

TUNABLE-FOCUS MICROFLUIDIC LENSES FOR SMART EYEGLASSES

by

Nazmul Hasan

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

The University of Utah

December 2017

Copyright © Nazmul Hasan 2017

All Rights Reserved

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of Nazmul Hasan
has been approved by the following supervisory committee members:

Carlos H. Mastrangelo, Chair 6/9/2017
Date Approved

Hanseup Kim, Member 6/9/2017
Date Approved

Rajesh Menon, Member 6/15/2017
Date Approved

Berardi Sensale Rodriguez, Member 6/9/2017
Date Approved

Ashutosh Tiwari, Member 6/12/2017
Date Approved

and by Gianluca Lazzi, Chair/Dean of
the Department/College/School of Electrical and Computer Engineering

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Refractive errors of vision are very common in all human beings, namely myopia, hyperopia, presbyopia, and astigmatism. Worldwide, over 1 billion people were estimated to suffer from presbyopia with around 410 million of them suffering from near vision loss. The result of refractive errors is blurred vision, affecting our ability to focus on near or far objects. The utilization of conventional fixed, uniform, or graded power eyeglasses is generally unsatisfactory as fixed power eyepieces cannot provide any accommodation restoration.

In this dissertation, we demonstrate compact tunable-focus liquid lenses suitable for ophthalmic adaptive eyeglasses. These lightweight, low footprint tunable-focus lenses augment the accommodation range of vision, thus restoring normal vision function.

First, a tunable-focus large aperture liquid lens is constructed using shape memory alloy (SMA) springs as actuators. The lens mainly consists of a shallow liquid-filled cylindrical cavity bound by a thin compressible annular rim and encapsulated by a flexible circular membrane on the top of the rim and a rigid circular plate at the rim bottom. The lens optical power is adjusted by a controlled compression of the annular rim in vertical direction via actuation of the three shape memory alloy (SMA) springs.

Second, we report a compact tunable-focus liquid lens with large aperture actuated by piezo-electric bimorph actuators. The lens consists of a rigid annular sealing rim encapsulated by two membranes forming a sealed chamber with a fixed volume of

high index optical fluid filled in it. When a normal force is applied to the bottom piston via piezo-electric bimorph actuators, the shape of the top membrane is changed, causing the change of focal length.

We did the simulation using Python to improve the lens optical quality, and lens parameters were determined from the simulation. While simulation and fabrication of the tunable-focus liquid lens using piezo-electric bimorph actuators, we noticed the effect of tension over lens membrane to determine lens optical power and optical quality. This gave us the idea of implementing tunable-focus liquid lens by changing the tension of the membrane. The theory, simulation, fabrication, and experimentation for these three different lenses are discussed in this dissertation.

TO MY PARENTS

TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGMENTS	ix
Chapters	
1. REFRACTIVE VISION ERRORS AND CORRECTIVE DEVICES	1
1.1 Motivation and background	1
1.2 Corrective eyeglasses.....	3
1.3 Smart adaptive eyeglasses.....	6
1.4 Tunable-focus lenses for eyeglasses and thesis organization	8
1.5 References and links	11
2. TUNABLE-FOCUS LENS TECHNOLOGIES	13
2.1 Alvarez lens	13
2.1.1 Lens mechanism.....	13
2.1.2 Drawbacks.....	14
2.2 Liquid crystal (LC) lens	14
2.2.1 Working principle	14
2.2.2 Drawbacks.....	16
2.3 Variable focus lenses with shape changing mechanism	16
2.4 Applicability of liquid lenses for eyeglasses and need for new research	19
2.5 References and links	20
3. TUNABLE-FOCUS EYEGLASS LIQUID LENS ACTUATED BY SHAPE MEMORY ALLOY SPRINGS.....	22
3.1 Basic lens configuration.....	22
3.2 Lens power versus rim deflection	24
3.3 Actuator selection	26
3.4 Actuator and the external load	28
3.5 SMA spring actuation	31
3.6 Lens fabrication	34
3.7 Experiments	36
3.7.1 Lens optical power measurements	36
3.7.2 Lens wavefront measurements, lens quality, and discussion.....	39
3.8 Limitations	40
3.9 Further work.....	41

3.10 Summary	42
3.11 References and links	44
4. TUNABLE-FOCUS EYEGLASS LIQUID LENS ACTUATED BY PIEZOELECTRIC BIMORPHS.....	46
4.1 Variable focus eyeglasses	46
4.2 Piston actuation mechanism.....	52
4.3 High-voltage driver circuits	55
4.4 Fabrication	56
4.4.1 Lens frame and pistoned liquid chamber	56
4.4.2 Lens chamber liquid filling.....	59
4.4.3 Curved bimorph actuators.....	61
4.5 Results and discussion	62
4.5.1 Lens optical test setup.....	62
4.5.2 Optical power measurement	62
4.6 Wavefront profiling and aberration measurements.....	65
4.7 Autofocusing eyeglasses.....	67
4.8 Testing of autofocusing eyeglasses.....	69
4.9 Conclusion	70
4.10 References and links	71
5. TUNABLE-FOCUS EYEGLASS LIQUID LENS ACTUATED VIA MEMBRANE TENSION-ADJUSTING SMA CONTRACTION RING.....	72
5.1 Lens concept	72
5.2 Lens structure.....	74
5.3 Lens fabrication	77
5.4 Experimental setup.....	78
5.5 Results and discussion	81
5.6 Conclusion	85
5.7 References and links	86
6. CONTROL OF LENS ABERRATION.....	87
6.1 Effect of gravity on lens optical quality.....	87
6.2 Spherical aberration and astigmatism	88
6.3 Zernike polynomials	89
6.4 Simulation	90
6.4.1 Installing Python and other software	92
6.4.2 Results and discussion	93
6.5 Summary	97
6.7 References and links	99
7. EXPERIMENTAL SETUP FOR OPTICAL PROFILING	100
7.1 Shack-Hartmann (SH) wavefront sensor	100
7.2 Measurement of lens focal length.....	101

7.3 Profiling lens full aperture for measuring wavefront aberrations	103
7.4 References and links	105
8. CONCLUSION.....	106
8.1 Summary of the research	106
8.2 Future work.....	108
Appendices	
A. PROGRAMMING CODES USED IN THE DISSERTATION	109
B. SOFTWARE INSTALLATION PROCEDURE FOR SIMULATION.....	155

ACKNOWLEDGMENTS

It was a wonderful journey for last four years. Through this journey, I have come to realize that one has to be truly passionate about his/her PhD dissertation. The person who instigated this passion inside me is Dr. Carlos H. Mastrangelo. He is a true mentor and an outstanding researcher. Both my knowledge and skill developed under his guidance. I cannot thank him enough, and it is impossible to express my gratitude towards him. I would like to thank all my committee members for their guidance and advice as well. Dr. Hanseup Kim's constant advice put a big impact in this research. I am eternally grateful to my parents especially to my mother who dreamed for this. It is her constant support and inspiration which motivated me towards a PhD. All my family members from Bangladesh cheered me and supported me throughout my whole life. I love you all. And last of all, a big thank you to all of my lab mates, Aishwryadev Banerjee, Shashank Pandey, Niladri Banerjee, Fariha Khan, Mohit Karkhanis, Mehedi Hasan, Rugved Likhte, and Mashad Saleh. You guys rock. The research was sponsored by the US National Institute of Health NIBIB 1U01EB023048-01 co-operative agreement.

CHAPTER 1

REFRACTIVE VISION ERRORS AND CORRECTIVE DEVICES

1.1 Motivation and background

Degradation of vision is common in all human beings. The biological lenses in our eyes start to degrade creating refractive errors of vision by the age of 45. The four most common refractive errors of vision are myopia (nearsightedness); hyperopia (farsightedness), where far and nearby objects are seen out of focus, respectively; astigmatism, where vision is distorted by an irregularly curved cornea; and presbyopia which leads to loss of focal accommodation and difficulty in reading at arm's length [1-5]. Fig. 1.1 shows the vision impairment caused by myopia and hyperopia.



Fig. 1.1 Myopia and hyperopia, where people cannot focus at far and nearby objects, respectively.

In particular, presbyopia is the irreversible loss of the accommodative ability of the eye that occurs due to aging. Accommodation refers to the ability of the eye to increase its refractive power of the crystalline lens in order to focus objects on the retina. The most significant decrease in accommodative power occurs in between the ages of 20 and 50. In the first two decades of life accommodative amplitude has been shown to be relatively stable in the range of 7-11 Diopters. By the age of 50, accommodative amplitude typically decreases to about 2 Diopters which is typically 11 Diopters for 20 years of age as shown in Fig. 1.2 [6,7].

This decline occurs as a natural result of aging and ultimately affects any person reaching advanced enough age. Despite its ubiquity, the exact mechanism behind presbyopia remains unknown. Presbyopia is primarily an inevitable, age-related condition and accordingly its prevalence in a given population is related to the percentage of individuals surviving to old age. Worldwide in 2005, over one billion people were

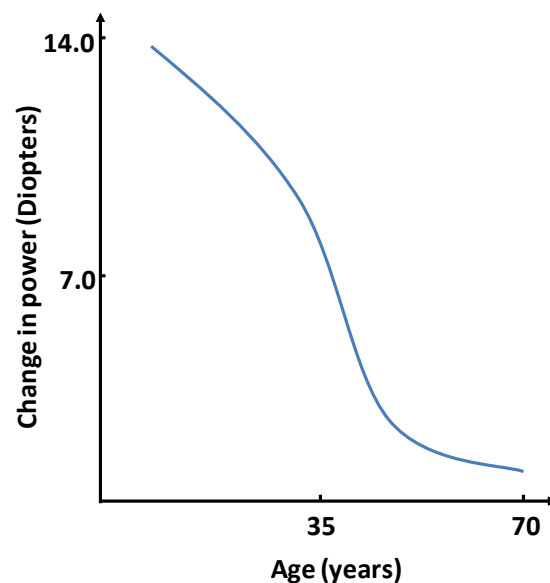


Fig. 1.2. Decrease in accommodation range as a function of age.

estimated to suffer from presbyopia alone with around 410 million of them suffering from near vision loss due to lack of vision correction [8].

1.2 Corrective eyeglasses

Refractive errors cannot be prevented, but they are treated with corrective devices such as glasses, contact lenses, and refractive surgery. The most common and inexpensive tool for correcting refractive errors of vision is prescription eyeglasses. Corrective eyeglasses are ancient devices that originated during the middle ages in Europe. Most historians believe that the first form of eyeglasses was produced in Italy by monks or craftsmen around 1285-1289. Reading eyeglasses were shaped like two small magnifying glasses and set into bone, metal, or leather mountings that could be balanced on the bridge of the nose. The first known artistic representation of the use of eyeglasses was Tommaso da Modena's painting in 1352 [9-12].

Modena's painting depicts monks reading and writing manuscripts. The first eyeglasses could only be used to treat hyperopia and presbyopia. Eyeglasses for myopia appeared much later, sometime in the early 1400s. Hinged glasses were not made until the 1750s. Bifocal eyeglasses were invented by Benjamin Franklin in 1784. Eyeglasses correct refractive errors by shifting the focal plane by a fixed diopter amount, but often these tools do not provide satisfactory solutions. Bifocal eyeglasses split the field of view which causes the vision impairment as well. This is illustrated by the example below.

Fig. 1.3 shows a schematic of a human eye [13]. The image is produced at the retina which is immersed in vitreous humor, a watery fluid with index of refraction $n_i = 1.33$. The lens equation for the human eye imaging system is approximately,

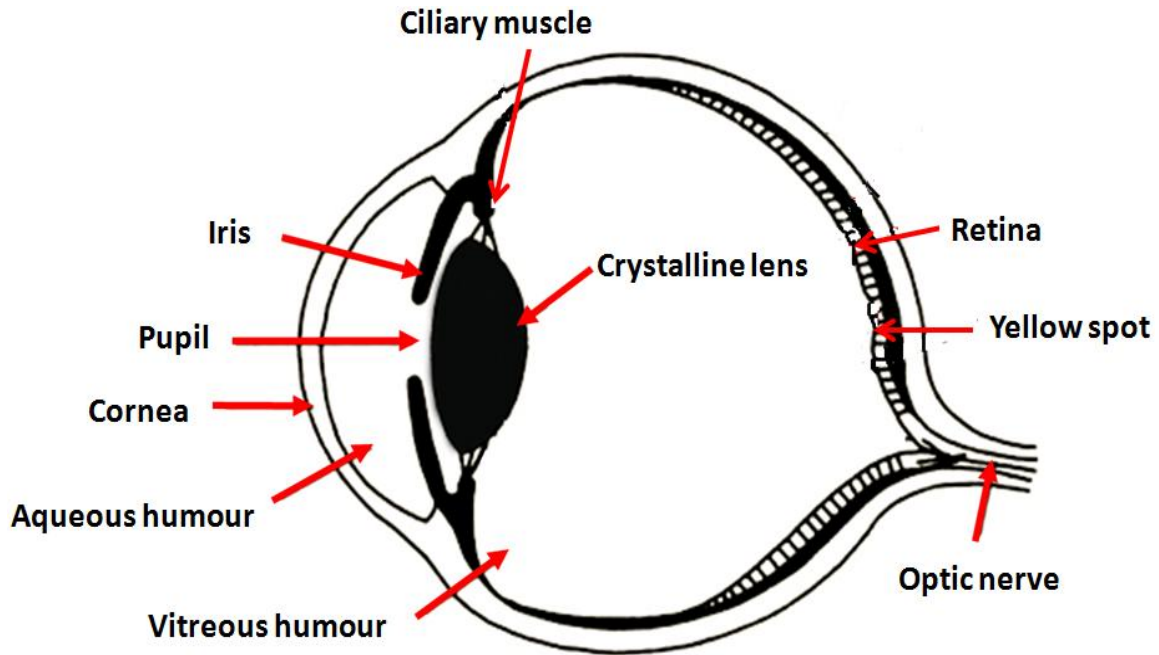


Fig. 1.3. Structure of a human eye. Images are formed at the retina inside the vitreous humor.

$$\frac{1}{s_o} + \frac{n_i}{s_i} = \frac{n_i}{f_i} = \frac{1}{f_o} \quad (1.1)$$

where, s_o is the distance between the object and the lens, s_i is the distance between the lens and the image, f_i is the image focal distance and f_o is the object medium referred focal distance. A healthy eye automatically adjusts the lens focal length to produce a sharp image at the retina independent of the object distance.

$$f_o(s_o, s_i) = \frac{s_o \cdot s_i}{s_i + n_i \cdot s_o} \quad (1.2)$$

The image focal length of the eye with an object at infinity ($s_o = \infty$) is approximately

$(f_i)_{\max} = s_i = 22$ mm corresponding to an object lens power (the inverse of $(f_o)_{\max}$) of +60 diopters. The normal eye can adjust its focal length to see objects between ~10 cm to infinity. Therefore, $(f_i)_{\min} \approx 19$ mm corresponds to a maximum object lens power of +70 diopters. The normal accommodation range of the human eye is $(\Delta P_{eye})_{normal} = 7-10$ diopters.

If the eye loses its ability to accommodate its lens focal length, objects in some regions of the 10 cm to infinity range will be projected on the retina out of focus producing refractive errors. Conventional eyeglasses correct these errors by placing a fixed focus lens of power P_{lens} in close proximity to the eye-lens between the object and the eye. For two or more thin lenses placed closely, the optical power of the combined lens system is approximately equal to the sum of the optical powers of each lens: $P = P_1 + P_2$. Therefore, the net corrective effect is approximately,

$$\frac{1}{f_o} \approx \frac{1}{(f_o)_{eye}} + P_{lens} \Rightarrow f_o = \frac{(f_o)_{eye}}{1 + P_{lens} \cdot (f_o)_{eye}} \quad (1.3)$$

For presbyopia and hyperopia, $(f_o)_{eye} > (f_o)_{\min} \sim 14$ mm and the image is projected behind the retina; hence we use a corrective lens with a positive power, $P_{lens} > 0$ to bring the image at the retina back in focus. For myopia, the situation is reversed as $(f_o)_{eye} < (f_o)_{\max} \sim 17$ mm projecting the image in front of the retina; hence we use a corrective lens with a negative power, $P_{lens} < 0$ for focusing. Note that, all that a fixed power corrective lens does is to provide a fixed shift in the effective object lens power of the eye to produce the "in focus" focal distance of Eq. (1.3). It does not change the

accommodating power range of the defective eye.

$$\Delta P_{eye} = \left(\frac{1}{(f_o)_{\min}} - \frac{1}{(f_o)_{\max}} \right) < (\Delta P_{eye})_{normal} \quad (1.4)$$

This is the reason why a fixed focus corrective lens can bring objects within a range in focus, but cannot correct the focus for the entire range of normal vision.

The modern conventional approach for vision correction over the entire object distance range is based on the utilization of bifocal or multifocal lenses where different regions of the vision field have different focal lengths. For example, in bifocal glasses it is common to increase the lens power on the lower half of the visual plane for reading, and so on. With this scheme it is not possible to image objects clearly over the entire visual field and these solutions still cannot restore the full accommodation. Many myopia and hyperopia sufferers often suffer from presbyopia as well; thus, they require several sets of eyeglasses with different monofocal, bifocal, trifocal, and progressive lenses [14-15]. Lens zoning greatly reduces the effective field of view, thus resulting in significant visual impairment.

1.3 Smart adaptive eyeglasses

The full field of view can be restored if the eyeglass lenses have a variable power that adaptively accommodates to the object distance. Fig. 1.4 shows the schematic of a smart adaptive eyeglass system. The smart adaptive eyeglasses consist of two tunable-focus lenses that can change their focal lengths. The eyeglasses need to have a control circuit and system with portable battery to provide power to the driving circuit. The

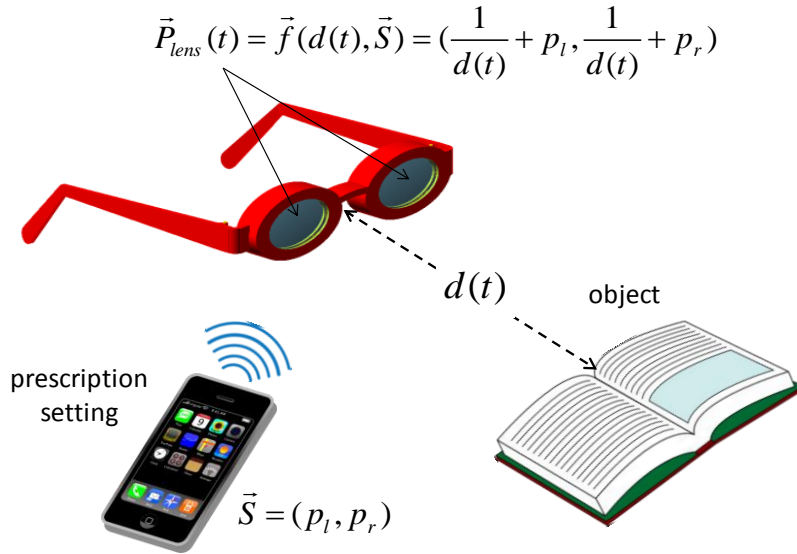


Fig. 1.4. Schematic of a smart eyeglass system. The adaptive lenses continuously adjust the lens power to bring the object in focus to the observer.

system settings and default lens power will be controlled by a phone application where the user enters his specific eyeglass prescription vector $\vec{S} = (p_l, p_r)$, consisting of the corrective fixed powers for the left and right eyes.

Once set, the prescription vector is wirelessly sent to the smart eyeglasses where it is registered. The smart eyeglasses continuously measure the distance from the lens to the object plane via distance sensors. Next, it calculates the required corrective lens power to bring the object into focus and changes the adaptive lens power accordingly.

In order to address the need of most eyeglass corrected problems, the lenses should have an accommodation range from -4 to +4 Diopters [4]. Furthermore, the lenses should be light and thin with the aperture of 30-45 mm in diameter [16,17]. The lens power should be adjusted at a minimum electrical power expense to ensure long battery operation. The lens has to be lightweight with small footprint.

1.4 Tunable-focus lenses for eyeglasses and thesis organization

None of the lenses discussed in the literature or commercially available have the characteristics suitable for adaptive eyeglass application. In this dissertation, we demonstrate several approaches to build such lenses with characteristics suitable for eyeglass applications. This is the main subject of this dissertation and the main contribution of this research.

The dissertation is organized as follows. Chapter 2 reviews the present technologies used for the construction of tunable-focus lenses and their various trade-offs that determine the maximum practical lens size and realizable optical power variation. The following three chapters discuss three different liquid lens implementations that could be utilized for ophthalmic eyeglass applications. The lenses must be lightweight and must have adequate aperture (~30 mm diameter) while providing significant accommodation (~4-8D). These tunable lenses are actuated by different electromechanical mechanisms.

In Chapter 3, we present a tunable eyeglass liquid lens actuated by low-profile shape memory alloy (SMA) spring actuators [18]. The liquid lens has aperture diameter of 34 mm, weighs 16.7 gm, and thickness 9 mm. This lens satisfied the weight and optical power goals for ophthalmic lenses, but the shape memory alloy springs are too energy inefficient. The lens actuation mechanism consumes a significant amount of electrical power which can drain batteries quickly. This problem motivated us to make tunable-focus lenses which are more energy efficient.

In Chapter 4, we demonstrate an eyeglass liquid lens actuated by low profile piezoelectric actuators [19]. This lens has an aperture diameter of 30 mm, overall

footprint diameter of 52 mm, optical power range ~6 Diopters, weight 14.2 gm, and thickness 8.4 mm. The lens has RMS wavefront aberration between 0.73 μm and 0.956 μm . The lens consumes electrical power less than 20 mW which is good for battery powered applications. This is our best performing lens suitable for ophthalmic applications.

While designing and testing the lens described in Chapter 4, we found the effect of pre-tension on the liquid lens. Tension plays a significant role in both optical power range and lens optical quality. This gave the idea for the final lens discussed in this dissertation. We varied the value of tension of the elastic membrane using a microshape memory alloy wire embedded in the membrane. By varying the tension, we tuned the focal length of the lens. This kind of lens is demonstrated in Chapter 5. The tension was varied using a shape memory alloy ring placed at the periphery of the membrane, which made the implementation of this lens extremely light. The lens has aperture diameter of 30 mm with optical power range of 2.2 D. The thickness and weight of this lens are 5.6 mm and 8 g, respectively, which are better than the previously discussed two lenses. However, slow response time and large electrical power consumption are the disadvantages for this lens.

The following chapters are dedicated to the modeling of optical quality and aberrations of ophthalmic-grade liquid lenses followed by experimental setup for lens profiling. Liquid lenses intrinsically exhibit coma aberrations due to the hydrostatic pressure effect caused by gravity. When placed in a vertical position, the elastic membrane of the lens bulges out more at the bottom than the top for the weight of the liquid, causing a significant amount of coma aberration. Furthermore, the actual shape of

the lens is also determined by the interplay of tension and flexural rigidity. Thicker membranes behave more like plates producing nonspherical deformations and a significant amount of spherical aberrations. Both of these aberrations reduce the lens optical quality. In Chapter 6, we discuss both coma and spherical aberration in liquid lenses from the theoretical standpoint and with simulation tools. We performed finite element analysis of membrane deflection for different pre-tension value and pressure. We also modified the structure of the membrane in order to reduce coma aberrations which works well for focusing, non-imaging applications. The simulation method and results are shown in this chapter.

In Chapter 7, we present in detail the experimental setup utilized for the measurements of the lens tunability and optical performance. The setup was constructed using a combination of beam expanders and a Shack-Hartmann wavefront sensor which is able to fully characterize the lens aberration parameters in terms of their corresponding Zernike polynomial expansions. Chapter 8 presents a summary of this work and a discussion on future directions.

The dissertation includes two appendices that describe the simulation and optical characterization programs utilized throughout this work for the sake of completeness and continuation of this research by others in future generations of ophthalmic tunable eyeglass lenses.

1.5 References and links

1. W. Tasman and E. A. Jaeger, *Duane's Ophthalmology* (LLW, 2013).
2. M. P. Keating, *Geometric, Physical and Visual Optics* (Butterworth-Heinemann, 2002).
3. S. H. Schwartz, *Geometrical and Visual Optics* (McGraw-Hill, 2002).
4. D. A. Goss and R. W. West, *Introduction to the Optics of the Eye* (Butterworth-Heinemann, 2001).
5. https://www.vsp.com/myopia.html?site_preference=SMARTPHONE
6. http://www.medicinenet.com/image-collection/farsightedness_picture/picture.htm
7. <https://www.framesemporium.com/print-page/how-to-read-your-prescription>
8. S. Resnikoff, D. Pascolini, S. Mariottia, and G. Pokharel, "Global magnitude of visual impairment caused by uncorrected refractive errors in 2004," *Bulletin of the World Health Organization* **86**(1), 63-70 (2008).
9. <http://www.historyofinformation.com/expanded.php?id=1755>
10. https://en.wikipedia.org/wiki/Tommaso_da_Modena
11. C. E. Letocha, "The invention and early manufacture of bifocals," *Survey of Ophthalmology* **35**(3), 226-235 (1990).
12. <http://www.ushistory.org/franklin/science/bifocals.htm>
13. <https://www.zigya.com/study/search/>
14. L. Johnson, J. Buckley, A. Scally, and D. Elliott, "Multifocal spectacles increase variability in toe clearance and risk of tripping in the elderly," *Investigative Ophthalmology & Visual Science* **48**, 1466-1471 (2007).
15. S. Lord, J. Dayhew, and A. Howland, "Multifocal glasses impair edge-contrast sensitivity and depth perception and increase the risk of falls in older people," *Journal of American Geriatric Society* **50**(11), 1760-1766 (2002).
16. T. Callina and T. Reynolds, "Traditional methods for the treatment of presbyopia: spectacles, contact lenses, bifocal contact lenses," *Ophthalmology Clinics of North America* **19**(1), 25-33 (2006).
17. <https://adlens.com>

18. http://www.deepoptics.com/do_site/
19. N. Hasan, H. Kim, and C. H. Mastrangelo, "Large aperture tunable-focus liquid lens using shape memory alloy spring," *Optics Express*, **24**(2), 13334-13342 (2016).
20. N. Hasan, A. Banerjee, H. Kim, and C. H. Mastrangelo, "Tunable-focus lens for adaptive eyeglasses," *Optics Express*, **25**(2), 1221-1233 (2017).

CHAPTER 2

TUNABLE-FOCUS LENS TECHNOLOGIES

An essential element of smart adaptive eyeglasses is the use of tunable-focus lenses. In this chapter, we review the technologies used to make tunable-focus lenses. There is a large amount of literature regarding the construction of variable focus lenses [1-17]. Tunable-focus lens technologies can be classified into three major categories: (1) Alvarez lens, (2) Liquid crystal lens, and (3) Liquid lens with shape changing mechanism.

2.1 Alvarez lens

2.1.1 Lens mechanism

One of the earliest attempts for the construction of variable focus eyeglasses is the Alvarez lens as that shown in Fig. 2.1 [3-5,18]. The Alvarez lens consists of two plano-convex-concave lenses sliding against each other. Each of the curved surfaces contributes to a positive or negative power for each interface of the lens. The net power depends on their relative position which can be adjusted to a positive or a negative power with a slider or screw. Alvarez lenses are commercially available with adjustable powers between -6 to +3 Diopters.

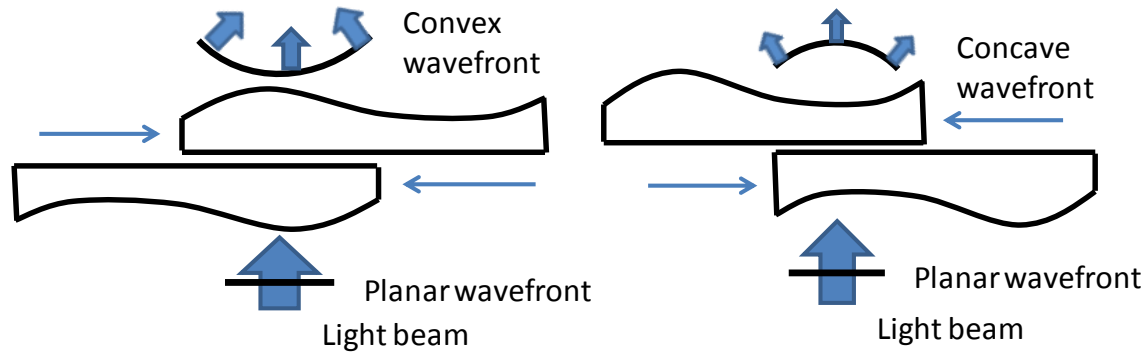


Fig. 2.1. The Alvarez lens concept. It consists of two shaped half lenses each with sections of opposing curvature. The lens power depends on the relative position.

2.1.2 Drawbacks

The Alvarez lens has unfortunately many visually disturbing issues including the presence of a visible gap, imperfections, and friction in the sliding glasses that virtually renders them useless for all practical situations. An additional problem with these types of lenses is that the field of view is severely reduced due to continuously varying resulting optical power.

2.2 Liquid crystal (LC) lens

2.2.1 Working principle

Variable focus lenses can be implemented using liquid crystals. The index of refraction in liquid crystal materials is a function of their applied electric field. One may therefore implement a lens simply by changing the voltage of a liquid crystal layer trapped between two pieces of glasses. Unlike regular shaped-surface lenses, LC variable focus lenses are flat and are based on graded index lenses (or GRIN lenses) which are commonly used in the fiber optics industry. Fig. 2.2 shows an example of a typical LC variable focus lens implemented by Sato [19].

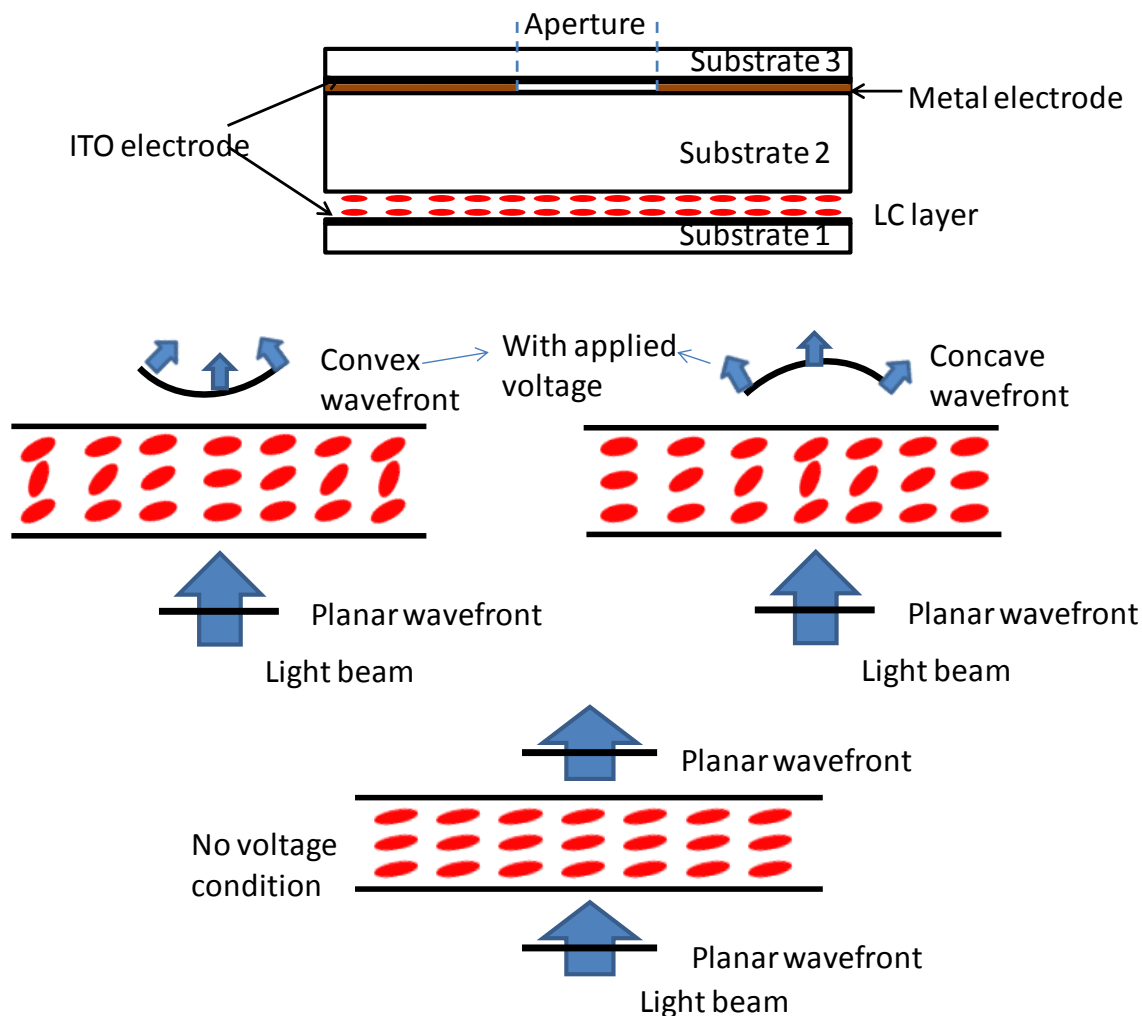


Fig. 2.2. LC graded-index type variable focus lens. The lens power is a function of the applied voltages.

The change in the phase with radius is formed by the resulting electric fields produced by a specific transparent Indium Tin Oxide (ITO) electrode shape. The aberration distortions produced by LC lenses are small, and Sato developed a lens arrangement that produces focusing for a wide range of light polarizations. In principle, LC lens is very attractive for eyeglass application because it takes very little electrical power to change the phase of the light. This implementation method indeed works very

well for lens with aperture diameter few millimeters and for single wavelength application. In the eyeglass application, however the aperture is large since the LC lens power, P_{opt} is

$$P_{opt} = \frac{\lambda \Delta \phi}{\pi \cdot r^2} = \frac{2 \cdot t \cdot \Delta n}{r^2} \quad (2.1)$$

where r is the lens radius, λ is the light wavelength, $\Delta \phi$ is the change in phase, Δn is the change in index of refraction, and t is the LC layer thickness.

2.2.2 Drawbacks

Because of the quadratic radius dependence, it is difficult to make a high optical power large aperture LC lens. A proposed method to avoid excessively large LC thickness is the use of Fresnel configuration [20-23]. Fresnel configurations generally have image quality issues related to the visibility of the grooves and circular noise due to diffraction [21,24].

2.3 Variable focus lenses with shape changing mechanism

Variable focus lenses (VFL) with larger apertures have also been realized with fluidic, flexible lenses by changing the shape of peripheral rim, inserting liquid in or out, and by changing aperture [1-12]. A liquid lens usually consists of a cylindrical bladder with flexible front or back surfaces filled with a transparent optical fluid. The shape of the lens is changed by either pumping fluid in and out or by squeezing of the bladder.

Fig. 2.3 shows the simple structure of liquid lens by pumping fluid in and out. The

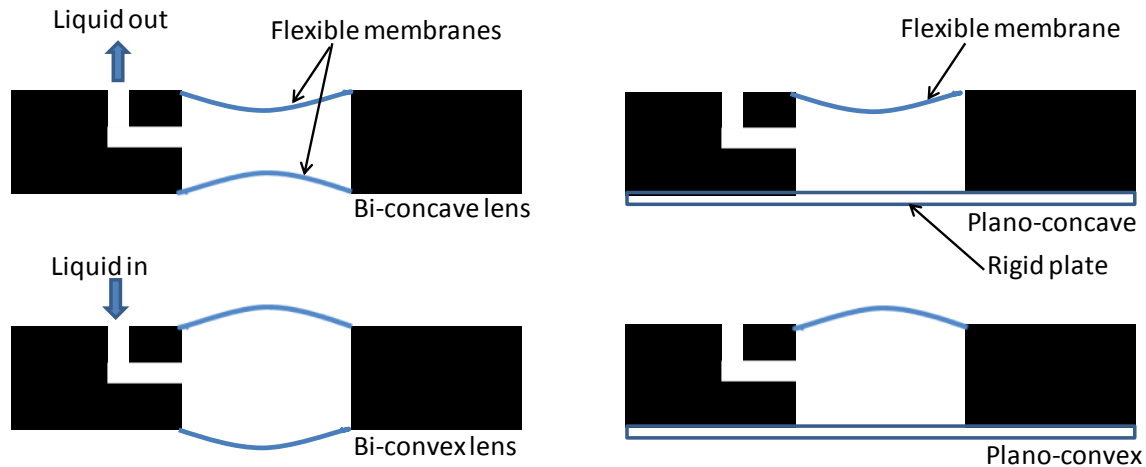


Fig. 2.3. Changing the shape of the lens by pumping fluid in and out.

flexible membranes of the lens bulge up or down as a result of fluid pumping in or out, thus changing the shape of the lens and producing focus change. A major challenge in liquid VFLs is the selection of the actuation mechanism. Several approaches have been reported with various degrees of successes including electrical motors, electrostatic forces, electrophoretic motion, and more recently piezoelectric actuators [1,6]. Ren et al. changed the shape of the lens' front elastic membrane by squeezing the peripheral sealing rim shown in Fig. 2.4(a) [7]. Oku et al. changed the shape of lens by pressing the flexible side membrane using a stack of piezo-electric actuators shown in Fig. 2.4(b) [25]. Son et al. changed the shape of the lens using an antagonistic winding-type shape memory alloy (SMA) actuator and piston [26]. All of these structures and choices of actuators make these tunable-focus lenses bulky, heavy, and large footprint which are inappropriate for ophthalmic especially eyeglass application. SMA winding-type actuator consumes large electrical power which drains the battery which is essential for longer operating time. Liquid lenses with dielectrophoresis forces have been demonstrated changing the shape of liquid droplet. Cheng et al. demonstrated lenses with two different immiscible liquids

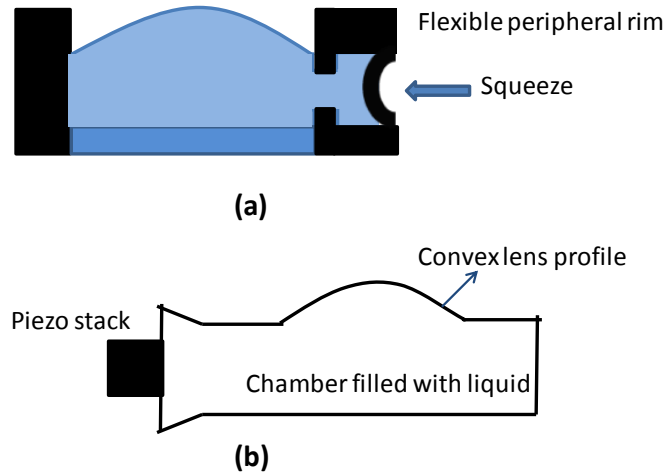


Fig 2.4. Different liquid lenses with shape changing mechanism.

in a cavity as depicted in Fig. 2.5 [27]. The two different liquids have different permittivities and refractive indices, but similar densities to minimize the effect of gravity. By applying the voltage, a non-uniform electric field is constructed and the shape of the droplet is changed as the liquids with different permittivities are attracted to the electric field in different ways. However, the tunable focus range of this lens depends on the difference of refractive indices of two liquids which makes the focus change very small. The small aperture size makes this lens inappropriate for eyeglass application as well.

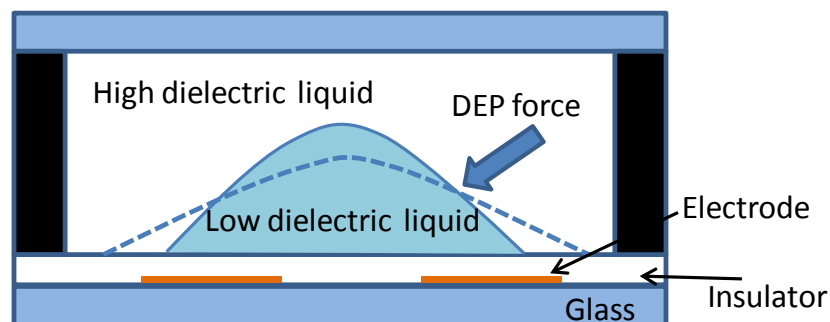


Fig. 2.5. Liquid lens with two immiscible liquids actuated by dielectrophoresis force.

2.4 Applicability of liquid lenses for eyeglasses and need for new research

The largest aperture electrically controlled variable-focus liquid lens commercially available today is manufactured by Optotune with a clear aperture of 16 mm. This aperture is too small for eyeglass applications [28]. Liquid VFL systems with larger apertures (~30 mm) have been recently demonstrated under table top laboratory conditions, but they have not been yet realized for light weight applications [29]. The above discussion in this chapter makes it clear that none of the lenses reported in literature or commercially available is appropriate for adaptive eyeglass application. The limitations in existing tunable-focus lens technologies prompted us to make tunable lenses appropriate for ophthalmic application. The following three chapters discuss such new implementations.

2.5 References and links

1. H. Ren and S.-T. Wu, *Introduction to Adaptive Lenses* (Wiley, 2012).
2. H. Jiang and X. Zeng, *Microlenses: Properties, Fabrication and Liquid Lenses* (CRC Press, 2013).
3. L. Alvarez, "Two-element variable power spherical lens," U.S. patent application 3305294 (1967).
4. L. Alvarez, "Development of a variable-focus lenses and a new refractor," *Journal of the American Optometric Association* **49**(1), 24-29 (1978).
5. O. Aves, "Improvements in and relating to multifocal lenses and the like, and the method of grinding same," GB patent no. 15735 (1908).
6. C.-P. Chiu, T.-J. Chiang, J.-K. Chen, F.-C. Chang, F.-H. Ko, C.-W. Chu, S.-W. Kuo, and S.-K. Fan, "Liquid lenses and driving mechanisms: A review," *Journal of Adhesion Science and Technology* **26**:12-17, 1773-1788 (2012).
7. H. Ren, D. Fox, P. A. Anderson, B. Wu, and S.-T. Wu, "Tunable-focus liquid lens controlled using a servo motor," *Optics Express* **14**(18), 8031-8036 (2006).
8. F. Carpi, G. Frediani, S. Turco, and D. D. Rossi, "Bioinspired tunable lens with muscle-like electroactive elastomers," *Advanced Functional Materials* **21**(21), 4152-4158 (2011).
9. S. Shian, R. M. Diebold, and D. R. Clarke, "Tunable lenses using transparent dielectric elastomer actuators," *Optics Express* **21**(7), 8669-8676 (2013).
10. M. Niklaus, S. Rosset, and H. Shea, "Array of lenses with individually tunable focal-length based on transparent ion implanted EAPs," *Proc. SPIE*, **7642** (2010).
11. N. Hasan, H. Kim, and C. H. Mastrangelo, "Large aperture tunable-focus liquid lens using shape memory alloy spring," *Optics Express* **24**(12), 13334-13342 (2016).
12. H. Ren and S.-T. Su, "Variable-focus liquid lens by changing aperture," *Applied Physics Letters* **86**, 211107 (2005).
13. N. Peyghambarian, G. Li, and P. Ayras, "Adaptive electro-active lens with variable focal length," U.S. patent application 0164593 (2006).
14. C. W. Fowler and E. S. Pateras, "Liquid crystal lens review," *Ophthalmic Physiol. Opt.* **10**(2), 186-194 (1990).
15. A. V. Naumov, G. Love, M. Yu. Loktev, and F. L. Vladimirov, "Control optimization

of spherical modal liquid crystal lenses,” *Optics Express* **4**(9), 344–352 (1999).

16. H. Ren, D. W. Fox, B. Wu, and S.-T. Wu, "Liquid crystal lens with large focal length tunability and low operating voltage," *Optics Express* **15**(18), 11328-11335 (2007).

17. S. Sato, "Applications of liquid crystals to variable-focusing lenses,” *Optical Review* **6**(6), 471–485 (1999).

18. <https://adlens.com/technology/alvarez-lens/>

19. M. Ye, B. Wang, and S. Sato, "Liquid crystal lens with focal length variable from negative to positive values," *IEEE Photonics Technology Letters* **18**, 78-81 (2006).

20. Fresnel Technologies Inc., "Fresnel lenses brochure,” (2003)
<http://www.fresneltech.com/pdf/FresnelLenses.pdf>

21. H. Ren, Y. Fan, and S.-T. Wu, "Tunable Fresnel lens using nanoscale polymer-dispersed liquid crystals,” *Applied Physics Letter* **83**(8), 1515-1517 (2003).

22. S. Sato, A. Sugiyama, and R. Sato, "Variable-focus liquid-crystal fresnel lens," *Japanese Journal of Applied Physics* **24**(8), 626-628 (1985).

23. G. Li, D. L. Mathine, P. Valley, P. Ayras, J. N. Haddock, M. S. Giridhar, G. Williby, J. Schwiegerling, G. R. Meredith, B. Kippelen, S. Honkanen, and N. Peyghambarian, "Switchable electro-optic diffractive lens with high efficiency for ophthalmic application,” *Proceedings of the National Academy of Sciences of the USA* **103**(16), 6100-6104 (2006).

24. Edmund Optics, "Fresnel lenses,” <http://www.edmundoptics.com/optics/optical-lenses/fresnel-lenses/2040/>

25. H. Oku, K. Hashimoto, and M. Ishikawa, "Variable-focus lens with 1-kHz bandwidth,” *Optics Express* **12**(10), 2138-2149 (2004).

26. H. Son, M. Kim, and Y. Lee, "Tunable-focus liquid lens system controlled by antagonistic winding-type SMA actuator,” *Optics Express* **17**(16), 14339-14350(2009).

27. C.-C. Cheng and J. A. Yeh, "Dielectrically actuated liquid lens,” *Optics Express* **15**(12), 7140-7145 (2007).

28. <http://www.optotune.com/products/focus-tunable-lenses>

29. L. Wang, H. Oku, and M. Ishikawa, "Variable-focus lens with 30 mm optical aperture based on liquid–membrane–liquid structure,” *Applied Physics Letters* **102**, 131111 (2013).

CHAPTER 3

TUNABLE-FOCUS EYEGLOSS LIQUID LENS ACTUATED BY SHAPE MEMORY ALLOY SPRINGS

3.1 Basic lens configuration

Fig. 3.1 shows the structure of the tunable eyeglass lens. A rigid lens back plate and a flexible membrane are attached to two sides of a compressible annular sealing rim implementing a variable-focus plano-convex or plano-concave lens. The membrane and back plate are 41 mm in diameter. The annular rim width and height are 3.5 mm and 5 mm, respectively. This produces a 41 mm diameter lens with a 34 mm aperture. The top membrane consists of 0.840 mm-thick layer of polymethylsiloxane (PDMS) and the bottom rigid back plate is made with 1.5 mm thick acrylic. The rigid washer, 1.5 mm thick of the same width as the rim, is used to hold the membrane in place. The lens is actuated by compression. As the rim of the lens is fairly wide (~3.5 mm) and the vertical displacement required to attain the desired optical accommodation is in the ~1 mm range, a relatively large force is needed for actuation. The low weight of the lens thus requires the use of actuators that can provide a high-density of actuation energy per unit volume and weight. Shape memory alloy (SMA) actuators are attractive in this application because of high driving force and high energy density applied to the load. Trading of displacement with force can be achieved compactly through the use of SMA springs as

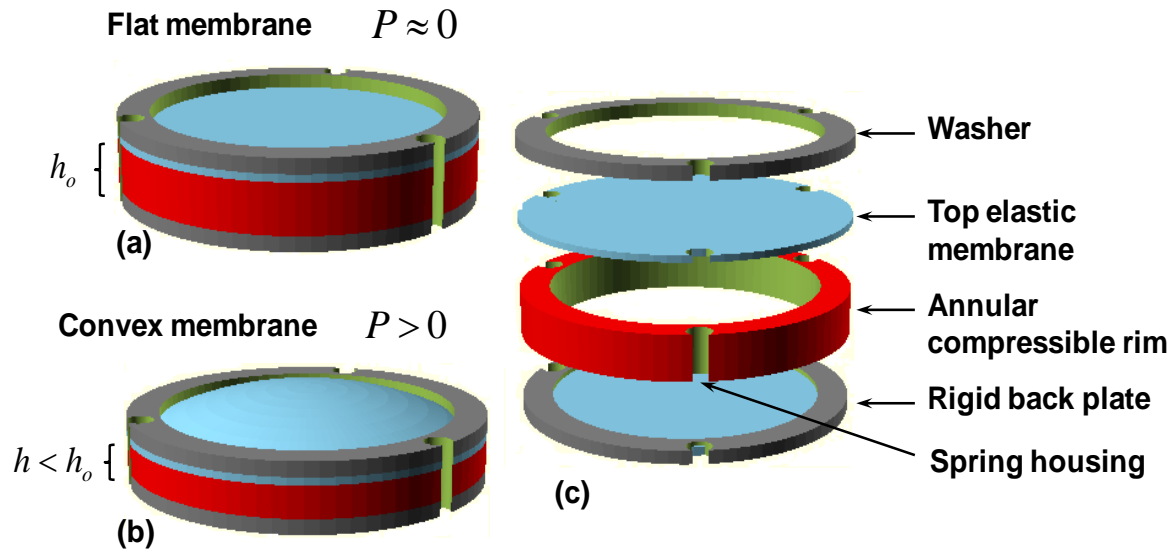


Fig. 3.1. Schematic showing the fluid lens construction. The lens consists of a top elastic membrane that produces the lens curved surface, a rigid bottom plate and a compressible notched annular rim of height, h_o . The interior of the lens cavity is filled with an optical fluid and the cavity is compressed at the rim with three vertical spring actuators spaced 120 degrees apart. (a) Lens in planar state, (b) Lens in convex state with compressed rim height $h < h_o$, and (c) exploded view showing the lens components (excluding the spring actuators).

is shown in Section 3.3 and 3.5 below.

The lens rim has three cylindrical notches spaced 120° apart, used for housing three SMA springs. The diameter of each spring wire is $500 \mu\text{m}$ and postdeformation length (martensite state) is 9 mm. Each spring and holder weighs 1.2 g. Both ends of the SMA springs are soldered to rigid flat copper contact pads. These contact pads are also tightly attached to the washer and back plate by screws. The lens liquid is inserted into the lens chamber by a small hole in the back plate which is later hermetically sealed and plugged by a screw.

The height of the annular sealing rim is the initial height, h_o . The amount of fluid introduced into the chamber sets the initial shape and power of the lens. If the initial power is set to zero, fluid is introduced until the elastic membrane is made flat. In the

absence of any applied force, the rim retains its initial height, h_o . When the SMA springs are electrically actuated, they squeeze and compress the elastic rim and the height, h of the fluid chamber decreases to, $h < h_o$. Since the liquid trapped inside the lens chamber cannot leak outside, to maintain the total volume of the liquid constant, the flexible membrane bulges outward (thus producing a positive optical power) as shown in the example of Fig. 3.1(b). The higher the compression is, the higher the optical power of the lens.

3.2 Lens power versus rim deflection

According to Fig. 3.1, the lens is cylindrical shaped in no focusing state. After squeezing the side ring, the flexible membrane becomes convex making the lens a plano-convex one. The lens can also be constructed as a plano-concave one. The lens optical power is given by the lens maker's equation,

$$P = \frac{1}{f} = (n-1) \left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)d}{nR_1R_2} \right] = \frac{n-1}{R} \quad (3.1)$$

Here, R_1 and R_2 are the radii of curvatures of the lens at each interface surface, d is the mid plane lens thickness, and n is the index of refraction of the liquid medium. For a plano-convex lens, $R_2 = \infty$ and $R_1 = R$. This makes the lens optical power independent of the thickness. The liquid volume of a lens with rim inner radius r_r and height h is [1],

$$V_l = \frac{\pi}{3} (R - \sqrt{R^2 - r_r^2})^2 \cdot (2R - \sqrt{R^2 - r_r^2}) + \pi \cdot r_r^2 \cdot h \approx \pi \cdot r_r^2 \cdot \left(h + \frac{r_r^2}{4 \cdot R} \right) \quad (3.2)$$

where $R \gg r_r$. Using Eq. (3.1), the fluid volume is expressed in terms of the optical power as,

$$V_l(P) \approx \pi \cdot r_r^2 \cdot \left(h + \frac{r_r^2 \cdot P}{4 \cdot (n-1)} \right) \quad (3.3)$$

For a variable focus lens, since the liquid volume is constant, any change in the optical power must originate from the change in the rim height. Taking the differential of Eq. (3.3) and setting it to zero, we obtain a direct relationship between lens power and height of the annular sealing rim,

$$\Delta P = - \frac{4(n-1) \cdot \Delta h}{r_r^2} \quad (3.4)$$

For example, a lens with rim inner radius of 17 mm (34 mm aperture) with optical liquid of index 1.33, a power change of 4 Diopters requires a change in rim height of $\Delta h \approx 0.9$ mm. This is the minimum thickness of the rim possible. The force required to squeeze the rim is proportional to the rim strain ($\Delta h/h_o$), the rim area, and the elastic modulus of the rim. A lower compression force is required if the rim height is made large, but doing so makes the lens heavy. This illustrates a basic trade-off between the actuator force requirements and the lens weight. For the lens implemented here, the lens thickness measured from the top washer to the bottom back plate is 9 mm.

3.3 Actuator selection

The vertical force squeezing the rim is applied using actuators. If we ignore the force required to deflect the thin elastic membrane, the actuator force required to produce strain, $\Delta h/h_0$ on the annular sealing ring is,

$$F_{act} = -\pi \cdot ((r_r + w_r)^2 - r_r^2) \cdot E_r \cdot \frac{\Delta h}{h_0} = -k_r \cdot \Delta h \quad (3.5)$$

where r_r is the annular ring's inner radius, w_r is the width of the annular ring, E_r is the Young's modulus of the rim material, and k_r is the spring constant of the rim and reasonably assumed as constant. If we utilize a very flexible 3M VHB tape rim with Young's modulus of ~ 200 kPa, it serves the purpose of both robustness and compressibility [2-3]. For, $r_r = 17$ mm, rim width $w_r = 3.5$ mm, rim height $h_0 = 5$ mm, and $\Delta h/h_0 = 0.1$, the calculated required actuator force, F_{act} is 8.2 N, and the average actuation energy is $E_{act} = 4.1$ mJ. This large force and strain requirement excludes almost all common driving micromechanisms [4]. Fig. 3.2 shows various actuators' power output per volume as a function of operating frequency [4].

Since a low weight is desired, we selected an actuation mechanism that has a high actuation energy density. Shape memory alloy (SMA) materials can produce actuation energy densities of ~ 7 J/g; thus yielding actuators that have very low mass. SMA materials have the ability to return to their predetermined shape from their deformed state when heated [5]. The shape change is caused by a phase change from austenite to martensite state or vice-versa which reconfigures the atomic lattice. Below the transformation temperature, SMA has low yield strength and can be deformed easily. In

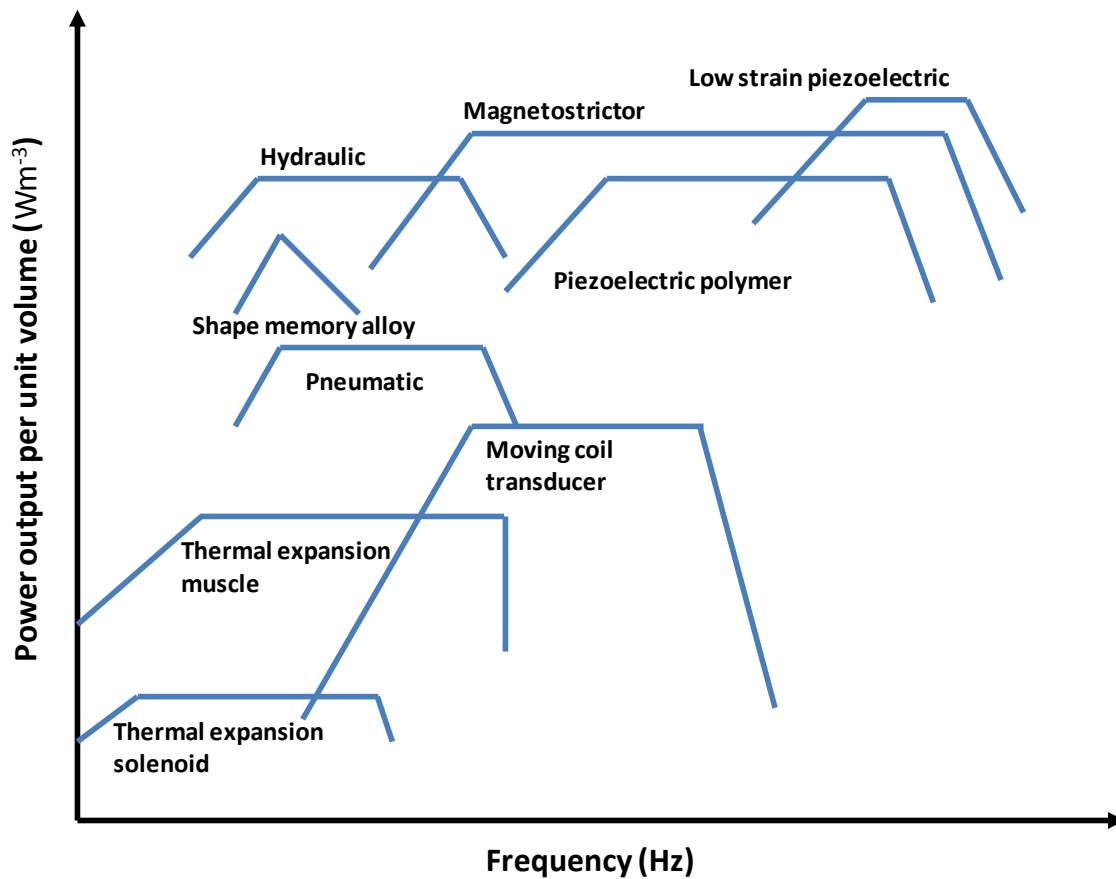


Fig. 3.2. Actuators' power output per unit volume as a function of frequency.

the most common SMA actuators configuration, the SMA length is set to the initial desired dimension, and when heated, the SMA shrinks producing a negative strain. The shape memory alloy, NiTi is capable of $\epsilon_{SMA} \sim 1\%$, reversible linear strain recovery for an effective life as high as 10^7 cycles (provided the detwinning strain threshold is not exceeded) [6-7]. The Young's modulus of NiTi is typically 40 GPa in cold state and 85 GPa in the hot state [5]. A 0.5 mm diameter SMA wire can thus produce as much as 8 kN of force simply due to the change in elastic modulus between the two phases.

3.4 Actuator and the external load

For the selection of any actuation mechanism, we need to evaluate its load curve first. For a fixed actuation voltage, the force and displacement for that actuator lies on that curve as shown in Fig. 3.3 [8]. The load line for the actuator is AB. Depending on the actuators; load lines can be of different shapes. The intersection points along with AB along the force axes and displacement axes give blocked force, F_b and free displacement, δ_{fr} . For a fixed applied voltage, the force given by the actuator can be expressed as,

$$F_o = F_b \left[1 - \frac{\delta_o}{\delta_{fr}} \right] \quad (3.6)$$

$$F_o = F_b - \delta_o K_{act}$$

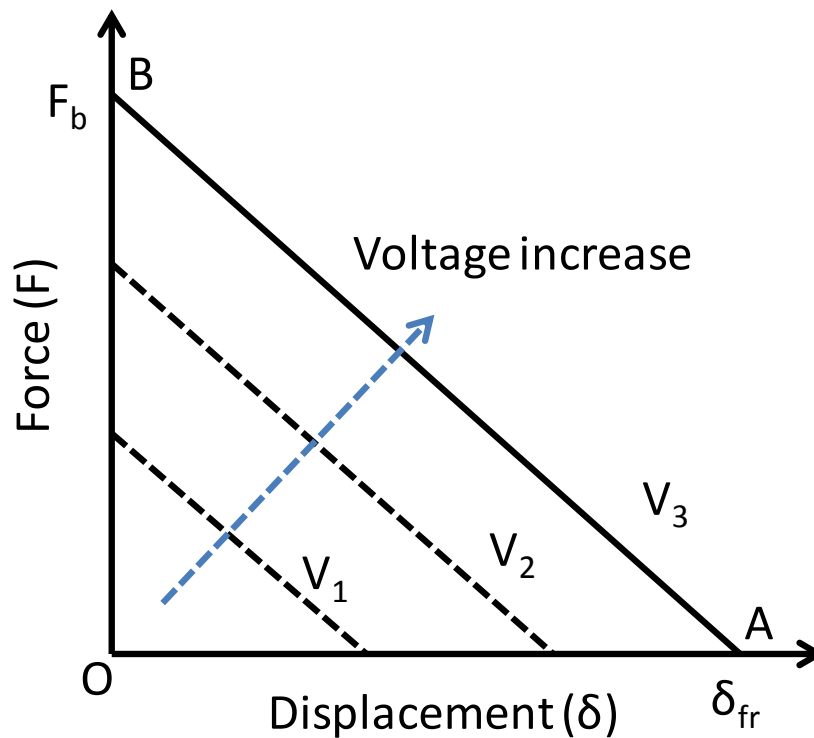


Fig. 3.3. Load line for actuator.

and the displacement can be expressed as,

$$\delta_o = \delta_f - \frac{F_o}{K_{act}} \quad (3.7)$$

Here, K_{act} is the stiffness of the actuator. When an external load is connected to the actuator, the equivalent system can be described as shown in Fig. 3.4 [8]. The external load with stiffness K_{ext} can be modeled as two springs in series exerting force upon each other. Fig. 3.5 shows the force-displacement curves of the actuator load system. The spring load is OC, which intersects the load line AB at point C. C is the equilibrium point of the system [8].

The force and displacement values at point C are,

$$\begin{aligned} F_o &= F_b - \delta_o K_{act} \\ F_o &= K_{ext} \delta_o \end{aligned} \quad (3.8)$$

and

$$\delta_o = \frac{F_b}{K_{act} + K_{ext}} \quad (3.9)$$

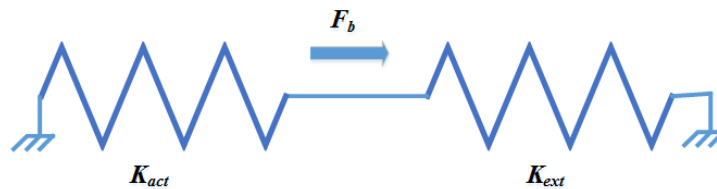


Fig. 3.4. Actuator with external load model.

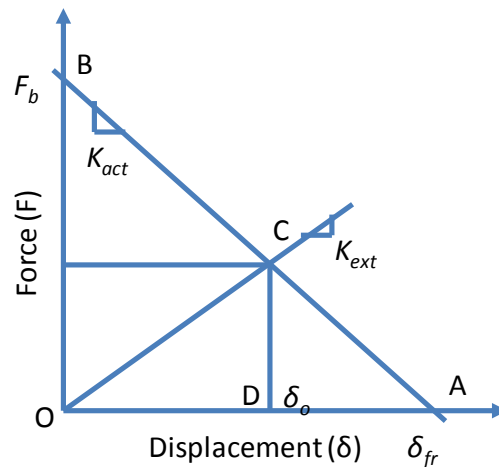


Fig. 3.5. Load line intersection point with external load.

The maximum energy can be exerted from the actuator if the stiffness of the external load matches with the stiffness of the actuator.

For SMA spring actuators, the exerted force versus displacement plot has two lines OA and OB, as the Young's Modulus changes as a function of thermal energy (shown in Fig. 3.6) [9]. Here two different load lines give two different forces and displacements at two different temperatures for a fixed load. The fixed load is represented

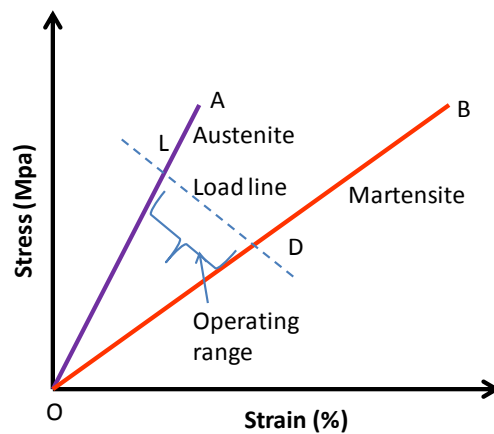


Fig. 3.6. SMA spring stress as a function of strain in linear zone. Load line excursion is reversible if the strain is kept below plastic deformation regime.

by the dashed line LD. The intersection points L and D with the load line OA and OB gives us the operating stress and strain range for any particular SMA spring actuator and load system. Note that, load line excursion is reversible if the strain difference is sufficiently low and kept below plastic deformation regime.

3.5 SMA spring actuation

SMA wires can produce very large forces but cannot produce large strains. Our lens requires strains of about 10% and much lower forces; therefore, it is necessary to use an SMA actuator with a leverage mechanism that trades strain with force. A compact leverage mechanism is an SMA helical spring [9-10]. A helical spring expands or contracts along its main axis by torsion of a wire wound up in a helix [11]. The easiest way to analyze the spring behavior is to consider the behavior of the unwound spring as shown in Fig. 3.7 [12]. The spring consists of a wire of diameter d wound up in a helix of n loops and diameter $D=2R$. The unwound spring is subject to torque $P \cdot R \cdot \cos(\alpha)$ causing

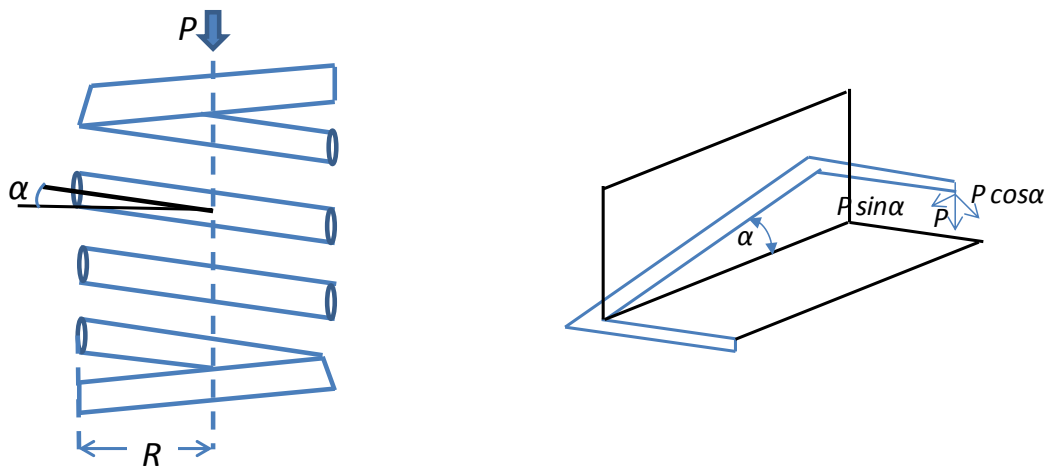


Fig. 3.7. Analysis of helical spring deflection. The observed deflection is equal to that of the unwound beam subject to torque.

it to deflect. The spring constant of the helical spring is,

$$k = \frac{G \cdot d^4}{8 \cdot N \cdot D^3} \quad (3.10)$$

Here, $G=E/2(1+\eta)$ is the spring shear modulus, E is the Young's modulus, and η is Poisson's ratio. The larger the number of turn is, the smaller the spring constant. The relationship between the spring internal shear strain θ and the spring linear strain ε_s is,

$$\varepsilon_s \approx \frac{\pi}{W} \cdot \left(\frac{D}{d}\right)^2 \cdot \theta \leq \frac{\pi}{2 \cdot W} \cdot \left(\frac{D}{d}\right)^2 \cdot \varepsilon_D \quad (3.11)$$

where, $W \approx 1.5$ is the spring Wahl correction factor. By making $D > d$, the maximum linear strain of the helical spring can be made many times larger than the SMA linear strain detwinning threshold $\varepsilon_D < \sim 1\%$. Helical SMA springs provide temperature-dependent deflections with large force as its Young's modulus, E is a function of temperature as shown in Fig. 3.8 [9].

At low temperature, the three SMA springs of default length, h_s , are stretched to length (h_o+h_a) matching the initial rim thickness, h_o plus the thickness of incompressible top washer, membrane and back plate, lumped together as h_a . After the springs are mounted on the elastic rim, they are released compressing the rim and reaching an equilibrium length $h_C < (h_o+h_a)$. The equilibrium length is determined from a force balance between the cold spring force and the rim restoring force. The governing parameters here are the spring constants of both the SMA spring and compressible rim.

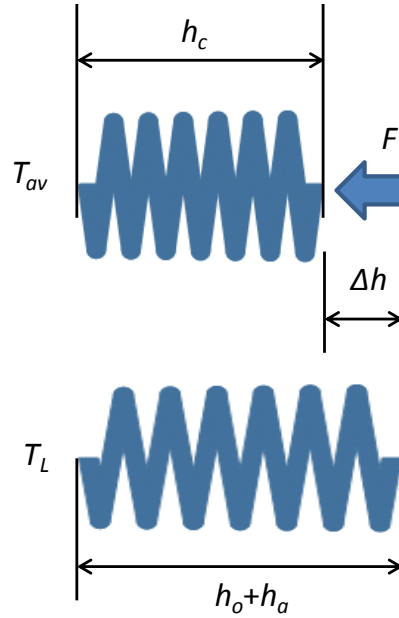


Fig. 3.8. SMA extension spring. The spring is extended to length (h_o+h_a) at low temperature T_L . When heated to average hot temperature state T_{av} , the spring produces a pulling force F with deflection Δh .

The cold temperature length of the spring is,

$$h_c = \frac{(k_r \cdot h_s + 3 \cdot k_c \cdot (h_o + h_a))}{(3 \cdot k_c + k_r)} \quad (3.12)$$

where k_c is the cold SMA spring constant. Note that, $h_o+h_a \geq h_c \geq h_s$ and the maximum displacement is less or equal to $(h_c - h_s)$. If the SMA springs are next heated, the springs stiffen compressing the rim further. The shrinkage produces a change on the rim height of,

$$\Delta h = -\frac{3 \cdot k_r \cdot (k_H(T_{av}) - k_c) \cdot (h_o + h_a - h_s)}{(3 \cdot k_H(T_{av}) + k_r) \cdot (3 \cdot k_c + k_r)} \quad (3.13)$$

where, T_{av} is the average spring temperature and $k_H(T_{av})$ is the average hot spring constant from Eq. (3.10). Eq. (3.13) can be combined with Eq. (3.4) to determine the optical power change for a given average spring temperature. Note that when heated, a real SMA spring is subject to temperature gradient, hence resulting in a smooth variation of k_H with average temperature T_{av} . Eq. (3.13) tells us that this reversible actuation mechanism requires both the initial elastic stretching of the cold SMA spring and a sufficient actuation force in the hot SMA state. If the springs are much softer than the rim, they will produce very little compression force and negligible Δh . On the other extreme, if the springs are much stiffer than the rim, the initial cold spring stretching, $(h_C - h_s)$ is small also producing negligible Δh . For maximum Δh , the spring constants for the cold SMA springs and rim must be appropriately matched. For our lens, the spring constant for the VHB rim used was ~ 8.2 kN/m. The NiTiCu SMA springs used (Kellogg's Research Labs) had wire diameter of 500 μm , mandrel radius of 900 μm , a pitch of 500 μm and 15 turns. The transition temperature for NiTiCu SMA wire was 45 $^\circ\text{C}$. The effective net spring constant, $3 \cdot k_c$ was 9.2 kN/m in the cold state. Finally the SMA springs are electrically heated; hence their average temperature is approximately proportional to the square of the applied voltage.

3.6 Lens fabrication

The adaptive lens fabricated is shown in Fig. 3.9(a). The annular sealing rim is constructed from 3M VHB 4910 acrylic elastomer which can sustain strains as high as 77% [2-3]. Five layers of VHB 4910, 1 mm-thick tape were bonded together to produce an initial undeformed rim thickness of 5 mm. The annular shape of the rim was laser cut

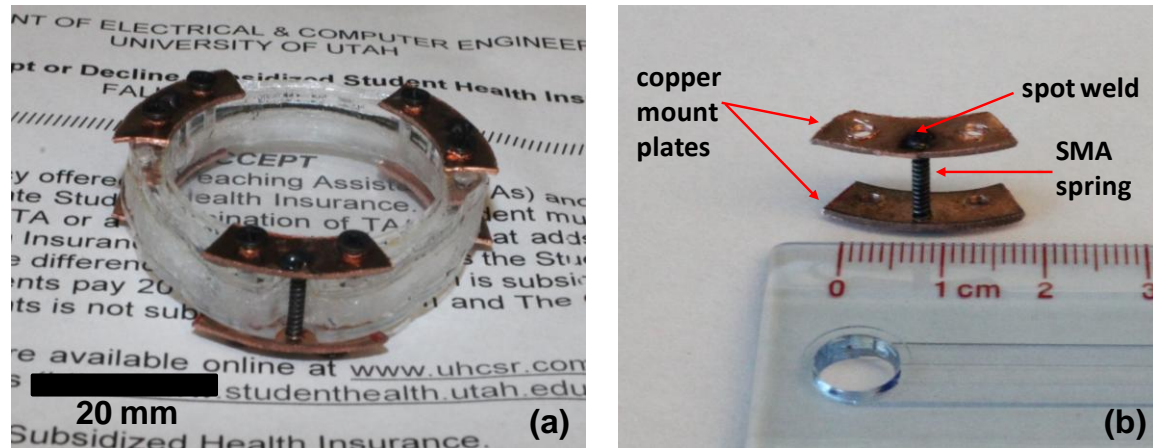


Fig. 3.9. Photograph of the complete lens and SMA actuator assembly. (a) Lens assembly kept on a reading object and (b) Single NiTiCu SMA spring actuator spot welded to flat copper tabs used to mount the springs on the rim top washer and bottom plate. The copper tabs serve as electrical contacts for the SMA springs.

from the tape stack using a VLS 3.60 Laser Platform 60W CO₂ laser with laser power 100% and speed 1%. As the rims cut from the VHB tape were very sticky, a 1 μm thick parylene-C film was deposited around the rim for both physical and chemical insulation. The rigid back plate and top washer were made with 1.5 mm thick transparent acrylic. The elastic membrane was made with polydimethylsiloxane (PDMS). SYLGARD 184 Silicone Elastomer was (10:1 ratio of base and curing agent) coated on an acrylic petri dish. The PDMS uncured mixture was cured in an oven at 60 $^{\circ}\text{C}$ for 6 hours to form 840 μm thick PDMS membranes. The PDMS membrane and acrylic back plate are adhesively attached to the annular VHB ring, and the entire assembly has better than 90% optical transmittance. Deionized water was used ($n=1.33$) as the optical liquid. The method we used was immersion followed by weak evacuation. After venting, the cavity is sealed while completely immersed in deionized water. This produces bubble-free chambers. We next examine the lens before actuation and testing using a Shack-Hartmann sensor to determine the flatness of the lens.

The SMA helical springs were laser spot welded to flat 0.6 mm thick copper tabs as shown in Fig. 3.9(b). Each SMA spring assembly was inserted at the outer rim notches and screwed to the top rim washer and the back plate as shown in Fig. 3.9(a). The weight of the completed lens and actuator assembly was 16.7 g.

3.7 Experiments

Lens power, focal length and wavefront measurements were made using a Shack-Hartmann wavefront sensor (SHS) (WFS150-7AR from ThorLabs) and a collimated LED light source (M625L3-C1 from ThorLabs) with wavelength 625 nm. We utilized the proximity technique for measurement of the focal length and a relay lens system for measuring the wavefront aberrations [13,14]. The experimental setup for lens optical profiling is discussed at length in Chapter 7.

3.7.1 Lens optical power measurements

Lens optical power measurements were made using the proximity technique at the lens center under various actuation conditions. At zero voltage, the liquid lens was in planar state and lens optical power was approximately zero. When a DC voltage, V is applied to the SMA springs, resistance heating occurs producing a temperature increase proportional to the electrical power $= 3V^2/R_s$. Here, R_s is the resistance of each SMA spring. The initial value of each spring resistance was around 0.96 Ω . Fig. 3.10(a) shows the measured lens power as a function of applied voltage V and Fig. 3.10(b) shows lens power as a function of lens height change.

The three springs were connected in parallel with the power supply. Three MOS

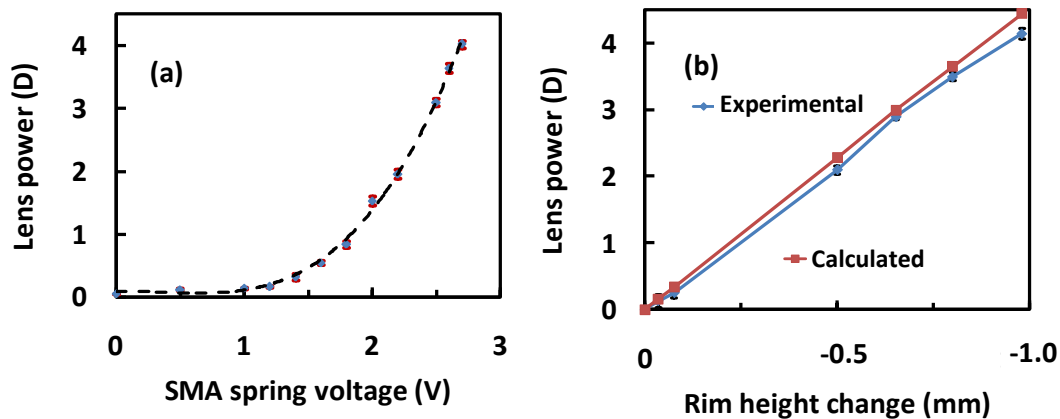


Fig. 3.10. Lens optical power plots (a) Lens power versus SMA spring voltage and (b) Lens power versus rim height. The standard deviation of the lens power was less than 3%.

switches were used to control the pulse width of the applied voltage. Each voltage is applied for 10 seconds to measure the focal length. Fig 3.10(a) shows that the lens power increases slowly with voltage in the beginning and the slope increases rapidly after 1.8 V. This is attributed to the nonlinearity of power generation versus applied voltage and the temperature dependence of the SMA resistivity which increases in the beginning but falls to a lower value with further temperature increase [15].

Fig. 3.10(b) shows the lens power as a function of height change of the annular sealing rim. The rim height was measured using a caliper. The measured power was close to the calculated value. For a 1 mm deformation (20% strain), the VHB acrylic rim behaved elastically. Hysteresis was not observed upon cyclic actuation, and all measurements were made with the lens in a stable state. Considering other factors constant, the change in lens optical power depends on the actuation voltage and current compliance. With 2.2 V, it took approximately 4 seconds to change lens power from 0 to +1 D. The response time of the SMA actuators is determined by the thermal time constant

of the springs and roughly proportional to the square of the unwound spring length and inversely proportional to the SMA thermal diffusivity. We have not made any attempts to improve the speed of the actuator as that is beyond the scope of this dissertation. The drift of the lens deflection under cyclic excitation was also measured as follows. The actuation voltage was set to 2.7 volts corresponding to a power of 4.02 diopters. The lens was actuated on and off for several hundred cycles, each cycle 10 seconds on and 60 seconds off. The deflection of the lens power drifted approximately 1.2% after 500 actuation cycles. This gives an estimation of lens life time for extreme actuation condition.

Next we captured video images recorded through the actuated liquid lens. The lens was attached to a digital single-lens reflex camera with 40 mm focal length. The target object was placed 14 cm apart from the camera-liquid lens setup. The autofocus camera adjusts its focus as the lens changes its focal length. Fig. 3.11(a) and 3.11(b) show two images recorded with SMA lens at zero power and at a power of +3 D.

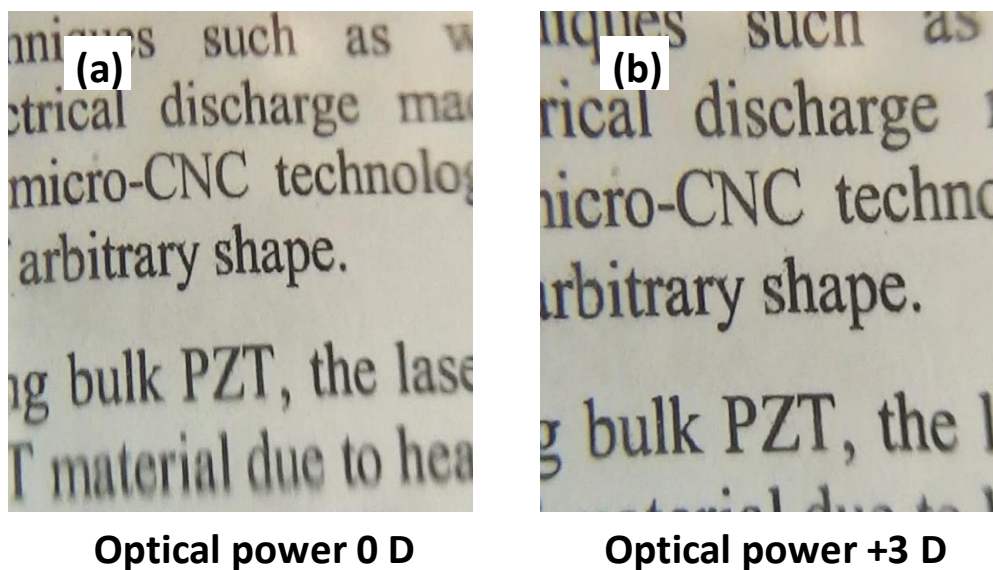


Fig. 3.11. Photos taken through our fabricated lens (a) Image of text recorded through liquid lens at its default state and (b) at a lens power +3 D.

The bottom part of the Figure 3.11(a) has some distortion. The root cause is gravity-induced vertical coma aberration. The weight of the liquid makes the membrane deformation at the bottom of the lens somewhat higher than at the top. The amount of coma can be much reduced by using a thicker PDMS membrane or a high membrane tension [16]. For example, doubling the PDMS thickness reduces the coma by 8-fold. At higher optical power, vertical coma is also reduced which is apparent in Figure 3.11(b) [17]. In addition, we took photographs of a USAF 1951 1X R1DS1P resolution target through the lens with fixed focus camera at test lens power of +3 D. The camera was at fixed focus with 55 mm lens and the test lens was attached with the camera. The target bar was kept 10 cm apart from the test lens. The chart resolution cutoff was about 28 lp/mm.

3.7.2 Lens wavefront measurements, lens quality, and discussion

To evaluate the lens optical performance, the Zernike coefficients of the lens were measured using the $4f$ relay lens system for both lens optical powers 0 and +4 D. The Zernike coefficients at lens power 0 D were: (Astig-45 = -0.108 μm , Astig-90 = -0.492 μm , Trefoil-X = 0.03 μm , Coma-Y = -0.536 μm , Coma-X = 0.236 μm , Trefoil-Y = -0.083 μm , Spherical = -0.006 μm) producing an RMS wavefront error of 0.77 μm . For +4 D optical power the Zernike coefficients were (Astig-45 = -0.738 μm , Astig-90 = -0.81 μm , Trefoil-X = 0.992 μm , Coma-Y = 0.67 μm , Coma-X = 0.166 μm , Trefoil-Y = -0.317 μm , Spherical = -0.292 μm) producing an RMS wavefront error of 1.68 μm .

The 80% encircled energy radius of the point spread function (PSF) at 0 and +4 D were 0.079° and 0.107° , respectively. The largest source of aberrations was coma at 0 D

and Trefoil at +4 D. The coma aberration is produced by gravity as the weight of the liquid produces a larger pressure at the bottom of the membrane than at the top, and the coma decreases with increased lens power. The Trefoil aberration at +4 D was caused by the three SMA actuators spaced 120° apart. Spherical aberration at 0 D was very small and had a larger value at +4 D as the lens profile became more parabolic. The lens profiles that are obtained with liquid-filled diaphragm lenses are not spherical but bell-shaped [18]. Because of the bell-shape membrane deflection, these types of lenses experience spherical aberrations, especially at higher powers.

While the measured wavefront errors are larger than desired ($< 0.5 \mu\text{m}$), both Coma and Trefoil aberrations can be mostly eliminated with the utilization of a thicker membrane and a more rigid, thicker lens rim. Coma can also be largely eliminated if the lens diameter is reduced. Since the diaphragm deflections are proportional to the fourth power of the radius, a 5-fold reduction in the radius reduces aberrations by ~ 600 fold. Such improvements have been observed in very small aperture liquid lenses [19]. Although we have used water as the lens fluid, a higher refractive index liquid can be used to produce a higher optical power for the same height change. Glycerol ($n= 1.47$) and ethylene glycol ($n=1.43$) are improved choices that do not swell the elastic PDMS lens membrane.

3.8 Limitations

The lens has several limitations which motivated us to construct better lens. The electrical power consumption for this lens is very large, around 3-4 W per actuation. Shape memory alloy takes thermal energy to change the phase of material from

martensite to austenite state. The change of phase produces deflection with large force. The energy conversion for shape memory alloy is 4-5% which makes the lens very electrical power hungry. Large electrical power consumption makes this lens inappropriate for portable eyeglass application. Another problem for SMA spring actuated lens is very small operating frequency. From Fig. 2.2, it is evident that an SMA wire can produce a large force for very small volume of actuator but the operating frequency is small. The cooling time for SMA spring wire is large (~90 seconds) which makes the operating frequency small. The lens has an optical quality issue as the trefoil aberration at higher optical power is large. It also suffers from significant coma aberration which can be reduced by making the elastic membrane thicker. We construct the next lens which consumes little electrical power, has faster response time and operating frequency, and has better optical quality.

3.9 Further work

The results discussed in section 3.5 were experimented in open loop control system using simple switching circuit. We further developed the system by making it a closed loop control. From Eq. (3.4), we can see that lens optical power is a function of annular sealing rim height. Therefore, by controlling the height of annular sealing rim, we can control the lens optical power. We used capacitive sensor (FDC 1004 from Texas Instruments) to control the height of the annular sealing rim [20]. One plate of the capacitor is placed upon the top washer and another plate is placed below the rigid back plate. The flexible VHB 4950 elastomer and rigid acrylic work as the dielectric materials between capacitor's parallel plates. Three such capacitors were placed 120° apart from

each other along the periphery of the lens. Capacitance, C , is inversely proportional to the height of the annular sealing rim, h .

$$C = \frac{\varepsilon A}{h} \quad (3.10)$$

Here, ε is the permittivity of the medium between two plates of the capacitor, A is the area of the parallel plate, and h is the height of the annular sealing rim. As the SMA springs are compressed, the height of the annular sealing rim, h decreased and the capacitance value, C increased. We interfaced FDC 1004 I²C (inter-integrated circuit) with microcontroller (Arduino Due) and the switching circuit to control the height of the annular sealing rim. We implemented the sliding mode control (SMC) system here [21]. SMC is a nonlinear control scheme which uses a discontinuous control signal to force the system to slide along the cross section of system's normal behavior. The microcontroller code is given in Appendix A1.

3.10 Summary

A tunable focus large aperture liquid lens actuated by shape memory alloy springs has been fabricated and tested. The liquid lens has a 34 mm aperture, 9 mm thickness and weighs 16.7g which is around 10 times lighter than similar aperture liquid lenses. The lens is capable of changing its optical power between 0-4 diopter with a low 3V voltage operation and the response time of the lens was approximately 4 seconds. The RMS wavefront aberrations for this lens at 625 nm wavelength light were 0.77 μm and 1.68 μm at powers of 0 D and +4 D, respectively. With utilization of thicker PDMS

membranes, the aberration of the lens can be further reduced. The speed of the lens can also be increased by careful actuator design. These lightweight lenses have many potential applications for replacement of compound zoom lenses in portable imaging applications where large lens apertures are needed with smaller footprint and light weight.

3.11 References and links

1. H. Ren, D. Fox, P. A. Anderson, B. Wu, and S.-T. Wu, "Tunable-focus liquid lens controlled using a servo motor," *Optics Express* **14**(18), 8031-8036 (2006).
2. M. Hossain, D. K. Vu, and P. Steinman, "A comprehensive characterization of the electromechanically coupled properties of VHB 4910 polymer," *Archive of Applied Mechanics* **85**, 523-537 (2015).
3. M. Wissler and E. Mazza, "Mechanical behavior of an acrylic elastomer used in dielectric elastomer actuators," *Sensors and Actuators A: Physical* **134**(2), 494–504 (2007).
4. J. E. Huber, N. A. Fleck, and M. F. Ashby, "The selection of mechanical actuators based on performance indices," *Proceedings of the Royal Society A* **453**(1965), 2185-2205 (1997).
5. M. Kohl, *Shape Memory Microactuators* (Springer, 2004).
6. E. Hornbogen, "Thermo-mechanical fatigue of shape memory alloys," *Journal of Materials Science* **39**, 385-399 (2004).
7. Y. Liu and Z. Xie, "Detwinning in shape memory alloy," *Progress In Smart Materials and Structures*, 29-65 (2007).
8. I. Chopra and J. Sirohi, *Smart Structures Theory* (Cambridge, 2013)
9. E. Khan and S. M. Srinivasan, "A New Approach to the Design of Helical Shape Memory Alloy Spring Actuators," *Smart Materials Research* 2011(167195), (2011).
10. T. C. Waram, *Actuator Design Using Shape Memory Alloys* (Tom Waram, 1993).
11. R. C. Juvinall and K. M. Marshek, *Machine Component Design* (Willey, 2012).
12. Y. Prawotto et al., "Failure analysis of automotive suspension coil springs," *Iron and Steel Tech*, Sep. 2008, pp. 35-48.
13. D. R. Neal, J. Copland, D. A. Neal, D. M. Topa, and P. Riera, "Measurement of lens focal length using multi-curvature analysis of Shack-Hartmann wavefront data," *Proc. SPIE* 5523 (2004).
14. C. Li, G. Hall, X. Zeng, D. Zhu, K. Eliceiri, and H. Jiang, "Three-dimensional surface profiling and optical characterization of liquid microlens using a Shack–Hartmann wave front sensor," *Applied Physics Letters* **98**, 171104, 2011.

15. H. Song, E. Kubica, and R. Gorbet, "Resistance modeling of SMA wire actuators," *International Workshop on Smart Materials, Structures & NDT in Aerospace Conference*, Montreal, Canada (2011).
16. M. D. Giovanni, *Flat and Corrugated Diaphragm Design Handbook* (CRC Press, 1982).
17. N. Sugiura and S. Morita, "Variable-focus liquid-filled optical lens," *Applied Optics* **32**(22), 4181-4186 (1993).
18. Q. Yang, P. Kobrin, C. Seabury, S. Narayanaswamy, and W. Christian, "Mechanical modeling of fluid-driven polymer lenses," *Applied Optics* **47**(20), 3658–3668 (2008).
19. F. Schneider¹, J. Draheim, R. Kamberger, P. Waibe, and U. Wallrabe, "Optical characterization of adaptive fluidic silicone-membrane lenses," *Optics Express* **17**(14), 11813–11821 (2009).
20. <http://www.ti.com/tool/fdc1004evm>
21. V. I. Utkin, *Sliding Modes and Their Application in Variable Structure Systems* (Mir Publishers, 1978).

CHAPTER 4

TUNABLE-FOCUS EYEGLASS LIQUID LENS ACTUATED BY PIEZOELECTRIC BIMORPHS

4.1 Variable focus eyeglasses

In this chapter, we demonstrate the realization of a liquid lens for eyeglass applications driven by piezoelectric bimorph actuators. Piezoelectric bimorph actuators are attractive because of high driving force and high electrical efficiency. The key features are its compact low weight and low profile design. Fig. 4.1 shows a schematic cross-section of our lens without the actuators. The lens consists of a rigid annular sealing rim of gap, g_r , encapsulated by two membranes forming a sealed chamber. This chamber is filled with a fixed volume of a high index optical fluid (glycerol, $n=1.47$). The top membrane has uniform thickness, t_t , and radius, r_t . The bottom membrane has a rigid flat central piston of radius, r_p , supported by a flexible annular membrane of thickness, t_b , and radius, r_b . The thickness of the bottom membrane is made very thin such that the force required to flex it is negligible compared to that required to deform the top membrane. When a normal force, F_{piston} , is applied to the bottom piston, the shape of the top membrane is changed. It bulges out or in depending on the direction of the force. This action thus produces a plano-convex or a plano-concave lens. The radius of the entire device is defined by the outer support rim, and the lens thickness is defined by the piston

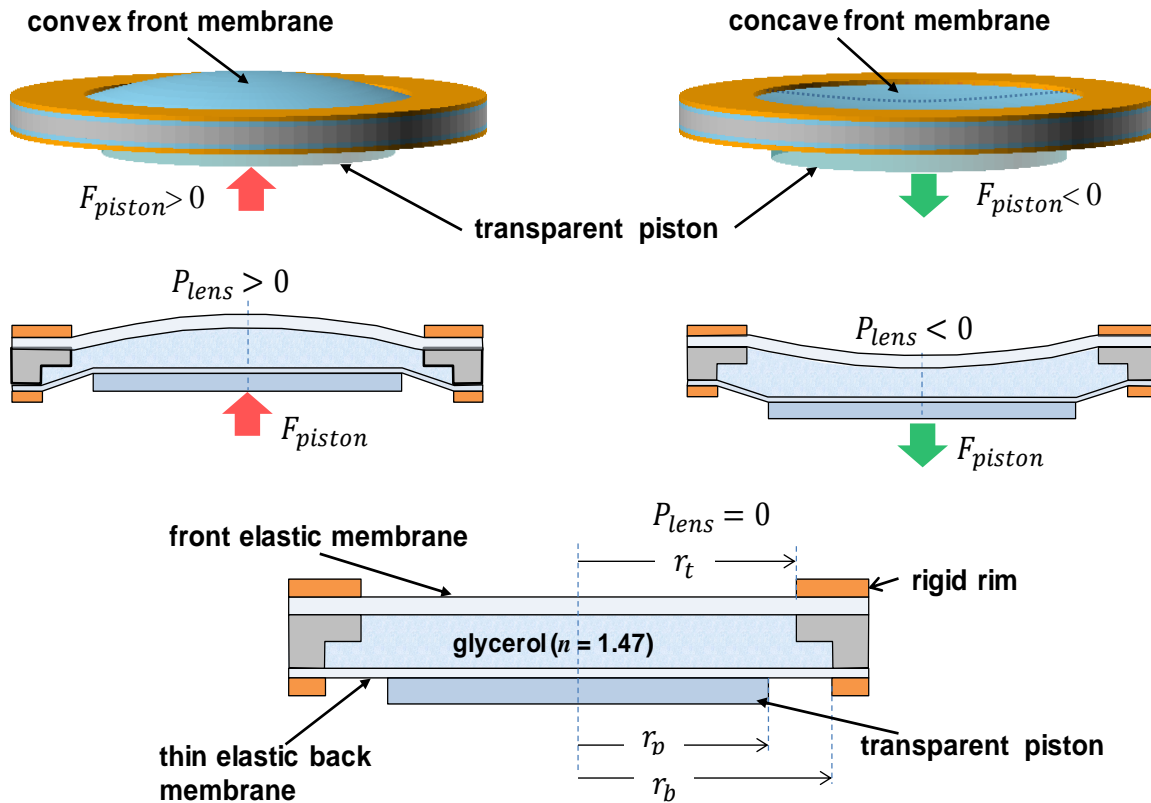


Fig. 4.1. Simplified schematic of soft membrane liquid lens excluding actuators. The lens optical power, P_{opt} , is adjusted by vertically displacing the fluid, deflecting the top membrane thus changing its curvature.

displacement required for a given optical power change. The deflection requirement depends on the shape of the top membrane.

The top membrane deflection, u_o for a circular membrane of constant thickness under radial tension, T and uniform pressure, q_o satisfies the modified biharmonic Eq. [1],

$$D \cdot \nabla^4 u_o - T \cdot \nabla^2 u_o = q_o. \quad (4.1)$$

Here, D is the flexural rigidity of the membrane. The Eqs. for D and T are,

$$D = \frac{E \cdot t_i^3}{12(1 - \mu^2)} \quad (4.2)$$

$$T = \varepsilon_i \cdot E \cdot t_i$$

where ε_i is the initial membrane stretch, E is the membrane Young's modulus, and μ is the membrane Poisson's ratio.

The solution of Eq. (4.1) for any T and D for a circular diaphragm with clamped edge boundary condition is well known [1],

$$u_o = \frac{q_o \cdot r_i^2}{4 \cdot T} \left((1 - \rho^2) + \frac{2}{\beta \cdot I_1(\beta)} (I_0(\beta\rho) - I_0(\beta)) \right) \quad (4.3)$$

where $I_0()$ and $I_1()$ are the zero and first order modified Bessel functions of the first kind,

$\beta = r_i \cdot \sqrt{\frac{T}{D}}$ is the normalized ratio of tension over rigidity, and $\rho = \frac{r}{r_i}$ is the normalized

diaphragm radius. This solution has two well-known limits for tension and rigidity dominated regimes. The maximum deflection height, h at the membrane center ($r=0$) is,

$$h = u_o(0) = \frac{q_o \cdot r_i^2}{4 \cdot T} \left(1 + \frac{2(1 - I_0(\beta))}{\beta} \right) \quad (4.4)$$

Note that, if tension is very large ($\beta \gg 1$), Eq. (4.4) converges to, $h = \frac{q_o \cdot r_i^2}{4 \cdot T}$. In order to

form a liquid lens, a spherical surface of radius of curvature, R is desired. Although the deformed membrane is not fully spherical, we approximate the deflection as a quadratic

in ρ corresponding to a spherical cap of radius R and maximum height h as shown in Fig. 4.2 satisfying the relationship,

$$(R-h)^2 + r_t^2 = R^2 \quad (4.5)$$

For typical lenses used for eyewear, $h \ll r_t$; hence $R \approx r_t^2 / 2h$. Thus the lens optical power is,

$$P_{opt}(q_o) = \frac{(n-1)}{R} \approx \frac{2h(n-1)}{r_t^2} = \frac{q_o}{2 \cdot T} (n-1) \left(1 + \frac{2(1-I_0(\beta))}{\beta I_1(\beta)} \right) \quad (4.6)$$

The lens power is thus proportional to the pressure q_o . The top membrane displaced volume is the volume of the spherical cap,

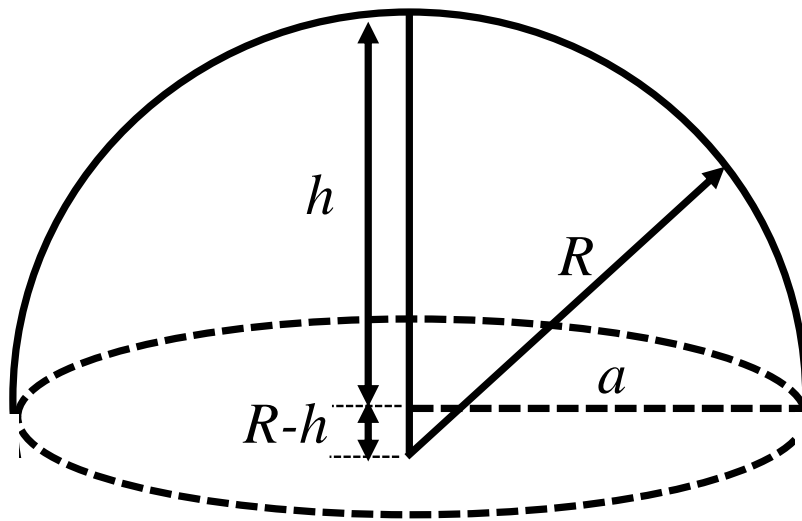


Fig. 4.2. A circular diaphragm under uniform tension, T , and pressure, q_o , forms approximately a spherical cap.

$$\Delta V_{front}(q_o) = \frac{1}{6} \pi h (3r_t^2 + h^2) \approx \frac{1}{2} \cdot \pi \cdot h \cdot r_t^2 \quad (4.7)$$

Since the chamber volume is fixed, the same liquid volume is displaced by the back membrane. If the back membrane is thin, narrow, and of negligible rigidity,

$$\Delta V_{back}(q_o) \approx \frac{1}{2} \pi d_p (r_b^2 + r_p^2) = \Delta V_{front}(q_o) \approx \frac{1}{2} \pi h r_t^2 \Rightarrow h \approx \frac{(r_b^2 + r_p^2)}{r_t^2} d_p \quad (4.8)$$

where d_p is the piston displacement. The piston force is, $F_{piston} = \pi r_b^2 q_o$. Combining Eq. (4.6) and (4.8), one obtains expressions for the piston spring constant, k_p

$$k_p = \frac{F}{d_p} \approx 4\pi T \cdot \frac{r_b^2 (r_b^2 + r_p^2)}{r_t^4} \frac{1}{\left(1 + \frac{2(1 - I_0(\beta))}{\beta I_1(\beta)}\right)} \quad (4.9)$$

Eq. (4.9) allows us to determine the piston displacement, d_p from the applied force. The optical power versus piston displacement is obtained from Eqs. (4.8) and (4.6),

$$P_{opt}(d_p) \approx 2(n-1) \frac{(r_b^2 + r_p^2)}{r_t^4} d_p \quad (4.10)$$

At the default lens position, the two membranes are flat and the minimum rim gap is selected such that the membranes are not in contact for the largest piston displacement, or

$(g_r)_{\min} \approx d_p$. This relationship defines the minimum volume and weight of liquid in the chamber as a function of the maximum lens power such that

$$V_{liquid} \geq \frac{\pi \cdot r_t^2 \cdot P_{\max}}{2(n-1)} \quad (4.11)$$

Eqs. (4.9), (4.10), and (4.11) are useful to estimate some of the liquid lens parameters. For example, for an optical power change of +3D with glycerol as the optical fluid and using top membrane radius of 18 mm, piston radius of 16 mm, and bottom membrane radius of 20 mm, the required piston displacement is 0.511 mm which is also the minimum gap. The minimum glycerin volume is thus $\approx 1.3 \text{ cm}^3$. For glycerin with density, $\rho_o = 1.26 \text{ g/cc}$, this corresponds to a minimum liquid weight of 1.64 gr. In practice, the lens weight will also be affected by the thickness and weight of the frame. The force required to move the piston depends on the initial tension parameter, T . The top and bottom membranes are made of polydimethylsiloxane (PDMS) with thicknesses of 1.3 mm and 0.2 mm, respectively. The Young modulus and tension of these membranes can vary significantly depending on the PDMS mixture formulation and curing cycle [2]. We measured these parameters using the deflection method described in Yang et al. [3]. The value of Young's modulus, Poisson's ratio, and prestrain were 987.6 kPa, 0.49, and 2.83%, respectively. This prestrain yields a pre-tension of 36 N/m. The calculated piston force required for the optical power change of +3 D was 0.8N or 80 gm consistent with these parameters.

4.2 Piston actuation mechanism

The central part of the lens (the bossed piston) is transparent and unobstructed; therefore, the piston actuators are placed along the lens periphery as shown in Fig. 4.3. The piston moves up and down driven by three low-profile curved piezoelectric bimorph actuators. The points of contact of the three bimorphs with the piston thus define the backside plane of the liquid lens.

The vertical and angular deflections of curved bimorphs are [4,5],

$$U(R, s) = \frac{M_b \cdot R^2}{E_b \cdot I_b} \cdot \left(1 - \cos\left(\frac{s}{R}\right)\right) \quad (4.12)$$

$$\varphi(s) = \frac{U(s)}{R}$$

where $U(s)$ is the vertical deflection at the mid radius R as a function of the length s along the bimorph midradius, and $\varphi(s)$ is the bimorph tilt angle as shown in Fig. 4.4.

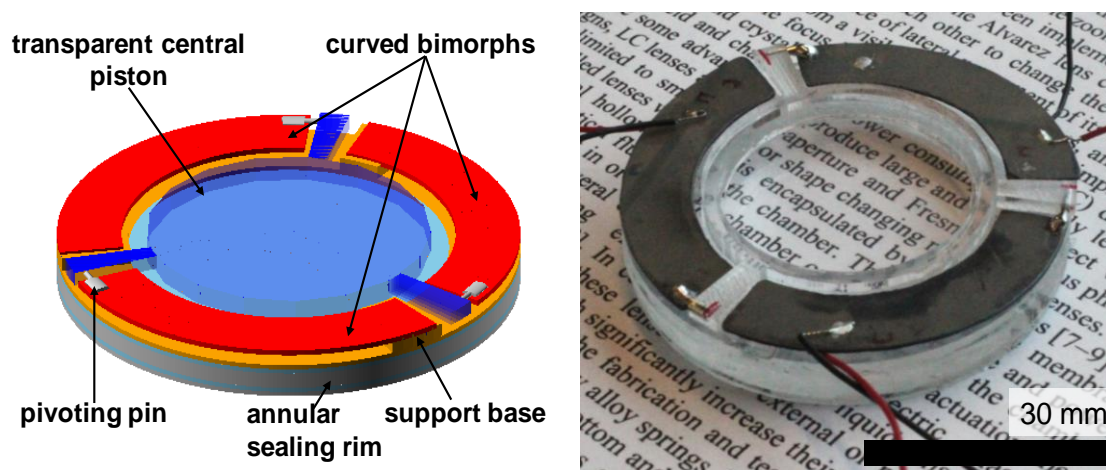


Fig. 4.3. Schematic of the bimorphs actuating the bossed membrane lens (left) and photo of the actual device (right).

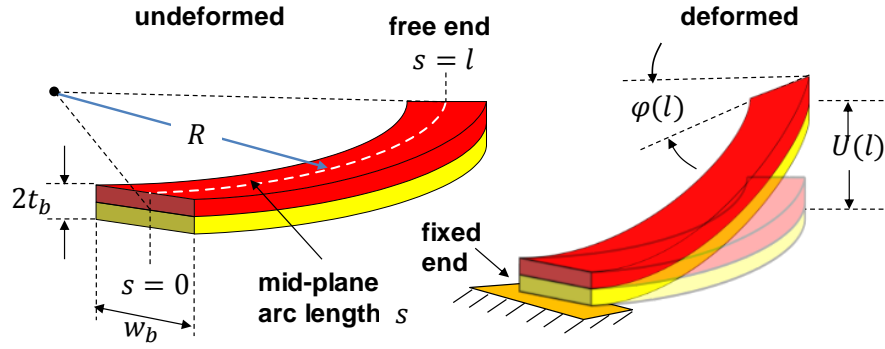


Fig. 4.4. Deflection of a curved bimorph. Since the outer edge is longer than the inner one, this type of actuator not only bends but also rotates at its tip.

The parameters E_b and I_b are the bimorph's Young's modulus and moment of inertia, respectively and M_b is the bimorph's piezoelectric moment,

$$M_b = w_b \cdot E_b \cdot d_{31} \cdot t_b \cdot V_b, \quad (4.13)$$

where w_b is the bimorph's beam width, t_b is the thickness of each bimorph layer, d_{31} is the bimorph's piezoelectric coefficient and V_b is the applied voltage across the bimorph actuator.

For efficient deflection, the end support of the bimorph must pivot about the highest elevation point, the end of the bimorph outer radius. For example, a 22.5 mm radius actuator 5 mm wide and an angle of 115° , a 1 mm midradius deflection yields a rotation angle of 2.5° and a difference in deflection of 20% between the inner and outer radius. In order to get maximum deflection from the actuators without compromising the actuator force, we used a pinning-hole end configuration. Rigid pins are thus attached near the highest elevation point of the actuators (the outer farthest corner). The pins pivot inside cylindrical holes drilled inside three extension tabs connected to the central piston.

The curved actuators were implemented using thin PZT 5H4E bimorphs with dimensions and characteristics shown in Table 4.1 Each bimorph actuator weighs less than 1 gm.

Combining Eqs. (4.10), (4.12), and (4.13), we can obtain an expression for the lens optical power as a function of actuator parameters, liquid refractive index, lens dimensions, and operating voltage. If we drive each lens with three bimorphs, one obtains the optical power,

$$P_{opt} \approx \frac{2(n-1) \cdot U(R, s) \cdot (r_b^2 + r_p^2)}{r_t^4 \cdot \left(1 + \frac{k_p}{3k_b}\right)} \quad (4.14)$$

where k_p is the piston spring constant of Eq. (4.9) and k_b is the vertical spring constant of one bimorph at the pinhole support. The bimorph spring constant was measured as $k_b \sim 390$ N/m. The optical power in Eq. (4.14) increases linearly with the actuator voltage V_b . As the membrane and piston are made stiffer, the spring constant of the bimorph actuator k_b itself becomes important, and the lens membrane deflection and observed optical power are gradually reduced.

Table 4.1 Curved Bimorph Characteristics

Bimorph Material	PZT 5H4E
Layer Thickness, t_b	270 μm
Width, w	8.2 mm
Young's Modulus, E	5×10^{10} N/m ²
Piezoelectric Strain coefficient, d_{31}	-320×10^{-12} m/V
Radius of Curvature, R	21 mm
Angle of Cosine, (s/R)	110 -113
Voltage Range	0-250 V

4.3 High-voltage driver circuits

The bimorphs require multiple high voltage control signals. For evaluation purposes, a fixed 250V DC voltage source was used to power the driver circuits. The high voltage DC can also be generated using miniature 3V DC-to-DC converters (EMCO-A series) suitable for battery driven setups. The fixed DC voltage was converted to variable voltages using pulse width modulators (PWMs). Each PWM modulator was implemented using a high voltage half-bridge driver circuit (ST Micro L6384E), two high voltage NMOS transistors (ST Micro IRF820) and a high-voltage 100 nF capacitor. The pulsed half-bridge drivers were driven by a microcontroller through opto-isolators. The circuit diagram is shown in Fig. 4.5. The lens was driven in a bipolar two-terminal series configuration using two unipolar PWM high-voltage circuits in differential drive configuration. A software open loop control system was implemented to control the bimorph deflection and the lens optical power. The bimorph bending magnitude and direction were changed by adjusting the duty cycle of the PWM signal and driving from only one of the opposing PWM drivers at any given time.

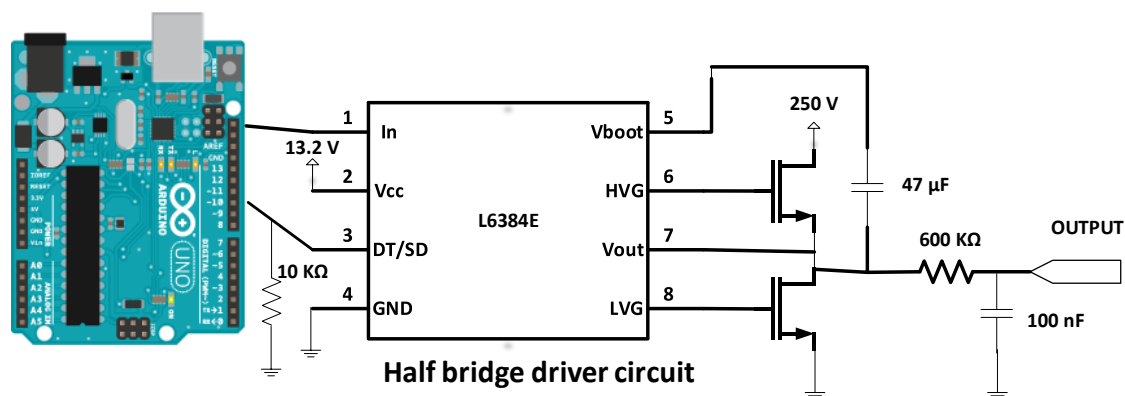


Fig. 4.5. Schematic diagram of half bridge driver circuit. Two half-bridge driver circuits in differential configuration make a full bridge driver circuit.

Fig. 4.6 shows oscilloscope waveform from the half bridge driver circuit using the software open loop control system. To show the output voltage increase with the increase of pulse width, we continuously increased the duty cycle from 0.1 to 0.9. Fig. 4.6(a) shows the increasing pulse width without the lens actuator and 100 nF capacitor connected at the output. Fig. 4.6(b) shows the ramping up of output voltage as the lens bimorph actuators and 100nF capacitor are connected at the output. The capacitance value of three piezoelectric bimorph actuators in parallel configuration is 64 nF. A 1 M Ω resistance is also connected in parallel with the lens actuators for discharging purpose while doing this experiment.

In addition to series bimorphs, other more efficient three-terminal configurations are also possible [6]. Y-poled (polling direction same) three-wire bimorphs were also tested for actuation in a bipolar configuration [6]. The bipolar configuration provides 30% more deflection with higher actuation force but requires higher voltages. Lenses were made and tested with both configurations, but the series bimorph configuration was our preferred implementation due to its simplicity. Fig. 4.7 shows the generalized schematic diagrams of bimorph driving circuit for both series and bipolar configurations.

4.4 Fabrication

4.4.1 Lens frame and pistoned liquid chamber

The lens rim was constructed by cutting acrylic sheet with laser (VLS 3.60, Universal Laser Systems) with 100% power and 10% speed for through cut and with 20% speed for making notches. The lens rim height was 2.6 mm and the lens rim had a notch of 1 mm in one side. The inner and outer radius of the rim was 18 mm and 26 mm,

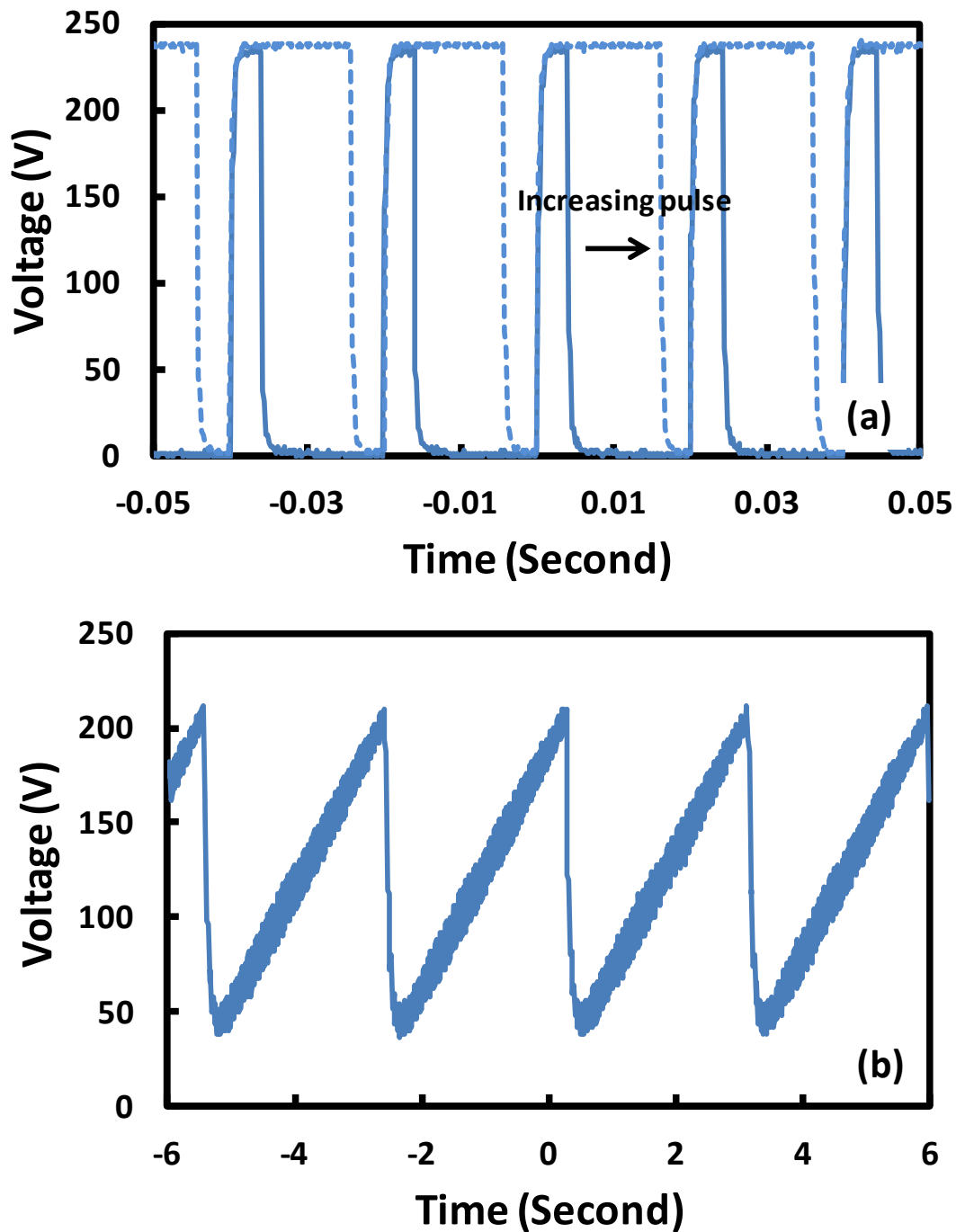


Fig. 4.6. Oscilloscope output of half bridge driver circuit. (a) Increasing pulse width probed after $600\text{ k}\Omega$ resistance without the lens and 100 nF capacitor connected to output and (b) ramping up of voltage from 0 to 220 V across the lens and 100 nF capacitor.

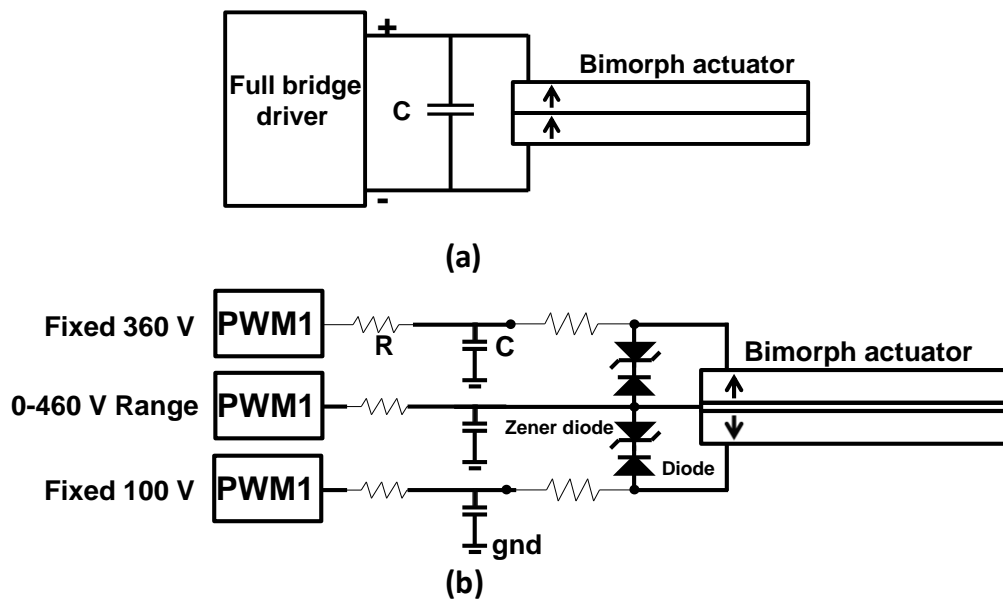


Fig. 4.7. Bimorph actuator driving circuit: (a) Series configuration of bimorph actuator and (b) Bipolar driving configuration.

respectively. The radius in the notched part of the rim, r_b was 20 mm. The front and back elastic membranes were made of polydimethylsiloxane (PDMS). The thicknesses of front and back membranes were 1.3 mm and 0.2 mm, respectively. A PDMS silicone elastomer (Sylgard 184, Dow Corning) was used in 7.5:1 ratio of base and curing agent to fabricate the PDMS membranes at 60° C for 6 hours. After fabricating the membranes, the 0.2 mm thick back membrane was attached to the notched side of the rim and front membrane to the other side. Both membranes were attached to the acrylic rim using a thin layer of silicone sealing adhesive from Dow Corning (734 Flowable Sealant). A thin solid transparent piston was next attached to the back membrane with optically clear urethane rubber (Clear Flex). The urethane rubber mixture was spun for 550 rpm for 1 minute to get a uniform thin layer (<0.1 mm) and the piston was kept over the thin back membrane for 16 hours to let the adhesive set. The transparent piston was 4 mm thick and it had

three extending arms with three pinning holes as shown in Fig. 4.3.

The second type of lens, which uses a rimmed piston instead of solid piston, has a different type of back PDMS membrane. The central part inside the rim of the piston has thickness 1.4 mm and the rest of the membrane is 0.2 mm thick. To fabricate this back membrane, a mold was made using acrylic sheet. The thick central portion of the back membrane ensures much less optical aberration. The rimmed piston is attached to the 0.2 mm thickness portion of the membrane which ensures the same elasticity of the membrane and actuator force. The schematic diagram of the rimmed lens is shown in Fig. 4.8. The rimmed lens is lighter than the solid piston lens.

4.4.2 Lens chamber liquid filling

Two holes were drilled on the annular sealing rim for insertion of the optical lens fluid and venting of air. Glycerol is used as the liquid because it has both high refractive index ($n=1.47$) and does not swell the PDMS membrane, but other higher index optical fluids (SantoLight 5267, $n=1.67$) are available as well [7,8]. In spite of having lower

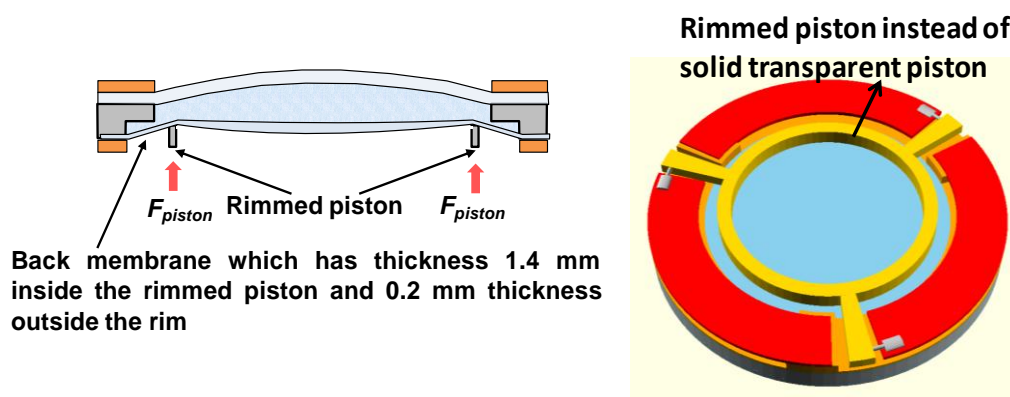


Fig. 4.8. Cross-sectional view of the rimmed piston lens excluding the actuators (left) and 3-Dimensional view of the rimmed piston piezoelectric lens (right).

refractive index than glycerol, water has another problem with PDMS membrane interface. PDMS membrane is porous. Therefore, water evaporates through PDMS membrane over the time. To cease the evaporation through PDMS membrane, we deposit Parylene C over the membrane for various thicknesses. It solves the water evaporation problem, but even depositing very thin Parylene makes the membrane cloudy (shown in Fig. 4.9). For a liquid lens filled with glycerol with density of 1.26 g cm^{-3} , a lens with vertical height of 36 mm can produce a maximum hydrostatic pressure difference of $P_{hyd} = g \cdot \rho_g \cdot h = 444.5 \text{ Pa}$ between the inside and outside of the lens. Therefore, if glycerol is inserted into the chamber at atmospheric conditions, the front membrane bulges significantly outward, which makes the initial lens optical power high. The hydrostatic pressure drop deflection is significantly reduced if the lens reservoir is pressure equilibrated and hermetically sealed. Pressure equilibration is achieved when the lens cavity is filled in by bath immersion, in this case in a mixture of 3:2 glycerol and water,



Fig. 4.9. Transparency of PDMS membrane. Membrane with optical clarity without depositing Parylene (left) and PDMS membrane becomes opaque after depositing 200 nm Parylene (right).

such that at any given point pressure inside and outside the lens are almost equal thus producing little deformation of the membranes during the fill operation. The two holes are hermetically sealed while the lens is submerged. The lens is next pulled out of the glycerin bath, rinsed and dried. The hermetic seal produces a vacuum head pressure that counteracts fluid motion driven by gravity producing a much smaller lens deformation. After sealing of the lens chamber, a 0.5 mm thick acrylic washer was attached to the front side of the lens. A second 1 mm thick washer was attached to the back side with raised supports for the bimorphs.

4.4.3 Curved bimorph actuators

The pinned piston-actuator design configuration overcomes the twisting problem of the curved actuators and provides maximum vertical deflection without compromising force. Curved actuators not only deflect up and down vertically, but also rotate at their free ends. If we use the whole width of the curved actuator for providing deflection and force, the rotation creates a coupling of force which ultimately gives negligible force with little vertical deflection. Therefore, we used the outermost tip of the free end of the curved actuator to provide deflection and force.

We constructed curved bimorph actuators from thin sheets of pre-poled PZT-4H (T223-H4CL-503X for two terminal actuators and T220-H4-503Y for three terminal actuators from Piezo System Inc.). The actuator sheet was cut into the curved shape using a diamond rotary saw followed by grinding and soldering of the end pin joint. Strong nickel plated steel pins were soldered to the outer most point of the free ends of the actuators. One end of each curved actuator was glued to the actuator supports on the rim.

The other pinned end of the actuator is a free moving end which was inserted in the piston tab receiving holes.

4.5 Results and discussion

4.5.1 Lens optical test setup

A Shack-Hartmann wavefront sensor (SHS) (WFS150-7AR from ThorLabs) and a collimated LED light source (M625L3-C1 from ThorLabs) with wavelength 625 nm were used for measuring lens optical power and wavefront aberration. The setup is described in Chapter 7. All optical measurements were recorded with the lens standing in vertical position which is the worst-case scenario for coma aberration. In order to measure the lens focal length as a function of applied voltage, we utilized the proximity technique [9].

For measuring the lens wavefront aberration, we profiled the central 25 mm diameter of the possible 32 mm aperture of the lens as discussed in Yang et al. [3]. As the diameter of the SHS sensor is small (~4.6 mm), the SHS sensor cannot profile the above-mentioned aperture of our test lens [10]. For this, a $4f$ afocal relay lens system was constructed that feeds all lens light into the sensor as discussed in Chapter 7 [11].

4.5.2 Optical power measurement

The lens operation is straightforward. If the voltage is applied and increased in the positive direction, the three bimorph actuators along with the piston move in an inward direction, which makes the front membrane convex. For negative actuator voltage, the piston moves outward making the lens concave. The proximity technique was used to

measure the lens optical power at the center of the lens under various actuation voltages.

Fig. 4.10 shows the lens optical power as a function of actuator voltage.

The lens has an offset power of +0.78 D when unpowered. The lens optical power ranged between -2.03 D to +3.57 D for a voltage range of -250 V to +250 V, which was below the depolarization voltage for our bimorph actuators. The lens optical power is linearly proportional with the actuator voltage as expected. To demonstrate the quality of the lens image, the lens focal length was tuned continuously using the driver circuit and photos were taken at different lens power. The test lens was attached to a single lens reflex camera with 40 mm focal length. The target object was placed 35 cm apart from the test lens. Fig. 4.11 shows two photos taken through our fabricated lens at different lens power.

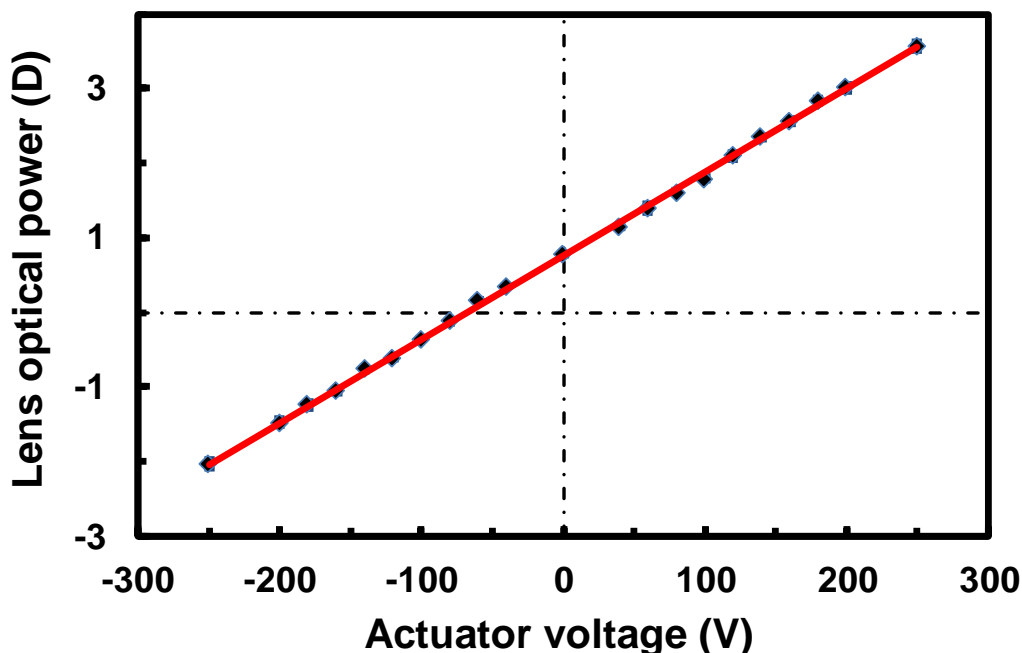


Fig. 4.10. Lens optical power (at the lens center) as a function of voltage. The standard deviations of lens power are below 1.3%.

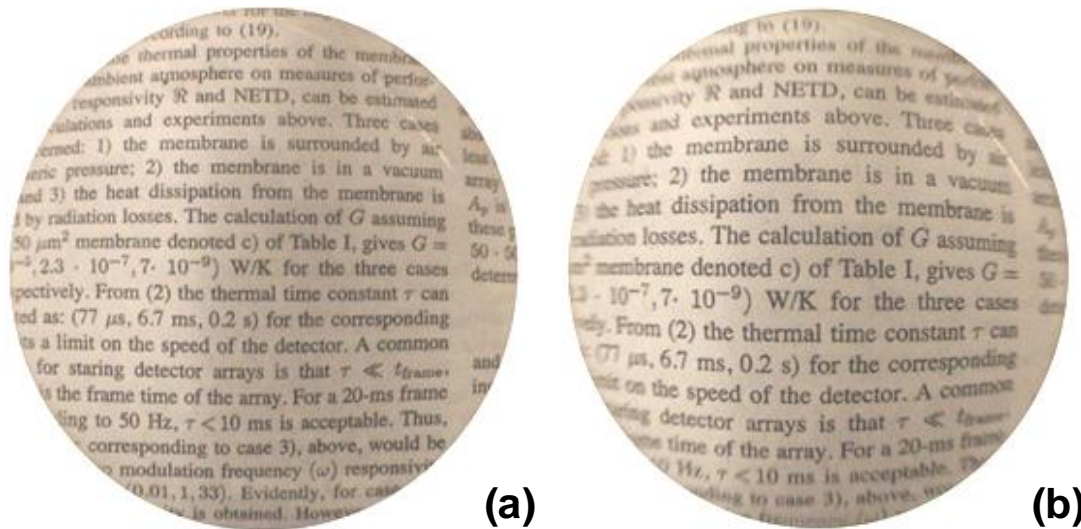


Fig. 4.11. Target object photos taken through the VFL lens at (a) -1.2 Diopter (b) +3 Diopter.

The lens electrical power consumption and its mechanical resonance were also measured. The electrical power dissipation for the lens was very small, in the range of 10-20 mW. This is good low power performance as these lenses can be operated from lightweight rechargeable portable batteries. With an 8 gm, 110 mAhLiPo battery and aDC-DC high voltage converter (EMCO A series), the lens can continuously operate for about 6 hours with control circuit. One of the important advantages for piezoelectric bimorph actuators is the zero-static power consumption; hence battery lifetime can be significantly extended if the focal change frequency is reduced. The power consumption and resonant frequency as a function of frequency are shown in Fig. 4.12.

The mechanical resonance of the structure determines the speed of response for the lens. The frequency response of the lens was measured by observing the deflection of the lens piston (via a bouncing laser beam) projected onto a screen as a function of bimorph driving frequency. The lens displayed a resonant frequency of about 70 Hz

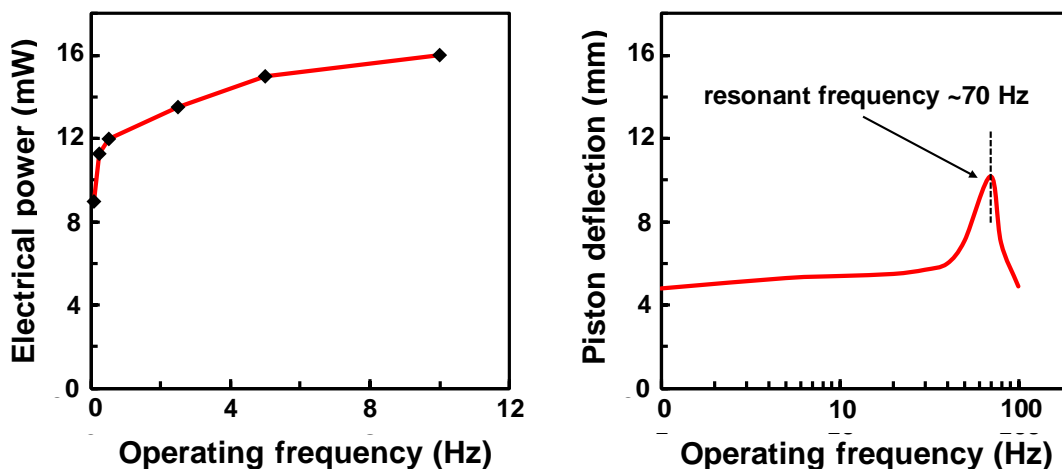


Fig. 4.12. Electrical characterization of the lens (left) Lens electrical power consumption (at 160 V) as a function of switching frequency and (right) lens actuators' mechanical displacement as a function of frequency.

which makes the effective response time about 15 milliseconds. The lens was operated continuously for more than 500 cycles with the driving circuit in the voltage range of -220 V to +220 V without observing any failure or significant performance degradation as well. The code for continuously operating the lens for ramping up and down the voltage is given in Appendix A2. The lens was also actuated intermittently for more than 6 months without any failure.

4.6 Wavefront profiling and aberration measurements

To evaluate lens optical performance and image quality, the lens aberrations were measured with the lens in the upright position by the SHS using $4f$ optical test setup discussed in Chapter 6. The wavelength of the collimated test light source was $0.625 \mu\text{m}$. The aberration values at no actuation (+0.78 D), lens convex, and concave states are reported in Table 4.2.

Table 4.2 Lens Aberrations at Different Optical Power

Aberration	Optical power +0.78 D	Optical power +3 D	Optical power -1 D
Astigmatism 45°	-0.364 μm	-0.354 μm	0.376 μm
Astigmatism 90°	-0.380 μm	0.343 μm	0.400 μm
Trefoil X	-0.008 μm	-0.035 μm	-0.056 μm
Trefoil Y	-0.007 μm	-0.385 μm	0.070 μm
Coma X	0.298 μm	-0.208 μm	-0.153 μm
Coma Y	-0.7400 μm	-0.264 μm	-0.608 μm
Spherical	-0.068 μm	-0.188 μm	-0.119 μm
RMS Aberration	0.958 μm	0.733 μm	0.846 μm

The main contributor for RMS wavefront aberration at +0.78 D (no actuation) was coma. Coma aberration is proportional to the third power of the radius of the lens and inversely proportional to the membrane tension which will be discussed in detail in the next chapter [1]. Coma aberration was worst at no actuation and it decreased significantly at higher optical power. Spherical aberration was very small at no actuation, and it increased a little at high optical power. The low values of spherical aberration are indicative of the tension dominated membrane deflection [3]. The value of trefoil aberration also increased from negligible value to 0.385 μm as the lens optical power increased because the actuators applied forces at three different points 120° apart along the periphery. All these aberration values except coma were relatively small and below 0.5 μm which is the approximate RMS value of human lens aberration [12]. The 80% encircled energy radii of the point spread function (PSF) were measured at three different lens optical powers as well. The values were 0.1°, 0.065°, 0.065° at lens optical power +0.78 D, +3 D and -1D, respectively.

4.7 Autofocusing eyeglasses

We further developed autofocusing eyeglasses using the piezoelectric lenses discussed in this chapter. The first generation autofocusing eyeglasses have been made and more efficient next generation eyeglasses are in progress. The autofocusing eyeglasses consist of two tunable-focus eyepieces, a distance sensor, a microcontroller supervisory processing unit, a microprocessor controlled actuator voltage driving circuit, a wireless bluetooth low energy (BLE) module and a set of LiPo rechargeable batteries. Fig. 4.13(a) shows a photograph of the smart eyeglasses we have developed. Fig. 4.13(b) shows a schematic of the control and communication electronics. The set is controlled by a smartphone application used to set the observer prescription, the type of refractive error and other operating settings. This application sends the prescription data to the master processor through the BLE module (HM-10, JNHuaMao). The processing unit consists of two microcontrollers (Arduino Pro-mini), one acting as a supervisory processor and the other as a slave controlling the actuator voltage generation. A time of flight (ToF) distance sensor is embedded above the nose support of the eyeglasses frame to measure

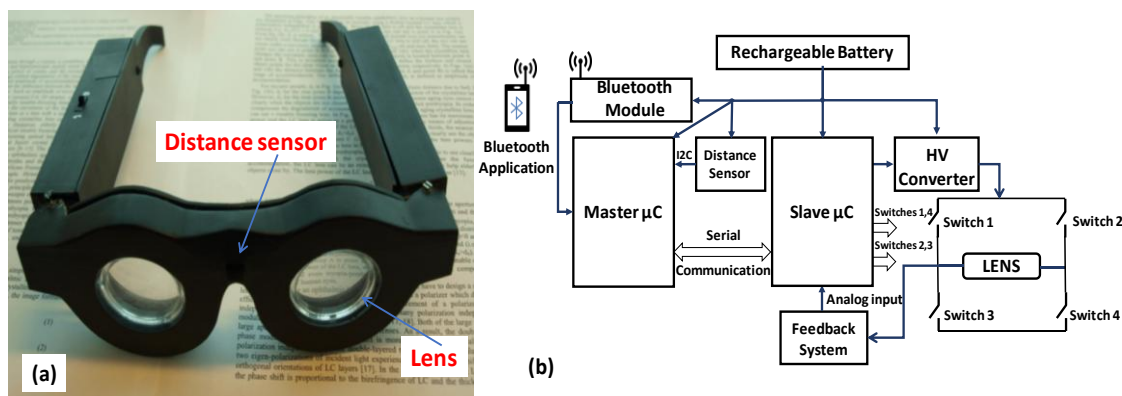


Fig. 4.13. First-generation Smart Eyeglasses (a) Photograph of our developed eyeglasses and (b) block diagram of the eyeglass control system.

the forward distance, d to the object in front of the observer. The supervisory board continuously calculates the two-component (one for each eyepiece) optical power vector \vec{P}_{lens} required to produce sharp images of the object ahead by combining the prescription setting vector \vec{S} and object distance d . In the simplest mode of operation the supervisor microcontroller calculates the following farsighted (reading) or nearsighted (far distance) adaptive corrections, respectively.

$$\vec{P}_{lens}(d) = \begin{bmatrix} P_{left} \\ P_{right} \end{bmatrix} = \vec{S}_{far} + \left(\frac{1}{d} - \frac{1}{d_{read}} \right) \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (4.15)$$

$$\vec{P}_{lens}(d) = \vec{S}_{near} + \frac{1}{d} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (4.16)$$

The supervisory processor maps the required optical powers into actuator voltages needed for the eyepieces sending these voltage values to the slave board. The slave microcontroller board drives a high voltage, high-efficiency circuit that produces those voltages utilizing an ultraminiature high-voltage DC-to-DC converter in a feedback loop configuration. This increases the electrical power efficiency as the high voltage converter is not turned on all the time. The output of the high voltage converter is connected to the tunable lens actuators in a H-bridge configuration using four high-voltage semiconductor switches that control the polarity of the voltage delivered to the actuators. The overall system is extremely flexible and capable of producing software-controlled complex corrections beyond those of Eq. (4.15) and (4.16).

4.8 Testing of autofocus eyeglasses

The setup shown in Fig. 4.14(a) consisting of an 18MP image sensor (MU1803, AMScope) with a fixed power lens followed by our tunable lens was constructed to mimic the response of a fully presbyopic farsighted eye (focused at infinity) with zero accommodation. The setup demonstrates how our adaptive eyeglasses eyepiece changes optical power as a function of distance restoring accommodation function. To check the feasibility of our lens and control system, we first placed one object in front of our eyepiece and varied the object distance from 28 cm to 1.5 m. The distance sensor measured the distance to the object and varied the eyeglasses optical power according to Eq. (4.15). Fig. 4.14(b) and (c) show two photos of far and near objects taken at +0 D and +3.4 D. When the object is at infinity the lens optical power becomes +0 D, and when the object is 28 cm away from the eyepiece, the lens optical power becomes +3.4 D. From the photos it is clear that the eyeglasses eyepiece changes its focal length according to the

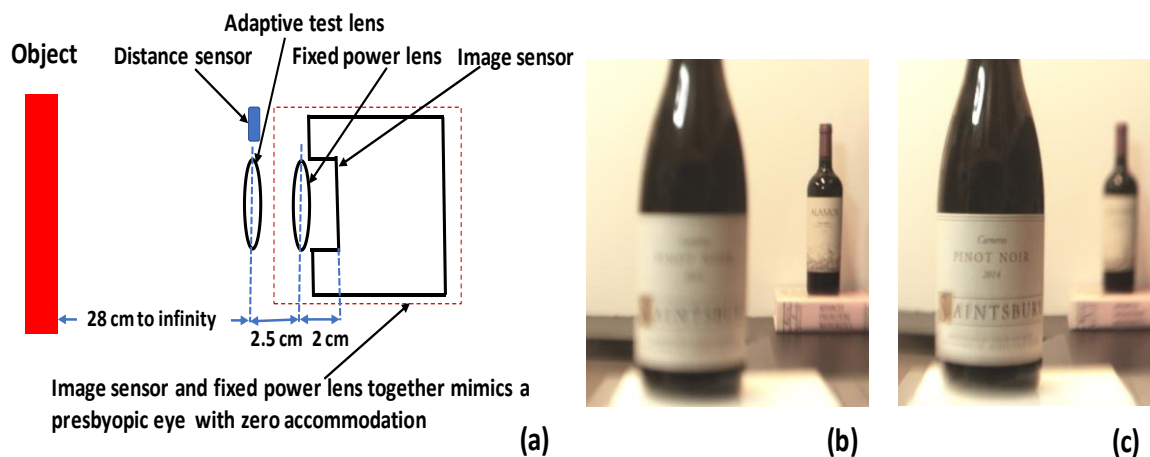


Fig. 4.14. Smart Eyeglasses experimentation. (a) Simplified diagram of the test setup emulating the imaging of objects in a presbyopic eye with accommodation provided by the variable power eyepiece. (b) Images recorded using our test eyepiece at optical power +0 D, and (c) at +3.4 D.

object distance that the observer wants in focus. We have also measured the MTF of the setup using a standard slanted knife edge target and ImageJ software from NIH [13]. For a modulation factor of 0.5, the resolution was 25 lp/mm.

4.9 Conclusion

A tunable-focus liquid lens actuated by low-profile piezoelectric bimorph actuators has been demonstrated. The lens has aperture diameter 32 mm, footprint diameter 52 mm, optical power range 5.6 D, electrical power consumption less than 20 mW, and resonant frequency 70 Hz. The lens weighs 14.4 gm. The lens RMS wavefront aberration is in the range of 0.73 to 0.95 μm . All these criteria make this lens suitable for adaptive eyeglass application.

4.10 References and links

1. E. H. Mansfield, *The Bending and Stretching of Plates* (Cambridge University, 1989).
2. D. Armani, C. Liu, and N. Aluru, "Re-configurable fluid circuits by PDMS elastomer micromachining," in *Proceedings of Twelfth IEEE conference on MEMS* (IEEE, 1999).
3. Q. Yang, P. Kobrin, C. Seabury, S. Narayanaswamy, and W. Christian, "Mechanical modeling of fluid-driven polymer lenses," *Applied Optics* **47**(20), 3658-3668 (2008).
4. M. S. Weinberg, "Working equations for piezoelectric actuators and sensors," *Journal of Microelectromechanical Systems* **8**(4), 529- 533 (1999).
5. S. Pal and H. Xie, "Analysis and simulation of curved bimorph microactuators," *Proceedings of NSTI-Nanotech* **2**, 685-688 (2010).
6. S. A. Rios and A. J. Fleming, "A new electrical configuration for improving the range of piezoelectric bimorph benders," *Sensors and Actuators A* **224**, 106-110 (2015).
7. Y. Wang, J. Balowski, C. Phillips, R. Phillips, CE. Sims, and NL. Allbritton, "Benchtop micromolding of polystyrene by soft lithography," *Lab Chip* **11**(18), 3089-3097 (2011).
8. L. Wang, H. Oku, and M. Ishikawa, "Variable-focus lens with 30 mm optical aperture based on liquid-membrane-liquid structure," *Applied Physics Letters* **102**, 131111 (2013).
9. D. R. Neal, R. J. Copland. D. A. Neal, D. M. Tropa, and P. Riera, "Measurement of lens focal length using multicurvature analysis of Shack-Hartmann wavefront data," *Proc. SPIE* **5523**, 243-255 (2004).
10. <https://www.thorlabs.com/thorproduct.cfm?partnumber=WFS150-7AR>
11. C. Li, G. Hall, X. Zeng, D. Zhu, K. Eliceiri, and H. Jiang, "Three-dimensional surface profiling and optical characterization of liquid microlens using a Shack-Hartmann wave front sensor," *Applied Physics Letters* **98**(17), 171104 (2011).
12. J. Porter, A. Guirao, I. G. Cox, and D. R. Williams, "Monochromatic aberrations of the human eye in a large population," *Journal of the Optical Society of America A* **18**(8), 1793-1803 (2001).
13. <https://imagej.nih.gov/ij/>

CHAPTER 5

TUNABLE-FOCUS EYEGLASS LIQUID LENS ACTUATED VIA MEMBRANE TENSION-ADJUSTING SMA CONTRACTION RING

5.1 Lens concept

In Chapter 4, we noticed the effect of pre-tension over flexible front membrane regarding lens optical quality and its optical power. The pre-tension value of the flexible membrane, T depends on initial membrane stretch ε_i , membrane's Young's modulus E , and membrane's thickness t .

$$T = \varepsilon_i \cdot E \cdot t \quad (5.1)$$

Depending on the membranes' thicknesses, the pre-tension values vary between 30-70 Nm^{-1} for our fabricated membranes. These high values of pre-tension give us an idea for actuating the lens membrane to tune the lens focus by modifying tension of the front membrane. The relationship between lens optical power, P_{opt} and pre-tension, T we found from Chapter 4 is,

$$P_{opt}(q_o) = \frac{q_o}{2 \cdot T} (n-1) \left(1 + \frac{2(1-I_0(\beta))}{\beta I_1(\beta)} \right) \quad (5.2)$$

where $I_0()$ and $I_1()$ are the zero and first order modified Bessel functions of the first kind, $\beta = r_i \cdot \sqrt{\frac{T}{D}}$ is the normalized ratio of tension over rigidity, $\rho = \frac{r}{r_i}$ is the normalized diaphragm radius, and q is the applied pressure over the membrane. By varying the tension value, T and keeping all other parameters constant, we can get tunable lens optical power as a function of tension. This gives us the idea for our third type of lenses. Adjustment of the tension can be achieved by a compression or expansion of actuator ring or coil placed around the periphery of the membrane and near the membrane anchor region. As the ring type actuator can be as thin as the membrane thickness, one could realize an extremely low profile and weight actuating mechanism in this fashion.

We demonstrate lensing mechanism that uses a fixed pressure as the driving force and a membrane of electrically controlled tension and stiffness that determines the deflection magnitude. The uniform pressure is provided by a pressurized sealed liquid chamber which has flexible membranes on both front and back sides. The tension of the front membrane is electrically changed through the addition of a shape memory alloy (SMA) coil that contracts when electrically heated. This causes variable deflection of the front membrane as a function of SMA coil voltage which results in variable lens power. Low actuation voltage, thinner, and lighter than the previous two lenses discussed in Chapter 3 and 4 are the main advantages for this lens. However, high electrical power consumption and slower response time are the main problems for these lenses. The natural cooling time for shape memory alloy (SMA) wire is large which makes this lens response time slow.

5.2 Lens structure

Fig. 5.1 shows an example of conventional thin membrane that is being deflected under uniform pressure, P in the case when the membrane deflection is sufficiently small and it is stretched by tension T . If the tension in the diaphragm is large and the diaphragm is thin, the membrane deflection Δz is,

$$\Delta z \approx \frac{P}{2T} \quad (5.3)$$

Note that one can achieve a displacement Δz not only by changing, P but also by changing the tension parameter, T . In this chapter, we present a new type of lens where the tension, T is variable and electrically controlled which changes the deflection of the front membrane, Δz as a result the focus of the lens. Fig. 5.2 shows a schematic of the lens with new type of actuation mechanism consisting of a sealed, pressurized microfluidic chamber enclosed by two flexible membranes. The top membrane includes a very fine SMA wire coil near its anchoring perimeter. The tension is made large by

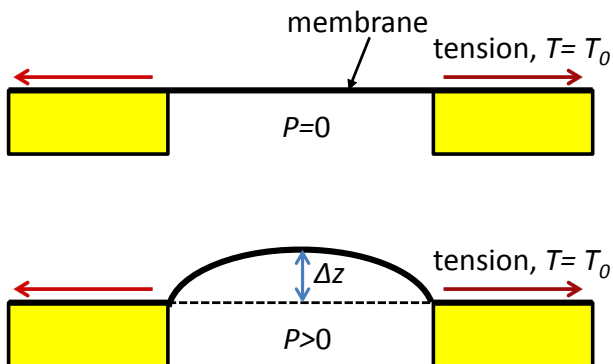


Fig. 5.1. Conventional membrane actuator driven by pressure changes.

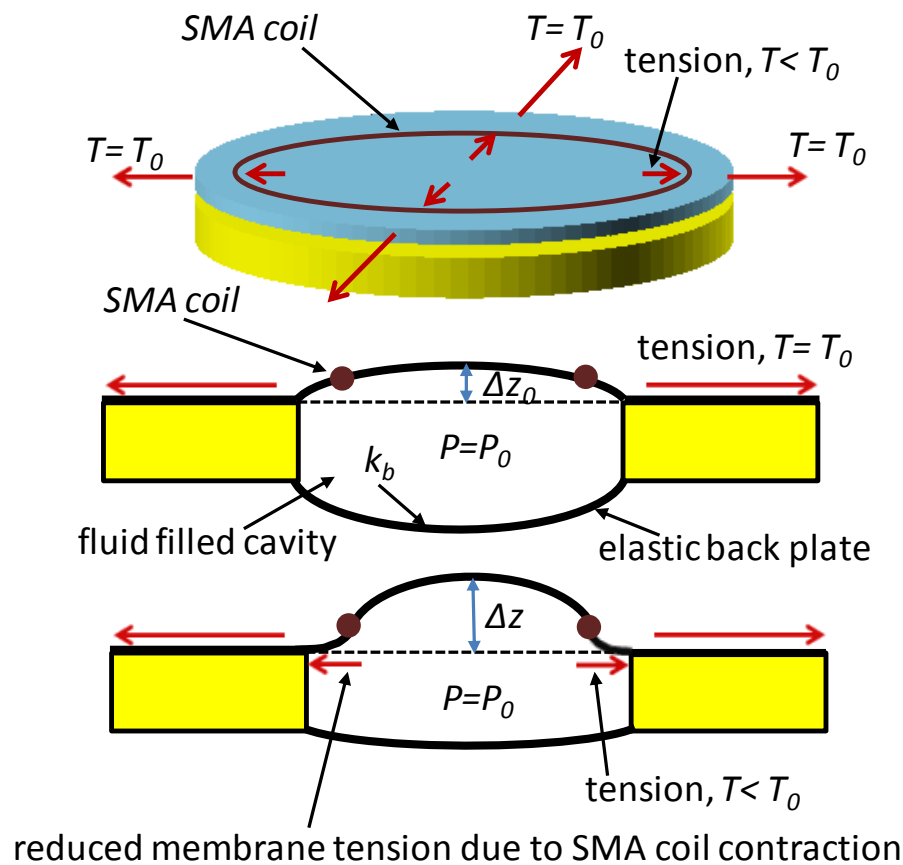


Fig. 5.2. New actuator deflection driven by tension changes which in return creates lensing effect. The tension is reduced by the SMA coil contraction.

prestretching the top membrane which as a result prestretches the SMA wire. The bottom of the sealing cavity consists of an elastic diaphragm of spring constant, k_b . When the coil of SMA wire is electrically heated, the SMA wire contracts in lengths, thus producing a large force. The contraction of the SMA wire results in an inward force along the plane of the membrane that reduces the tension on the bulk of the thin membrane. Effectively the diaphragm is thus subject to a reduced tension which causes change in deflection,

$$\Delta z \approx \frac{P_o}{2(T_o - T_w)} \approx \frac{P(\Delta z)}{2(T_o - aV^n)} \quad (5.4)$$

where T_w is the electrically controlled coil tension, a is a constant, and V is the coil voltage. The pressure P_o is the initial chamber pressure. For a fixed prestretching, this inward force depends on the diameter of the SMA wire and the number of turns. If the SMA contraction force is large the wire contraction introduces an effective SMA coil induced tension,

$$T_w = \frac{\Delta l}{l} \cdot E_m \cdot t_m \quad (5.5)$$

which is subtracted from the membrane prestretched tension. Here, Δl is the contraction of the coil in radial direction; l is the initial distance of the coil from the center, E_m is the Young's modulus of the membrane material and t_m is the membrane thickness.

The change in deflection of the front membrane causes change in lens power as indicated in Eq. (5.2). The relationship between front membrane deflection and lens optical power, P_{opt}

$$P_{opt} = \frac{2\Delta z(n-1)}{r^2} \quad (5.6)$$

Δz is the deflection of front membrane, n is the refractive index of fluid trapped inside the lens chamber, and r is the radius of lens either front or back membrane.

5.3 Lens fabrication

Several variations of the lenses were fabricated with different turn numbers and initial pressure for experimentation. To implement the spiral SMA coil, a PDMS membrane was fabricated with spiral ridges. The membrane mold was made using 1.5 mm-thick acrylic sheets. A laser cutter (VLS 3.6) was used to define the spiral grooves over the acrylic sheet shown in Fig. 5.3. Each groove is ~ 0.5 mm deep, ~ 170 μm wide, and the distance between two neighboring grooves is ~ 450 μm .

After making the mold, PDMS was poured in a mixture of 10:1 with base to curing agent (SYLGARD 184 Silicone Elastomer). The mixture was kept in an oven at 45°C for 5 hours to cure. Next, 100-310 μm diameter SMA wires (Dynalloy, Inc) were carefully inserted in between the ridges, and a second layer of PDMS was spun cast over the first membrane. Both the front and back membranes were 1.5 mm thick, but only the front membrane had wire coils. The cylindrical hollow chamber was machined from a

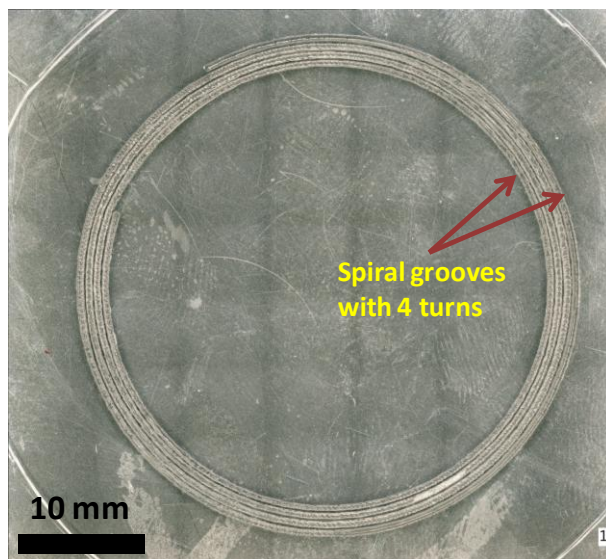


Fig. 5.3. Optical photo of the mold which was used to fabricate PDMS membrane with spiral grooves.

2.6 mm thick acrylic sheet using a laser cutter. The inner radius of the annular sealing rim is 18 mm and outer radius is 21 mm. The PDMS membranes were attached to both sides of the cylindrical chamber using silicone adhesive (Dow Corning). Two small holes were kept in the cylindrical chamber in order to insert the liquid. Glycerin was used as the pressurization liquid as it doesn't swell the PDMS membrane nor evaporate through the membranes [1]. After inserting glycerin, the microfluidic chamber was pressurized and the access holes were sealed. Fig. 5.4 shows a photo of the finished lens.

5.4 Experimental setup

The Young's modulus of PDMS varies depending on curing cycle and base to curing agent mixture ratio. We measured the Young's modulus and initial tension of the PDMS membrane which are 900 kPa and 29 N/m by the method described in Yang et al.

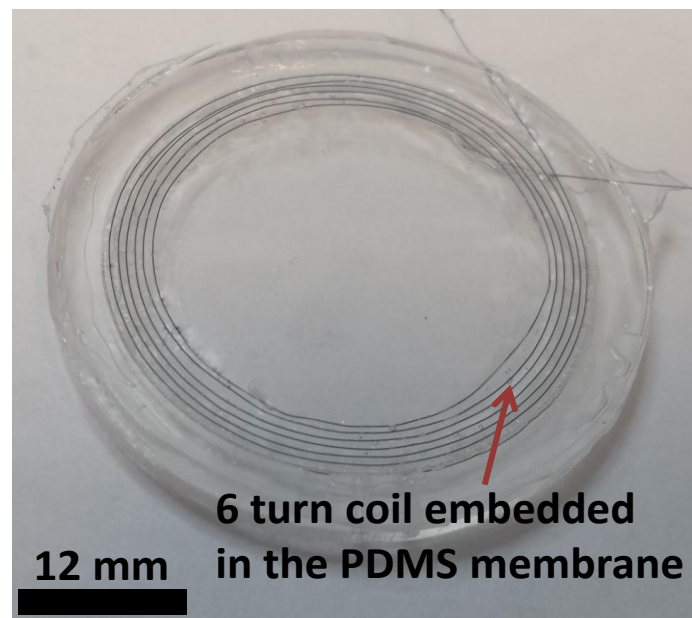


Fig. 5.4. Photograph of lens with top SMA coil. The coil has 6 turns and SMA wire diameter is 100 μm for this lens.

[2]. After fabricating 1.5 mm thick PDMS membrane with 6 turns SMA wire coil embedded in it, the membrane was attached to a hollow rim. The other side of the rim is attached to a fixed base as shown in Fig 5.5. The base has a hole for pumping air into the chamber. Both the syringe and a pressure gauge are connected to the hole of the base through a T-connector. When air is pumped into the chamber via the syringe, the flexible PDMS membrane deflects upward and the pressure gauge gives the reading of the corresponding pressure. The deflection of the membrane is measured using a Keyence VHX-5000 digital microscope.

The relationship among membrane's deflection, Δz with membrane's Young's modulus, E and prestrain, ε_i is,

$$\Delta z = -\frac{1}{3} \left[\sqrt[3]{\left(\frac{-27N + \sqrt{72N^2 + 108M^2}}{2} \right)} - \frac{3M}{\sqrt[3]{\left(\frac{-27N + \sqrt{729N^2 + 108M^3}}{2} \right)}} \right] \quad (5.7)$$

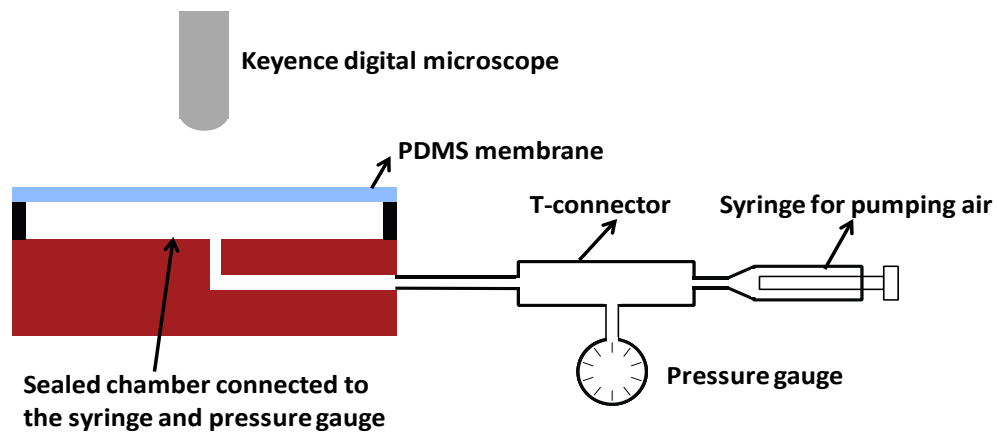


Fig. 5.5. Setup for deflection vs pressure measurement to determine the Young's modulus of PDMS and membrane pre-tension.

Here,

$$M = \frac{56t_i^2 + 42r^2(1+\nu)\varepsilon_i}{23 + 14\nu - 9\nu^2} \quad (5.8)$$

and

$$N = \frac{21P(1-\nu)r^4}{(46 + 28\nu - 18\nu^2)Et_i} \quad (5.9)$$

t_i is the thickness of the membrane, P is the applied pressure measured by pressure gauge, r is the radius of the membrane, and ν is the Poisson's ratio. The measured pressure vs. deflection values are given in Table 5.1.

To fit the deflection data as a function of pressure, we did the curve fitting in both MATLAB and Python varying the values of Young's Modulus and prestrain. Poisson's ratio for PDMS is used as 0.495. From the curve fitting, we got the values of Young's modulus and prestress 900 Kpa and 29 N/m, respectively. The example Python code is given in Appendix A3.

For the lens, the pressure difference across the PDMS membrane was 700 Pa. The SMA coil wire diameter is 100 μm and it has 6 turns. The lens optical power was measured using a Shack-Hartmann (SH) sensor from Thorlabs (WFS150-7AR). The

Table 5.1 Deflection of the Membrane at Different Pressure

Pressure (Pa)	Deflection (mm)
0	0
140	0.134
210	0.184
540	0.413
700	0.574
1000	0.753
1530	1.079
1970	1.272
2250	1.53

optical test setup is shown in Fig. 5.6. A collimated light source from Thorlabs (M625L3-C1) with wavelength 625 nm was placed 50 cm apart from the device. The device was placed 1.4 cm far from the SH sensor. For measuring the lens aberration, we used the 4f afocal system discussed in Chapter 7.

5.5 Results and discussion

When a voltage is applied to the SMA coil, resistive heating occurs and the SMA wire changes its phase from martensite to austenite state, thus contracting in length. The wire contraction changes the net tension of the membrane making it more convex according to Eq. 5.2. For a 6 turn coil of 100 μm diameter wire, the initial resistance was 120.2 Ω .

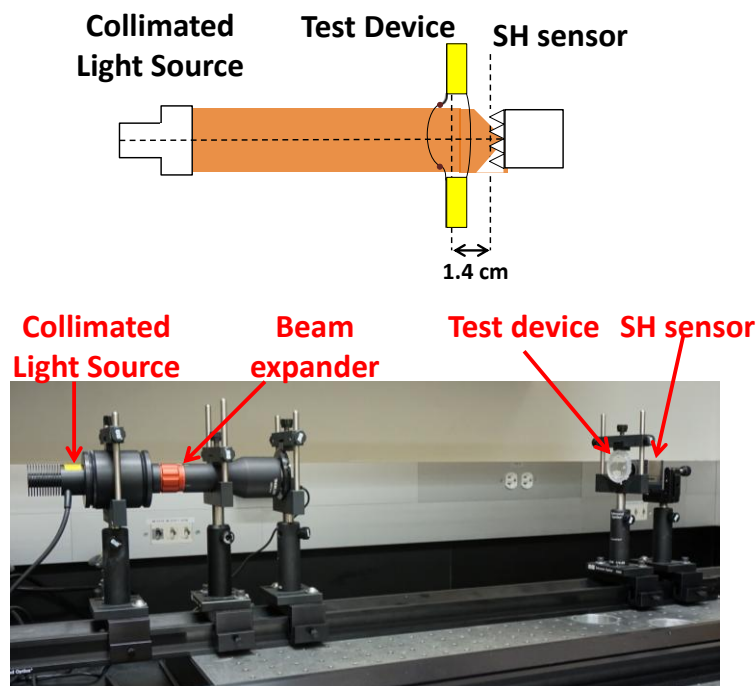


Fig. 5.6. Optical test setup for measuring the lens optical power. Schematic on top and photograph of the setup at the bottom.

The lens optical power as a function of voltage is shown in Fig. 5.7. At lower voltages, resistive heating and the optical power are small. The maximum optical power measured is 3.34 D. The voltage was applied to the coil for 10 seconds before taking the optical data. The SMA wire resistance decreases rapidly after a certain increase in temperature; hence we see a rapid change of optical power after 10 V [3].

To demonstrate the effect of initial pressure in the bulging of the front membrane, we measured the front membrane deflection at two different pressures. For this measurement, we used a device with just one turn of SMA coil wire diameter of 310 μm . This wire provides 8 times larger force than 100 μm diameter wire. The deflection of the front PDMS membrane for two different initial pressures is shown in Fig. 5.8.

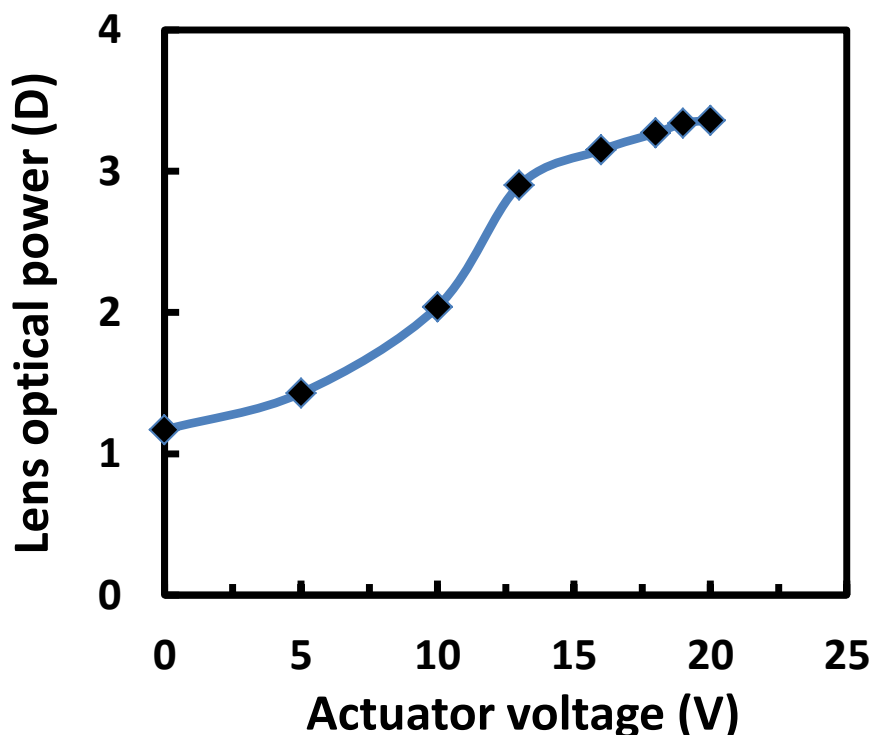


Fig. 5.7. Lens optical power as a function of SMA coil voltage. The standard deviation of lens power is below 1%.

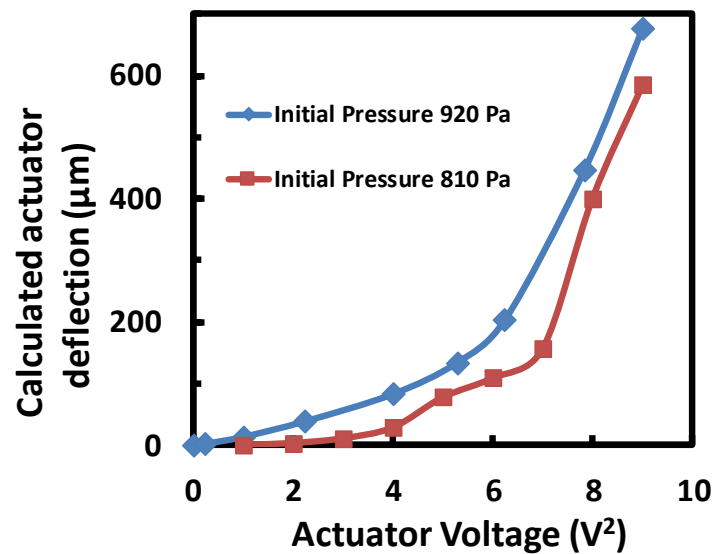


Fig. 5.8. Calculated membrane's central displacement using the measurement of SH sensor as a function of SMA wire voltage for different initial pressure.

We also took photos using the lens at different optical power. The lens which has 1 turn of SMA wire coil was kept in a horizontal direction attached with digital single lens reflex camera with a 40 mm lens. A reading sample was kept 25 cm away from the device. By applying the voltage across the SMA coil, the optical power of the lens was changed. The four photos shown at Fig. 5.9 are taken at 2.56 D, 2.79 D, 3.5 D, and 5.4 D, respectively, indicative of progressively larger optical power and central membrane deflection. The response bandwidth was ~ 1 Hz.

The lens aberrations were also measured using the $4f$ afocal system in conjunction with Shack-Hartmann sensor discussed in Chapter 7 [4]. As the lens has thicker membrane (1.5 mm thick), lens optical quality is better than the first two lenses discussed in Chapter 3 and 4. The aberration values are shown in Table 5.2. As the tension for the front membrane is larger, effect of coma aberration is less prominent for this lens. It is also independent of trefoil aberration.

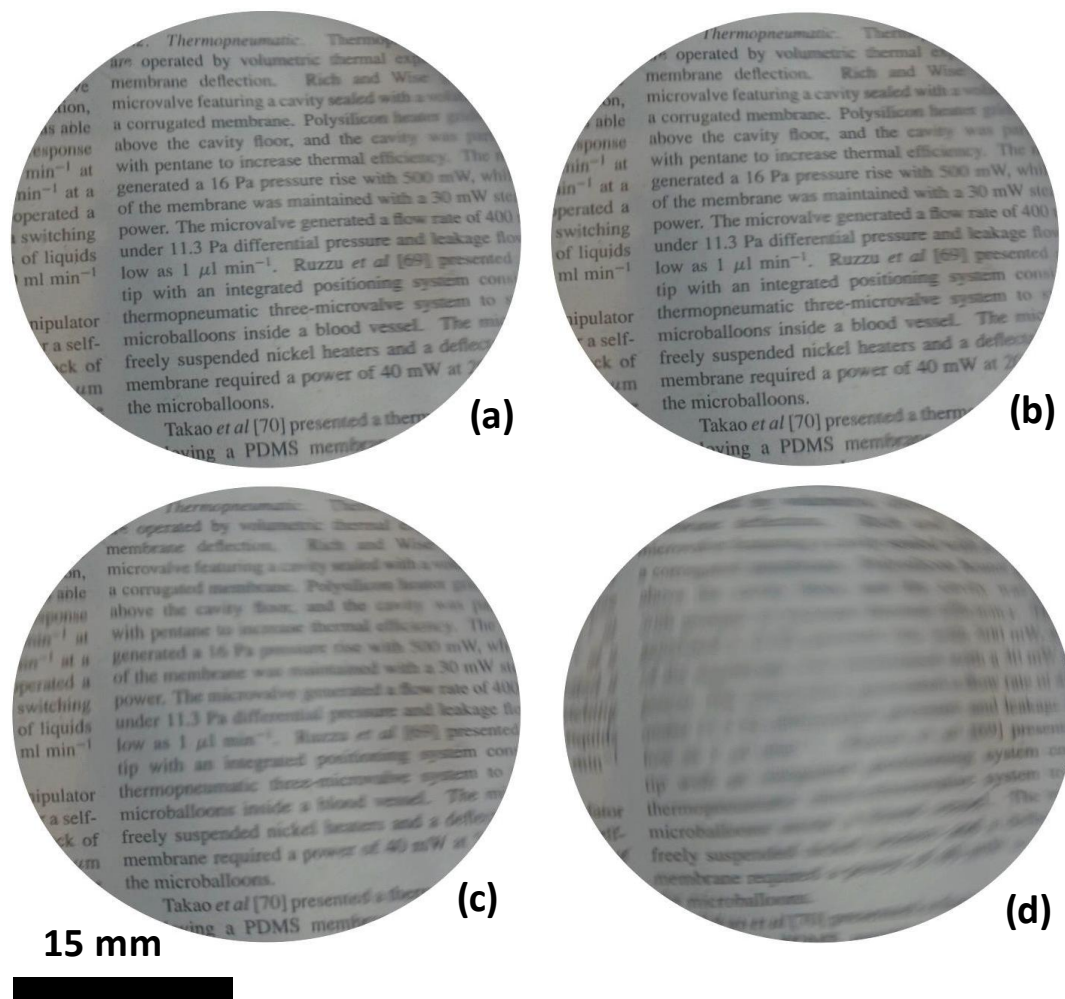


Fig. 5.9. Optical lensing at four different SMA voltages for the one turn device. Photos were taken at optical power (a) 2.56 D, (b) 2.79 D, (c) 3.5 D, and (d) 5.4 D.

Table 5.2 Lens Aberration

Aberration	Optical power 1.1 D	Optical power 3.3 D
Astigmatism 45°	0.340 μm	0.420 μm
Astigmatism 90°	0.410 μm	0.432 μm
Trefoil X	0.003 μm	0.004 μm
Trefoil Y	0.001 μm	0.003 μm
Coma X	0.145 μm	0.149 μm
Coma Y	0.492 μm	0.480 μm
Spherical	0.080 μm	0.314 μm
RMS Aberration	0.745 μm	0.845 μm

5.6 Conclusion

We have demonstrated a simple microfluidic lens which can focus at low voltages based on membrane tension modification and voltage controlled stiffness. The device has very simple construction. The thickness of the lens is 5.6 mm while weighing 8 g. The lens consists of a pressurized membrane under tension. The tension of the membrane is reduced by heating of an SMA wire coil embedded within the top membrane. When the SMA wire-coil contracts, the overall membrane tension is reduced causing significant amounts of focus change.

5.7 References and links

1. Y. Wang, J. Balowski, C. Phillips, R. Phillips, CE. Sims, and NL. Allbritton, "Benchtop micromolding of polystyrene by soft lithography," *Lab Chip*, **11**(18), pp. 3089-3097, 2011.
2. Q. Yang, P. Kobrin, C. Seabury, S. Narayanaswamy, and W. Christian, "Mechanical modeling of fluid-driven polymer lenses," *Applied Optics*, **47**(20), pp. 3658-3668, 2008.
3. H. Song, E. Kubica, and R. Gorbet, "Resistance modeling of SMA wire actuators," in *International Workshop on Smart Materials, Structures & NDT in Aerospace Conference* (2011).
4. C. Li, G. Hall, X. Zeng, D. Zhu, K. Eliceiri, and H. Jiang, "Three-dimensional surface profiling and optical characterization of liquid microlens using a Shack-Hartmann wave front sensor," *Applied Physics Letters* **98**(17), 171104 (2011).

CHAPTER 6

CONTROL OF LENS ABERRATION

6.1 Effect of gravity on lens optical quality

The front diaphragm deformation of the lens is not only subject to the piston force but also the effect of gravity. If the lens is standing upright on its edge, gravity produces hydrostatic pressure which increases linearly from the top to the bottom of the lens. This hydrostatic pressure adds to that of the piston thus producing a nonspherical deformation and asymmetric bulging of the diaphragm. This lens shape distortion produces a significant amount of coma aberration that must be minimized for acceptable optical performance [1,2]. The deformation of membranes under symmetric hydrostatic pressure is given in [3],

$$u_h = \frac{\rho_o g r_i^3 (1 - \rho^2) \cos(\theta)}{8T} \left(1 - \frac{2(I_1(\beta) - I_2(\rho\beta))}{\beta(1 - \rho^2)I_2(\beta)} \right) \quad (6.1)$$

where g is the gravitational acceleration (9.8 m/s^2), θ is the angle respect to the vertical axes, and $I_2(\cdot)$ is the second order modified Bessel functions of the first kind. The hydrostatic pressure produces an S -type deflection that adds to the symmetric deflection of Eq. (4.3) of the previous chapter. The effect of S -shaped coma aberration with symmetric spherical bulging is shown in Fig 6.1. The net result of the distortion is that

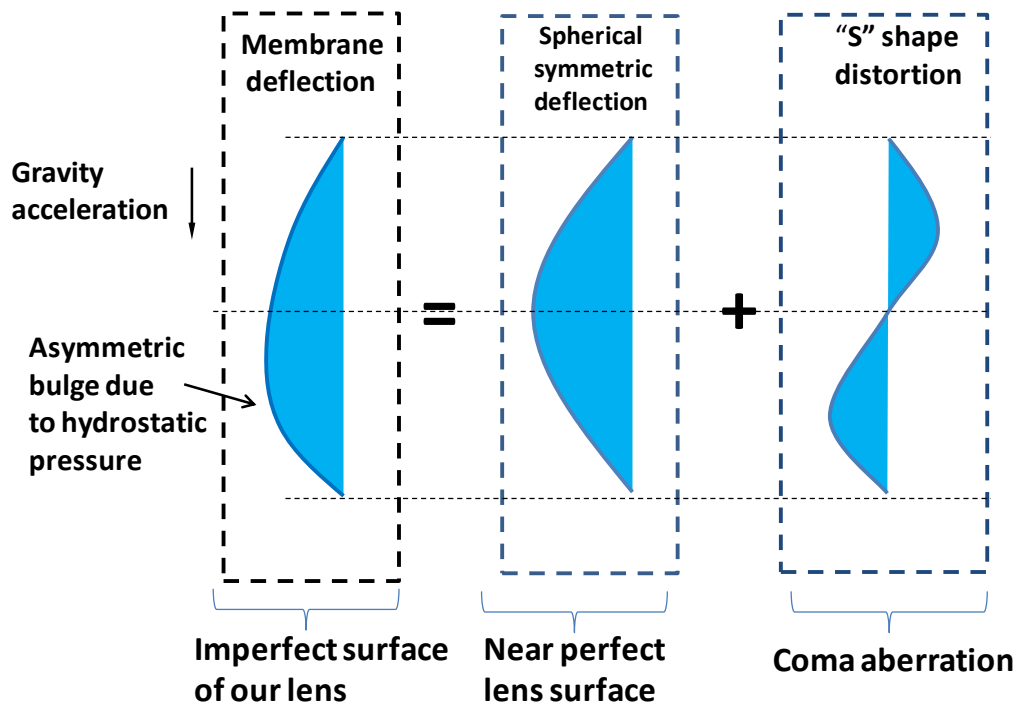


Fig. 6.1. Hydrostatic asymmetric bulging for our lens is the superposition of coma aberration and spherical symmetric bulging.

the optical power at the top is lower than at the bottom of the lens.

6.2 Spherical aberration and astigmatism

Spherical aberration occurs when paraxial rays and marginal rays for a lens system focuses at different spots on the optical axis. This is typically produced by a degree of plate-like behavior of the membrane. The deviation causes the optical power to increase or decrease radially causing objects to be in focus in the center but out of focus on the periphery of the image. The symmetrical deviation of the membrane deflection from the spherical curve is the superposition of lens spherical aberration and symmetric spherical bulging as shown in Fig. 6.2. Spherical aberration for our piezoelectric lens is below $0.2 \mu\text{m}$ as the front membrane is tension dominated and it has aspherical shape to a

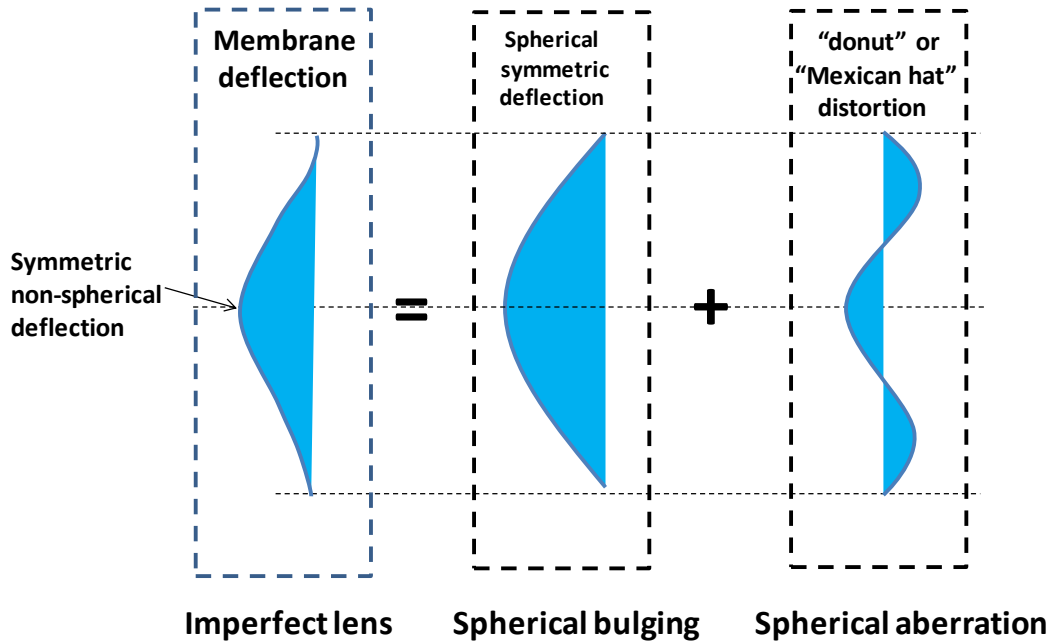


Fig. 6.2. The membrane nonspherical profile is decomposed into a perfect spherical deflection and the spherical aberration.

moderate extent [4].

Lens astigmatism aberration occurs as a result of different lens curvatures at different directions. The optical power variation due to astigmatism along vertical and horizontal axes for the piezoelectric lens is, $\frac{P_x}{P_y} = \left(\frac{a}{b}\right)^2 \approx 0.15$ D [5]. Here, a and b are the radii of curvature of the lens front membrane along X and Y axes.

6.3 Zernike polynomials

Zernike polynomial is a set of orthogonal polynomials which can be used to describe a wavefront in optics for circular pupil with unit radius.

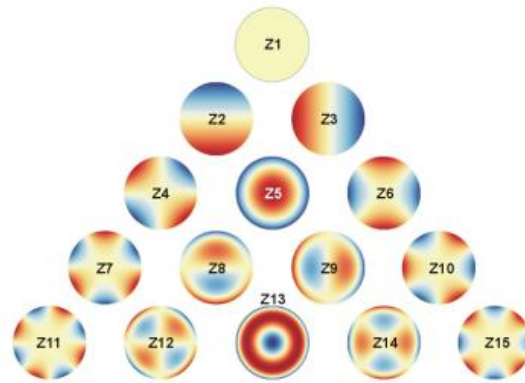
$$W(x, y) = \sum_0^{\infty} c_n Z_n(x, y) \quad (6.2)$$

Here, $W(x,y)$ is the wavefront, $Z_n(x,y)$ is the orthogonal Zernike function weighted by their amplitudes or Zernike coefficients c_n . Zernike modes are expressed in either polar or rectangular coordinate reference forms. Polar coordinate Zernike polynomials are expressed as $Z_n^m(\rho,\theta)$. Here, $\rho=r/a$, a is the pupil radius, r is the physical radial coordinate, n is radial order, m is angular frequency, and θ is angular coordinate. Various schemes for ordering Zernike sequence have been used in literature. Here, we used the ordering depicted in Thorlabs WFS 150-7AR manual. Zernike coefficients, c_n associated with different Zernike order represent different wavefront properties and aberration. For further clarification in this dissertation, we wrote down the names of the aberrations instead of representing them by Zernike coefficients. In the simulation codes, we gave necessary notations to point out Zernike coefficients representing aberrations. Fig. 6.3 shows different Zernike modes representing different optical aberrations and optical attributes depicted in Thorlabs manual [6].

6.4 Simulation

The power uniformity and quality of the lens image can be arbitrarily improved if the tension is increased at the expense of optical power range. As coma is the dominant aberration for our liquid lenses, we did both simulation and experimentation to understand the effect of gravity and other parameters on coma aberration. In the following section, we discussed the simulation procedure regarding the effect of tension, membrane radius, and liquid on coma aberration and optical power range.

For simulation purpose, we consider lens front membrane as a flexible flat plate deflecting in accordance to varying applied pressure. We ignore the effect of back plate



(a)

Zernike Term	Shortcut	Description
Z 1	-	Piston
Z 2	-	Tip Y
Z 3	-	Tilt X
Z 4	Ast45	Astigmatism 45°
Z 5	Def	Defocus
Z 6	Ast0	Astigmatism 0°
Z 7	TreY	Trefoil Y
Z 8	ComX	Coma X
Z 9	ComY	Coma Y
Z 10	TreX	Trefoil X
Z 11	TetY	Tetrafoil Y
Z 12	SAstY	Secondary Astigmatism Y
Z 13	SAb3	3rd Order Spherical Abberation
Z 14	SAstX	Secondary Astigmatism X
Z 15	TetX	Tetrafoil X

(b)

Fig. 6.3. Sequence of Zernike coefficients representing different aberrations according to Thorlab Shack-Hartmann sensor [6].

because for the SMA spring actuated lens discussed in Chapter 3, it has a rigid back plate which is not affected by gravity or tension. For the piezoelectric actuated lens discussed in Chapter 4, the thin lens back membrane is attached to solid piston which greatly eliminates the effect of gravity as well. For doing the simulation, we used programming language Python [7]. Python is an open source language with large and comprehensive library. We particularly did two kinds of simulations in Python.

First, we simulated the front membrane behavior under different tension and found corresponding coma aberration values. With more pre-tension, coma aberration decreases but higher tension value also reduces the optical power range of the lens. Therefore, we did the trade-off in between acceptable coma aberration and lens optical power range, and chose the lens parameters according to that.

Second, we did finite element analysis of the membrane using Python. Our goal was to incorporate circular array of stiff ribs (i.e., glass ribs) into PDMS membrane to strengthen the membrane elasticity in places in order to reduce coma aberration without compromising lens optical power range. We constructed the front membrane with variable shape and number of ribs embedded in it. The pre-tension of the membrane was also variable. By applying the force generated by piezoelectric actuators, we found the aberration values (especially coma aberration) and lens optical power range.

6.4.1 Installing Python and other software

We installed Python XY with the specialized libraries pygmsh, pycad, opticspy, easyprocess, and trimesh. The full installation instruction is given in Appendix B.

Openscad is a free software to design and create 3D cad objects. We used python to draw membrane plate with and without ribs utilizing Openscad. Gmsh is a free three-dimensional mesh generator. After drawing the membrane by Openscad, we used python to generate mesh through gmsh. Getfem++ is a library for finite element analysis which has a Python interface. Calculix is the finite element solver. The codes in Appendix A4, A5, and A6 were written and ran after installing the software discussed above.

6.4.2 Results and discussion

Without applying any external pressure to the flexible membrane of the lens, first we calculated the deflection of the lens front membrane. The simulation code is given in Appendix A4. From the simulation, it is clearly shown that coma aberration reduces with higher tension. Fig. 6.4 shows membrane deflection at two different pre-tension values. Keeping the membrane material as PDMS, we can increase the tension of the membrane by thickening it or by prestretching it. However, with higher tension lens optical power range reduces. For lens fabrication, we used membrane thickness from 1.3 mm to 1.8 mm. For these parameters, coma aberration is below 1 μm from simulation. By applying tension and piston pressure, coma aberration reduces further for actual lens.

We also ran the simulation to see the deflection of the membrane in 3D as a function of different pre-tension. Fig. 6.5 shows two instances of the membrane deflection from the animation.

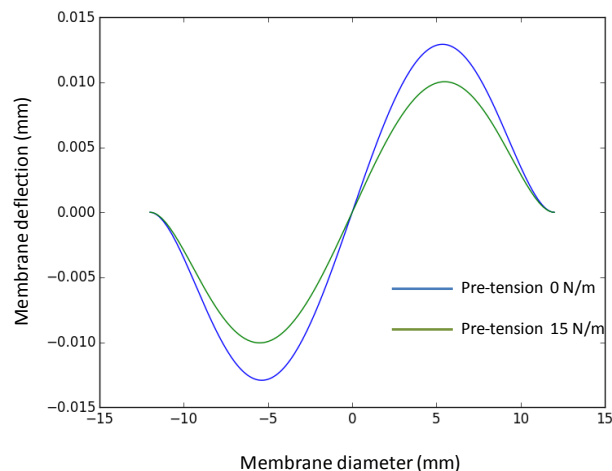


Fig. 6.4. Membrane deflection for different pre-tension simulated by Python. No external pressure is applied here. Membrane deflects producing “S” shaped curve for hydrostatic pressure.

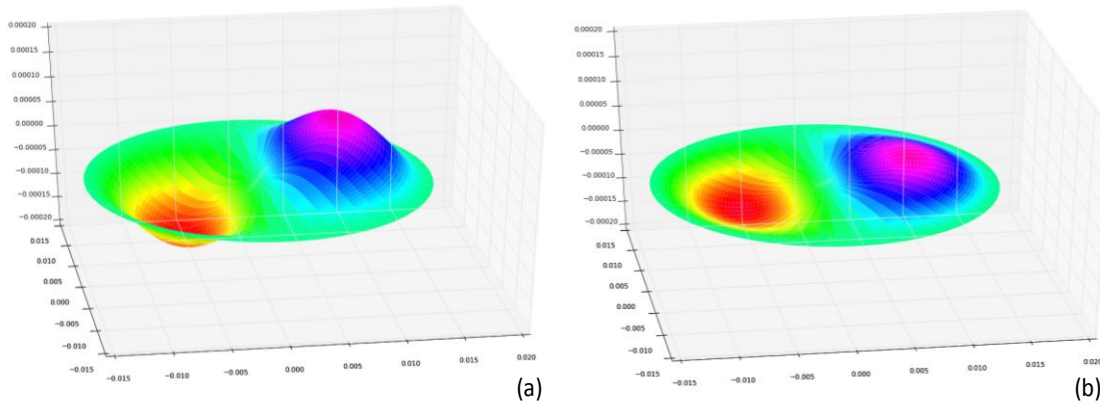


Fig. 6.5. Lens front membrane deflection for hydrostatic pressure (a) Large membrane deflection at lower tension producing huge coma aberration, (b) small deflection at higher pre-tension producing smaller coma aberration.

When the lens is put in vertical position, the slope of the distorted lens power versus height, y at the lens center can be calculated from the mean curvature of Eq. (6.3) as,

$$\left. \frac{\partial P_{opt}}{\partial y} \right|_{r=0} \approx -\frac{\rho_o g}{2T} (n-1) \left(1 - \frac{\beta^2}{8I_2(\beta)} \right) \quad (6.3)$$

The calculated and measured slopes of the optical power at the center of the lens are -0.084 D/mm and -0.09 D/mm, respectively. Fig. 6.6 shows the measured optical power of the lens chamber (excluding piston and actuators) as a function of height for the lens chamber placed in upright position. The measured pre-tension for the front membrane in that case is 35 N/m.

Next, we did the finite element analysis using Python. The python code is given in Appendix A5. First, we made the front membrane by Openscad using Python. We used Gmsh to create triangular mesh for the membrane. Calculix was used to do the numerical analysis.

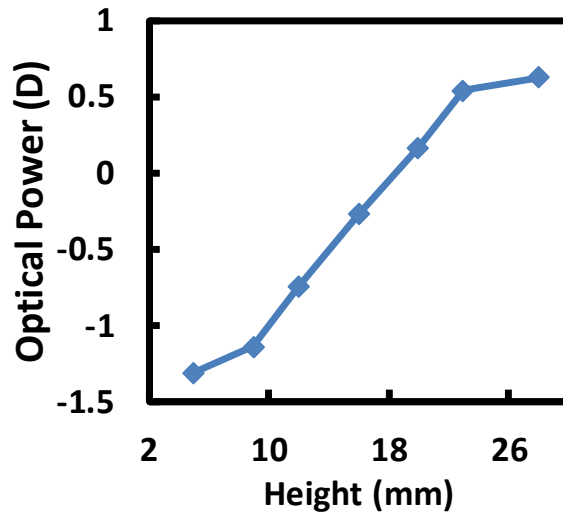


Fig. 6.6. Optical power profile of the lens chamber (excluding piston and actuators) as a function of height when the chamber is placed in vertical direction.

All the parameters were given according to the actual lens material and dimensions. Fig. 6.7 shows front membrane Openscad plot and created mesh from Gmsh. Lens membrane material was PDMS with thickness 1.3 mm. The Young's modulus and Poisson's ratio for PDMS were used as 1 Mpa and 0.49. We used Glycerol as the lens liquid with refractive index 1.47 and density 1260 Kg/m³. The piston force was also compatible with the actual force we get from piezoelectric actuators.

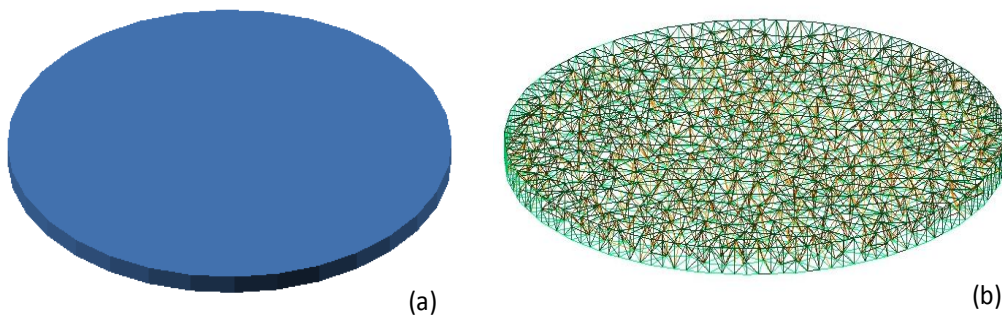
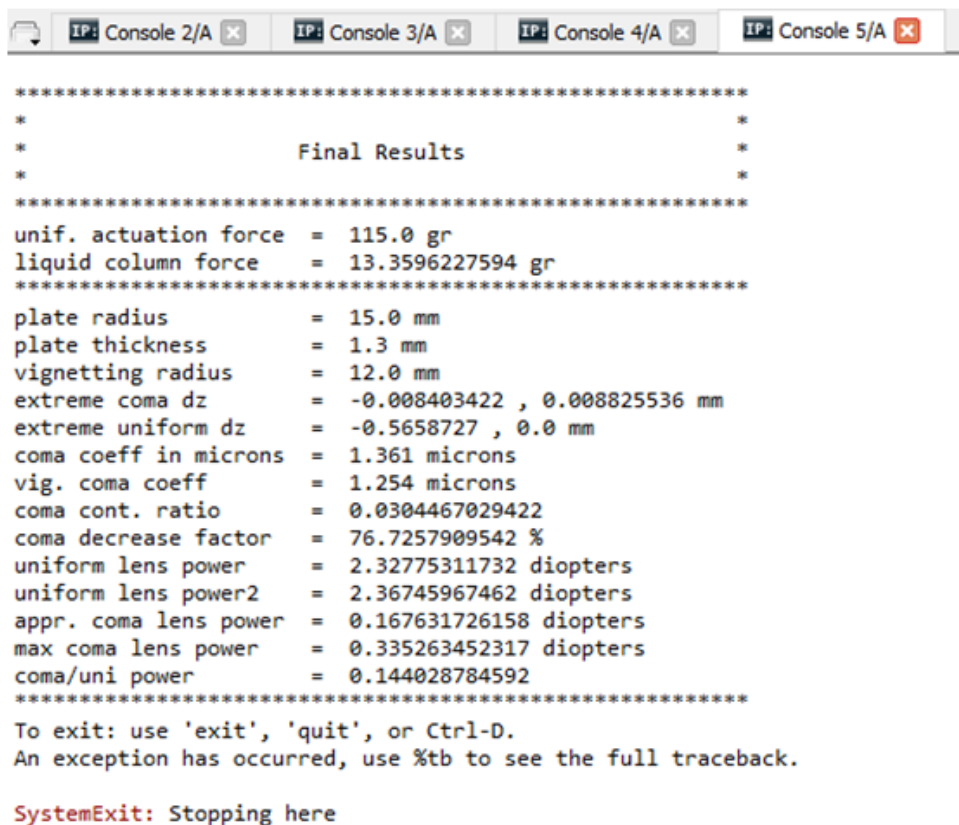


Fig. 6.7. Simulation of lens front membrane. (a) Drawing of the membrane geometry via Openscad and (b) creating triangular mesh using Gmsh. Both Openscad and Gmsh were run using Python.

The result from the simulation is given below in Fig. 6.8. For the lens discussed in Chapter 4, the value of coma aberration is lower than the simulated result here as the pre-tension for the front membrane in actual lens is higher. The front membrane tension also changes for the piston-actuator system which is attached to the back membrane in the case of an actual piezoelectric lens. Our fabricated lens has astigmatism and trefoil aberration as well which have negligible values in the simulation. From the simulation, we get good estimation of the lens fabrication parameters which we incorporated in fabrication.

To further reduce the coma without compromising lens optical power range, we ran the simulation incorporating glass ribs into front PDMS membrane. We varied the rib



```

*****
*                                     *
*                               Final Results                               *
*                                     *
*****
unif. actuation force = 115.0 gr
liquid column force  = 13.3596227594 gr
*****
plate radius          = 15.0 mm
plate thickness       = 1.3 mm
vignetting radius    = 12.0 mm
extreme coma dz      = -0.008403422 , 0.008825536 mm
extreme uniform dz   = -0.5658727 , 0.0 mm
coma coeff in microns = 1.361 microns
vig. coma coeff      = 1.254 microns
coma cont. ratio     = 0.0304467029422
coma decrease factor = 76.7257909542 %
uniform lens power   = 2.32775311732 diopters
uniform lens power2  = 2.36745967462 diopters
appr. coma lens power = 0.167631726158 diopters
max coma lens power  = 0.335263452317 diopters
coma/uni power       = 0.144028784592
*****
To exit: use 'exit', 'quit', or Ctrl-D.
An exception has occurred, use %tb to see the full traceback.

SystemExit: Stopping here

```

Fig. 6.8. Simulation result showing lens output optical power and coma.

size, shape, and number. Fig. 6.9 shows an example of simulated front membrane which has glass ribs (dashed line box) embedded in it. Although incorporating glass ribs reduces coma, it produces higher order optical aberration. Therefore, lenses made with PDMS membrane which has glass ribs embedded in it are only good for focusing purpose, but not for any imaging or vision application. Edges of glass ribs also impair vision. The code for front membrane simulation with glass ribs is given in Appendix A6.

6.5 Summary

From the simulation we found the estimation of lens fabrication parameters and lens characteristics as well. We chose the membrane parameters and tension values according to that. The aberration values we got from the lens discussed in Chapter 4 were used to calculate the point spread function (PSF) for the worst case scenario. We get the

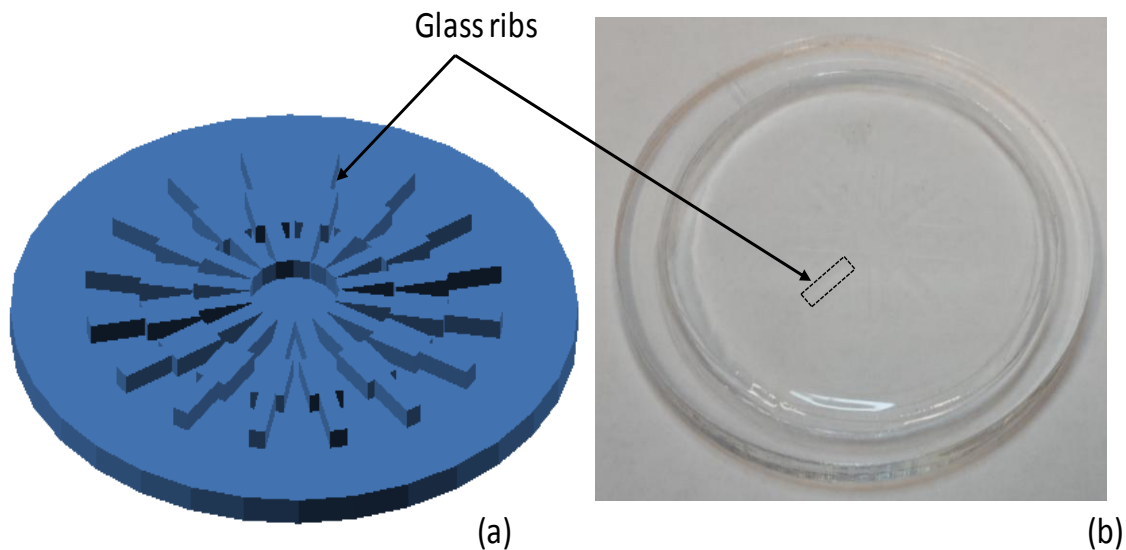


Fig. 6.9. Front membrane simulation and fabrication with embedded glass ribs array. (a) Construction of membrane via openscad for simulation and (b) fabricated PDMS front membrane with embedded glass ribs.

80% encircled radius of the PSF 0.1° which is 30 times larger than abbe limit. Fig. 6.10 shows the PSF for the worst case scenario of the piezoelectric lens discussed in Chapter 4. Point spread function shows the “comet” like figure because coma is the dominant aberration for this lens. From the simulation discussed in this chapter, we found the factors which determine coma aberration for liquid lenses and kept them in acceptable range for our lens. Coma can be reduced further or nearly eliminated with liquid-membrane-liquid lenses [8]. These lens configurations rely on the utilization of liquids of equal densities but different index on both sides of deflecting membranes.

The primary disadvantage of this coma elimination method is that optical power is severely reduced as it is proportional to the refractive index difference between two similar liquids (~ 0.2) compared to much larger index differences between optical liquids and air (0.5-0.7).

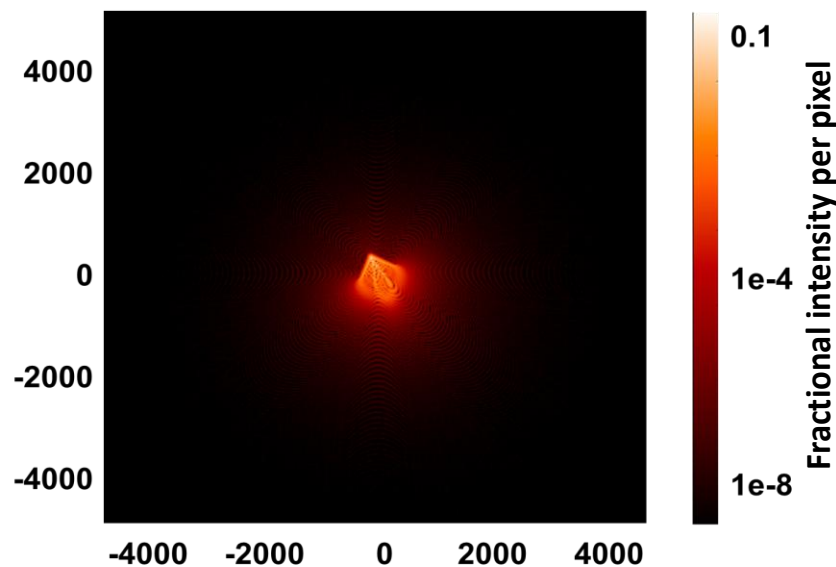


Fig. 6.10. Point spread function (PSF) constructed using Python from the aberration values found for piezoelectric lens at worst case scenario (0.78 D). The “comet” like PSF proves coma as the dominant optical aberration.

6.7 References and links

1. N. Sugiura and S. Morita, "Variable-focus liquid-filled optical lens," *Applied Optics* **32**(22), 4181-4186 (1993).
2. F. Schneider, J. Draheim, R. Kamberger, P. Waibel, and U. Wallrabe, "Optical characterization of adaptive fluidic silicone-membrane lenses," *Optics Express* **17**(14), 11813-11821 (2009).
3. E. H. Mansfield, *The Bending and Stretching of Plates* (Cambridge University, 1989).
4. Q. Yang, P. Kobrin, C. Seabury, S. Narayanaswamy, and W. Christian, "Mechanical modeling of fluid-driven polymer lenses," *Applied Optics* **47**(20), 3658-3668 (2008).
5. H. L. Dryden, F. D. Murnaghan, and H. Bateman, *Hydrodynamics* (Dover Publications, 1965).
6. <https://www.thorlabs.com/drawings/a56915a09662ca5c-562B793C-B2CB-29D0-F3BCB7E75E688230/WFS150-7AR-Manual.pdf>
7. <https://www.python.org/>
8. L. Wang, H. Oku, and M. Ishikawa, "Variable-focus lens with 30 mm optical aperture based on liquid-membrane-liquid structure," *Applied Physics Letters* **102**, 131111 (2013).

CHAPTER 7

EXPERIMENTAL SETUP FOR OPTICAL PROFILING

7.1 Shack-Hartmann (SH) wavefront sensor

Lens power, focal length, and wavefront measurements are made using a Shack-Hartmann wavefront sensor (SHS) (WFS150-7AR from ThorLabs). The working principle of Shack-Hartmann wavefront sensor was developed by Johannes Franz Hartmann in 1900 [1]. In 1960, Ronald Shack and Ben Platt modified the design for Shack-Hartmann sensor to its current form [2]. The schematic of Shack-Hartmann sensor is shown in Fig. 7.1. A lenslet array is placed in front of a CCD or CMOS image sensor. Each microlens in the lenslet array generates a single spot on image sensor. The position of the spot created on the image sensor depends on the incoming wavefront into the

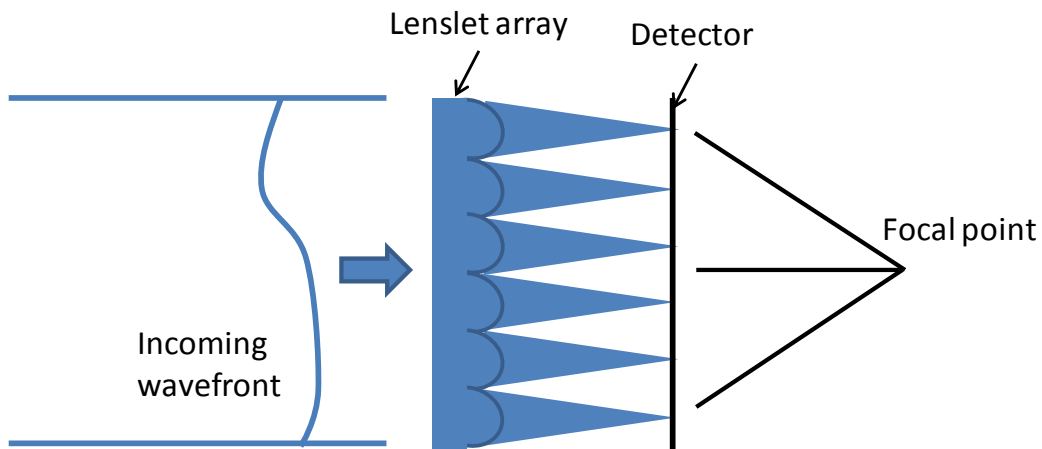


Fig. 7.1. Schematic of Shack-Hartmann (SH) sensor.

lenslet array. There are predefined spot positions for each microlens when the incoming wavefront is planar wave. By comparing the current spot position with the reference spot position, local slope can be calculated. With sufficient number of local slope measurements, the incoming wavefront can be constructed numerically. The WFS 150-7AR uses Zernike coefficients to reconstruct the wavefront. Zernike polynomials are set of orthogonal polynomials which were discussed in Chapter 6.

7.2 Measurement of lens focal length

To measure the lens focal length, a collimated light source (M625L3-C1 from ThorLabs) with wavelength 625 nm was used. We utilized the proximity technique for measurement of the focal length [3]. In order to measure the focal length, the collimated light source was placed 50 cm apart from the lens as shown in Fig. 7.2.

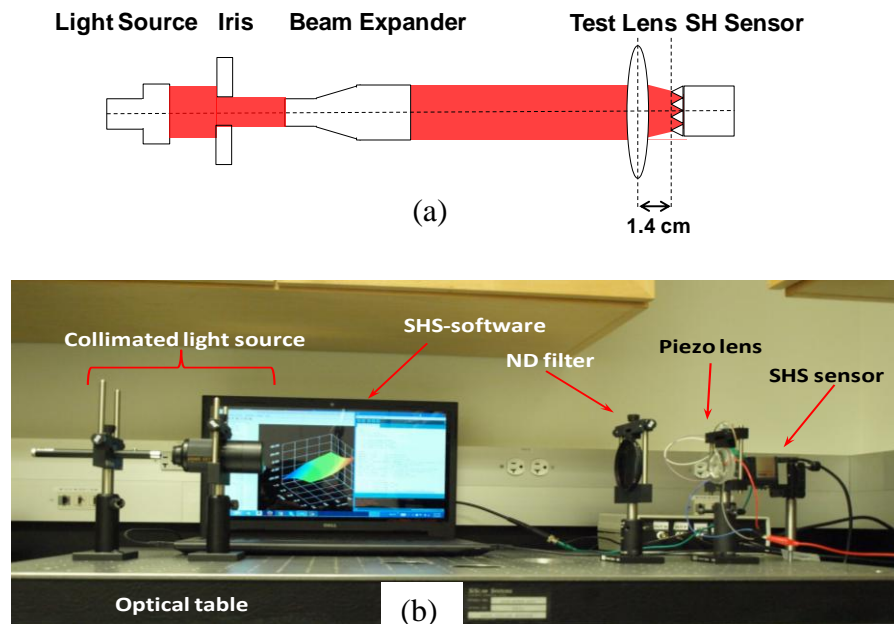


Fig. 7.2. Experimental setup. (a) Schematic of proximity technique to measure lens focal length and (b) photo of the test setup.

The test lens was clamped in vertical orientation in close proximity to the wavefront sensor (~ 1.4 cm between test lens and SHS lenslet array). SH sensor can measure the radius of curvature of the incoming light. Fig. 7.3 shows the basic geometry for measuring radius of curvature of incoming light for the SH sensor. A source S is placed at Z in front of a test lens with focal length f , Z_o is the position of collimated light source, L is the separation between lenslet array and test lens, and R is the measured radius of curvature using the algorithm of SH sensor. According to thin lens formula,

$$\frac{1}{f} = \frac{1}{f + Z - Z_o} + \frac{1}{R + L} \quad (7.1)$$

For our experimental setup, the incoming light to the test lens is collimated, which makes value of $Z_o = \infty$. Therefore, Eq. (7.1) reduces to the lens focal length, $f = R - L$ (radius of curvature of incoming light) $- L$ (separation of test lens and sensor) [3]. From the graphical user interface of the WFS 150-7AR, we got the value of RoC (radius of

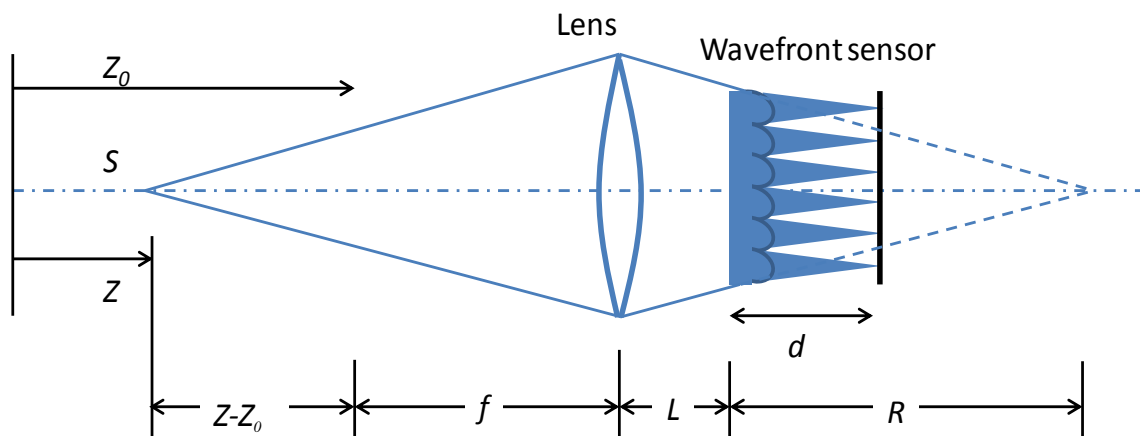


Fig. 7.3. Ray optics for measuring the focus of the test lens using SH sensor.

curvature) of incoming light to the SH sensor. By subtracting 1.4 cm from those values, we got the focal lengths of the test lens. For measuring focal lengths of the test lens, we passed the collimated light to the central part of the test lens. As the lens front membrane profile is nearly spherical, profiling the central part of the lens gives accurate focal lengths. Fig. 7.4 shows the SH sensor output with RoC indicated in a box.

7.3 Profiling lens full aperture for measuring wavefront aberrations

The proximity focus measurement technique works very well but only captures a small portion of the lens light, as the diameter of the SHS sensor (~ 4.6 mm) is much smaller than the liquid lens aperture. In order to approximately capture the entire light field from the lens, we also constructed a $4f$ afocal relay lens system that feeds all lens

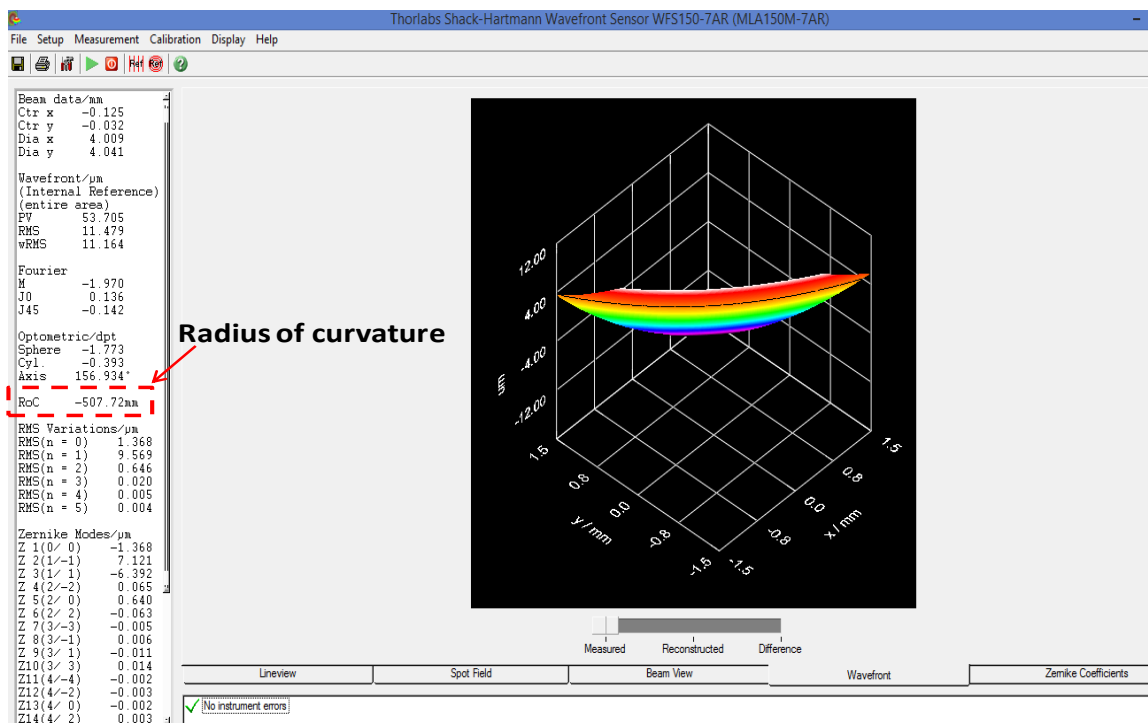


Fig. 7.4. SH sensor (WFS 150-7AR) graphical user interface.

light into the sensor as shown in Fig. 7.5 [4]. As the LED light source goes through an iris and a beam expander (Thor Labs GBE 10A) combination, it produces a highly collimated light beam 30 mm in diameter. For measuring the lens wavefront aberration, we profiled the central 25 mm diameter of the possible 32 mm aperture of the lens. The focal lengths of the first and second lens of the relay system are $f_1=20$ cm and $f_2=3.5$ cm, respectively. The test lens is placed f_1 away from the first relay lens. After the light beam passes through the test lens, the afocal relay lenses, placed $f_1+f_2=23.5$ cm apart, collect all the light and reduce the beam diameter by ~ 5.7 fold. The SHS is placed f_2 away from the second relay lens. It is also possible to measure the focal length of the test lens using the relay lens system but the observed focal lens is $f_2(1+(f_L \times f_2/f_1^2))$. Note that, since a beam reduction is required ($f_2/f_1 \ll 1$), the influence of the lens focal length on the entire system focal length (as measured by the SHS) is greatly reduced; hence the proximity technique is preferred for focal measurements. The relay lens setup (with no test lens) produced aberrations with RMS wavefront error $\sim 0.15 \mu\text{m}$. The optical setups discussed in this section were used to get all the aberration values reported in Chapter 3, 4, and 5.

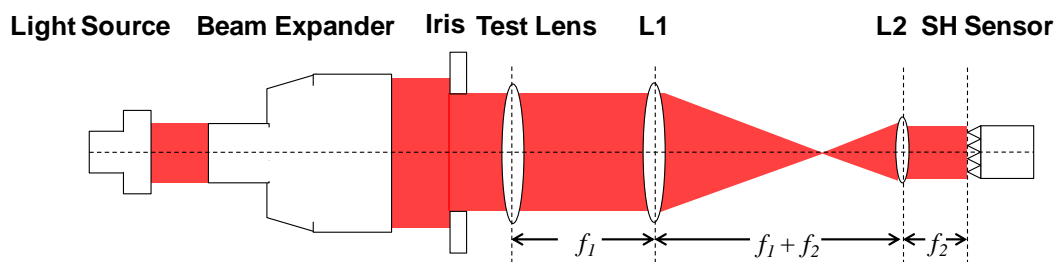


Fig. 7.5. $4f$ optical setup to measure lens aberration using Shack-Hartmann sensor.

7.4 References and links

1. Hartmann, J. (1900). "Bemerkungen über den Bau und die Justirung von Spektrographen," *Zeitschrift für Instrumentenkunde*. Berlin: Julius Springer. **20**: 17–27, 47–58.
2. Platt, Ben C.; Shack, Ronald (October 2001). "History and principles of Shack-Hartmann wavefront sensing," *Journal of Refractive Surgery*. **17**(5): S573–7.
3. D. R. Neal, J. Copland, D. A. Neal, D. M. Topa, and P. Riera, "Measurement of lens focal length using multi-curvature analysis of Shack-Hartmann wavefront data," *Proc. SPIE* **5523** (2004).
4. C. Li, G. Hall, X. Zeng, D. Zhu, K. Eliceiri, and H. Jiang, "Three-dimensional surface profiling and optical characterization of liquid microlens using a Shack–Hartmann wave front sensor," *Applied Physics Letters* **98**, 171104, 2011.

CHAPTER 8

CONCLUSION

8.1 Summary of the research

The full field view for people who have lost accommodation can only be restored by using adaptive eyeglasses which have variable optical power. In this dissertation, we report three new approaches for constructing variable optical power lenses. In Chapter 3, we report the first lens we made using shape memory alloy (SMA) spring actuators. Here, a flexible annular sealing rim is used as lens side wall. A rigid back plate is attached to one side of the rim and a flexible PDMS membrane is attached to the other side of the rim. Three SMA springs were housed in the annular sealing rim. When voltage is applied to the SMA springs, electrical energy converts to thermal energy contracting the SMA springs. As the volume of the trapped liquid inside the lens chamber is constant, flexible membrane bulges up changing the focal length of the lens. This lens is compact, lightweight, thin, and low footprint, hence suitable for ophthalmic application. However, SMA actuators have poor energy conversion which makes this lens electrical power hungry. The operating frequency of this lens is also very low because SMA springs take a long time to cool down naturally. These drawbacks motivated us to construct better lenses.

In Chapter 4, we report the second lens we constructed using curved piezo-

electric bimorph actuators. Here, an annular sealing rim was encapsulated with very thin flexible membrane in one side and thicker flexible membrane in another side. Glycerin was used as liquid here. Three piezo-electric bimorph actuators were placed along the periphery of the rim on the softer membrane side. One end of the actuators was fixed and the other end was free moving. The free moving ends of the actuators were attached to a clear piston. When the voltage is applied across the actuators, the free ends of the actuators move up and down along with the piston. The piston moves the flexible back membrane up and down making the front membrane convex and concave. This lens is thin, lightweight, low footprint, and has a large aperture. The electrical power consumption of the lens is less than 20 mW and the lens can change its focal length in less than 16 milli seconds. The lens front membrane has significant pre-tension which improves lens optical quality. To our knowledge, this is the most appropriate lens for ophthalmic application reported so far.

While experimenting with the lens discussed in Chapter 4, we got the idea for implementing tunable-focus lens using tension modification. In Chapter 5, we report the lens which was tuned by modifying the tension of the front membrane. The lens uses a fixed pressure as the driving force and a membrane of electrically controlled tension to change the deflection of the front membrane. An annular sealing rim was encapsulated by flexible membranes on both sides and liquid was inserted into it. A shape memory alloy (SMA) coil was embedded in the flexible front membrane which had initial tension. When voltage is applied across the coil, the coil heated up causing the contraction of the membrane. This changes the tension of the front membrane which results in variable lens power.

In Chapter 6, we discussed the simulation procedure to determine the lens parameters. We used Python to do finite element analysis. Coma aberration is the dominant aberration for any liquid lens in vertical direction. The effect of pre-tension over lens optical power range and lens quality is simulated as well. We utilized the optimum pre-tension and other parameters for lens fabrication. The experimental setup for optical profiling is discussed in Chapter 7 followed by a brief summary and future work in Chapter 8.

8.2 Future work

We successfully constructed tunable-focus lens suitable for ophthalmic especially eyeglass application. The lenses we fabricated are circular in size. Typical eyeglasses have lenses of different shapes and sizes. Eyeglasses have oval, rectangular, and circular lenses. We are developing tunable-focus lenses of rectangular shape as well, which requires rigorous simulation before fabrication as a rectangular shape lens needs variable tension and force in different portions of the lens.

We are developing adaptive eyeglasses using the tunable-focus lenses discussed in Chapter 4. The tunable eyepieces are driven by a microcontroller system that adaptively calculates the optical power required to produce sharp images from the prescription of the wearer and object distance range measurements. The microcontroller system wirelessly communicates with a mobile app that is used to enter the wearer prescription and other operating parameters. The driving circuits and microcontroller system are powered by 3.7 V rechargeable batteries embedded inside the eyeglasses frame temples. The whole system development is under way.

APPENDX A

PROGRAMMING CODES USED IN THE DISSERTATION

A1. Code for FDC 1004 capacitive sensor and sliding mode control

```
#include <Wire.h>

float cinit1,cinit2,cinit3;

float cap11;

float cap22;

float cap33;

float t1;

float c[5];

int i;

float delta;

float dC;

float ctarget1,ctarget2,ctarget3;

void ReadCapacitanceVector( float *a);

void PrintCapacitanceVector( float *a);

void PrintCapacitances( float a, float b, float c);

int FD = B1010000; //i2c initialization

void setup()

{
```

```
pinMode(3, OUTPUT); // pin which will give voltage to spring
pinMode(4,OUTPUT);
pinMode(5,OUTPUT);
digitalWrite(3,LOW);
digitalWrite(4,LOW);
digitalWrite(5,LOW);
Serial.begin(9600);

Wire.begin();

//initFD();
Serial.println("Setup and start capturing");
//Serial.println(rc,3);

// read initial values
t1=millis(); //time when we take n th capacitance value
for (i=1; i<=10;i++) {
  ReadCapacitanceVector(c);
  cap11 = c[1];
  cap22 = c[3];
  cap33 = c[4];
}
// PrintCapacitanceVector(c);
Serial.print("Initial values: ");
PrintCapacitances(cap11,cap22,cap33);
// here wait some time
delay(50);
```



```
cinit1 = cap11; // these are the initial zero values
cinit2 = cap22;
cinit3 = cap33;

// set the desired change in capacitance
dC = 0.03; // desired capacitance change
ctarget1 = cinit1+dC;
ctarget2 = cinit2+dC;
ctarget3 = cinit3+dC;
}

// run the loop

void loop()
{
  void ReadCapacitanceVector( float *a);
  void PrintCapacitanceVector( float *a);
  void PrintCapacitances( float a, float b, float c);
  void PrintSlidingVector( float a, float b, float c);

  // float c[5], cap11,cap22,cap33,capn11,capn22,capn33;
  float capn11,capn22,capn33;

  // Serial.println("new set of Capacitance");
  float t2=millis(); //time when we take n th capacitance value
  ReadCapacitanceVector(c);
  capn11 = c[1];
```

```

capn22 = c[3];
capn33 = c[4];
PrintCapacitanceVector(c);
//PrintCapacitances(capn11,capn22,capn33);
// PrintCapacitanceVector(c);
// delay(5000);
float dt=t2-t1;
float s1=(0.1*((capn11-cap11)/dt))+(capn11-ctarget1); //sliding plane 1

float s2=(0.1*((cap11+cap22-capn11-capn22)/dt))+(cap11+cap22-ctarget1-ctarget2);
//sliding plane 2

float s3=(0.1*((cap11+cap22+cap33-capn11-capn22-capn33)/dt))+(cap11+cap22+cap33-
ctarget1-ctarget2-ctarget3); //sliding plane 3

//float s3 = +1;

//float s2=(0.1*((capn22-cap22)/dt))+(capn22-ctarget2); //sliding plane 1
PrintSlidingVector(s1,s2,s3);

// control first spring
if (s1<0) digitalWrite (3,HIGH);
else digitalWrite (3,LOW);
// control second spring
if (s2<0) digitalWrite (4,HIGH);
else digitalWrite (4,LOW);
// control third spring
if (s3<0) digitalWrite (5,HIGH);
else digitalWrite (5,LOW);
// Serial.println("end of one set");
//delay(1000);

```

```

t1 = t2;
cap11 = capn11;
cap22 = capn22;
cap33 = capn33;
delay(1);
}

```

```

void ReadCapacitanceVector(float *(capvec) )

```

```

{

float d=pow(2,19);

write2byte(0x0C,0x05,0x80);
write2byte(0x08,0x1C,0x00);// select first channel
uint32_t cap1 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); //(n-1)th capacitance of
cap1
delay(20);
write2byte(0x0C,0x05,0x80);
write2byte(0x08,0x1C,0x00);// select first channel
uint32_t cap2 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); //(n-1)th capacitance of
cap1
delay(20);
write2byte(0x0C,0x05,0x80);
write2byte(0x08,0x1C,0x00);// select first channel
uint32_t cap3 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); //(n-1)th capacitance of
cap1
delay(20);

```

```
write2byte(0x0C,0x05,0x80);  
write2byte(0x08,0x1C,0x00); // select first channel  
uint32_t cap4 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of  
cap1  
delay(20);  
write2byte(0x0C,0x05,0x80);  
write2byte(0x08,0x1C,0x00); // select first channel  
uint32_t cap5 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of  
cap1  
delay(20);  
write2byte(0x0C,0x05,0x80);  
write2byte(0x08,0x1C,0x00); // select first channel  
uint32_t cap6 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of  
cap1  
delay(20);  
write2byte(0x0C,0x05,0x80);  
write2byte(0x08,0x1C,0x00); // select first channel  
uint32_t cap7 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of  
cap1  
delay(20);  
write2byte(0x0C,0x05,0x80);  
write2byte(0x08,0x1C,0x00); // select first channel  
uint32_t cap8 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of  
cap1  
delay(20);  
write2byte(0x0C,0x05,0x80);  
write2byte(0x08,0x1C,0x00); // select first channel  
uint32_t cap9 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of  
cap1
```

```
delay(20);

write2byte(0x0C,0x05,0x80);

write2byte(0x08,0x1C,0x00); // select first channel

uint32_t cap0 = (((read2byte(0x00)<<16)+read2byte(0x01))>>8); // (n-1)th capacitance of
cap1

delay(20);

//Serial.println(cap_mux,BIN);

float cap11=cap1/d; // capacitance after conversion in picofarad

float cap12=cap2/d;

float cap13=cap3/d;

float cap14=cap4/d;

float cap15=cap5/d;

float cap16=cap6/d;

float cap17=cap7/d;

float cap18=cap8/d;

float cap19=cap9/d;

float cap21=cap0/d;

float capvect =
(cap11+cap12+cap13+cap14+cap15+cap16+cap17+cap18+cap19+cap21)/10;

capvec[1]= capvect-cap11;

}
```

```
void PrintCapacitanceVector(float *(capvec) )
{
    char str[20];
    sprintf(str,"%+7.5f",capvec[1]);
    Serial.print("c = ");
    // Serial.println(capvec[1],5);
    Serial.println(str);
}
```

```
void PrintCapacitances(float a, float b, float c)
{
    Serial.print("c = (");
    Serial.print(a,5);
    Serial.print(",");
    Serial.print(b,5);
    Serial.print(",");
    Serial.print(c,5);
    Serial.println(")");
}
```

```
void PrintSlidingVector(float a, float b, float c)
{
    //Serial.print("S = (");
    //Serial.print(a,5);
    //Serial.print(",");
    //Serial.print(b,5);
```

```
    //Serial.print(",");  
    //Serial.print(c,5);  
    //Serial.println("");  
}
```

```
uint16_t read2byte(uint8_t address) {
```

```
    Wire.beginTransmission(FD);  
    Wire.write(address);  
    Wire.endTransmission();  
    Wire.requestFrom(FD, 2);  
    uint8_t msb = Wire.read();  
    uint8_t lsb = Wire.read();  
    //data = (data<<8)+Wire.read();  
    uint16_t shift=msb<<8;  
    uint16_t data= shift|lsb;  
    return data;  
    //delay(10000);  
}
```

```
void write2byte(uint8_t address1,uint8_t msb1,uint8_t lsb1) {
```

```
    Wire.beginTransmission(FD);  
    Wire.write(address1);  
    Wire.write(msb1);  
    Wire.write(lsb1);
```

```

Wire.endTransmission();

//delay(1);

}

```

A2. Microcontroller code for piezoelectric lens control circuit

Code for control circuit to continuously ramp up and down the voltage for piezoelectric

actuator lens

```

unsigned long time0;
unsigned long time1;
float pwm_min;
float pwm_max;
float pwm_step;
float pwm;
int Nsteps;
float time_step;
float period;
float ton;
float toff;
float rate;
float cycle;
int state;
int up;
int down;
int driver;
int positive_pwm = 6;
int negative_pwm = 5;
int sync_pwm = 9;
int positive_enable = 8;
int negative_enable = 7;
int positive = 1;
int negative = 0;

void ramp_up() // increase pwm step if reached time step
{
    pwm = pwm + pwm_step;
    ton = period*pwm;
    toff = period*(1.0-pwm);
}

```



```

void ramp_down() // decrease pwm step if reached time step
{
    pwm = pwm - pwm_step;
    ton = period*pwm;
    toff = period*(1.0-pwm);
}

void ramp_reset() // decrease pwm step if reached time step
{
    pwm = pwm_min;
    ton = period*pwm;
    toff = period*(1.0-pwm);
}

void setup() {
    up = 1;
    down = 0;
    pinMode(13, OUTPUT);
    pinMode(sync_pwm,OUTPUT);

    pinMode(positive_pwm, OUTPUT);
    pinMode(positive_enable, OUTPUT);
    pinMode(negative_pwm, OUTPUT);
    pinMode(negative_enable, OUTPUT);

    time0 = millis();
    pwm_min = 0.001; // minimum duty cycle
    pwm_max = 0.999; // maxium duty cycle
    Nsteps = 70; // number of steps in ramp
    pwm_step = (pwm_max-pwm_min)/(Nsteps-1); // this is the increment in the duty cycle
    cycle = 0.01; // cycle of ramp time in seconds
    pwm = pwm_min; // start ramp at minimum
    time_step = cycle/Nsteps*1000; // time in each step in milliseconds
    period = 2000.0; // period of repeating duty cycle signal in us
    ton = period*pwm; // this is the high time
    toff = period*(1.0-pwm); // this is the low time

    // pin enabling signals
    digitalWrite(positive_pwm, LOW);
    digitalWrite(positive_enable, LOW);
    digitalWrite(negative_pwm, LOW);
    digitalWrite(negative_enable, LOW); // everything is off
    // delay (10*1000);

    // define intial state

```

```

state = up;
driver = positive;
//
// enable positive driver
//
digitalWrite(negative_enable, HIGH); // enable positive driver
digitalWrite(positive_enable, HIGH); // enable negative driver

time0 = millis(); // starting time
}

// the loop function runs over and over again forever rmping UP
void loop() {
  if ( driver == positive ) {
    digitalWrite(positive_pwm, HIGH); // turn the LED on (HIGH is the voltage level)
    digitalWrite(sync_pwm, HIGH); // turn the LED on (HIGH is the voltage level)
    delayMicroseconds(ton); // wait for ton
    digitalWrite(positive_pwm, LOW); // turn the LED off (LOW is the voltage level)
    digitalWrite(sync_pwm, LOW); // turn the LED on (HIGH is the voltage level)
    delayMicroseconds(toff); // wait for a second
  } else if (driver == negative ) {
    digitalWrite(negative_pwm, HIGH); // turn the LED on (HIGH is the voltage level)
    digitalWrite(sync_pwm, HIGH); // turn the LED on (HIGH is the voltage level)
    delayMicroseconds(ton); // wait for ton
    digitalWrite(negative_pwm, LOW); // turn the LED off (LOW is the voltage level)
    digitalWrite(sync_pwm, LOW); // turn the LED on (HIGH is the voltage level)
    delayMicroseconds(toff); // wait for a second
  }
}

time1 = millis();
if (time1 > time0 + time_step) { // do changes here
  if ( state == up ) {
    ramp_up();
  } else if (state == down) {
    ramp_down();
  }
}
// reverse states when maximum or minumum are reached
if (pwm >= pwm_max ) state = down;
if (pwm <= pwm_min ) {
  state = up;
  // flip driver if reached minimum
  if ( driver == positive) {
    driver = negative;
  } else {
    driver = positive;
  }
}
}

```

```

    time0 = time1;
  }
}

```

A3. Code for curve fitting of deflection data

Python code for measuring membrane Young's modulus and initial tension by fitting

deflection data

```

from lmfit import minimize, Minimizer, Parameters, Parameter, report_fit
import numpy as np
from scipy.special import i1,i0

# create data to be fitted
#x = np.linspace(0, 15, 301)
#data = (5. * np.sin(2 * x - 0.1) * np.exp(-x*x*0.025) +
#  np.random.normal(size=len(x), scale=0.2) )

data = np.loadtxt('deflection_data.txt')
x = data[:, 0]
y = data[:, 1]
q = x[:]
hdata = y[:] * 1.0e-3

# define objective function: returns the array to be minimized
def nondeflcur(params, q, hdata):
    """ model circular plate subtract data """
    eps0 = params['eps0'] #pretension
    E = params['E'] #young's modulus
    tt = params['tt'] #thickness of the membrane
    rt = params['rt'] #radius of the membrane
    mu = params['mu'] #Poisson's ratio
    D = E*tt*tt*tt/12.0/(1.0-mu*mu)
    T = eps0*E*tt
    M = (56.0*tt*tt+42.0*rt*rt*(1.0+mu)*eps0)/(23.0+14.0*mu-9.0*mu*mu)

    N = 21.0*q*rt*rt*rt*rt*(1.0-mu)/(46.0+28.0*mu-18.0*mu*mu)/E/tt
    ## beta = rt*np.sqrt(T/D)
    ## model = q*rt*rt/4.0/T*(1.0+2.0/beta/i1(beta)*(i0(0)-i0(beta)))
    Q = (-27.0*N+np.sqrt(729.0*N*N+108.0*M*M*M))/2.0
    Q1 = np.power(Q,1.0/3.0)
    P = Q1-3.0*M/Q1
    model = -1/3.0*P

```

```
return model - hdata

# create a set of Parameters
params = Parameters()
params.add('eps0', value= 1.0e-3, min=1.0e-4, max=6.0e-2, vary=True)
params.add('E', value= 1.0e6, min=0.5e6, max=0.9e6, vary=True)
params.add('tt', value= 1.5e-3, vary=False)
params.add('rt', value= 18.0e-3, vary=False)
params.add('mu', value= 0.49, vary=False)
#params.add('mu', value= 0.2, min=0.1, max=0.4, vary=True )

# do fit, here with leastsq model
minner = Minimizer(nondeflcur, params, fcn_args=(q, hdata))
kws = {'options': {'maxiter':100}}
result = minner.minimize()

# calculate final result
final = hdata + result.residual

# write error report
report_fit(result)

# try to plot results
try:
    import pylab
    pylab.plot(q, hdata, 'k+')
    pylab.plot(q, final, 'r')
    pylab.show()
except:
    pass
```

A4. Python code for calculating membrane deflection

```

import math
import pylab
import numpy as np
import matplotlib.pyplot as plt

import opticspy
from scipy.special import *

def sumf(x):
    sum = 0
    fl = 1
    for i in range(0,50):
        p = 2*i+1
        sum = sum + fl*2.0/p/math.pi*math.cos(2.0*math.pi*x*p)*2
        fl = -fl
        #print sum
    sum = sum + 1.0
    return(sum)

def comaT(r,th,i_strain,E,R,rho,g,t):
    n = r/R
    f1 = (1.0-n*n)*n*math.cos(th)
    T = E*i_strain*t
    a0 = rho*g*R*R*R/8.0/T
    sum = a0*f1
    return(sum)

def comaD(r,th,E,R,rho,g,t):
    n = r/R
    f1 = (1.0-n*n)*(1.0-n*n)*n*math.cos(th)
    D = E*t*t*t/12.0
    a0 = rho*g*R*R*R*R/R/192.0/D
    sum = a0*f1
    return(sum)

def comaEx(r,th,i_strain,E,R,rho,g,t):
    n = r/R
    D = E*t*t*t/12.0
    T = i_strain*E*t
    b = R*math.sqrt(T/D)
    # print b
    f1 = math.cos(th)*(n-n*n*n+2.0*i1(b))/((1.0+1.0)*i1(b)-b*i0(b))*(n-i1(b*n)/i1(b))

```

```

a0 = rho*g*R*R*R/8.0/T
sum = a0*f1
return(sum)

f = open('output1','w')

th = 0.0
xmin = -12.0e-3
xmax = 12.0e-3
N = 200
rho = 1000.0
g = 9.8
t = 1.5e-3
R = 12.0e-3
eps_i = 0.01
E = 1.0e6
#eps_i1=0.03

dx = (xmax-xmin)/(N-1)

x = xmin
for i in range(0,N):
    y = comaT(x,th,eps_i,E,R,rho,g,t)
    v = comaD(x,th,E,R,rho,g,t)
    w = comaEx(x,th,eps_i,E,R,rho,g,t)
    print x,y,v,w
    f.write(str(x*1.0e3))
    f.write(" ")
    f.write(str(y*1.0e3))
    f.write(" ")
    f.write(str(v*1.0e3))
    f.write(" ")
    f.write(str(w*1.0e3))
    f.write('\n')
    x = x + dx
f.close()

q = 0.7

x, y, v, w = np.loadtxt('output1', delimiter=' ', unpack=True)

s = []
##for i in range(0,len(x)):
##    s1 = q*y[i]+(1.0-q)*z[i]
##    s1 = 1.0/(q/y[i]+(1.0-q)/z[i]+.0001)

```

```

## s.append(s1)

#plt.plot(x,y)
##
plt.plot(x,v)

plt.plot(x,w)

fig = plt.figure()

plt.show()
#Z = opticspy.spherical_surf(200)
#fitlist,C1
opticspy.zernike.fitting(w,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
#C1.zernikesurface(zlim=[-1,2])

```

=

A5. Code for calculating optical power and aberration with applied force

```

from solid import *
from scad import *
import math

import subprocess
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter

import numpy.ma as ma
import numpy as np
import sys
import os

import re
import shutil
import time
#import shlex
import opticspy

from scipy.interpolate import griddata
from easyprocess import EasyProcess
from getfem import *

```

```

#
# this is the library of modules
#

import rccalc_lib
rccalc_lib = reload(rccalc_lib)
#
# Main program
#
#####
#
# geometry description
#

number_of_ribs = 1

plate_radius = 15.0 # mm
plate_thickness = 1.3 # mm
rib_width = 0 # mm

rib_thickness = 0 # mm

rib_length = 0 * plate_radius

rib_radius = 0 * plate_radius

# rib_radius = 1.2*rib_radius

rib_boss_length = 0.005 * rib_length

# rib_boss_length = 0.1*rib_length

rib_boss_width = 1.0 * rib_width
flare_len = 0
flare_center_r = 0
flare_angle = 0

#####
# optical parameters
#
refractive_index = 1.47
wavelength = 0.59e-6
vignetting_radius = 12.0 # mm in sensor

```



```
#####
# material properties
#
matname = 'PDMS'
E = 1e3 # PDMS Young modulus (Sylgard 184)
Nu = 0.49 # PDMS Poisson's ratio

#####
# loading force conditions
#
commonfactor = 1.0
magfactor2 = 1.0

#

weight1 = 135.0 # uniform actuator force in gr

#
# piston pressures (for weight1 gr) in mN/mm^2

ppiston_uni = -weight1 * commonfactor * 1.0e-3 * 9.81 / (math.pi
    * plate_radius * plate_radius) * 1.0e3

ppiston_coma = 0.0

#
# calculate the density pressure due to radius of glycerol
# need pressure gradient parameter p/r (see Timoshenko,
# Theory of plates and shells, pg. 285)
#
rhog = 1260.0 # glycerol density in kg/m^3
pres_gly = -rhog * 9.81 * plate_radius * 1.0e-3 * 1.0e-6 * 1.0e3 # in mN/mm^2
pgmax = commonfactor * magfactor2 * pres_gly # maximum pressure difference to center

a0_coma = pgmax / plate_radius # coefficient for pressure gradient
a0_uni = 0.0 # coefficient for uniform pressure

liquid_weight = rhog*plate_radius*math.pi*plate_radius*plate_radius*1e-6

#####
#

solfolder = 'solution_folder'
fileprefix = 'rplate'
```

```

#
run_aborted = False

#
# make the plate here
#

rib_plus_boss = []
plate = cylinder(r=plate_radius, h=plate_thickness, center=True,
                 segments=60)
rib = cube([rib_length, rib_width, rib_thickness], center=True)
rib_boss = cube([rib_boss_length, rib_boss_width, rib_thickness],
               center=True)
rib_boss2 = cube([rib_boss_length / 2.0, rib_boss_width * 2.0,
                 rib_thickness], center=True)
flare = cube([flare_len, rib_width, rib_thickness])
flare = Translate(y=-rib_width / 2.0, z=-rib_thickness / 2.0)(flare)
flare_up = Rotate(z=flare_angle)(flare)
flare_down = Rotate(z=-flare_angle)(flare)
flare_up = Translate(x=flare_center_r - rib_radius - flare_len
                    / 2.0)(flare_up)
flare_down = Translate(x=flare_center_r - rib_radius - flare_len
                       / 2.0)(flare_down)

# rib_boss2 = Translate(x=rib_length/6.0) (rib_boss2)

rib_plus_boss.append(rib)

# rib_plus_boss.append(rib_boss)

rib_plus_boss.append(rib_boss2)

# rib_plus_boss.append(flare_up)
# rib_plus_boss.append(flare_down)

rib_plus_boss = Union>(*rib_plus_boss)
rib_plus_boss = Translate(x=rib_radius, z=(rib_thickness
    + plate_thickness) / 2.0
    - 0.05)(rib_plus_boss)
ribbed_plate = []
ribbed_plate.append(plate)
for i in range(0, number_of_ribs):
    _rib = Rotate(z=360.0 / number_of_ribs * i)(rib_plus_boss)
    ribbed_plate.append(_rib)

total_shape = Union>(*ribbed_plate)

```

```

#
# now make a folder
#
# clean it if it exists
#

if os.path.exists(solfolder):
    shutil.rmtree(solfolder, ignore_errors=True)

# now make it again

if not os.path.exists(solfolder):
    os.mkdir(solfolder)

# oscad_file = solfolder + "\\\" + fileprefix + ".scad"
# stl_file = solfolder + "\\\" + fileprefix+".stl"

oscad_file = fileprefix + '.scad'
stl_file = fileprefix + '.stl'

sys.stdout.write('printing to oscad file ... ')
total_shape.render(oscad_file)

# add the cylinder segment number option

add_segment_number_to_oscad(oscad_file, 32)
print 'done\n'

## if file exists, delete it ##

if os.path.isfile(stl_file):
    os.remove(stl_file)

sys.stdout.write('printing stl file ... ')
sys.stdout.flush()

# openscad to STL conversion

oscad_to_stl_cmd = 'openscad.exe -o ' + stl_file + ' ' + oscad_file

status = subprocess.call(oscad_to_stl_cmd, shell=True)
if status != 0:
    aborted_run = True
    print 'oscad to stl failed !'
    sys.exit('Stopping here')

```

```

print 'done\n'

tetgen_initial_time_tick = time.time()
print 'starting tetgen at ', \
    time.asctime(time.localtime(tetgen_initial_time_tick))
sys.stdout.write('meshing with tetgen ... ')
sys.stdout.flush()

# stl_to_mesh_tetgen_meshing_cmd = "tetgen.exe -pQgqa2.0 " + stl_file

stl_to_mesh_tetgen_meshing_cmd = 'tetgen.exe -pgqa2.0 ' + stl_file

# impose a maximum time limit for tetgen
pr = EasyProcess(stl_to_mesh_tetgen_meshing_cmd).call(timeout=10)
returncode = pr.return_code
stdoutdata = pr.stdout
stderrdata = pr.stderr

tetgen_end_time_tick = time.time()

status = returncode
print stdoutdata

if status != 0:
    aborted_run = True
    print 'stl to tetgen mesh failed !'
    print
    sys.exit('Stopping here')

print 'done\n'
sys.stdout.flush()
print 'ending tetgen at ', \
    time.asctime(time.localtime(tetgen_end_time_tick))

# converting .mesh to .msh file

fileposttet = fileprefix + '.1.mesh'
sys.stdout.write('converting mesh to msh file with gmsh ... ')
sys.stdout.flush()
mesh_to_msh_cmd = 'gmsh ' + fileposttet + ' -O -o ' + 'mesh3d.msh'
p = subprocess.call(mesh_to_msh_cmd, shell=True)
print 'done\n'

# here strip triangles

sys.stdout.write('stripping triangles from mesh ... ')

```

```

# strip_2D_elements_from_mesh("mesh3dproc.msh","meshsolid.msh")

strip_2D_elements_from_mesh('mesh3d.msh', 'meshsolid.msh')
print 'done\n'

# here convert back to .mesh format using gmsh

filepoststrip = 'meshsolid.msh'
meshfile = 'meshsolid.mesh'

sys.stdout.write('converting msh to mesh file with gmsh ... ')
sys.stdout.flush()
msh_to_mesh_cmd = 'gmsh ' + filepoststrip + ' -0 -o ' + 'meshsolid.mesh'
p = subprocess.call(msh_to_msh_cmd, shell=True)
print 'done\n'

# here convert mesh to abaqus to .inp format using gmsh

filepoststrip = 'meshsolid.msh'
abaqusmeshfile = 'meshsolid.inp'

sys.stdout.write('converting msh to inp file with gmsh ... ')
sys.stdout.flush()

# set tolerance to 1e-4 mm so it prints a float which calculix needs
# msh_to_inp_cmd = "gmsh " + filepoststrip + " -0 -tol 0.00001 -o " + "meshsolid_t.inp"

msh_to_inp_cmd = 'gmsh ' + filepoststrip + ' -0 -o ' + 'meshsolid_t.inp'
p = subprocess.call(msh_to_inp_cmd, shell=True)
print 'done\n'

sys.stdout.write('cleaning inp file ... ')
sys.stdout.flush()
clean_inp_file('meshsolid_t.inp', 'meshsolid.inp')
print 'done\n'

#####
#
# here read mesh with getfem interface to find boundary pts
#
#####

sys.stdout.write('importing mesh ... ')
m = Mesh('import', 'gmsh', 'meshsolid.msh')
print 'done!'

```

```

# first collect the mesh points

P = m.pts()
num_el = m.nbcvs() # this is the number of tetrahedra
print 'mesh has ', num_el, 'tetrahedra'

# find the centroid coordinates for all of the mesh points

centroids = find_solid_centroids(m)

#
#   boundary selection
#
# P[2] contains the z coordinate of the points
#
# anything z >= plate_thickness/2 belongs to top

ctop = P[2, :] - plate_thickness / 2.0 > -1.0e-5 * plate_thickness

# anything at z=-plate_thickness/2 is part of bottom

cbot = abs(P[2, :] + plate_thickness / 2.0) < 1.0e-5 * plate_thickness

# anything at x^2+y^2 >=r^2 is part of side
# all points from the faces must be recognized
# hence it must be on a band

R = (P[0, :] * P[0, :] + P[1, :] * P[1, :]) ** 0.5

# cside=(abs(R-plate_radius) < 0.015*plate_radius);

cside = abs(R[:] - plate_radius) < 0.05 * plate_radius

#
# now find bottom faces centroids
#

border = m.outer_faces()

#

pidtop = compress(ctop, range(0, m.nbpts()))
pidbot = compress(cbot, range(0, m.nbpts()))
pidside = compress(cside, range(0, m.nbpts()))

```

```

#

fside = m.faces_from_pid(pidside)

ftop = m.faces_from_pid(pidtop)
fbot = m.faces_from_pid(pidbot)

#

fnor = m.normal_of_faces(fside)
fnor1 = m.normal_of_faces(fbot)

#
# find the bottom facet centroid coordinates for all of the solids
# return(ct,ta,fac_pts)
#

(bottom_facet_centroids, area_facets, bot_el_pts, pcp) = \
    find_solid_bottom_facet_centroids1(m, fbot, -plate_thickness / 2.0,
    1.0e-5)

#
# Here identify and refine the edge BC
# to make sure that they are outer faces.
#

fside2 = []
fside1 = fside.tolist()
borderlist = border.tolist()

#
# correct the edge boundary
#

for index in range(0, len(fside1[0])):
    if abs(fnor[2, index]) < 0.1 and inlist(fside1[0], borderlist[0],
        index) == True:
        str1 = [fside[0, index], fside[1, index]]
        fside2.append(str1)

fside3 = array(fside2)
fside4 = fside3.transpose()

#
# here are the point IDs for the side boundary
#

```

```

side_pts_ids = m.pid_in_faces(fside4)

#

fnor2 = m.normal_of_faces(fside4)
fnor4 = m.normal_of_faces(fbot)

#
# Set the boundaries and multiple forces
#
# now get the points for each set
#

fix_side_point_ids = m.pid_in_faces(fside4)

#
# here we get the bottom elements concentrated forces
# first we get the points from every bottom face element
# and assign the corresponding 1/3 force
# from the centroid pressure force and element area
# and keep adding force values to account for pressure
# of adjacent elements
#

bot_pts = m.pid_in_faces(fbot)

#####
#
# first find the coma deflection
#

print 'coma loading ...'
print 'ppiston =', ppiston_coma, 'mN/mm^2, a0*rad =', a0_coma * plate_radius, \
      'mN/mm^2'

#
# find the pressure distribution and concentrated loads
#

(press_centroid, pmsh) = pressure_at_centroids(bottom_facet_centroids,
      ppiston_coma, a0_coma)
force_centroid = force_at_centroids(press_centroid,
      bottom_facet_centroids, area_facets)
plot_pressure(pmsh, 20, plate_radius)

```



```

(bottom_pts_ids, bottom_pts_cload) = find_solid_bottom_facet_cloads(m,
    fbot, force_centroid)
cmsh = force_at_bottom_points(m, bottom_pts_ids, bottom_pts_cload)
#plot_cload(cmsh, 20, plate_radius)

#
#   now assemble the calculix input deck and files
#
calculix_input_deck = 'calculix_coma_run.inp'
sys.stdout.write('assembling coma input deck ... ')
sys.stdout.flush()
assemble_input_deck(
    num_el,
    matname,
    E,
    Nu,
    calculix_input_deck,
    abaqusmeshfile,
    side_pts_ids,
    bottom_pts_ids,
    bottom_pts_cload,
)
print 'done\n'

#
# here run the solver
#

calculix_jobname = 'calculix_coma_run'
sys.stdout.write('running calculix ... ')
sys.stdout.flush()
calculix_solve_cmd = 'ccx -i ' + calculix_jobname

p = subprocess.Popen(calculix_solve_cmd, shell=True,
    stdout=subprocess.PIPE)
result = p.communicate()[0]
print 'done\n'
print result

#
# find the extreme dz displacement for uniform pressure
#

(dzmin_coma, dzmax_coma) = calculix_extreme_dz('calculix_coma_run.dat')

(nn,zdata_coma) = calculix_dz('calculix_coma_run.dat')
(xcoor,ycoor,coma_surf_dz) = surface_dz(P,bot_pts,zdata_coma)

```

```

# plotting coma deflection

plot_dz(xcoor,ycoor,coma_surf_dz,20.0,plate_radius)

# now make the Zernike coefficient fit
# first construct a uniform matrix for the fit

ddz= RadiallyNormalizedMatrix(xcoor,ycoor,coma_surf_dz,plate_radius,120)

# wavefront displacement in microns
ddwf=
RadiallyNormalizedWavefrontMatrix(xcoor,ycoor,coma_surf_dz,plate_radius,120,refractive_index)
ddwf_vig=
VignettedRadiallyNormalizedWavefrontMatrix(xcoor,ycoor,coma_surf_dz,plate_radius,vignetting_radius,120,refractive_index)

#Begin Fitting
coma_fitlist,C1=
opticspy.zernike.fitting(ddwf,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
C1.zernikesurface(zlim=[-1,2])

#Begin Vignetted Fitting
coma_fitlist_vig,C1v=
opticspy.zernike.fitting(ddwf_vig,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
C1v.zernikesurface(zlim=[-1,2])

fig = plt.figure()
ax = fig.gca(projection='3d')
#X = np.arange(-5, 5, 0.25)
#Y = np.arange(-5, 5, 0.25)
#X, Y = np.meshgrid(X, Y)
#R = np.sqrt(X**2 + Y**2)
#Z = np.sin(R)
Xa = np.arange(-1, 1, 2.0/120)
Ya = np.arange(-1, 1, 2.0/120)
#Y = np.arange(-5, 5, 0.25)
#X, Y = np.meshgrid(X, Y)
#R = np.sqrt(X**2 + Y**2)
#Z = np.sin(R)
Za = ddwf
surf = ax.plot_surface(Xa, Ya, Za, rstride=1, cstride=1, cmap=cm.coolwarm,
                      linewidth=0, antialiased=False)
plt.show()

#####

```

```

#
#   Next find the uniform deflection
#
#####

print 'uniform loading ...'
print 'ppiston =', ppiston_uni, 'mN/mm^2, a0*rad =', a0_uni * plate_radius, \
      'mN/mm^2'

#
#   find the pressure distribution and concentrated loads
#

(press_centroid, pmsh) = pressure_at_centroids(bottom_facet_centroids,
      ppiston_uni, a0_uni)
force_centroid = force_at_centroids(press_centroid,
      bottom_facet_centroids, area_facets)
plot_pressure(pmsh, 20, plate_radius)

#

(bottom_pts_ids, bottom_pts_cload) = find_solid_bottom_facet_cloads(m,
      fbot, force_centroid)
cmsh = force_at_bottom_points(m, bottom_pts_ids, bottom_pts_cload)
#plot_cload(cmsh, 20, plate_radius)

#
#   now assemble the calculix input deck and files
#
#

calculix_input_deck = 'calculix_uni_run.inp'
sys.stdout.write('assembling uniform pressure input deck ... ')
sys.stdout.flush()
assemble_input_deck(
    num_el,
    matname,
    E,
    Nu,
    calculix_input_deck,
    abaqusmeshfile,
    side_pts_ids,
    bottom_pts_ids,
    bottom_pts_cload,
)
print 'done\n'

```

```

#
# here run the solver
#

calculix_jobname = 'calculix_uni_run'
sys.stdout.write('running calculix ... ')
sys.stdout.flush()
calculix_solve_cmd = 'ccx -i ' + calculix_jobname

#

p = subprocess.Popen(calculix_solve_cmd, shell=True,
                    stdout=subprocess.PIPE)
result = p.communicate()[0]
print 'done\n'

print result

# this is the deflection ratio (coma/uni) for 1 mm plate
coma_uniratio_1mm = 0.0396824882006

# find the extreme dz displacement for uniform pressure

(dzmin_p, dzmax_p) = calculix_extreme_dz('calculix_uni_run.dat')

(nn,zdata) = calculix_dz('calculix_uni_run.dat')
(xcoor,ycoor,uni_surf_dz) = surface_dz(P,bot_pts,zdata)

plot_dz(xcoor,ycoor,uni_surf_dz,20.0,plate_radius)

# now make the fit
# first construct a uniform matrix for the fit

ddz= RadiallyNormalizedMatrix(xcoor,ycoor,uni_surf_dz,plate_radius,120)

# calculare wavefront in microns
ddwf = RadiallyNormalizedWavefrontMatrix(xcoor,ycoor,uni_surf_dz,plate_radius,120,refractive_index)
#Begin second Fitting
uni_fitlist,C2=
opticspy.zernike.fitting(ddwf,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
C2.zernikesurface(zlim=[-3,3])

lens_power = 4.0*sqrt(3.0)*uni_fitlist[4]*1.0e-6 /(plate_radius*plate_radius)/1.0e-6
lens_power2 = (refractive_index-1.0)*2.0*abs(dzmin_p)/(plate_radius*plate_radius-
dzmin_p*dzmin_p)/1.0e-3

```

```
coma_lens_power = 4.0*sqrt(3.0)*coma_fitlist[8]*1.0e-6/(plate_radius*plate_radius)/1.0e-6*4.0
```

```
print
print '*****'
print '*
print '*          Final Results          *
print '*
print '*****'
print 'unif. actuation force = ', weight1, "gr"
print 'liquid column force = ', liquid_weight, "gr"
print '*****'
print 'plate radius = ', plate_radius, "mm"
print 'plate thickness = ', plate_thickness, "mm"
print 'vignetting radius = ', vignetting_radius, "mm"
print 'extreme coma dz = ', dzmin_coma, ", ", dzmax_coma, "mm"
print 'extreme uniform dz = ', dzmin_p, ", ", dzmax_p, "mm"

avgc = (abs(dzmin_coma) + abs(dzmax_coma)) / 1.0
maxdzp = abs(dzmin_p)
#print
ratio_coma = avgc / maxdzp
#####
print 'coma coeff in microns = ', coma_fitlist[8], "microns"
print 'vig. coma coeff = ', coma_fitlist_vig[8], "microns"
print 'coma cont. ratio = ', ratio_coma
print 'coma decrease factor = ', ratio_coma / coma_uniratio_1mm * 100.0, '%'
print 'uniform lens power = ', lens_power, 'diopters'
print 'uniform lens power2 = ', lens_power2, 'diopters'
print 'appr. coma lens power = ', coma_lens_power, 'diopters'
print 'max coma lens power = ', 2.0*coma_lens_power, 'diopters'
print 'coma/uni power = ', 2.0*coma_lens_power/lens_power

print '*****'
```

```
sys.exit('Stopping here')
```

```
print 'You can view the tripod with (for example) mayavi:'
print 'mayavi -d ./rplate.vtk -f WarpVector -m BandedSurfaceMap'
print 'or'
print 'mayavi2 -d rplate.vtk -f WarpScalar -m Surface'
print 'or'
print 'gmsht rplate.pos'
```

A6. Python simulation code for front membrane with glass ribs

Code for estimating coma and lens optical power range for PDMS membrane with glass

ribs embedded in it

```

from solid import *
from scad import *
import math

import subprocess
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
from matplotlib.ticker import LinearLocator, FormatStrFormatter

import numpy.ma as ma
import numpy as np
import sys
import os

import re
import shutil
import time
#import shlex
import opticspy

from scipy.interpolate import griddata
from easyprocess import EasyProcess
from getfem import *

#
#  this is the library of modules
#

import rccalc_lib
rccalc_lib = reload(rccalc_lib)
#
#  Main program
#
#####
#
#      geometry description
#

number_of_ribs = 16

```

```

plate_radius = 15.0 # mm
plate_thickness = 1.3 # mm
rib_width = 0.9 # mm

rib_thickness = 1.0 # mm

rib_length = 0.585 * plate_radius

rib_radius = math.sqrt(1.0 / 5.0) * plate_radius

# rib_radius = 1.2*rib_radius

rib_boss_length = 0.65 * rib_length

# rib_boss_length = 0.1*rib_length

rib_boss_width = 1.0 * rib_width
flare_len = plate_radius - rib_length / 2.0 - rib_radius
flare_center_r = plate_radius - flare_len / 2.0
flare_angle = 360.0 / number_of_ribs / 2.0 * 3.0

#####
# optical parameters
#
refractive_index = 1.47
wavelength = 0.59e-6
vignetting_radius = 12.0 # mm in sensor

#####
# material properties
#
matname = 'PDMS'
E = 1e3 # PDMS Young modulus (Sylgard 184)
Nu = 0.49 # PDMS Poisson's ratio

#####
# loading force conditions
#
commonfactor = 1.0
magfactor2 = 1.0

#

```

```

weight1 = 115.0 # uniform actuator force in gr

#
# piston pressures (for weight1 gr) in mN/mm^2

ppiston_uni = -weight1 * commonfactor * 1.0e-3 * 9.81 / (math.pi
    * plate_radius * plate_radius) * 1.0e3

ppiston_coma = 0.0

#
# calculate the density pressure due to radius of glycerol
# need pressure gradient parameter p/r (see Timoshenko,
# Theory of plates and shells, pg. 285)
#
rhog = 1260.0 # glycerol density in kg/m^3
pres_gly = -rhog * 9.81 * plate_radius * 1.0e-3 * 1.0e-6 * 1.0e3 # in mN/mm^2
pgmax = commonfactor * magfactor2 * pres_gly # maximum pressure difference to center

a0_coma = pgmax / plate_radius # coefficient for pressure gradient
a0_uni = 0.0 # coefficient for uniform pressure

liquid_weight = rhog*plate_radius*math.pi*plate_radius*plate_radius*1e-6

#####
#

solfolder = 'solution_folder'
fileprefix = 'rplate'

#
run_aborted = False

#
# make the plate here
#

rib_plus_boss = []
plate = cylinder(r=plate_radius, h=plate_thickness, center=True,
    segments=60)
rib = cube([rib_length, rib_width, rib_thickness], center=True)
rib_boss = cube([rib_boss_length, rib_boss_width, rib_thickness],
    center=True)
rib_boss2 = cube([rib_boss_length / 2.0, rib_boss_width * 2.0,
    rib_thickness], center=True)

```



```

flare = cube([flare_len, rib_width, rib_thickness])
flare = Translate(y=-rib_width / 2.0, z=-rib_thickness / 2.0)(flare)
flare_up = Rotate(z=flare_angle)(flare)
flare_down = Rotate(z=-flare_angle)(flare)
flare_up = Translate(x=flare_center_r - rib_radius - flare_len
                    / 2.0)(flare_up)
flare_down = Translate(x=flare_center_r - rib_radius - flare_len
                    / 2.0)(flare_down)

# rib_boss2 = Translate(x=rib_length/6.0) (rib_boss2)

rib_plus_boss.append(rib)

# rib_plus_boss.append(rib_boss)

rib_plus_boss.append(rib_boss2)

# rib_plus_boss.append(flare_up)
# rib_plus_boss.append(flare_down)

rib_plus_boss = Union>(*rib_plus_boss)
rib_plus_boss = Translate(x=rib_radius, z=(rib_thickness
    + plate_thickness) / 2.0
    - 0.05)(rib_plus_boss)
ribbed_plate = []
ribbed_plate.append(plate)
for i in range(0, number_of_ribs):
    _rib = Rotate(z=360.0 / number_of_ribs * i)(rib_plus_boss)
    ribbed_plate.append(_rib)

total_shape = Union>(*ribbed_plate)

#
# now make a folder
#
# clean it of it exists
#

if os.path.exists(solfolder):
    shutil.rmtree(solfolder, ignore_errors=True)

# now make it again

if not os.path.exists(solfolder):
    os.mkdir(solfolder)

```

```

# oscad_file = solfolder + "\\\" + fileprefix + ".scad"
# stl_file = solfolder + "\\\" + fileprefix+".stl"

oscad_file = fileprefix + '.scad'
stl_file = fileprefix + '.stl'

sys.stdout.write('printing to oscad file ... ')
total_shape.render(oscad_file)

# add the cylinder segment number option

add_segment_number_to_oscad(oscad_file, 32)
print 'done\n'

## if file exists, delete it ##

if os.path.isfile(stl_file):
    os.remove(stl_file)

sys.stdout.write('printing stl file ... ')
sys.stdout.flush()

# openscad to STL conversion

oscad_to_stl_cmd = 'openscad.exe -o ' + stl_file + ' ' + oscad_file

status = subprocess.call(oscad_to_stl_cmd, shell=True)
if status != 0:
    aborted_run = True
    print 'oscad to stl failed !'
    sys.exit('Stopping here')
print 'done\n'

tetgen_initial_time_tick = time.time()
print 'starting tetgen at ', \
    time.asctime(time.localtime(tetgen_initial_time_tick))
sys.stdout.write('meshing with tetgen ... ')
sys.stdout.flush()

# stl_to_mesh_tetgen_meshing_cmd = "tetgen.exe -pQgqa2.0 " + stl_file

stl_to_mesh_tetgen_meshing_cmd = 'tetgen.exe -pgqa2.0 ' + stl_file

# impose a maximum time limit for tetgen
pr = EasyProcess(stl_to_mesh_tetgen_meshing_cmd).call(timeout=10)
returncode = pr.return_code

```

```

stdoutdata = pr.stdout
stderrdata = pr.stderr

tetgen_end_time_tick = time.time()

status = returncode
print stdoutdata

if status != 0:
    aborted_run = True
    print 'stl to tetgen mesh failed !'
    print
    sys.exit('Stopping here')

print 'done\n'
sys.stdout.flush()
print 'ending tetgen at ', \
    time.asctime(time.localtime(tetgen_end_time_tick))

# converting .mesh to .msh file

fileposttet = fileprefix + '.1.mesh'
sys.stdout.write('converting mesh to msh file with gmsh ... ')
sys.stdout.flush()
mesh_to_msh_cmd = 'gmsh ' + fileposttet + ' -O -o ' + 'mesh3d.msh'
p = subprocess.call(mesh_to_msh_cmd, shell=True)
print 'done\n'

# here strip triangles

sys.stdout.write('stripping triangles from mesh ... ')

# strip_2D_elements_from_mesh("mesh3dproc.msh","meshsolid.msh")

strip_2D_elements_from_mesh('mesh3d.msh', 'meshsolid.msh')
print 'done\n'

# here convert back to .mesh format using gmsh

filepoststrip = 'meshsolid.msh'
meshfile = 'meshsolid.mesh'

sys.stdout.write('converting msh to mesh file with gmsh ... ')
sys.stdout.flush()
msh_to_mesh_cmd = 'gmsh ' + filepoststrip + ' -O -o ' + 'meshsolid.mesh'
p = subprocess.call(mesh_to_msh_cmd, shell=True)

```

```

print 'done\n'

# here convert mesh to abaqus to .inp format using gmsh

filepoststrip = 'meshsolid.msh'
abaqusmeshfile = 'meshsolid.inp'

sys.stdout.write('converting msh to inp file with gmsh ... ')
sys.stdout.flush()

# set tolerance to 1e-4 mm so it prints a float which calculix needs
# msh_to_inp_cmd = "gmsh " + filepoststrip + " -0 -tol 0.00001 -o " + "meshsolid_t.inp"

msh_to_inp_cmd = 'gmsh ' + filepoststrip + ' -0 -o ' + 'meshsolid_t.inp'
p = subprocess.call(msh_to_inp_cmd, shell=True)
print 'done\n'

sys.stdout.write('cleaning inp file ... ')
sys.stdout.flush()
clean_inp_file('meshsolid_t.inp', 'meshsolid.inp')
print 'done\n'

#####
#
# here read mesh with getfem interface to find boundary pts
#
#####

sys.stdout.write('importing mesh ... ')
m = Mesh('import', 'gmsh', 'meshsolid.msh')
print 'done!'

# first collect the mesh points

P = m.pts()
num_el = m.nbcvs() # this is the number of tetrahedra
print 'mesh has ', num_el, 'tetrahedra'

# find the centroid coordinates for all of the mesh points

centroids = find_solid_centroids(m)

#
# boundary selection
#
# P[2] contains the z coordinate of the points

```

```

#
# anything z >= plate_thickness/2 belongs to top

ctop = P[2, :] - plate_thickness / 2.0 > -1.0e-5 * plate_thickness

# anything at z=-plate_thickness/2 is part of bottom

cbot = abs(P[2, :] + plate_thickness / 2.0) < 1.0e-5 * plate_thickness

# anything at x^2+y^2 >=r^2 is part of side
# all points from the faces must be recognized
# hence it must be on a band

R = (P[0, :] * P[0, :] + P[1, :] * P[1, :]) ** 0.5

# cside=(abs(R-plate_radius) < 0.015*plate_radius);

cside = abs(R[:] - plate_radius) < 0.05 * plate_radius

#
# now find bottom faces centroids
#

border = m.outer_faces()

#

pidtop = compress(ctop, range(0, m.nbpts()))
pidbot = compress(cbot, range(0, m.nbpts()))
pidside = compress(cside, range(0, m.nbpts()))

#

fside = m.faces_from_pid(pidside)

ftop = m.faces_from_pid(pidtop)
fbot = m.faces_from_pid(pidbot)

#

fnor = m.normal_of_faces(fside)
fnor1 = m.normal_of_faces(fbot)

#
# find the bottom facet centroid coordinates for all of the solids
# return(ct,ta,fac_pts)

```

```

#

(bottom_facet_centroids, area_facets, bot_el_pts, pcp) = \
    find_solid_bottom_facet_centroids1(m, fbot, -plate_thickness / 2.0,
        1.0e-5)

#
# Here identify and refine the edge BC
# to make sure that they are outer faces.
#

fside2 = []
fside1 = fside.tolist()
borderlist = border.tolist()

#
# correct the edge boundary
#

for index in range(0, len(fside1[0])):
    if abs(fnor[2, index]) < 0.1 and inlist(fside1[0], borderlist[0],
        index) == True:
        str1 = [fside[0, index], fside[1, index]]
        fside2.append(str1)

fside3 = array(fside2)
fside4 = fside3.transpose()

#
# here are the point IDs for the side boundary
#

side_pts_ids = m.pid_in_faces(fside4)

#

fnor2 = m.normal_of_faces(fside4)
fnor4 = m.normal_of_faces(fbot)

#
# Set the boundaries and multiple forces
#
# now get the points for each set
#

fix_side_point_ids = m.pid_in_faces(fside4)

```

```

#
# here we get the bottom elements concentrated forces
# first we get the points from every bottom face element
# and assign the corresponding 1/3 force
# from the centroid pressure force and element area
# and keep adding force values to account for pressure
# of adjacent elements
#

bot_pts = m.pid_in_faces(fbot)

#####
#
# first find the coma deflection
#

print 'coma loading ...'
print 'ppiston =', ppiston_coma, 'mN/mm^2, a0*rad =', a0_coma * plate_radius, \
      'mN/mm^2'

#
# find the pressure distribution and concentrated loads
#

(press_centroid, pmsh) = pressure_at_centroids(bottom_facet_centroids,
      ppiston_coma, a0_coma)
force_centroid = force_at_centroids(press_centroid,
      bottom_facet_centroids, area_facets)
plot_pressure(pmsh, 20, plate_radius)

(bottom_pts_ids, bottom_pts_load) = find_solid_bottom_facet_loads(m,
      fbot, force_centroid)
cmsh = force_at_bottom_points(m, bottom_pts_ids, bottom_pts_load)
#plot_load(cmsh, 20, plate_radius)

#
# now assemble the calculix input deck and files
#
calculix_input_deck = 'calculix_coma_run.inp'
sys.stdout.write('assembling coma input deck ... ')
sys.stdout.flush()
assemble_input_deck(
    num_el,
    matname,
    E,
    Nu,
    calculix_input_deck,

```

```

    abaqusmeshfile,
    side_pts_ids,
    bottom_pts_ids,
    bottom_pts_cload,
    )
print 'done\n'

#
# here run the solver
#

calculix_jobname = 'calculix_coma_run'
sys.stdout.write('running calculix ... ')
sys.stdout.flush()
calculix_solve_cmd = 'ccx -i ' + calculix_jobname

p = subprocess.Popen(calculix_solve_cmd, shell=True,
                    stdout=subprocess.PIPE)
result = p.communicate()[0]
print 'done\n'
print result

#
# find the extreme dz displacement for uniform pressure
#

(dzmin_coma, dzmax_coma) = calculix_extreme_dz('calculix_coma_run.dat')

(nn,zdata_coma) = calculix_dz('calculix_coma_run.dat')
(xcoor,ycoor,coma_surf_dz) = surface_dz(P,bot_pts,zdata_coma)

# plotting coma deflection

plot_dz(xcoor,ycoor,coma_surf_dz,20.0,plate_radius)

# now make the Zernike coefficient fit
# first construct a uniform matrix for the fit

ddz= RadiallyNormalizedMatrix(xcoor,ycoor,coma_surf_dz,plate_radius,120)

# wavefront displacement in microns
ddwf=
RadiallyNormalizedWavefrontMatrix(xcoor,ycoor,coma_surf_dz,plate_radius,120,refractive_ind
ex)
ddwf_vig=
VignettedRadiallyNormalizedWavefrontMatrix(xcoor,ycoor,coma_surf_dz,plate_radius,vignettin

```



```

g_radius,120,refractive_index)

#Begin Fitting
coma_fitlist,C1=
opticspy.zernike.fitting(ddwf,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
C1.zernikesurface(zlim=[-1,2])

#Begin Vignetted Fitting
coma_fitlist_vig,C1v=
opticspy.zernike.fitting(ddwf_vig,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
C1v.zernikesurface(zlim=[-1,2])

fig = plt.figure()
ax = fig.gca(projection='3d')
#X = np.arange(-5, 5, 0.25)
#Y = np.arange(-5, 5, 0.25)
#X, Y = np.meshgrid(X, Y)
#R = np.sqrt(X**2 + Y**2)
#Z = np.sin(R)
Xa = np.arange(-1, 1, 2.0/120)
Ya = np.arange(-1, 1, 2.0/120)
#Y = np.arange(-5, 5, 0.25)
#X, Y = np.meshgrid(X, Y)
#R = np.sqrt(X**2 + Y**2)
#Z = np.sin(R)
Za = ddfw
surf = ax.plot_surface(Xa, Ya, Za, rstride=1, cstride=1, cmap=cm.coolwarm,
                      linewidth=0, antialiased=False)
plt.show()

#####
#
#   Next find the uniform deflection
#
#####

print 'uniform loading ...'
print 'ppiston =', ppiston_uni, 'mN/mm^2, a0*rad =', a0_uni * plate_radius, \
      'mN/mm^2'

#
#   find the pressure distribution and concentrated loads
#

(press_centroid, pmsb) = pressure_at_centroids(bottom_facet_centroids,
      ppiston_uni, a0_uni)
force_centroid = force_at_centroids(press_centroid,

```

```

        bottom_facet_centroids, area_facets)
plot_pressure(pmesh, 20,plate_radius)

#

(bottom_pts_ids, bottom_pts_cload) = find_solid_bottom_facet_cloads(m,
    fbot, force_centroid)
cmsh = force_at_bottom_points(m, bottom_pts_ids, bottom_pts_cload)
#plot_cload(cmsh, 20,plate_radius)

#
#   now assemble the calculix input deck and files
#
#

calculix_input_deck = 'calculix_uni_run.inp'
sys.stdout.write('assembling uniform pressure input deck ... ')
sys.stdout.flush()
assemble_input_deck(
    num_el,
    matname,
    E,
    Nu,
    calculix_input_deck,
    abaqumeshfile,
    side_pts_ids,
    bottom_pts_ids,
    bottom_pts_cload,
)
print 'done\n'

#
# here run the solver
#

calculix_jobname = 'calculix_uni_run'
sys.stdout.write('running calculix ... ')
sys.stdout.flush()
calculix_solve_cmd = 'ccx -i ' + calculix_jobname

#

p = subprocess.Popen(calculix_solve_cmd, shell=True,
    stdout=subprocess.PIPE)
result = p.communicate()[0]
print 'done\n'

```

```

print result

# this is the deflection ratio (coma/uni) for 1 mm plate
coma_uniratio_1mm = 0.0396824882006

# find the extreme dz displacement for uniform pressure

(dzmin_p, dzmax_p) = calculix_extreme_dz('calculix_uni_run.dat')

(nn,zdata) = calculix_dz('calculix_uni_run.dat')
(xcoor,ycoor,uni_surf_dz) = surface_dz(P,bot_pts,zdata)

plot_dz(xcoor,ycoor,uni_surf_dz,20.0,plate_radius)

# now make the fit
# first construct a uniform matrix for the fit

ddz= RadiallyNormalizedMatrix(xcoor,ycoor,uni_surf_dz,plate_radius,120)

# calculate wavefront in microns
ddwf=
RadiallyNormalizedWavefrontMatrix(xcoor,ycoor,uni_surf_dz,plate_radius,120,refractive_index)
#Begin second Fitting
uni_fitlist,C2=
opticspy.zernike.fitting(ddwf,12,remain2D=1,remain3D=1,barchart=1,interferogram=1)
C2.zernikesurface(zlim=[-3,3])

lens_power = 4.0*sqrt(3.0)*uni_fitlist[4]*1.0e-6/(plate_radius*plate_radius)/1.0e-6
lens_power2      =      (refractive_index-1.0)*2.0*abs(dzmin_p)/(plate_radius*plate_radius-
dzmin_p*dzmin_p)/1.0e-3
coma_lens_power  =  4.0*sqrt(3.0)*coma_fitlist[8]*1.0e-6/(plate_radius*plate_radius)/1.0e-
6*4.0

print
print '*****'
print '*
          *'
print '*          Final Results          *'
print '*          *'
print '*****'
print 'unif. actuation force = ', weight1, "gr"
print 'liquid column force = ', liquid_weight, "gr"
print '*****'
print 'plate radius      = ', plate_radius, "mm"
print 'plate thickness   = ', plate_thickness, "mm"
print 'vignetting radius  = ', vignetting_radius, "mm"
print 'extreme coma dz    = ', dzmin_coma, ", ", dzmax_coma, "mm"

```

```

print 'extreme uniform dz  = ', dzmin_p, ",", dzmax_p, "mm"

avgc = (abs(dzmin_coma) + abs(dzmax_coma)) / 1.0
maxdzp = abs(dzmin_p)
#print
ratio_coma = avgc / maxdzp
#####
print 'coma coeff in microns = ', coma_fitlist[8], "microns"
print 'vig. coma coeff      = ', coma_fitlist_vig[8], "microns"
print 'coma cont. ratio    = ', ratio_coma
print 'coma decrease factor = ', ratio_coma / coma_uniratio_1mm * 100.0, '%'
print 'uniform lens power  = ', lens_power, 'diopters'
print 'uniform lens power2 = ', lens_power2, 'diopters'
print 'appr. coma lens power = ', coma_lens_power, 'diopters'
print 'max coma lens power  = ', 2.0*coma_lens_power, 'diopters'
print 'coma/uni power      = ', 2.0*coma_lens_power/lens_power

print '*****'

sys.exit('Stopping here')

print 'You can view the tripod with (for example) mayavi:'
print 'mayavi -d ./rplate.vtk -f WarpVector -m BandedSurfaceMap'
print 'or'
print 'mayavi2 -d rplate.vtk -f WarpScalar -m Surface'
print 'or'
print 'gmsht rplate.pos'

```

APPENDIX B

SOFTWARE INSTALLATION PROCEDURE FOR SIMULATION

1. Install Python XY, full distribution
 - a. <http://python-xy.github.io/>
2. Download gmsh from <http://gmsh.info>
 - a. Add the path of the .exe file to the “Path” Environment Variable by editing the Environment Variables in the Advanced System Settings of the computer.
3. Install openSCAD development version from www.openscad.org/downloads.html
 - a. Add the path of the .exe file to the “Path” Environment Variable by editing the Environment Variables in the Advanced System Settings of the computer.
4. Go to C:\Python27\Scripts through the command prompt to install various modules:
 - a. Type: pip install pygmsh
 - b. Type: pip install pysolid
 - c. Type: pip install pycad
 - d. Type: pip install opticspy
 - e. Type: pip install easyprocess
 - f. Type: pip install trimesh
5. Download getfem++ binary from

<http://download.gna.org/getfem/html/homepage/download.html>

a. Run the file by opening it

6. Install the bConverged version of Calculix software. The setup file is provided in the “Calculix Installation Files” folder.

7. The installation files folder has a “tetgen” executable. Edit “Path” environment variable to include the path to this executable. Once this is done, run command prompt and type “tetgen”.

8. Edit Path environment variable to include the path of ccx.bat and cgx.bat files (found in “bin” folder in the place where you installed calculix). Check by typing “ccx” and “cgx” in the command prompt.

9. Finally, go to the Python27 folder. There is a “DLLs” folder inside. Open the folder and make a copy of the “freetype-2.5-vc90-mt.dll” file and rename the copy to “freetype.dll”