China. PRC

2007………………………………B.S. Electronic Engineering, Tsinghua University

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>II</td>
</tr>
<tr>
<td>Acknowledgement</td>
<td>III</td>
</tr>
<tr>
<td>Vita</td>
<td>IV</td>
</tr>
<tr>
<td>List of Tables</td>
<td>VII</td>
</tr>
<tr>
<td>List of Figures</td>
<td>VIII</td>
</tr>
<tr>
<td><strong>Chapters:</strong></td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Overview</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Motivation for beam steering devices</td>
<td>1</td>
</tr>
<tr>
<td>1.3 Review of current beam steering designs</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1 Single beam steering devices</td>
<td>2</td>
</tr>
<tr>
<td>1.3.2 Split, manipulate and combine</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2.1 Grating designs</td>
<td>5</td>
</tr>
<tr>
<td>1.3.2.2 Phase delay</td>
<td>9</td>
</tr>
<tr>
<td>1.4 True time delay design approach</td>
<td>15</td>
</tr>
<tr>
<td>1.4.1 Concept of true time delay</td>
<td>15</td>
</tr>
<tr>
<td>1.4.2 How to increase steering angle</td>
<td>17</td>
</tr>
<tr>
<td>1.4.3 White Cell and Fourier Cell</td>
<td>19</td>
</tr>
<tr>
<td>1.4.4 Comparison with other designs</td>
<td>23</td>
</tr>
<tr>
<td>1.5 Organization of this thesis</td>
<td>23</td>
</tr>
<tr>
<td>2. Calculation of diffraction pattern</td>
<td>24</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>24</td>
</tr>
<tr>
<td>2.2 Integral of diffraction pattern</td>
<td>25</td>
</tr>
<tr>
<td>2.3 FFT in Matlab</td>
<td>27</td>
</tr>
<tr>
<td>2.3.1 Comparison between FFT and Fourier transform</td>
<td>28</td>
</tr>
<tr>
<td>2.3.2 Revised FFT for close distance diffraction</td>
<td>31</td>
</tr>
<tr>
<td>2.4 Brief introduction to the program</td>
<td>33</td>
</tr>
<tr>
<td>2.4.1 Parameter setting in the program</td>
<td>33</td>
</tr>
<tr>
<td>2.4.2 Examples of simulation in the program</td>
<td>35</td>
</tr>
<tr>
<td>2.4.3 Angle Limit of our program</td>
<td>39</td>
</tr>
<tr>
<td>3. Discussion on the TTD design based on White Cell</td>
<td>43</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>43</td>
</tr>
</tbody>
</table>
3.2 The reimaging process of White Cell based TTD piston array design…….44
3.3 Simulation process of the design......................................................48
  3.3.1 Determine the simulation method and basic parameters. ...............48
  3.3.2 Simulation process and result....................................................59
3.4 Simulation result analysis and introduction of Fourier Cell..............64
  3.4.1 Simulation result analysis.......................................................64
  3.4.2 Introduction to Fourier Cell structure.......................................65
4. Discussion on the TTD design based on Fourier Cell.........................68
  4.1 Introduction to Fourier Cell and this chapter..................................68
  4.2 Validation of Fourier Cell imaging performance................................69
  4.3 TTD device design based on Fourier Cell......................................74
    4.3.1 Description of TTD device structure based on Fourier cell............74
    4.3.2 Consideration of lens material and package technology................77
    4.3.3 Detailed design parameters................................................79
  4.4 Performance analysis.....................................................................90
    4.4.1 Steering angle.......................................................................91
    4.4.2 Energy loss..........................................................................91
5. Conclusion..........................................................................................98
  5.1 Summary of previous work.............................................................98
  5.2 Future work..................................................................................101
Appendices.............................................................................................102
  A. Derivation of the steered beam far field pattern array factor.............102
  B. The code for the Matlab simulation program....................................106
References.............................................................................................112
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Fill factor comparison</td>
<td>54</td>
</tr>
<tr>
<td>3.2 Simulation parameters</td>
<td>61</td>
</tr>
<tr>
<td>4.1 Simulation parameters</td>
<td>71</td>
</tr>
<tr>
<td>4.2 Available glass and crystal material</td>
<td>78</td>
</tr>
<tr>
<td>4.3 Lens radius</td>
<td>83</td>
</tr>
<tr>
<td>4.4 Simulation parameters</td>
<td>85</td>
</tr>
<tr>
<td>4.5 Lens weight</td>
<td>88</td>
</tr>
<tr>
<td>5.1 Design parameters</td>
<td>99</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>One micro mirror upon gimbal ring structure</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Diagrams of electro wetting prisms</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Schematic of gimbal-less beam steering design and 1D array of micro mirrors</td>
<td>6</td>
</tr>
<tr>
<td>1.4</td>
<td>Blazed-grating deflector</td>
<td>7</td>
</tr>
<tr>
<td>1.5</td>
<td>Basic concept of 1D phase delay array using liquid crystal</td>
<td>9</td>
</tr>
<tr>
<td>1.6</td>
<td>Sketch of element factor, array factor (a) and their product (b)</td>
<td>11</td>
</tr>
<tr>
<td>1.7</td>
<td>Steered array factor, element factor and combined form</td>
<td>13</td>
</tr>
<tr>
<td>1.8</td>
<td>Sketch of phase shift design</td>
<td>14</td>
</tr>
<tr>
<td>1.9</td>
<td>Sketch of piston MEMS based TTD device</td>
<td>16</td>
</tr>
<tr>
<td>1.10</td>
<td>Demagnification</td>
<td>18</td>
</tr>
<tr>
<td>1.11</td>
<td>White Cell structure</td>
<td>19</td>
</tr>
<tr>
<td>1.12</td>
<td>Revised White Cell TTD beam steering system</td>
<td>21</td>
</tr>
<tr>
<td>1.13</td>
<td>Fourier Cell TTD beam steering system</td>
<td>21</td>
</tr>
<tr>
<td>1.14</td>
<td>Comparison between White Cell and Fourier Cell</td>
<td>22</td>
</tr>
<tr>
<td>2.1</td>
<td>Physical situation of a diffraction process</td>
<td>25</td>
</tr>
<tr>
<td>2.2</td>
<td>Profile of the object on the MEMS surface</td>
<td>36</td>
</tr>
<tr>
<td>2.3</td>
<td>Profile and diffraction pattern of another object</td>
<td>37</td>
</tr>
<tr>
<td>2.4</td>
<td>Calculation of diffraction pattern</td>
<td>40</td>
</tr>
<tr>
<td>3.1</td>
<td>Revised White Cell TTD beam steering system</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Illuminated pattern on MEMS surface</td>
<td>46</td>
</tr>
<tr>
<td>3.3</td>
<td>Profile and diffraction pattern of the object</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Field lens</td>
<td>48</td>
</tr>
<tr>
<td>3.5</td>
<td>Diffraction pattern of plane wave with and without lens</td>
<td>50</td>
</tr>
<tr>
<td>3.6</td>
<td>Reflection pattern arrangements</td>
<td>53</td>
</tr>
<tr>
<td>3.7</td>
<td>Cross section view of the diffraction pattern</td>
<td>54</td>
</tr>
<tr>
<td>3.8</td>
<td>Diffraction pattern with lens matrix at 300mm</td>
<td>55</td>
</tr>
<tr>
<td>3.9</td>
<td>Comparison of fill factors in steered beam case</td>
<td>57</td>
</tr>
<tr>
<td>3.10</td>
<td>Steered array factor, element factor and combined form</td>
<td>58</td>
</tr>
<tr>
<td>3.11</td>
<td>Simulation process</td>
<td>59</td>
</tr>
<tr>
<td>3.12</td>
<td>Theoretic sketch of the White Cell mechanism</td>
<td>60</td>
</tr>
<tr>
<td>3.13</td>
<td>Reflection light intensity</td>
<td>62</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Overview

This thesis introduces an innovative design of a laser beam deflection system: the true time delay (TTD) beam steering system based upon first a White cell and then a Fourier Cell. Using multiple bounces of light beams on a piston micro-electro-mechanical systems (MEMS) surface, this design has advantages such as wide steering angle, minimized side lobes and no dispersion. Detailed analysis of these properties will be given in later chapters.

1.2 Motivation for beam steering devices

Laser beam steering devices are widely used in various applications, such as for optical cross-connect components in communication systems. Such manipulation in the “optical domain” avoids the requirement for opto-electronic and electro-optic transforming processes and devices, and thus reduces time delay and system
complexity.

1.3 Review of current beam steering designs

Various approaches for beam steering have been suggested, making use of different mechanisms, such as mechanical steering, phase shifting and true time delay. This section will give a brief review of them. All these beam-steering designs have been classified into categories according to the similarities and differences between their mechanisms. Designs of each category will be reviewed in the sections below, and it will be seen that both mechanical steering and phase delay have some disadvantages compared to true time delay.

1.3.1 Single beam steering devices

In the first class of designs, a single device, rather than an array of devices, is used to manipulate the incident laser beam. This requires the device to be big enough to accept the input beam. Generally speaking, such device is usually as wide as a few hundred microns or a few millimeters, depending on the laser sources commonly used in optical labs.

*Dual-gimbal analog MEMS mirror*

One method used to deflect a laser beam is by using a MEMS micro mirror shown in figure 1.1. [1] Here a micro mirror is connected to a gimbal ring through two torsional springs. The gimbal ring itself is connected to a fixed frame by two
additional springs. Electrodes are fabricated below the gimbal ring. When voltage is applied to these electrodes, the resulting electro-static force can tilt the mirror around two axes. The beam is then steered by reflection.

According to the scale marker in the figure, however, it can be seen that the micro-mirror diameter is on the scale of a few hundred µm. Due to the big and heavy mirror and gimbal ring, this device is sensitive to forces such as acceleration and shock. Other problems like fabrication complexity and response speed also limit its application.

Electro wetting prisms

Another innovative beam steering design uses electro wetting prisms. The structure can be seen in figure 1.2. [6]
This device, as shown above in the figure, is a container partially filled with a water: glycerin: KCl solution of 80:20:0.1 mixture by weight. The top surface and bottom of container are made of glass. The left and right side walls are electrodes made of silicon. The hydrophobic surface of the side walls creates a convex liquid surface when no voltage is applied to the side wall electrodes. When voltage is introduced, the liquid surface curvature and tilt angle can be changed, thus changing the deflection angle of a laser beam passing through this liquid layer by refraction.

One limitation of this design is the difficulty of fabrication. The sidewalls of each cell, according to this report, need to be fabricated individually and then bonded to the substrate. This could become expensive in a mass production process. Moreover, this current design is on the scale of mm, while the validity of this design is still an open problem when it is magnified or demagnified. For example, it is already known that the shape of liquid surface and its interaction with electrostatic forces will change significantly when the device is scaled down to the µm level.
1.3.2 Split, manipulate and combine

Another group of beam steering devices use multiple cells to steer a single beam. Each cell is controlled independently to manipulate a small part of the wave front. The treated parts of wave front, when they are emitted from the device, interfere with each other in far field region to form a single steered beam. We call this “split, manipulate, and combine,” because the input beam can be thought of as being split into pieces, the pieces manipulated individually, and finally combined together in the far field.

1.3.2.1 Grating designs

This first of this class of designs to be discussed is gratings. Each groove of the grating manipulates the corresponding part of the incident wave front. The emitted wave fronts interfere with each other coherently in the far field region, forming a steered beam.

Gimbal-less micro mirror

One example of a grating structure is illustrated in Figure 1.3a, which shows the structure of a single gimbal-less micro mirror. Figure 1.3b is a photomicrograph of a 1D gimbal-less micro mirror array. According to the figure, this design is similar to the dual-gimbal micro mirror design mentioned above, but the gimbal ring and torsional springs around the micro mirror are replaced by springs below the mirror. Thus, the micro mirrors can be fabricated close together, resulting in a high fill factor.
The parameter “fill factor” is defined as the mirror size (96 um) over cell size (100um). [2]

When a voltage is applied to the electrodes below the micro mirrors, the mirrors are tilted, steering the reflected beam aside. Like in other kinds of gratings, the output beam direction is one of several discrete angles. The possible output angles are given by constructive interference condition of the micro mirror array:

\[ \theta = n \frac{\lambda}{a} \]  

In the formula above, “\( \theta \)” is the output angle (defined as the angle between output peak direction and the direction normal to device surface) in the near-axis condition, “\( \lambda \)” is wave length and “\( a \)” is the cell size (100um here). The letter “\( n \)” represents the order of interference maximum, which could be any integer.

Figure 1.3 Schematic of gimbal-less beam steering design and 1D array of micro mirrors [2]
The disadvantages of gimbal-less micro mirror design, and other kinds of gratings, are due to the limit that only the discrete output angles that satisfy equation (1.1) could be accessed with high efficiency. Besides, the output side lobes are difficult to suppress, causing beam energy leakage away from the required output angle.

**Blazed grating**

Figure 1.4 Blazed-grating deflector [5]
Another example of a grating solution is shown in Figure 1.4. There is a stack of two layers of blazed grating deflector. In each layer, a saw-tooth-shaped piece of poly-methyl-methacrylate (PMMA) is bonded to the indium tin oxide (ITO) electrode on the layer bottom. Another ITO electrode is on the top of this layer. Liquid crystal (LC) fills the space between top ITO electrode and the PMMA. [5] When no voltage is applied to the electrodes, the refractive index difference between the LC and the PMMA causes this structure to behave as a blazed grating. The sawtooth grooves can locally tilt the phase of the passing wave front. The emitted wave thereafter diffracts in the far field region, creating a single peak steered by a fixed angle from the direction normal to ITO surface. However, when a specific voltage is applied to the electrodes, the LC molecules are rotated to a specific direction. This causes the LC layer refractive index to change to be the same as that of PMMA, cancelling out the diffraction at interface between LC layer and PMMA layer, and the output beam is steered back to the direction normal to ITO surface. A binary beam steering control is achieved in this way. Similar structures of different grating periods could be stacked and controlled individually to acquire multiple steering angles, as shown in figure 1.4.

The disadvantage of blazed grating is the same as those of gimbal-less micro mirror. Only discrete steering angles are accessible with this design. An even worse thing is that one layer of blazed grating could only steer a beam between two directions, which is even fewer than a gimbal-less mirror could steer. Increasing the number of stacked layers of blazed gratings, on the other hand, always means great increase in system complexity and cost.
1.3.2.2 Phase delay

The last class of beam steering devices also uses multiple cells to manipulate the wave front of the incident beam. But in these devices, the resulting wave front is stepped with flat steps as opposed to tilted steps in the blazed grating deflector. Due to different phase delays introduced into each cell, the final output diffraction pattern can be steered by a required angle.

The mechanism of one such design is shown in Figure 1.5. This example uses an array of liquid crystal cells. Each cell can induce a specific phase delay to the wave front passing through it, which is achieved by changing the refractive index via an applied voltage. When the wave fronts are emitted from each cell, the beams diffract into the far field region, and interfere coherently with the emitted far field from other cells.

![Figure 1.5 Basic concept of 1D phase delay array using liquid crystal [3]](image)

If it is assumed that each LC cell is a square and all cells induce the same phase delay, the far field pattern of a 1D array of square LC cells in the $x$ direction is given
by [8]

\[ I(x_i) = I_o N^2 \left( \sin^2 \left( \frac{b a k}{2L} \frac{x_i}{\sqrt{1 + \left(\frac{x_i}{L}\right)^2}} \right) \right) \left( \sin^2 \left( \frac{N k a}{2L} \frac{x_i}{\sqrt{1 + \left(\frac{x_i}{L}\right)^2}} \right) \right) \]

\[ = N^2 \sin^2 \left( \frac{k a}{2L} \frac{x_i}{\sqrt{1 + \left(\frac{x_i}{L}\right)^2}} \right) \]

(1.2)

In the equation above, \( I \) represents the light intensity. The coordinate \( x_i \) shows the observation point location on the observation plane. Besides, \( N \) is the number of pixels (cells) in the array; \( L \) gives the distance between LC plane and observation plane. And \( a \) is pixel pitch of each LC cell, while \( b \) is the fill factor of these cells and \( k \) is the wave number for the given wavelength.

In this equation, the term

\[ \left( \frac{b a k}{2L} \frac{x_i}{\sqrt{1 + \left(\frac{x_i}{L}\right)^2}} \right)^2 \]

is called the element factor, which the far field diffraction pattern of the emitted wave from a single cell. This term is not changed no matter how much phase delay is induced in each cell. The term

\[ \frac{N k a}{2L} \frac{x_i}{\sqrt{1 + \left(\frac{x_i}{L}\right)^2}} \]

\[ N^2 \sin^2 \left( \frac{k a}{2L} \frac{x_i}{\sqrt{1 + \left(\frac{x_i}{L}\right)^2}} \right) \]

is the array factor, showing the interference between cells, and will be affected by the relative phase delay between cells.
Figure 1.6 Sketch of element factor, array factor (a) and their product (b)

Figure 1.6(a) is a plot showing the relation between the element factor and array factor. The dashed line (also called “envelope”) shows the element factor, which, as mentioned above, is the far field diffraction form of a single cell. The solid line shows the array factor when there are 11 cells in the array ($N=11$), and the shape is exactly the interference pattern of 11 slots. Thus it is easy to see that the far field of the entire
array given by (1.2) will be the product of two curves in figure 1.6(a), which is shown in 1.6(b)

When an electric field is applied to the LC layer, its refractive index is changed, inducing extra phase delay to the optical wave passing through it. When different electric fields are applied to different LC cells, the LC layer refractive index changes are different, thus inducing relative phase delay between cells. Assuming that two neighboring cells have a refractive index step $\Delta n$, the delay between these two cells is

$$\Delta \phi = 2\pi \frac{\Delta n}{\lambda} t$$  \hspace{1cm} (1.3)

In this equation “$t$” represents thickness of LC layer. The steering angle is given by

$$\theta = \frac{\Delta \phi \lambda}{2\pi a}$$  \hspace{1cm} (1.4)

where “$\Delta \phi$” is the introduced phase delay step between each of two neighboring cells in the LC layer. In order to avoid too much lowering of the peak intensity of the beam, the steering angle is limited to FWHM range of the envelope, which is $\frac{\lambda}{a}$ wide as shown in figure 1.5.

Considering equation (1.2), a phase delay step of $\Delta \phi$ in equation (1.3) will change the array factor into the form

$$\sin^2 \left( \frac{N k a}{2L} \frac{x_j}{\sqrt{1 + (x_j^2 / L^2)}} + \frac{N}{2} \frac{\Delta \phi}{\lambda} \right)$$

$$N^2 \sin^2 \left( \frac{k a}{2L} \frac{x_j}{\sqrt{1 + (x_j^2 / L^2)}} + \frac{1}{2} \frac{\Delta \phi}{\lambda} \right)$$

The derivation of this term will be described in detail in Appendix A. Using the new array factor, if $\Delta \phi = 0.3 \pi$, the new plot of the element factor and array factor is shown in Figure 1.7a, and the resulting beam is shown in Figure 1.7b.
It can be seen that the peak of far field energy distribution is steered an angle away from center. This is the mechanism of the designs in this class.

**Liquid crystal (LC) phase shift array**

The case in figure 1.5 is an example of an LC phase delay design. However, as in equation (1.3) and (1.4), the steering angle of such design is proportional to the refractive index change $\Delta n$ and LC layer thickness $t$. The problem is that $\Delta n$ is quite small for currently available LC materials. Besides, the LC thickness $t$ determines the speed of response of the LC phase delay device, so $t$ can not be increased too much.
These problems limit the possible phase delay acquired by the mechanism mentioned above. A phase shift design is suggested to solve this problem.

The idea of phase shift is shown in figure 1.8. The gray boxes are the LC cells. The dotted lines are ideal stepped phase fronts emitted out of the LC cells. The arrows show the steered beam direction. The dashed lines show the possible phase shift range that can be attained by adjusting the LC refractive index. So it can be seen that only group \( a \) cells could satisfy the requirement. The two cells in group \( b \) need to shift the phase by \( 2\pi \), group \( c \) needs \( 4\pi \) shift, and group \( d \) needs \( 6\pi \). In phase shifting, a shift of \( 4\pi \) is considered equivalent to \( 2\pi \) and so \( 2\pi \) is used. The actual phase fronts of each cell group are shown in solid lines. In this way, the wave front could still constructively interfere in the expected steering direction.

This design works well for single wavelength case. However, in actual applications, there are probably multiple wavelengths in a beam. Thus, the delay that
equals $2\pi$ phase shift for one wavelength might be $2.01\pi$ or $1.99\pi$ for another wavelength. This means that the far field beam of each wavelength can not interfere constructively in the expected direction. Instead, these wavelength components will be steered to slightly different directions. This is called squint, or dispersion. Elimination of squint became our motivation for a true time delay design.

1.4 True time delay design approach

After reviewing the typical beam steering technologies that already exist, a new design called true time delay approach is suggested below, including some simple discussion about its advantages comparing to phase shift technology.

1.4.1 Concept of true time delay

True time delay (TTD) is also a kind of phase delay design similar to phase shift technology. However, TTD induced phase delay does not have any phase shift reset every $2\pi$. As mentioned before, the necessity of phase reset comes from the limit of the LC phase delay by changing the refractive index. So the key of TTD is to replace the LC phase delay device by some other design that can create a much wider range of phase delays directly. The solution we suggest here is based on micro-electro-mechanical systems (MEMS). The idea is shown in Figure 1.9.
Each white bar in figure 1.9 is a piston (which may be also mentioned to as a pixel) that can change its height according to the electric signal applied to it. The dark gray top on each piston represents a polished reflective surface that can reflect an incident wave front with quite high efficiency. Thus, assuming that the input beam is incident on the MEMS surface normally, if the piston (and the mirror on it) is shortened, it will take a longer time for the incident wave to reflect back, and a longer time delay created. Similarly, if the piston height is increased, the reflection optical path will be reduced, resulting in a shorter time delay during reflection. So if all pistons are equally tall (like piston 4 and 5 in figure 1.9), the output wave front (light grey lines) is flat and parallel to the MEMS surface, and the far field diffraction pattern will be a non-steered beam. If pistons are stepped arranged like piston 1, 2 and 3, the output wave front will also be step-delayed and creating a steered beam in the
far field. The analysis of the far field pattern based on equation (1.2) is absolutely the same as in the phase shift design. We only introduce a simple equation to calculate the steering angle based upon constructive interference condition:

$$\theta = \arcsin \left( \frac{h}{a} \right) \quad (1.5)$$

Here “a” is the pixel pitch—the size of a piston, and $h$ is the step height between two neighboring pistons; $\theta$ is the steering angle, which is valid for near-axis steering case.

### 1.4.2 How to increase steering angle

From equation (1.5), it is known that the beam steering angle of piston array is determined by the step height $h$ and pixel pitch $a$. Assuming that the number of pistons on the MEMS array is $N*N$, and the stroke (defined as the maximum displacement of each piston) is $S$, it can be found that

$$h = \frac{S}{N - 1} \quad (1.7)$$

Thus, the maximum beam steering angle of one piston MEMS array is limited by $S$ and $a$, both of which are determined by the manufacturing technology. In our current device, the maximum stroke of the piston is only 2um. If a $32 \times 32$ pixels array is used to steer the beam, the step height between two neighboring pistons is $2\text{um}/31=0.065\text{um}$. And one bounce on the MEMS surface results in two times the time delay (once toward the MEMS and the other from the MEMS). Considering that the pixel pitch is around 0.15mm at present, the maximum steering angle will be $\arcsin(2 \times 0.065\text{um}/0.15\text{mm})=8.6 \times 10^{-4} \text{rad}$. This is too small for any practical device.
So we will try to find some ways to enlarge this angle.

**Beam demagnification**

The first method we want to introduce is to demagnify the beam width. From equation (1.5), we can see that if the phase delay step height $h$ is a constant, the steering angle can be increased if the pixel pitch is smaller. The pixel pitch itself is determined by the method of manufacturing the MEMS, and can not be changed so easily, but we can use lens groups to demagnify the beam width after all phase delay has been induce to the beam. This is equivalent to the demagnification of pixel pitch.

Figure 1.10 Demagnification

The concept of demagnification process is shown in figure 1.10. When the beam width is demagnified (equivalent to pixel pitch demagnification), the phase delay
induced by reflections on the MEMS is not changed, so the beam steering angle is increased.

Multiple bounces

There is also another way to increase $h$ value without changing $S$ and $a$. That is by multiple reflections. If the steered beam is turned back and reflected on the piston MEMS array once again, the delay is accumulated, equaling to a larger step height value. Thus, the final steering angle can be doubled. By increasing the number of reflections on the piston MEMS, the steering angle is increased until it reaches other limits, such as the element factor induced maximum steering range shown in figure 1.6.

1.4.3 White Cell and Fourier Cell

As mentioned above, the beam steering angle in TTD design could be increased by multiple reflections on MEMS surface. In Figure 1.11, a White Cell structure is introduced to meet this requirement. [7]

![Figure 1.11 White Cell structure [7]](image-url)
Mirrors A, B and M are spherical mirrors with similar curvature radius $R$. For spherical lenses, it is known that their focal length is half of $R$. Mirror M is placed with its center normal to the White Cell optical axis. Two mirrors A and B are placed on either side of the optical axis. The distance between A, B and M is $R=2f$. Thus, the curvature center of M (named CCM in figure) is just between mirror A and B. On the other hand, the curvature centers of A and B (named CCA and CCB) are also placed closely together on two sides of optical axis, and also, on the surface of M. Another flat input turning mirror (ITM) is placed near mirror M. When a beam is focused onto the surface of ITM, it will be reflected to B, and then be refocused to M and reflected to A, finally be refocused to M. Such a process could be repeated many times if the energy loss is low enough.

The White Cell could provide the multiple reflections that are needed for the TTD beam steering system. To make use of White Cell structure, mirror M in figure 1.11 is replaced by the piston MEMS array and a lens (called field lens) close to it. The focusing process is also changed, as beam will be focused onto mirror A and B instead of mirror M. The new structure is shown below.
However, during the course of the thesis work, this system was proven to be inappropriate for our application. Detailed analysis of the problem will be given in chapter 3. For now, we also introduce a different multiple-bounce design. The revised design is called Fourier Cell, which is shown in figure 1.13.

In figure 1.13 a spherical lens replaces the field lens in the White Cell. A Fourier mirror replaces mirror A and B in White Cell. Multiple piston MEMS arrays, instead
of a single big piston MEMS array, are used to provide beam steering, and they are moved to one focal length away from field lens (now the spherical lens). Conceptual sketches comparing the White Cell and the Fourier Cell are shown in figure 1.14.

![Conceptual sketch of White Cell](image)

(a) The White cell

![Conceptual sketch of Fourier Cell](image)

(b) The Fourier cell

Figure 1.14 Comparison between White Cell and Fourier Cell

In the White cell, the MEMS (object) and observation plane are, in ideal case, infinitely close to the lenses. But in the Fourier Cell, both the MEMS and observations plane are one focal length away from the lenses. A performance comparison of these two structures and the Fourier Cell based design will be discussed in Chapter 4.
1.4.4 Comparison with other designs

In the piston MEMS array TTD beam steering system (also referred to as TTD design for short), the beam is steered continuously. It does not have some inaccessible degrees like the gimbal-less micro mirror and blazed grating. Compared to the LC phase shift design, the TTD based device is insensitive to wavelength, which provides less energy loss and less information leakage during laser beam steering. A detailed comparison between the performance of the LC phase shift design and the Fourier Cell TTD design will be given in chapter 4.

1.5 Organization of this thesis

In chapter 2, we discuss how to calculate the diffraction pattern of a given optical field in our case. This, in the following chapters, will help us to evaluate how the MEMS array could affect the beam wave front, which will determine whether the design is usable or not.

Chapter 3 specifies our first design based upon White Cell. This design is proved to be inappropriate for our application.

In chapter 4, a new design based on Fourier Cell is introduced and validated. Detailed discussion on the system parameters selection follows.

Chapter 5 gives conclusions on our design and predicts its potential.
CHAPTER 2

CALCULATION OF DIFFRACTION PATTERN

2.1 Introduction

In later chapters, as we analyze first the White cell and later the Fourier cell for optical beam steering, it will be necessary to evaluate the coherently combined beams as they travel both inside the time delay device and also after they leave the cell. In this chapter we discuss the simulation tools used later.

To calculate the beam’s structure, we recall that in our true-time delay approach, the beams from a group of individual light emitters combine coherently as they propagate to produce a single beam. In the White cell, the emitter array is generated by reflecting an input plane wave (ideally) from multiple arrays of piston MEMS. Thus, to evaluate the beam deflection performance (such as insertion loss, beam quality and so on) of a beam reflected from a MEMS piston array, it is necessary to calculate the diffraction pattern of a reflected beam at various distances from the MEMS array. For example, in the White cell, the beams do not necessarily reach the far field before being returned for another visit to the MEMS. Furthermore, the angles
are not necessarily small.

To simplify the problem, the MEMS array is considered to be illuminated uniformly by a normally incident, monochromatic plane wave. Then the problem is transformed into: calculate the diffraction pattern of a given optical field at a given distance. The uniform optical field on a MEMS array surface is treated as the source--or to say, the object--of a diffraction process.

2.2 Integral of diffraction pattern

When we start to discuss the diffraction pattern of a light wave emitted from a TTD piston MEMS array, we try to find the most general expression of the diffraction pattern, which can be valid for the most cases and has as few limiting conditions as possible. This is because at first it is not known which limiting conditions can be met and which can not.

![Figure 2.1 Physical situation of a diffraction process](image)

Figure 2.1 [8] shows the physical picture. The plane on the left is the objective plane (that is to say, the MEMS plane), each point on which has “θ” in its coordinate
subscripts, such as a point coordinate \((x_0, y_0, z_0)\). The plane on the right is the observation plane, a point on which is shown as \((x_1, y_1, z_1)\). The light field on objective plane is the source of diffraction, and its corresponding electric field is written as \(E_0(x_0, y_0, z_0)\). The diffraction pattern in the observation plane is similarly written as \(E_1(x_1, y_1, z_1)\). Here \(\vec{r}_0\) is the vector from space origin (also in objective plane in figure 2.1) to the light source point \((x_0, y_0, z_0)\), and \(\vec{r}_1\) is the vector to the observation point \((x_0, y_0, z_0)\). Thus \(r_{01}\) is the distance between these two points. According to figure 2.1 the angle between vector \(\vec{r}_1 - \vec{r}_0\) and the optical axis is written as \(\theta\). From [8] and [9], the most general form of a specific object’s diffraction pattern is given by

\[
E_1(x_1, y_1, z_1) = \frac{ik}{2\pi} \int \int E_0(x_0, y_0, z_0) \cos \theta \frac{e^{-ikr_{01}}}{r_{01}} \, dx_0 \, dy_0 \quad (2.1a)
\]

where

\[
r_{01} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \quad (2.1b)
\]

and

\[
\cos \theta = \frac{z_1 - z_0}{r_{1} - r_{0}} \quad (2.1c)
\]

Simply by combining three parts of (2.1) together without any approximation, one can derive that:

\[
E_1(x_1, y_1, z_1) = \frac{ik}{2\pi} \int \int \frac{(z_1 - z_0)e^{-ikr_{01}}}{r_{01}^2} E_0(x_0, y_0, z_0) \, dx_0 \, dy_0 \quad (2.2)
\]
It is interesting to notice that in (2.2) the object and observation surfaces do not necessarily need to be planes. As the distance \( r_{01} \) between light source point \((x_0, y_0, z_0)\) and observation location \((x_1, y_1, z_1)\) has included the term of distance \((z_1 - z_0)\), formula (2.2) could apply to the cases when light source (MEMS surface) and observation are not on flat surfaces and \((z_1 - z_0)\) is a variable. But for simplicity, this thesis will continue to use the terms “objective plane” and “observation plane”.

Formula (2.2) will be applied in Matlab to simulate our design of White Cell and Fourier Cell.

2.3 FFT in Matlab

It is well known that the far field, near-axis diffraction pattern of an object is given simply by a Fourier transform of the object optical field function. Although our task is to calculate the diffraction pattern not just in the far field region, it is still worthwhile to see if the Fourier transform can be revised in some way to apply to near field diffraction pattern calculations. If so, it may simplify our programming and calculation. The Matlab software we use during our research is Matlab version R2008a, which already provides a FFT (Fast Fourier Transform) function. Thus, our task has been expanded to two steps: first is to find the relationship connecting FFT function in Matlab (which is all discrete) to the Fourier transform function for a continuous field. Second is to try to revise the Fourier transform function to simulate the near field diffraction pattern. This section will figure out these two sub-topics.
Before we start, the FFT function in Matlab should be given in advance. For the 1-dimensional (1D) case, the FFT function is

\[
X(k) = \sum_{n=1}^{N} x(n) \exp \left( -i2\pi \left( \frac{k-1}{N} \right)(n-1) \right) \quad 1 \leq k \leq N \quad (2.3)
\]

In this equation \(x(n)\) is the input 1D matrix, and \(X(k)\) is the output 1D matrix, which is the Fast Fourier Transform of \(x(n)\). The letter \(N\) gives the length of both matrices.

### 2.3.1 Comparison between FFT and Fourier transform (Fraunhofer diffraction)

Equation (2.3) is still far away from the diffraction integral we use in equation (2.2). So it is important to know (a) how to compare and convert between their calculation results, and (b) the difference in their limiting conditions.

We already know that the FFT in a discrete field approximates the Fourier transform in a continuous field. So the first step to connect (2.2) and (2.3) together is to find the relationship between the FFT and a Fourier transform. It is known that the Fraunhofer (far-field) diffraction equation for a continuous field is in the form of a Fourier transform shown in equation (2.4) [10]

\[
X(\nu_x) = \int E_0(x_0) e^{i2\pi x_0 \nu_x} dx_0 = F \left( E_0(x_0) \right) \quad (2.4)
\]

where \(\nu_x = \frac{x_1}{\lambda z_1}\), \(X(\nu_x)\) is the electric field of the diffraction pattern, and \(E_0(x_0)\) is the electric field of the diffraction source on the objective plane.

It should be noticed that two significant approximations are made in arriving at this expression. (a) First is the small angle limit that arises by assuming that \(\theta = \sin \theta = \tan \theta\). (b) The other is the far field requirement for Fraunhofer
diffraction. We will describe later a method for judging whether a diffraction case is in far field region or not, and how it will affect our program and device design. But now, we will assume that these two conditions are satisfied, and focus on Equations (2.3) and (2.4) themselves. Comparing (2.3) and (2.4), we can acquire the conversion relationship between the Fast Fourier Transform and the Fourier transform. It is expressed as an equation group below.

\[ E_0(x_0) = x(n) \quad (2.5a) \]

\[ \frac{(k-1)(n-1)}{N} = \frac{x_0 x_1}{z_1 \lambda} \quad (2.5b) \]

In (2.5b), \( k \) and \( x_1 \) are the coordinates of an observation point in the discrete field and continuous field respectively, while \( n \) and \( x_0 \) are the corresponding coordinates of an object point in these two fields. The relations shown by equation (2.5) could also be interpreted in a physical perspective: the continuously distributed electric field \( E_0 \) in reality is sampled, creating a discrete sequence \( x(n) \). Equation (2.5a) shows the equivalence between the original electric field and sampled value, while (2.5b) connects the coordinates of a sampling point in \( E_0 \) with the sequence number of the corresponding point in \( x(n) \).

However, all these coordinates \((k, n, x_1, x_0)\) are variables in the integral, and they are not convenient for mutual conversion. We will try to find a method that replaces these variables by constants. As we have mentioned, \( X(k) \) in the discrete field is a sequence sampled from \( X \left( \frac{x_1}{z_1} \right) \) in the continuous field, and similarly, \( x(n) \) is sampled from \( E_0(x_0) \). So \( k \) and \( x_1 \) are corresponding coordinates, and so are \( n \) and \( x_0 \).
Thus, we can acquire another similar relation from (2.5b).

\[
\frac{k(n-1)}{N} = \frac{x_0(x_1 + \Delta x_i)}{z_i \lambda}
\]  
(2.6a)

Here $\Delta x_i$ is the distance between two neighboring sampling points on the observation plane. Or in other words, $\Delta x_i$ is the observation resolution. By taking the difference of (2.5b) and (2.6a), we get (2.6b)

\[
\frac{(n-1)}{N} = \frac{x_0 \Delta x_i}{z_i \lambda}
\]  
(2.6b)

Then we apply similar steps to $n$ and $x_0$ in (2.6b). The result is

\[
\frac{1}{N'} = \frac{\Delta x_0 \Delta x_i}{z_i \lambda^2}
\]  
(2.6c)

Similarly, here $\Delta x_0$ is the sampling resolution on the objective plane. With the help of equation (2.6c), the calculation results from the FFT and the continuous field Fourier transform can be converted from one to the other.

Now we turn back to the limit condition of the far field. A typical requirement to judge whether a given case of diffraction can be treated as far field Fraunhofer diffraction [10] is by calculating the Fresnel Number $N_F$:

\[
N_F = \frac{a^2}{\lambda z}
\]  
(2.7)

where $a$ is the smallest feature size on the objective plane, and $z$ is the distance between the objective plane and observation plane. According to [10], if $N_F < 0.1$, it is a far field diffraction case.

Putting this in the context of our situation, the pixel size for our TTD piston array is 0.15mm on a side. The current experimental wavelength is 633nm, while the next
generation practical device will use 1.55um. In order to satisfy the limit of N_F<0.1, a 633nm wavelength corresponds to a diffraction distance larger than 355mm, and 1.55um wavelength corresponds to not less than 145mm. It will be seen in Chapter 3 that this is longer than typical distances between various optics in the White cell. During our design, it is necessary to analyze diffraction patterns at a few tens of millimeters. So the FFT and Fraunhofer diffraction cannot replace equation (2.2) simply. But in next section, it will be discussed whether the FFT could be revised and applied to shorter distance cases.

2.3.2 Revised FFT for close distance diffraction

We have already connected the FFT and the Fourier transform by equation (2.6c) in the last section. Then the second step is to connect the later with general diffraction form in equation (2.2).

Goodman [9] gives a good method to do this job, in which equation (2.2) is transformed into a new form shown below.

\[
E_1(x_1, y_1) = \frac{e^{jkz_1}}{j\lambda z_1} e^{-\frac{k}{2z_1}(x_1^2 + y_1^2)} \int \int E_0(x_0, y_0) e^{\frac{k}{2z_1}(x_0^2 + y_0^2)} \left[ e^{-j\frac{2\pi}{\lambda z_1}(x_0 x_1 + y_0 y_1)} \right] dx_0 dy_0 \quad (2.8)
\]

What interests us is that there is a Fourier transform integral in this expression. The electric field in objective plane \( E_0(x_0, y_0) \) is first multiplied with a factor \( e^{\frac{k}{2z_1}(x_0^2 + y_0^2)} \), and then is Fourier-transformed (transform variables are \( x_0, y_0, \nu_x = \frac{x_1}{\lambda z_1}, \nu_y = \frac{y_1}{\lambda z_1} \)), and at last multiplied by another factor \( \frac{e^{jkz_1}}{j\lambda z_1} e^{\frac{k}{2z_1}(x_1^2 + y_1^2)} \).
Then the final result is equal to the result given by equation (2.2). This process fills the gap between the Fourier Transform and the general form of the diffraction pattern calculation.

The last problem to mention here is the issue of “FFTshift” function. Due to the definition of FFT function in Matlab, the Fast Fourier Transform puts the “minus frequency” component of the transform result to the right half of the output sequence, which causes zero frequency point to be on both ends of the output sequence. But in a diffraction problem, “zero frequency” corresponds to the center of diffraction pattern, and it should be in the center of the output sequence. So Matlab has provided a function called FFTshift that can shift the left half of output sequence to the right, thus moving “zero frequency” to the output pattern center. Therefore, during our calculation using FFT, an extra step of FFTshift is always needed.

The connection between the Fourier transform and the FFT has been given by equation (2.6c). Now with equation (2.8), we can finally connect the FFT function with (2.2) and to calculate the diffraction pattern in short-distance cases. The complete process is like this: (a) input electric field is multiplied with \( e^{\frac{j k}{2z_1}(x^2+y^2)} \).

(b) The product is operated on with FFT2 (2D FFT function) in Matlab.

(c) The result of FFT2 is adjusted by FFTshift as mentioned above.

(d) The shifted result is multiplied by \( e^{\frac{j k}{2z_1}(x^2+y^2)} \).

However, the process we described above still has a few problems. The first and most critical problem is that the derivation in [9] has moved the “\( z_1 \)” coordinate out of
the integral. This process is valid only when “$z$” is a constant inside the integral. Thus, the calculation process excludes the non-flat objective plane cases. This prevents the FFT method from simulating the stepped pistons’ case, which is the main target of our program. The second problem is in equation (2.6b). The input of the FFT is a matrix representing the electric field on the objective plane, and its result is a matrix that represents the electric field in the observation plane. Each value in the matrix gives the electric field in one point of the plane. The distance between each two neighboring points (we defined it earlier as the “resolution”) is given by equation (2.6b). If the input resolution is fixed by $\Delta x_0$ and $N$, the output resolution $\Delta x_1$ can not be adjusted. This eliminates some calculation flexibility for the FFT method.

Considering all these problems mentioned above, we decided to turn back to use equation (2.2) to program our simulation in Matlab. The FFT-based calculation is still used in some simple cases to validate the result of our main program. But the main part of our future analysis will still be based upon our own program using equation (2.2).

2.4 Brief introduction to the program

The program settings, performance and limitations are discussed in this part.

2.4.1 Parameter setting in the program

This part explains the settings of our program, based on the design requirement of our TTD piston MEMS array.
The parameters of the MEMS device we use in the lab are these: The MEMS pixel (the size of a single piston on the MEMS) is a 0.148mm × 0.148mm square, on a 0.15mm pitch. The fill factor is thus 0.99. It would take a large number of data points in the object plane to accurately represent this fine structure. Thus, a lower fill factor is used in many of the simulations to reduce the calculation time. It will be shown in chapter 3 that the fill factor does not significantly change the simulation results we are concerned with, so a lower fill factor simulation can still give a believable result.

The pixel may have some curvature due to the piston MEMS manufacturing process. The curvature radius is currently 6mm (concave) for the current device, and may be improved to 24mm in the next generation MEMS. We mention this curvature issue here as it affects the electric field on top of MEMS array, and may strongly affect the diffraction pattern. So it needs to be taken into consideration in the simulation program.

The operation of the White cell will be explained in Chapter 3, but here we note that a light beam large enough to illuminate a subset of MEMS pixels is circulating in the cell. Depending on the steering angle required and the number of bounces, the sub-array size may be 4×4, 8×8 and 16×16, which are all possible in our current lab devices.

The stroke (maximum dislocation) of each pixel is 2µm. Our current working wavelength is 633nm, and the next generation device will use 1.55um, as we mentioned before. Other parameters adjusted in the program will be described when
each simulation case is analyzed in next two chapters.

*Calculation of diffraction in Matlab*

Equation 2.2 is implemented on the computer (in MATLAB) as an addition process:

\[
plane_1(x_1, y_1) = \sum \left\{ \sum i \cdot Const \cdot plane_{01}(x_0, y_0) \cdot \left( z_1 - plane_{02}(x_0, y_0) \right) \cdot \frac{\exp(-ikr)}{r^2} \right\}
\]

(2.9)

Each term inside the parentheses of (2.9) corresponds to a term in (2.2). The term Const represents \( \frac{k}{2\pi} dx_0 dy_0 \), the product of all constant terms in (2.2) except imaginary number \( i \). The term \( plane_{01} \) is an array that represents the reflected electric field on the MEMS surface, which arises from multiplying the reflectivity profile of the piston mirrors with the incident electric field. Finally, \( plane_{02} \) represents the height of each MEMS pixel, corresponding to \( z_0 \) in (2.2). As coordinates in physical space do not include any electric field phase information, the numbers in \( plane_{02} \) can only be real numbers.

**2.4.2 Examples of simulation in the program**

One typical case of \( plane_{01} \) and \( plane_{02} \) mentioned in equation (2.9) are plotted in Figure 2.2. Figure 2.2a shows \( plane_{01} \), representing a uniform plane wave multiplied by an array representing a 4×4 array of pixels. Figure 2.2b gives the height of each pixel, including some curvature.
In the two figures above, the array has 4*4 cells. The cells are taken to be concave with curvature radius 6mm. The step height between neighboring columns in x direction is 0.775um, and step height in y direction is 0.465um. No light is reflected in the gap between pixels, so there are gaps between pixels in figure 2.2a and 2.2b.

Another example is shown below and its diffraction pattern is simulated.
(a) Electric field on the surface of MEMS

(b) MEMS surface height profile

(c) Diffraction pattern

Figure 2.3 Simulation example
An array of 8*8 pixels is used. As an example, we use the simplest electric field incident on MEMS surface, which is of unit amplitude and 0 degree phase across the entire MEMS array, as shown in figure 2.3a. This figure shows the field after being reflected off the MEMS, e.g. the spaces between the pixels are considered to have zero reflectance. The 0.465um step height is then applied between pixels in the \( y \) direction but no step is applied in the \( x \) direction. For simplicity, no pixel top curvature is included in this example. A wavelength of 1.55um is used here. The distance between the observation plane and the MEMS surface is 400mm. The diffraction pattern (light intensity) is shown in figure 2.3c. The peak of the light intensity is steered in the \( y<0 \) direction as the step height is increased in the positive \( y \) direction. Side lobes are visible in this diffraction pattern, which matches our prediction in figure 1.7b. Figure 2.3d gives the cross section view of the diffraction pattern in the \( y \) direction when \( x=0 \), showing that the exact location of steered peak is
The simulation result is also validated quantitatively. Using equation (2.7), it could be calculated that the Fresnel number for this case is

\[ N_F = \frac{(0.15\text{mm})^2}{1.55\text{um} \times 400\text{mm}} = 0.036 < 0.1 \]

proving it is in far field region. So we can apply \( \theta = \frac{\Delta \phi \lambda}{2\pi a} \) (equation (1.4)) to find the beam steering angle in theory. The step height between neighboring pixel columns is 0.465um, corresponding to 0.6 \( \pi \) phase delay at 1.55um wavelength. The steering angle obtained by (1.4) is 0.00310rad. In the simulation result, we can see from figure 2.3d that the peak intensity is steered 0.1757 degree in the negative y direction, equaling 0.00307rad. The simulation result and theoretical prediction matches very well with an error no larger than 1%. This proves that our program works correctly.

2.4.3 Angle Limit of our program

Although our program could work very well in small angle cases, it could not be applied simply in large angle cases. In other words, the observation point should not be far away from the optical axis. The reason for this is shown in Figure 2.4, where the diffraction processes in the real world and in Matlab are compared.
Figure 2.4 shows the diffraction process that happens in the real world, and it is assumed as a far field case. The electric field (rectangular function) diffracts into a far field pattern ($sinc$ function). However, Matlab can only solve discrete field problems. So we sample the continuous electric field into a matrix. This sampling process is shown in the upper half of figure 2.4. The dotted line in the upper half of 2.4b represents a group of $\delta$ functions working as the sampling function. The dotted rectangular function in 2.4c is the sampling result, which is also the actual “object” of diffraction in Matlab. Thus, the simulation result obtained in the Matlab program is a group of diffraction patterns (2.4 c lower half). It is the convolution result of the real diffraction pattern (figure 2.4a lower half) and the diffraction pattern of the sampling function (figure 2.4b lower half). We define the central diffraction pattern as the “central pattern”, which is what we really care about, and other diffraction patterns as “side patterns” or “aliasing” which need to be disregarded. Then we analyze quantitatively the distance between these diffraction patterns. According to the
property of convolution, the angular distance between diffraction patterns in the lower half of 2.4c is determined by the angular distance between $\delta$ functions in the lower half of 2.4b. We already know that the lower half of 2.4b is the far field diffraction pattern of the sampling function, so the angular distance can be calculated by the equation below.

$$\varphi = \arcsin\left(\frac{\lambda}{d}\right) \quad (2.10)$$

In this equation $d$ represents the sampling resolution, or the distance between two sampling points in the objective plane. The smaller $d$ is, the larger the angle $\varphi$ we will get. We hope that $\varphi$ can be as large as possible, but it is limited by the sampling rate we can have, so we would use the lowest sampling rate to see what is the smallest angle limit in the worst case. Any result observed below this angle would be quite believable in all other higher sampling rate cases. Thus we choose the lowest sampling rate and largest $d$ to give the strictest confinement to $\varphi$: 20 sampling points/pixel, meaning $d = \frac{150 \mu m}{20} = 7.5 \mu m$. When a wavelength of 1.55um is used as in the figure 2.2 example, $\varphi$ will be 11.9 degrees. And for 633nm cases, $\varphi = 4.84^\circ$. Considering that this $\varphi$ gives the distance between the centers of two neighboring diffraction patterns, we can only use half of this $\varphi$ value to make sure that most of the observed light intensity is the contribution of “center pattern” we need, rather than the “side pattern” we want to disregard. Thus, in the worst case, only the diffraction pattern in $\pm 2.42^\circ$ should be observed and evaluated for a sampling rate of 20 samples/pixel. In cases where larger angles must be investigated, a higher sampling
rate is needed, which increases the computation time.

Although the analysis above is based on far field diffraction, it still gives a useful impression about how a serious “side pattern” would affect the observation result. In future analysis, we will still keep in mind the observation angle we use to reduce as much as possible the effect of “side patterns” or “aliasing”. And we will carefully evaluate their effect on the observation result.
CHAPTER 3

DISCUSSION ON THE TTD DESIGN BASED ON WHITE CELL

3.1 Introduction

In chapter 1, we gave a brief introduction to the structure of the White Cell, showing its potential to achieve multiple reflections that we need for the TTD design based upon piston MEMS array, because multiple bounces can provide stroke amplification and steer the beam by a large angle. So in this chapter, we are going to give a detailed analysis of a White Cell structure, to see if it can meet our detailed requirements for the TTD process. We will evaluate how the White Cell structure will affect the wave front during the bouncing process. Some simplifications (such as lower fill factor) mentioned in chapter 2 will also be verified. During this process, we learned that a problem with the phase front can arise in the propagation inside the White Cell. The solution to this problem results in a new design based on the Fourier Cell, which will be given in chapter 4.
3.2 The reimaging process of White Cell based TTD piston array design

First the structure of White Cell based design is shown in figure 3.1, which is the same as figure 1.11. In this figure, the incident beam passes through the field lens and is incident on the input turning mirror (ITM). We call this trip of the beam $i_0$. Then the beam is reflected through the field lens again and focuses onto White Cell Mirror 1 (WCM1). This focusing trip of the beam is named $o_0$. Then the beam is reflected from WCM1 to the field lens and incident on the piston MEMS surface. This trip is called $i_1$. Then reflected beam goes through field lens again and refocuses on WCM2, the trip called $o_2$. The next trip of the beam from WCM2 to MEMS through field lens is called $i_2$. The number of total bounces is set to a fixed number determined by the width of incident beam and the alignment of the mirrors WCM1 and WCM2. The bounce pattern will be discussed in detail later.

Figure 3.1 Revised White Cell TTD beam steering system

In order to make sure that each output beam from the MEMS to WCM is focused precisely on the surface of the WCM, we require that the beams incident on the
MEMS should have planar wave fronts, and the focal length of the field lens should be exactly the distance between the field lens surface and the WCM surface.

The illumination pattern on the MEMS surface can be illustrated by figure 3.2. [11] In this figure, the x’s named CC1 and CC2 represent the curvature centers of WCM1 and WCM2. “In” shows the location of the ITM. The trip in0 ends on the top of ITM. The beam is wide enough to illuminate a sub-array of pistons (3 × 6 pistons in this figure). Then the beam is reflected to WCM1. When it is reflected back once again, the illumination pattern of i1 will be symmetric with the pattern on the ITM about CC1, so it will be in the location marked as “1” in figure 3.2. As the focal length of WCM equals half of the distance between MEMS and WCM, this round trip of the beam between MEMS and WCM1 is actually an imaging process of the field lens via spherical mirror WCM1, where the pattern on ITM is the object and the pattern on location1 is the image of it. Then the beam is reflected from the MEMS again and reimaged to location “2” after a round trip between MEMS and WCM2. So “2” and “1” should be symmetric about CC2. This process continues until the beam finally reaches the output location marked as “out”. Notice the black dot on each sub-array. It shows where a specific part of beam images every time on the MEMS. The total number of bounces is determined by the size of MEMS array and the illumination pattern width, that is to say, how many sub-arrays can fit on the MEMS.
Figure 3.2 Illuminated pattern on MEMS surface [11]

Then if the pistons of one sub-array (let’s say at location1) are moved as described earlier in the figure 2.2b and repeated in figure 3.3a, it should be seen that the point at which the beam is focused on WCM2 surface is shifted, as shown in figure 3.3b (repeated from figure 2.2c). But when the beam reimages back to location2, if it is a precise reimagining process, the image should still be in the same location as it would have been without any pistoning. The phase delay induced in the beam at location1 should only produce a corresponding phase delay for the image at location2. Then assume the pistons at location2 are also shifted; the phase delay will be added up directly. This delay accumulation is the ideal mechanism of the White Cell based TTD piston array design.
In next step, we will simulate to see if it is possible to realize the process described above.
3.3 Simulation process of the design

The simulation method and the result are shown in this section.

3.3.1 Determine the simulation method and basic parameters

Before starting the simulation, some details of the imaging process in the White Cell will be described first. We will discuss how to treat the field lens, how to treat the WCM, and how to choose fill factor and other simulation parameters.

*Field lens*

A bi-convex thin lens is shown in figure 3.4. Assuming that a plane wave is propagating from left to right through the lens, the lens induces some phase delay for each part of the wave front. By doing this, the wave front will become a converging
spherical wave front after it goes out of the lens, as shown in the figure. The point of convergence of all these spherical waves is the focal length $f$.

Thus, we can simplify the lens into a matrix in Matlab. Each cell in the matrix is a complex number, with unit modulus, representing a point on the lens. The phase of each cell represents the phase delay induced by the lens at this point. Thus, when a matrix of ones (representing a planar incident wave) is multiplied with this matrix (the wave passes the lens), the output wave front becomes a spherical wave. In the paraxial cases (all angles can be considered to be small), each cell in the lens matrix is given by

$$P = \exp\left(jk\left(R - \sqrt{R^2 - x^2 - y^2}\right)\right) \quad (3.1)$$

As we said, $P$ on each point of the lens is a complex number representing the phase delay and its modulus is 1. We use $R$ to represent the focal length, because it is also the radius of the output spherical wave front if the input wave front is planar. Besides, $x$ and $y$ are coordinates of this point on the lens, and they give the distance between this point and the optical axis.

This method to simplify the function of a field lens is examined by simulation. We use a single flat (no curvature) pixel of $1\text{mm} \times 1\text{mm}$ size to represent the incident planar wave front. A wavelength of 1.55um is used. A lens matrix defined by equation (3.1) is of the same width as the wave front, and the focal length used is $R=100\text{mm}$ (notice that distance is not necessarily in the far field region). The plane wave front departing from the pixel surface is multiplied with the lens matrix, producing the object of diffraction. The diffraction pattern is observed at 100mm (the same as the
lens focal length), giving the result shown in figure 3.5(a). In figure 3.5(b), as a comparison, the plane wave diffracts without the lens matrix. No other parameters are changed in figure 3.5(b).

Figure 3.5 Diffraction pattern of plane wave with and without lens
The effect of the lens matrix is validated by this figure. When a lens is added in, the wave front focuses very well into a spot around 0.3mm wide with little side lobes, while the diffraction pattern without the lens is as wide as 1mm wide and much more obvious side lobes are visible.

The simulation result with the lens matrix is not a singularity. Actually, from Rayleigh criterion the smallest spot focused by a lens of such diameter (1mm × 1mm) could be calculated using \( \sin \varphi = 1.22 \frac{\lambda}{d} \). Here \( \varphi \) gives the first zero of the spot, and \( d \) is the aperture of the lens. Using a 1.55um wavelength and 1mm aperture, we get 0.00189rad, corresponding to a spot radius of 0.189mm. Thus the spot size in theory should be 0.378mm wide. Considering that the simulation result resolution is not very high and we can not give the exact width of the spot in simulation, this theoretic result can be seen as matching the simulation result quite well. This result proves that we can use the matrix defined in (3.1) to simulate a lens of given size and focal length.

**WCM**

As we can see from figure 3.1, the WCM performs simply as a mirror. It does not perform any other function. And we already know that an ideal mirror only reflects the incident beam, changing the propagation direction, but does not interfere the diffraction process. That is to say, we can assume that a beam departs from a light source and gets reflected by a mirror 100mm away. If we observe the diffraction
pattern of the reflected beam at the location of light source, what we see should be a pattern that diffracts at 200mm distance (it propagates forward and gets reflected backward; each optical path is 100mm long).

This implies a simple way to jump over the WCM reflection in the simulation. If we need to simulate the beam propagating from MEMS surface, getting reflected by WCM and reaching MEMS again, we can simply simulate the diffraction pattern at a distance equaling two time the distance between the MEMS and the WCM (in other words, the lens focal length). In this way, one time of simulation is reduced and so is the error induced during calculation.

*Pixel number selection*

From figure3.2 it was seen that except for one time of reflection on the ITM and one on the output mirror, the other reflections share the surface of MEMS array. Thus, if we use a wider illumination pattern on the MEMS, the total number of bounces needs be reduced to prevent possible cross talk between two neighboring patterns. As the device we use now in the lab is a 32×32 pixels MEMS array, the biggest illumination pattern of each reflection for four bounces will be 16×16, which become our basic simulation configuration, as we show below in figure 3.6. Still, a 4×4 array is also used to evaluate the effect of some factors, such as the fill factor of pixels we will discuss below. But the validation of White Cell structure design itself is always based on 16×16 array simulation.
Fill factor issue

According to our current lab device conditions, the actual fill factor of each pixel is 99%. However, such a high fill factor in the simulation will reduce the simulation speed a lot. Thus, we would like to check carefully whether a lower fill factor simulation could give believable result. If positive, our simulation could be greatly sped up.

Two cases are simulated to examine the effect of fill factor. First is the far field simulation. We will use two different fill factors, 99% and 95%. To see how their far field pattern would be different. The second case uses the imaging process we described in White Cell. That is to say, a field lens is included in the form of a lens matrix, and then the diffraction pattern at two times the lens focal length is simulated using the parameters below.
Other parameters are listed here: wavelength = 633nm (wavelength used in lab), array size = $4 \times 4$, diffraction distance = 400mm (already in the far field region for this case), and observation plane size = 32mm (corresponding to 2.42 degrees mentioned in chapter 2). Pixel curvature is not included to ease the evaluation. The lens matrix is not included, neither.

<table>
<thead>
<tr>
<th>simulation</th>
<th>Case A</th>
<th>Case B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill factor</td>
<td>99%</td>
<td>95%</td>
</tr>
</tbody>
</table>

Table 3.1 Fill factor comparison

Figure 3.7 Cross section view of the diffraction pattern
The result is shown in figure 3.7. The solid line represents the case with 99% fill factor, while the dash line is the case with 95% fill factor. It could be seen that when fill factor shifts from 99% to 95%, the diffraction pattern in far field is quite unchanged. To be precise, their difference in the central lobe with highest energy concentration is negligible, while in some side lobes, there is some difference, about 5% of local light intensity at that point.

Most parameters in the second simulation use the same values as in the first one, while two changes are made. The lens matrix is added in, representing a lens with the focal length of 150mm, which is the lens we use in the lab now. The diffraction distance, as described in WCM issue part, is $2 \times 150\text{mm}=300\text{mm}$.

![Diffraction pattern with lens matrix at 300mm](image)

(a) Fill factor 0.99

Figure 3.8 Diffraction pattern with lens matrix at 300mm

Continued
The simulation results are shown in figure 3.8. The diffraction pattern at 300mm using the lens matrix is not affected by fill factor. Detailed analysis of the light intensity of these figures give similar result as central lobes in figure 3.7, proving that fill factor would not change diffraction pattern greatly.

However, there is still another exception we need to consider in the future. That is for the steered beam case. A simulation based on another group of fill factors is made. They are also diffractions of $4 \times 4$ arrays at a 400mm distance using a 633nm wavelength. But step heights are added to columns of pixels to induce a steering angle of 0.18 degree. The patterns in this case are shown in figure 3.9.
In this case, three fill factors are used. The highest fill factor (0.99) gives the lowest side lobes (24% height of the center lobe); the lowest fill factor (0.9) gives the highest side lobes (31% of the center lobe height); and 0.92 fill factor gives a result between them (29% of center lobe height). This result can be explained if we remember figure 1.7 (it is copied below as figure 3.10). We remember that the envelope (element factor) is determined by the diffraction pattern of a single pixel, while the array factor is determined by the number of pixels in the array. Thus, changes of the fill factor equivalently change the element factor and the diffraction envelope. The smaller the fill factor is, the wider the FWHM of the envelope will be, which means that side lobes would be at a location with higher element factor value. This is the reason for lower fill factor corresponding to higher side lobes. This problem is not so obvious for the non-steered case, because the side lobes in non-steered cases are near the zeros of element factors, and are mostly suppressed by them.
After describing all these special parts of the simulation, we will show the entire simulation process and its result next.
3.3.2 Simulation process and result

Figure 3.11 sums up the simulation process we are going to use for multiple bounces in the White Cell.

The solid line represents the MEMS profile matrix. First \textit{MEMS1} is multiplied with a dashed line which represents the lens matrix (\textit{lens1out}, meaning that it is the first reflection on MEMS and the field lens is operating on the beam output from the MEMS). The product of this multiplication is the object of diffraction. The diffraction process is shown as the arrow to the right. Then, the diffraction pattern is multiplied with another lens matrix (\textit{lens2in}, meaning the second reflection and the beam is heading toward the MEMS), corresponding to the reflected beam from WCM through field lens toward MEMS. We assume the field lens and MEMS are infinitely close together (in the lab, the field lens is about 10mm away from the MEMS, which is still quite a small distance); the product of this multiplication is the incident field on MEMS surface. Next, the field is multiplied with the \textit{MEMS2} (the second illumination and reflection on the MEMS) profile and lens matrix (passing through \textit{lens2out}) again.
This process continues and we can analyze the result at each step.

Based on the description above, the theoretic wave front propagation based on ray tracing can be shown in figure 3.12. The planar wave front propagates from left to right and becomes converging spherical wave after the first lens. After passing one focal length, it becomes an expanding spherical wave, and it is planar again after passing through the second lens. It can be seen that in theory the output beam should be a precise mirror image of the incident beam.

![Figure 3.12 Theoretic sketch of the White Cell mechanism](image)

Then, we can show the simulation results based on all the settings above. As we mentioned before, this process uses parameters as below. They will be applied to all four steps of simulation.
<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>wavelength</td>
<td>633nm</td>
</tr>
<tr>
<td>MEMS array</td>
<td>$16 \times 16$ array of pixels</td>
</tr>
<tr>
<td>Fill factor</td>
<td>0.95</td>
</tr>
<tr>
<td>Diffraction distance</td>
<td>300mm ($2 \times$ lens focal length)</td>
</tr>
<tr>
<td>Surface curvature</td>
<td>Infinite*</td>
</tr>
<tr>
<td>Step height</td>
<td>zero*</td>
</tr>
</tbody>
</table>

Table 3.2 Parameters for simulation

*The surface curvature and step height of MEMS surface have not been added in.

We would like to first evaluate the performance of the White Cell based structure conceptually as a TTD device. So the more “advanced” parameters such as pixel surface curvature will be added in only if we can prove that this design does not have serious aberration or some other similar problems.

The simulation results are shown in Figure 3.13.
(a) Beam starting from ITM (as the input field)

(b) First bounce pattern on MEMS

(c) Second bounce pattern on MEMS

Continued

Figure 3.13 Reflection light intensity
Figure 3.13 continued

In the last four figures above, we simulate the reflected beam incident on the surface of MEMS after passing through the field lens. These figures actually show the light intensity illuminating the MEMS surface at each time of reflection. As we are using $16 \times 16$ array of pixels as the object, the pattern should equal $2.4\text{mm} = 0.15\text{mm} \times 16$, and this size is also the image size we use in figure 3.13. If the White Cell based design worked correctly, the image we see should be almost uniformly illuminated pattern (the most serious aberration should be induced by the calculation
error) filling the entire image, which is far from what we really see above. The reason for this will be analyzed in next section.

3.4 Simulation result analysis and introduction of Fourier Cell

The simulation result in last section is analyzed below and a new design called Fourier Cell is introduced.

3.4.1 Simulation result analysis

We see that the evolution of the beam in figure 3.13 is far from our expectation. So we would like to find out why the simulation result is far from the expected result in figure 3.12. From [9], the diffraction pattern in the central focal plane between two lenses in figure 3.12 is given by

\[ E_i(x_1, y_1) = \frac{\exp\left(\frac{ik}{2f} \left(1 - \frac{d}{f}\right)(x_1^2 + y_1^2)\right)}{i\lambda f} \iint E_o(x_0, y_0) \exp\left(-\frac{ik}{f}(x_0x_1 + y_0y_1)\right) dx_0 dy_0 \]

(3.2)

In this equation, \( E_o(x_0, y_0) \) is the object, or the field on the MEMS surface, and \( E_i(x_1, y_1) \) is the field on the central focal plane. Also, \( d \) is the distance between the MEMS surface and the lens surface. This equation actually describes a kind of two-step diffraction process. The first step is the diffraction from the object (MEMS) to the lens surface (the Fresnel diffraction equation was used in this step [9]), and the second step is that the wave front with the lens induced phase diffracts to one focal length away after going out of the lens. We remember that the equation (2.2) in
chapter2 is the core of our simulation, and it only gives diffraction pattern of a single step diffraction. Thus, the diffraction described by equation (3.2) equals two steps of diffractions using equation (2.2). Besides, these two equations use similar approximations, such as the paraxial condition. That’s why we can use equation (3.2) to analyze the result given by equation (2.2).

According to equation (3.2), for an ideal White Cell, \( d=0 \) (That’s how we jump over the diffraction from the MEMS to the lens surface). In the lab, \( d \) is not zero but still much smaller than the lens focal length. Thus, except the part inside the integral of equation (3.2) which is a Fourier Transform, there is still a phase factor that precedes the integral. If we only care about the intensity distribution on the focal plane itself, this phase factor is of no importance. But now we are working on the diffraction pattern after this focal plane, and the phase problem really matters. We have not acquired a satisfactory Fourier transform on the central focal plane, this is the reason for us not receiving expected high-precision image on the observation plane.

### 3.4.2 Introduction to Fourier Cell structure

Analysis of equation (3.2) gives us one possible solution to the problem we mention above. That is to switch to a Fourier Cell structure from the White Cell. The basic idea of Fourier Cell has been introduced in chapter 1.
Figure 3.14 is the same as figure 1.13 to show the difference between the White Cell and the Fourier Cell. In the Fourier Cell, the object (MEMS) is one focal length away from the first lens, and the observation plane is also one focal length away from the second lens. In equation (3.2), this means that $d = f$, and the phase factor preceding Fourier Transform is cancelled out. Thus, we receive a perfect Fourier Transform in the focal plane between two lenses. The second step, of diffraction from the central focal plane to the observation plane, is another Fourier transform. So two Fourier transforms together give a high-precision image of the object.
A parallel simulation was done by another member of our group, using the commercial lens design software Zemax. The result is shown in figure 3.15. The top figure shows the intensity after one round trip for the White Cell case, while the bottom figure gives the round trip result for the Fourier Cell. In these two simulations, single pixel of $2.4\text{mm} \times 2.4\text{mm}$ size is used. The wavelength is $633\text{nm}$ and the focal lengths of all lenses are $300\text{mm}$.

![Comparison between simulation results of White Cell and Fourier Cell](image)

**Figure 3.15** Comparison between simulation results of White Cell and Fourier Cell

An interesting result is that Fourier Cell really keeps the fine details of object. This gives us confidence to extend our research into the realm of the Fourier Cell deeper, the discussion of which is given in next chapter.
CHAPTER 4

DISCUSSION ON THE TTD DESIGN BASED ON FOURIER CELL

4.1 Introduction to Fourier Cell and this chapter

As we have mentioned in previous chapters, the basic difference between the Fourier cell and the White cell in their structures is that in the Fourier cell, the objective plane is one focal length away from the first lens, and the observation plane is also one focal length away from the second lens. In White Cell, however, these two planes are close to the lens surfaces. This difference, as seen in the last part of chapter 3, causes great difference in their imaging effects. The White Cell structure induces an extra phase factor to its Fourier Transform, resulting in aberrations, while in the Fourier cell, the phase information is well-kept, and creates an image with a clear and fine edge.

Thus, in this chapter, we will design a TTD device based on the Fourier Cell. First we would like to double-check the validity of the Fourier Cell structure in Matlab. Then a TTD device design based on the Fourier cell is developed. At last we
will evaluate the performance of this design.

As the Fourier cell based design is already a next-generation device, we will mainly discuss cases based on a 1.55um wavelength.

### 4.2 Validation of Fourier Cell imaging performance

In this section, we would like to use our own simulation program to verify the Fourier Cell concept. During further simulation, two factors need to be addressed to ensure simulation success. First is the angle limit we have mentioned before. Another new limit is induced by Fourier Cell discussion in Goodman [9]. To ensure the validity of equation (3.2), the object must be in the Fresnel diffraction range of the lens. In other words, as sketched in figure 4.1, the MEMS is one focal length away from the lens. If we let light propagate inversely from right to left, the MEMS must be in the Fresnel diffraction range of this lens to ensure that in the original propagation direction, the light wave can diffract in the form given by equation (3.2). [9] The Fresnel diffraction range is given by the Fresnel Number \( N_F \) we have defined in equation (2.7) (repeated as equation (4.1)).

\[
N_F = \frac{a^2}{f^2} \geq 1 \quad (4.1)
\]
In equation (4.1), \( a \) is the lens aperture (for our lens, it is the lens diameter). Luckily, this condition is not strict and our design can meet it very easily. For example, if we use a lens of an ordinary size, such as 25.4mm, and 1.55\( \mu \)m wavelength, then equation (4.1) will only be violated if the focal length of this lens is larger than 416m! So we do not need to worry about this condition too much during our future work.

From figure 4.1, it can be seen that the simulation of the Fourier Cell in Matlab is separated into three diffraction steps. First is from the MEMS surface to the front surface of the first lens. The second step is from this lens’ back surface to the front surface of the second lens. The last one is from the second lens’ back to the observation plane. The effects of two lenses are included by multiplying with the lens matrix, which is the same as in White Cell cases.

The parameters we use for this simulation are listed in Table 4.1
Table 4.1 Simulation parameters

<table>
<thead>
<tr>
<th>Pixel numbers (N×N)</th>
<th>16×16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill factor</td>
<td>0.95</td>
</tr>
<tr>
<td>Focal length of two lenses</td>
<td>300mm</td>
</tr>
<tr>
<td>Lens size</td>
<td>10mm×10mm</td>
</tr>
<tr>
<td>wavelength</td>
<td>1.55μm</td>
</tr>
</tbody>
</table>

Figure 4.2 (a) shows the intensity at the source (MEMS surface). The simulation result is shown in figure 4.2 (b), which gives an almost uniform illumination pattern. Only at the edge of the image can we see some peaks higher than the center. Actually, this result is still a little different from our expectation. We had expected the result to be a precise image of the object in figure 4.2 (a). This may be explained by the simulation precision limit. That is to say, as the simulation resolutions we used during these three steps are not high enough, the edges of each pixel can not be very sharp (which can be also seen at the edge of the image). As all pixels are close together, their lower spatial frequency diffraction elements overlap with each other, resulting in a light intensity that looks almost the same all around the image.
We therefore tried to simulate another case, changing some parameters to see if our explanation is valid. As using a high resolution is too time-intensive, we tried a lower fill factor. If our thinking is correct, lowering the pixel fill factor will finally make the pixels visible on the observation plane.

We tried a fill factor of 0.8. All other parameters are unchanged. The simulation result is shown in figure 4.3. We can now distinguish each pixel in figure 4.3 (b),
proving that our guess is reasonable. But we also see that even without overlapping, every pixel’s image is still a little uneven, which is similar to the edges we see in figure 4.2 (b). Luckily, these are only due to the discrete sampling in the simulation used to represent continuous fields in reality. This means that the real case in the lab is equivalent to a simulation with infinitely high sampling resolution. So we can expect these problems to disappear in the lab experiment. Note that the scale of the color bars is different in the two plots.

Figure 4.3 Validation of the Fourier cell concept
Although the imaging quality in the finite-resolution simulation of the Fourier cell is imperfect, it still gives much better result than the White cell does under the same conditions. So we can continue to realize a Fourier Cell structure and develop a detailed device design. And some methods will be introduced in chapter 5 to improve the performance of the actual design.

4.3 TTD device design based on Fourier Cell

The innovative design of a TTD device based on the Fourier cell is introduced. In this section, we set down a set of design specifications, develop a design and evaluate its performance.

4.3.1 Description of TTD device structure based on Fourier cell

In the last section we verified the effectiveness of the Fourier cell in the imaging process. Next we will suggest an actual structure of a TTD device design based on the Fourier cell. Its validity and properties will be discussed in detail.

Since we are using a multiple-bounce system, and since the pixel array size is assumed to be $64 \times 64$ in this design, we investigate a system in which a separate MEMS chip is used for every bounce. This is because it is difficult to manufacture very large piston MEMS array, primarily because as the array size increases, the number of pins required to control them becomes too large for a single package. David Rabb proposed an innovative idea, [12] in which a single ball lens replaces the two thin lenses. The original structure is shown in Figure 4.4a, and the new structure
is shown in figure 4.4b. The dashed line in 4.4.b represents the MEMS chips, each one of which is a packaged MEMS array, and the beam bounces only once on each chip. The output optics shown in Figure 4.4b will be discussed in later sections. The Fourier mirror consists of a set of flat mirrors arranged on the surface of a cylinder. The MEMS chips and Fourier mirror forms a circle concentric with the Fourier lens. The circle radius is \( R \) and the lens radius is \( R_l \).

The key point in this design is that the Fourier lens material must have a refractive index between 1 and 2 to make sure that the focal length of this Fourier lens is finite and larger than its own radius, such that its focal plane is accessible. [12] Then the focal length of the Fourier lens can be calculated using equation (4.2) as a length larger than lens radius \( R_l \). [12]

\[
f = -\frac{1}{C} = -\frac{nR_i}{2(1-n)} = \frac{nR_i}{2(n-1)} \quad (4.2)
\]

Here \( n \) is the lens material refractive index; all other parameters were defined earlier.

![Concept of Fourier cell (copy from figure 4.1)](image)

(a) Concept of Fourier cell (copy from figure 4.1)

Continued
In figure 4.4 the MEMS chips and Fourier mirror are all placed one focal length away from the Fourier lens center. The incident beam enters this device through an opening in the place of one Fourier mirror segment, and illuminates the first MEMS on the opposite side. Then the beam is reflected back to the Fourier mirror again and gets re-imaged onto the next MEMS chip. After each round trip between a MEMS and a Fourier mirror, the beam is incident onto the next MEMS chip, and accumulates the step-height-induced phase delays. At last the steered beam outputs from the edge of the Fourier mirror, where it goes into the demagnification optics. Recall that the steering angle is given by equation (1.5) \[ \psi = \arcsin \left( \frac{h}{a} \right), \] where \( a \) is the element (pixel)
size. Our current pixel pitch is 150um. However, we can use lens groups to demagnify the beam by, for example, a factor of 50 while maintaining the beam quality. This is equivalent to a new pixel pitch as small as 3um. So the beam steering angle will be expanded by the same factor (50) after this demagnification. We will calculate the effect of this demagnification in detail later in this chapter, but we will not address the demagnification optics itself in this thesis.

The detailed design of the Fourier cell will be discussed in the next sections.

4.3.2 Consideration of lens material and package technology

In this section we discuss the two basic factors that may affect device size and performance: the lens material and MEMS packaging method.

**Lens material**

From the sketch in figure 4.4 we can see that the field lens is wide and thick, so in the interests of reducing weight, we considered first some optical plastics as the lens material, such as PMMA, polycarbonate and some other organic materials. However, later on we learned that plastics are not resistant to scratches, and are difficult to coat with antireflection layers. So we switched back to traditional non-organic materials. We selected four candidates listed in the table below, each of which is highly transmissive at 1.55um.
<table>
<thead>
<tr>
<th>Material</th>
<th>Refractive index</th>
<th>Density (g/cm³)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BK7</td>
<td>1.50065</td>
<td>2.51</td>
</tr>
<tr>
<td>Fused Silica</td>
<td>1.444</td>
<td>2.203</td>
</tr>
<tr>
<td>Sapphire</td>
<td>1.7462</td>
<td>3.98</td>
</tr>
<tr>
<td>Fused quartz</td>
<td>1.4585</td>
<td>2.203</td>
</tr>
</tbody>
</table>

Table 4.2 Available glass and crystal material

It seems that sapphire has a disadvantage for our device design. It is much heavier than other materials. Its refractive index is also far away from that of other materials, which we do not know for now whether it is good or not. But we will pay attention to this issue below, discussing which material may be better for our design and which is not.

Packaging method

The MEMS chips we use now are 32×32 piston arrays, but it is highly possible that next generation device could use 64×64 piston arrays. So we used this larger size for the design. The piston pixel pitch is currently 0.15mm, resulting in a MEMS array size of 64×0.15mm=9.6mm. In current device technology, this MEMS would normally be packaged in a ceramic carrier that is 36mm×36mm. A custom package, however, could realistically allow for a total package width about 25% larger than the
MEMS itself. So our design would use packaged MEMS of $9.6\text{mm} \times 1.25 = 12\text{mm}$ size. This new packaging method results in an increase of the package height because of the pins arrangement issue, but height is not restricted in our design.

4.3.3 Detailed design parameters

Device radius

The radius of the Fourier lens is determined by how many MEMS chips we want to place around it. And the number of MEMS is determined by how large angle we want to steer the beam by, since the number of bounces set the maximum time delay. So we have to figure out the beam steering angle limit of our design.

Remember in chapter 1 we talked about the element factor and array factor. Their effects on a steered beam case are shown in figure 4.5 (repeated from Figure 1.7). The envelope is determined by the element factor, and sharp peaks are determined by the array factor.
From the figure we can find that when the beam is steered, the central lobe is steered away from the figure center, with its intensity falling down, and the side lobe of the array factor will move towards the center of element factor, and its intensity rises. So here we get two criteria for the steering angle limit: (a) the central lobe of
array factor should not steer out of the FWHM of the element factor; (b) the first order side lobe of the array factor should not get into the FWHM of the element factor.

So first we consider how wide the FWHM can be. Form the expression of the element factor

\[
\sin^2 \left( \frac{\text{bak} x_i}{2L \sqrt{1 + \left(\frac{x_i^2}{L^2}\right)}} \right)
\]

(4.3)

and given the MEMS’ actual fill factor \( b=0.99 \), pixel pitch \( a=0.15\text{mm} \), and \( k=2\frac{\pi}{1.55\text{um}} \), we can extract that \( \arctan(x_i/L) = \pm 0.00462\text{rad} \) when element factor equals 0.5. So the FWHM width is 0.00924rad. The angular distance between two peaks in the array factor is given by

\[
\psi = \arcsin \left( \frac{\lambda}{a} \right) \quad (4.4) [8]
\]

For 1.55um wavelength and \( a=150\text{um} \), \( \psi = 0.0103\text{rad} \). So the FWHM width is slightly larger than then angle between two peaks of the array factor. That is to say, when the central lobe is steered to the edge of the FWHM, the first order side lobe is still out of the FWHM, and both of our criterions are satisfied. Thus, the maximum steering angle is \( \pm 0.00462\text{rad} \), excluding the final demagnification step.

We also apply the criteria for the demagnified beam. After the demagnification, \( a'=3\text{um} \), and the FWHM width is still larger than two times of the angular distance between two peaks of the array factor. So our criteria above will not be violated during the demagnification process even when the beam is steered to the maximum
angle. The effect of demagnification on our steering angle range will be left to the performance analysis section below. For now, our calculation is still based on 0.15mm pixel pitch.

Considering that the input beam is a plane wave normally incident onto a 64×64 pixels array, and the output beam from the last MEMS array is steered, there must be a stepped time delay profile accumulated during each bounce on the pixel arrays. Solving equation (1.5) using a total steering angle of $\psi = 0.00462\,\text{rad}$, and pixel pitch 0.15mm, the total true time delay from one pixel column of the array to the next column, expressed in distance, is $h = 0.693\,\text{um}$, at the end of all the bounces. Assuming the maximum pixel stroke is 2um and the array is 64×64 pixels, then in a single MEMS array, the largest step height variation across two pixel columns (in a single bounce) is $2\text{um}/63 = 0.0317\,\text{um}$. One bounce on the MEMS would result in two times this time delay because of the round trip (to and from the retracted piston). So after each one bounce, the beam can accumulate a true time delay, again expressed as a step height, of $h_i = 2\text{um} \times 2/63 = 0.0635\,\text{um}$ between two neighboring pixel columns. Thus the total number of bounces $M$ on the MEMS surface to reach a total delay $h$ is $M = \frac{h}{2h_i} = 10.9$, so we choose the total number of bounces to be $M=11$. Thus 11 MEMS chips in total are needed. From figure 4.4 we see that these 11 chips will take up nearly half the perimeter of the device. Two extra slots needed for the beam to input and output, and some extra space is needed for the convenience of assembling. As we do not have enough information available about the assembly space we need,
we left 1.25 times extra space for each MEMS, which means that 15mm space is left between each two MEMS chips. Thus the entire device radius is \( R = \frac{2 \times (11 + 2) \times 12 \times \pi}{2 \times 15 \times (2.25 + 1)} = 111.7 \text{mm} \), no matter what kind of lens material will be used. Based on equation (4.2) and the refractive index list of all four candidate materials, we can calculate the radius of the Fourier lens for four different materials. They are listed in table 4.3.

<table>
<thead>
<tr>
<th>Glass name</th>
<th>Refractive index</th>
<th>Lens radius (R₁)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BK7</td>
<td>1.50065</td>
<td>74.5mm</td>
</tr>
<tr>
<td>Fused Silica</td>
<td>1.444</td>
<td>68.7mm</td>
</tr>
<tr>
<td>Sapphire</td>
<td>1.7462</td>
<td>95.5mm</td>
</tr>
<tr>
<td>Fused quartz</td>
<td>1.4585</td>
<td>70.2mm</td>
</tr>
</tbody>
</table>

Table 4.3 Lens radius

We will next examine the required lens thickness for each material, and then we will figure out which material will result in the heaviest lens.

**Lens thickness**

We would like to use smallest lens thickness to reduce the total weight. On the other hand, we would like to use a lens that is as thick as possible to collect the most incident light energy.
There will be some loss during the diffraction process. Although the diffraction pattern actually distributes over an infinitely wide angle, we can only collect some of this energy when using a size-limited mirror or lens.

Figure 4.6 shows the process of diffraction from the MEMS surface to the lens surface. The MEMS (the black square) is placed on the outer perimeter. The diffraction pattern (cross-like pattern) is projected onto the front surface of the lens (the cylinder). We are not going to use an entire ball for the lens, just a slice, so the lens thickness \( t \), rather than lens radius \( R_l \), is the key factor during our consideration on the energy loss. The thicker lens we use, the less loss we suffer from.

In order to analyze quantitatively the relationship between the loss and the lens thickness, we simulate the diffraction process shown in figure 4.6, get the diffraction pattern at a virtual observation plane that is at the same location as the lens front surface, integrate the total diffracted light energy on this virtual plane, and see the
relation between the virtual plane size and the total energy loss when the diffraction pattern is collected by this plane. Thus, if we use a physical lens that is thicker than the observation plane size, we will ensure a light collection loss level lower than what the simulation predicts.

Our simulation parameters are listed in the table below.

<table>
<thead>
<tr>
<th>Pixel numbers (N×N)</th>
<th>64×64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill factor</td>
<td>0.95</td>
</tr>
<tr>
<td>Diffraction distance</td>
<td>40mm</td>
</tr>
<tr>
<td>Observation plane size</td>
<td>From 10mm×10mm to 45mm×45mm</td>
</tr>
<tr>
<td>wavelength</td>
<td>1.55um</td>
</tr>
</tbody>
</table>

Table 4.4 Simulation parameters

Here we choose a 40mm diffraction distance because it is close to the distance (R-R_l) for the cases of BK7, fused silica and fused quartz. The case of sapphire will be discussed later on using a conservative approximation with an explanation of the reason.
Figure 4.7 Observation plane size VS energy loss

Figure 4.7 shows the light energy loss level for different observation plane, and it matches our expectation that when a larger observation plane is used, more light is collected, and a lower loss level is achieved. We arbitrarily choose a cutoff of 0.1dB as the acceptable loss level. So the simulation result shows that we will need a virtual observation plane as large as 30mm×30mm. Thus, if we use 30mm as the lens thickness, we can make sure that the light collection loss on the lens surface is lower than 0.1dB.

However, another factor affecting the required lens thickness needs to be considered: steering angle. The lens thickness shown in figure 4.6 only takes into account the diffraction pattern without steering. The maximum steering angle range is $\psi = \pm 4.62 \times 10^{-3}\text{rad}$. For a diffraction pattern 40mm away, this angle corresponds to a light spot steering range of $\pm 0.18\text{mm}$. We want to meet the 0.1dB loss level even when the beam is steered to the extremely large angles up and down. So our lens thickness needs to be extended 0.18mm to both up and down directions, resulting in a total lens thickness of 30.4mm.
Then we turn to sapphire. We will use a conservative approximation instead of
direct simulation, because the sapphire lens is much larger in diameter than the other
lenses. So the distance between the sapphire lens and the MEMS will be smaller, and
a simulation using 45mm × 45mm observation plane in such case would mean that the
diffraction angle we simulate is much larger than the angle limit of our simulation tool,
and gives invalid results. So the sapphire lens case can not be simulated directly.

In the simulation process described in the paragraphs above, we collected the
light energy by using a lens of a given thickness and collecting light within a specific
diffraction angle. The larger then lens radius is, the smaller distance between the
MEMS and the lens, and thus the thinner the lens can be to catch the same angle. So
we should be able to use a smaller thickness for sapphire lens.

We already know that in far field, the diffraction divergence angle is a constant,
but in near field, it is not. Thus, using a constant divergence angle, we can give a
conservative estimate to the beam size and the energy distribution width in near field
region.

Figure 4.8 Light diffraction estimate
Figure 4.8 shows how this estimate works. The observation plane1 is the virtual observation plane for BK7, fused silica and fused quartz, and it is 30.4mm wide to ensure a light collection loss level lower than 0.1 dB. Then observation plane2 is for sapphire; if it is wide enough to cover a divergence angle as large as observation plane1 does, the observation plane (and the sapphire lens of the same thickness as the plane width) can also achieve the same loss level. This minimum size for observation plane 2, according to our calculation using simple geometry, is 18.0mm. So for a sapphire lens, 18.0mm thickness is thick enough to achieve 0.1dB loss cut-off. Combining the lens radius in table 4.3, the lens thickness we calculated above, and the densities of materials in table 4.2, we get the volumes and weights of the lenses made of all four materials.

<table>
<thead>
<tr>
<th>Glass name</th>
<th>Lens thickness (mm)</th>
<th>Lens radius (mm)</th>
<th>Density (g/cm³)</th>
<th>Weight (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BK7</td>
<td>30.4</td>
<td>74.5</td>
<td>2.51</td>
<td>1312</td>
</tr>
<tr>
<td>Fused Silica</td>
<td>30.4</td>
<td>68.7</td>
<td>2.203</td>
<td>977</td>
</tr>
<tr>
<td>Sapphire</td>
<td>18.0</td>
<td>95.5</td>
<td>3.98</td>
<td>2047</td>
</tr>
<tr>
<td>Fused quartz</td>
<td>30.4</td>
<td>70.2</td>
<td>2.203</td>
<td>1021</td>
</tr>
</tbody>
</table>

Table 4.5 Lens weight

Then we can see that the disadvantage of sapphire is obvious. It is big and heavy. Fused silica and fused quartz will be better. But quartz is usually sensitive to light
polarization, so we will use fused silica for future work.

The last factor we need to mention is the internal reflection in the lens. (It does not need to be total internal reflection, as we do not give the incident angle range of the light inside the lens. However, as we expect all light to be paraxial, TIR is the most possible case.)

![Figure 4.9 Sketch of internal reflection](image)

Based on the geometric calculation, we find that internal reflection may corrupt our device. The ray-tracing concept of internal reflection is sketched in figure 4.9. We notice that no matter what is the lens thickness, the light from bottom of MEMS toward top edge of lens will be incident onto the lens top, or to say, the angle between refracted beam inside this lens and the optical axis is always larger than zero. So internal reflection would probably happen, and the reflected energy will affect the shape of the beam on the Fourier mirror, and thus subsequent images. To prevent this effect, we suggest that the top and bottom surface of the lens should be coated with light absorption layer or abraded to prevent internal reflection.
Summarized device parameters

Figure 4.10 Top view of the design (repeated from figure 4.4)

Here we summarize all parameters we considered before. There are 11 MEMS chips placed around half perimeter of the lens. Each MEMS is a $64 \times 64$ pixels array packaged into a 12mm wide chip carrier. The other half perimeter is taken up by a segmented cylindrical mirror. The entire system radius will be 111.7mm. The lens is a spherical lens called the Fourier lens, with a radius of 68.7mm if we use fused silica lens. The lens thickness will be 30.4mm, and will weigh 977g.

4.4 Performance analysis

In this part we will discuss the possible performance that our device can achieve. Two main performance properties are of concern. First is steering angle range. Second is the energy loss level.
4.4.1 Steering angle

The maximum steering angle of our design based on the multiple reflections is $\pm 4.62 \times 10^{-3} \text{ rad}$, which is far from enough. So we use optical demagnification process to increase it after the multiple bounces inside the Fourier Cell.

If beam is demagnified for 50 times, then each pixel of the Fourier cell is equivalently now as small as 3um, but the phase delay induced across the array is not affected. Thus, we have $a'=3\text{um}$, and $h'=h=0.693\text{um}$. From equation (1.5), the steering angle is enlarged to $\pm 13.4^\circ$. This is the final steering angle range achieved by our design.

4.4.2 Energy loss

The total loss can be separated into three parts. First is diffraction loss caused by limiting the lens size and cutting off some of the diffracted energy at large angles. The diffraction loss at the first lens surface was discussed earlier. The second loss mechanism is reflection loss caused by partial reflection of a beam on the lens surface. The last factor is reflection loss on the MEMS surface and Fourier mirror surface. We will discuss them one by one.
The diffraction loss can be analyzed using figure 4.11 above. The first diffraction process is from the MEMS surface to the front surface of the Fourier lens (step 1). Step 2 is from the lens front surface to its back surface. The third step is from the back surface of the first lens to the front surface of the second lens, passing through the Fourier mirror (in the actual case, the beam goes from the lens surface to the mirror, gets reflected and goes back to the lens surface again). Step 4 is inside the second lens and the last step is from the back surface of the second lens to the MEMS again.

We have already chosen a lens thickness such that that the diffraction loss in step 1 is 0.1dB. This cut-off part of the diffraction pattern is marked as light blue in figure 4.11. And we also remember that if the Fourier lens thickness is increased, this loss can be reduced.

However, we should remember that there is still diffraction loss in step 2, marked as red in figure 4.11. This loss of energy, as we mentioned above, is absorbed by the
anti-reflection coating on the lens top surface or the abraded lens surface. What we care about here is the level of this loss. Compared to the beam in step 1, the beam in step 2 should be more paraxial due to the converging effect of the lens front surface; the diffraction diverging angle will be smaller; and the light energy diffracted onto the lens top surface will be smaller than the diffraction pattern that is cut-off in step 1, thus the loss level should be much lower than 0.1dB. And also, if the lens thickness is increased, this part of the loss will be reduced, because the top surface area is reduced for thicker lens, and more energy can be collected by the lens back surface, which has a larger area. In the extreme case, if the lens thickness is as large as two times of the lens radius, the lens becomes a glass ball, there will be no top surface, and all light getting into the lens front surface will be collected by the back surface.

In step 3, the diffraction process is only a mirroring process from the first lens’ back surface to the second lens’ front surface. We expect the energy distributions on these two surfaces are the same. So the loss in this step is negligible. Step 4 and 5 are only the reversed processes of step 2 and 1. All diffraction patterns in these two steps should be converging (as drawn in figure 4.11) rather than diverging. So the diffraction loss will be negligible, too.

To sum up, the main diffraction loss part comes from the diffractions in step 1 and 2. And in these two steps, the step 1 loss is the dominant part. Thus, we can simply use the first step loss (0.1dB) as the lower limit of the loss in one round trip of the beam, and $2 \times 0.1\text{dB}=0.2\text{dB}$ (if we assume that step 2 has, in the worst case, the same loss level as step 1) as the upper limit. Totally we have 11 round trips inside the
Fourier cell, thus our total diffraction loss will be between 1.1dB and 2.2dB.

Reflection loss on the Fourier lens surface

In order to discuss the reflection loss issue, we first repeat figure 4.6 here.

![Figure 4.12 Lens thickness issue (repeated from figure 4.6)](image)

It can be seen from this figure that the diffraction is actually projected to a spherical surface (the front surface of the Fourier lens), there will be four side lobes spread up and down, right and left, but the central lobe is in the center, carrying most energy. The central lobe is paraxial, so we can treat it as a normal incidence case. But for side lobes, we have to find some way to simplify the problem.
Figure 4.13 Reflection coefficient VS incident angle (in degrees)

Figure 4.13 shows the relation between the reflection coefficient and the beam incident angle. The refractive indices are 1.444 and 1 on two sides of the interface. For $S$ polarization, this coefficient rises as the incident angle increases. For $P$ polarization, it falls down to zero and then rises. For the four side lobes on four sides of the central lobe in figure 4.12, if two of them are $S$ polarized, the other two are $P$ polarized. For example, if the side lobes up and down are $S$ polarized, we can know that the electric field is horizontal in figure 4.12. Thus the side lobes on the left and right are $P$ polarized. Besides, all these four side lobes have similar incident angles because the interface is a spherical surface.

Thus, we get four conditions. (a) These four side lobes carry the same amounts of energy, and (b) according to our design size, they are incident onto the interface in angles no larger than 40 degrees, and all of them have similar angles of incidence. (c) If two of them are $P$ polarized, the other two are $S$ polarized. (d) According to figure 4.13, when incident angles are no larger than 50 degrees, the reflection coefficients
for two polarizations are almost symmetric about the normal incident beam reflection coefficient.

With these four conditions, the normally incident reflection coefficient will be a good approximation to the averaged reflection coefficient of the four side lobes. Thus, we can use the normal incidence condition not only for the central lobe, but also to calculate the averaged reflection loss of four side lobes.

In normal incidence cases, our Fourier lens coated with an anti-reflection layer can have a transmission coefficient as high as 0.997, which corresponds to a reflection loss of 0.013dB. One round trip in the Fourier cell includes four penetrations through the lens-air interface, so 11 round trips in our design in total results in a reflection loss of $11 \times 4 \times 0.013\text{dB} = 0.57\text{dB}$.

*Reflection loss on the MEMS and the Fourier mirror*

Typically, the MEMS mirror surfaces are coated with gold, which has a reflection coefficient of 97% (corresponding to 0.13dB of reflection loss). So the reflection loss on the MEMS surface for 11 bounces is $11 \times 0.13\text{dB} = 1.43\text{dB}$.

The Fourier mirror can also induce some loss. But as it can easily be coated with a high-reflection layer, the reflection efficient can be as high as 0.999. Thus the loss is 0.0043dB per reflection. For 11 bounces, the total loss is 0.05dB.

So we sum up all the loss from four mechanisms now. The diffraction induced loss is between 1.1dB and 2.2dB. The reflection loss on the lens surfaces is 0.57dB.
The reflection loss on the MEMS surface is 1.43dB, and on the Fourier mirror surface is 0.05dB. The total loss is around 3.2dB-4.3dB for an 11-bounce system. The diffraction loss at the lens and the reflection loss on the MEMS surface are the main loss sources. The former one can be reduced by using a thicker lens, at the expense of a heavier lens. The latter loss can be reduced by a better reflective coating on the MEMS surface, but the manufacturing complexity will increase due to this extra step.

The performance of this design can be summarized as below: the maximum beam steering angle is $\pm 13.4^\circ$ in 2 dimensions. The diffraction and reflection induced losses together total between 3.2dB and 4.3dB. In the next chapter we will discuss how to further improve the performance of our design and the work plan for the future.
CHAPTER 5

CONCLUSION

5.1 Summary of previous work

In previous chapters, we suggested a possible design of true time delay laser beam steering system based on the Fourier cell, which has some advantages compared to other designs. It is less sensitive to acceleration than dual-gimbal micro mirror design. It is more compatible to current MEMS manufacture processes than the electro wetting prism design. It can suppress side lobes to a lower level than grating based designs and phase shift designs. Besides, it can steer the beam continuously, while the gratings can only output the beam to discrete directions. And it can work in multi-wavelength systems, without beam squint (dispersion). All these advantages make this design applicable in various applications such as in optical communication systems.

The design is drawn in figure 5.1, and all parameters of it are included in table 5.1.
Figure 5.1 The design sketch (repeated from figure 4.4)

<table>
<thead>
<tr>
<th>MEMS chips</th>
<th>64 × 64 pistons array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piston (pixel) pitch</td>
<td>0.15mm</td>
</tr>
<tr>
<td>Fill factor</td>
<td>0.99</td>
</tr>
<tr>
<td>Maximum stroke</td>
<td>2um</td>
</tr>
<tr>
<td>Package size</td>
<td>12mm</td>
</tr>
</tbody>
</table>

(a) MEMS parameters

Table 5.1 Design parameters
Table 5.1 continued

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bouncing times</td>
<td>11</td>
</tr>
<tr>
<td>Fourier lens radius</td>
<td>68.7mm</td>
</tr>
<tr>
<td>Fourier lens thickness</td>
<td>30.4mm</td>
</tr>
<tr>
<td>Fourier lens weight</td>
<td>977g</td>
</tr>
<tr>
<td>System radius</td>
<td>111.7mm</td>
</tr>
<tr>
<td>Optical demagnification</td>
<td>50 times</td>
</tr>
</tbody>
</table>

(b) Fourier Cell parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steering angle</td>
<td>±13.4°</td>
</tr>
<tr>
<td>Energy loss</td>
<td>3.2dB–4.3dB</td>
</tr>
</tbody>
</table>

(c) performance parameters

Compared to the previous design based on the White cell, the new design using Fourier cell was proven to conserve a higher output beam quality, as we have shown in the simulations of chapter 4. Also from the analysis in chapter 4, we know that the main sources of losses are the reflections on the MEMS piston surfaces and the diffractions from the MEMS surfaces onto the Fourier lens. The first part can be reduced by replacing the gold surface coating with some higher reflective surface coating. And the second kind of loss can be reduced by increasing the lens thickness.
5.2 Future work

In this thesis, the design performance is mainly validated by simulations we designed in Matlab. The simulations have some limitations, however, such as the inability to handle large angle cases.

The future work calls for experimental demonstration of optical beam steering using a Fourier cell and demagnification of the output beam to magnify the steering angle. A laboratory demonstration using a White cell is being carried out by our group, and steering has been demonstrated, albeit with the degraded phase front predicted in Chapter 3 of this thesis. Thus future work should focus on building and testing a full scale system using a Fourier cell. Its advantages and disadvantages need to be finally verified by experiments, which can also provide feedback to help improve our design and simulation methods.
APPENDIX A

DERIVATION OF THE STEERED BEAM FAR FIELD
PATTERN ARRAY FACTOR
If we consider a one dimensional array of \( N \) pixels and the beam illuminating it has a planar wave front, the electric field on the array surface will be periodic. It is given by the equation (35) in [8], like this

\[
\bar{E}_0(x_0 + na, y_0) = \bar{E}_0(x_0, y_0) \quad \text{(A.1)}
\]

where \( \bar{E}_0(x_0, y_0) \) is the electric field at position \((x_0, y_0)\) on the array, \( n \) is an integer, and \( a \) is the pixel pitch. However, if a phase delay \( \Delta \varphi \) is induced between the wave front at each pixel surface, the wave front will have a stepped profile. And the electric field described in (A.1) can be revised to the form below.

\[
\bar{E}_0(x_0 + na, y_0) = \bar{E}_0(x_0, y_0)e^{i n \Delta \varphi} \quad \text{(A.2)}
\]

The diffraction process of this pixel array is drawn in figure A.1. Vector \( r_0 \) is from the origin to an object point on the pixel array. Vector \( r_1 \) is from the origin to a point on the observation plane. So their difference \((r_1 - r_0)\) is the vector between the object point and the observation point.

![Figure A.1 Physical situation of a diffraction process](Repeated from figure 2.1)

For the case without phase delay, the electric field of the diffraction pattern in far field paraxial case is given by equation (36) of [8], which is
Here \( \overline{E_1(r_1)} \) gives the diffraction electric field on the observation point \((x_1, y_1, z_1)\). The array has 11 pixels, so \( n \) takes 11 values, adding up the diffraction patterns from 11 pixels. For the case with phase delay, the electric field expression on the objective plane is changed from (A.1) to (A.2), and 11 pixels is replaced by \( N \) pixels, so equation (A.3) is revised to (A.4) below

\[
\overline{E_1(r_1)} = \frac{ike^{-ik|r_1|}}{2\pi|r_1|} \sum_{n=-N/2}^{N-1} dy_0 \int_{-a/2}^{a/2} dx_0 \overline{E_0(r_0)} e^{ik\left(\frac{x_1x_0+y_1y_0}{|r|}\right)} e^{in\Delta\varphi} \tag{A.4}
\]

Using a substitution for each pixel, we have for pixel \( n \):

\[
\begin{align*}
\begin{aligned}
x'_n &= x_0 + na \\
y'_n &= y_0 + na
\end{aligned}
\end{align*} \tag{A.5}
\]

Equation (A.2) and (A.5) are used to substitute terms in (A.4), and get the equation below.

\[
\overline{E_1(r_1)} = \frac{ike^{-ik|r_1|}}{2\pi|r_1|} \int dy_0 \int_{-a/2}^{a/2} dx_0 \overline{E_0(r_0)} e^{ik\left(\frac{x_1x_0+y_1y_0}{|r|}\right)} \sum_{n=-N/2}^{N-1} e^{in\Delta\varphi} e^{inx'_n} \tag{A.6}
\]

We notice that the summation is in the form of a geometric series. So (A.6) can be simplified as (A.7) below.
\[
\mathcal{E}_1(r_i) = \frac{ike^{-ik|\vec{r}|}}{2\pi|\vec{r}_i|} \int dy_0 \int dx_0 \mathcal{E}_0(\vec{r}_0) e^{i\frac{1}{2} \Delta \varphi} e^{i\frac{1}{2} \Delta \varphi} e^{\frac{Nkax_1}{2|\vec{r}_i|}} \sin \left( \frac{N}{2} \Delta \varphi + \frac{Nkax_1}{2|\vec{r}_i|} \right) \left( \frac{1}{2} \Delta \varphi + \frac{kax_1}{2|\vec{r}_i|} \right)
\]

(A.7)

Then, following the derivation of [8], the first part of (A.7)

\[
\mathcal{E}_1(r_i) = \frac{ike^{-ik|\vec{r}|}}{2\pi|\vec{r}_i|} \int dy_0 \int dx_0 \mathcal{E}_0(\vec{r}_0) e^{i\frac{1}{2} \Delta \varphi} e^{i\frac{1}{2} \Delta \varphi} e^{\frac{Nkax_1}{2|\vec{r}_i|}} \sin \left( \frac{N}{2} \Delta \varphi + \frac{Nkax_1}{2|\vec{r}_i|} \right)
\]

becomes the intensity part and the element factor of equation (1.2), and

\[
\sin \left( \frac{N}{2} \Delta \varphi + \frac{Nkax_1}{2|\vec{r}_i|} \right) \left( \frac{1}{2} \Delta \varphi + \frac{kax_1}{2|\vec{r}_i|} \right)
\]

becomes the steered array factor in chapter 1.
APPENDIX B

THE CODE OF THE MATLAB SIMULATION PROGRAM
The code for the Matlab simulation program used in this work is shown below:

```matlab
function z=draw2vec(celldimension,curv,code,a,z1,input,ff,R)

%celldimension is the dimension of the MEMS array, ranging from 1 to 16
%curv is the curvature of the mirror in millimeter
%code is the index determine which plane to output
%a is the pixel pitch (mm)
%z1 is the distance between objective plane and image plane (mm)
%input plane is the same size as objective plane to show the phase and electric field distribution at the objective plane
%ff determines if fft calculation of image pattern is used
%R gives the curvature of the image plane

%%%%--set objective plane parameters-----%
pscale=40;
%the size of one pixel
plane0=zeros(celldimension*pscale,celldimension*pscale,2); %objective plane
pixel=zeros(pscale,pscale,2);
%one pixel of objective plane
b=1;
%fill factor
k=2*pi/1.55e-3;
%wave number
x0i=0;
y0i=0;
%coordinate counting variables
x0=0;
y0=0;
%coordinates in objective plane
phasex=0;
phasey=0;
%the phase shift between two neighboring pixels
bias=0;
%the internal variable used to correct pixel shape

%%%%--set image plane parameters-----%
x1=0;
y1=0;
%x1 and y1 are coordinates in image plane
imagesize=300;
%size of image plane
shape=zeros(imagesize,imagesize);
```
%the shape of image plane
plane1=zeros(imagesize,imagesize);
%image plane1
x1i=0;
y1i=0;
%x1i and y1i are coordinate counting variables in image plane
range=2*z1/3;
%2*z1/range defines image plane range in mm
peak=[0,0];
%record the peak height and location in the image plane
phase=zeros(imagesize,imagesize);

%-----draw one pixel in the objective plane-----%
pixel((pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),(pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),2) =1;
%define the pixel reflectivity
if curv~=-inf
    for x0i=(pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2)
        for y0i= (pscale/2-b*pscale/2+1) : (pscale/2+b*pscale/2)
            %x0i and y0i are coordinate counting variables in objective plane
            pixel(x0i,y0i,1)=curv-sqrt(curv^2-((x0i-0.5-pscale/2)/pscale*a).^2-((y0i-0.5-pscale/2)/pscale*a)^2);
        end
    end
    if curv>0
        bias=max(max(pixel(:,:,1)));
    else
        bias=min(min(pixel(:,:,1)));
    %distance of moving the spherical surface to form our MEMS surface profile
    end
end
pixel((pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),(pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),1) =pixel((pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),(pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),(pscale/2-b*pscale/2+1):(pscale/2+b*pscale/2),1)-bias;
% spherical surface
end

%-----draw the entire objective plane-----%
for x0i=1:celldimension
    for y0i=1:celldimension
        plane0((x0i*pscale-pscale+1):(x0i*pscale),(y0i*pscale-pscale+1):(y0i*pscale),1)=pixel(:,:,1)+x0i*phase_x+y0i*phase_y;
    end
end
plane0((x0i*pscale-pscale+1):(x0i*pscale),(y0i*pscale-pscale+1):(y0i*pscale),2)=pixel(1:pscale,1:pscale,2).*input((x0i*pscale-pscale+1):(x0i*pscale),(y0i*pscale-pscale+1):(y0i*pscale));

%copy each pixel onto the objective plane
end
end

%-----output the objective plane-----%
if code==1
mesh(0:celldimension*a/(celldimension*pscale-1):celldimension*a,0:celldimension*a/(celldimension*pscale-1):celldimension*a,plane0(:,:,1).*plane0(:,:,2));
%show the objective plane height
    title('MEMS surface profile');
    xlabel('x direction (mm)');
    ylabel('y direction (mm)');
    zlabel('pixel height (mm)');
    return;
else if code==2
imagesc(0:celldimension*a/(celldimension*pscale-1):celldimension*a,0:celldimension*a/(celldimension*pscale-1):celldimension*a,abs(plane0(:,:,2)));
%show the objective plane pixel reflectivity
axis image; colormap(jet(256)); colorbar
    title('MEMS surface electric field');
    xlabel('x direction (mm)');
    ylabel('y direction (mm)');
    zlabel('light intensity');
    z=plane0(:,:,2);
    return;
end
end

%-----draw the image plane-----%
x0i,y0i=meshgrid(1:pscale*celldimension, 1:pscale*celldimension);
x0=(x0i-pscale*celldimension/2-0.5)/pscale*a;
y0=(y0i-pscale*celldimension/2-0.5)/pscale*a;
[x1im,y1im]=meshgrid(1:imagesize,1:imagesize);
if ff==1
const1=(imagesize-1)*(imagesize-1)/imagesize*2*pi*z1/k/a/celldimension;
const2=imagesize/2+0.5;
x1m=(x1im-const2)./imagesize*const1;
y1m=(y1im-const2)./imagesize*const1;
plane0f=plane0(:,:,2).*exp(-i*k/2./(z1-plane0(:,:,1)).*(x0.^2+y0.^2));
plane1=fftshift(fft2(plane0f));

plane1=plane1.*exp(i*k/2./(z1-plane0(:,:,1)).*(x1m.^2+y1m.^2)).*exp(i*k.*(z1-plane0(:,:,1)))*k/2/pi/i./(z1-plane0(:,:,1));
else
    constant=k/2/pi*(a/pscale)^2;
    const1=2*z1/range;
    const2=imagesize/2+0.5;
    x1m=(x1m-const2).imagesize*const1;
    y1m=(y1m-const2).imagesize*const1;
    for x1i=drange(1:imagesize)
        for y1i=drange(1:imagesize)
            x1=x1m(x1i,y1i);
            y1=y1m(x1i,y1i);
            \%shape(x1i,y1i)=R-sqrt(R^2-x1^2-y1^2);
            \%spherical image plane
            r01=sqrt((z1+plane0(:,:,1)+shape(x1i,y1i)).^2+(x1-x0).^2+(y1-y0).^2);
            \%distance between objective point and image point
            plane1(x1i,y1i)=sum(sum(i*(constant*plane0(:,:,2).*(z1+plane0(:,:,1)+shape(x1i,y1i))./r01.*(exp(-i*k.*r01))./r01)));
        end
    percent=100*x1i/imagesize
end

z=plane1;
\%output eletric field
plane1=(abs(plane1)).^2;
\%take complex modulus and calculate the light intensity

\%-----output image plane,light collection and light intensity distribution cross section view-----%
figure;
if ff==1
    imagesc(-(imagesize-1)*pi/k*z1/a/celldimension:2*pi/k*z1/a/celldimension:(imagesize-1)*pi/k*z1/a/celldimension,-(imagesize-1)*pi/k*z1/a/celldimension:2*pi/k*z1/a/celldimension:(imagesize-1)*pi/k*z1/a/celldimension,plane1);
else
    imagesc(-z1/range:2*z1/(imagesize-1)/range:-z1/range:2*z1/(imagesize-1)/range:-z1/range,plane1);
end
end
axis image; colormap(jet(256)); colorbar
line1='image plane( light intensity)';
line2=['element number=',num2str(celldimension^2),',objective-image distance=',num2str(z1),'mm'];
title({line1;line2});
xlabel('x direction (mm)');
ylabel('y direction (mm)');
return;
REFERENCES


