1-30-2013

Distributed force model for arbitrarily shaped IMPC actuators

Manuel Martinez

Manuel Martinez

Candidate

Mechanical Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Ron Lumia, Chairperson

Juan Heinrich

Tariq Khraishi

DISTRIBUTED FORCE MODEL FOR ARBITRARILY SHAPED IPMC ACTUATORS

BY

Manuel N. Martinez

B.S., Mechanical Engineering, University of New Mexico, 2009

THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Masters of Science

Mechanical Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2012

ACKNOWLEDGEMENTS

I would first like to thank Dr. Ron Lumia for inviting me to participate in his research and then providing the advice and encouragement necessary to complete this thesis. He is the best teacher I've ever had; his influence extends far beyond academics.

I also need to express my gratitude to my wife Priscilla, who has given me love and a sense of meaning. I have an enjoyable and happy life because of her company and that of our two daughters, Araceli and Evalisse.

My mother, Victoria, has always been there for me. She has also always believed in me. When I have a hard problem she is the person I turn to for clarity. I could not have accomplished this without her support.

Lastly, to all the friends and coworkers I've had in the time I have studied at UNM. Although they are too numerous to mention individually, I want to thank them all for the conversations and good experiences we've shared. I would especially like to thank Brian Schmitt who first recommended me to Dr. Lumia for a position in the lab.

DISTRIBUTED FORCE MODEL FOR ARBITRARILY SHAPED IPMC ACTUATORS

by

Manuel N. Martinez

B.S., Mechanical Engineering, University of New Mexico, 2009

M.S., Mechanical Engineering, University of New Mexico, 2012

ABSTRACT

A model that describes the relationship of an arbitrarily shaped artificial muscle to the force it produces does not currently exist for actuators made of ionic polymer-metal composites (IPMC), a type of electroactive smart material. The model in this thesis couples a finite element force simulation for IPMC with a novel method of performing force measurements for IPMC actuators. The model is capable of predicting the blocked force output for IPMC actuators of arbitrary dimension. The ultimate goal of this work is to create a method of analysis that allows for the design of custom IPMC fingers that have specific force production and actuation properties.

TABLE OF CONTENTS

CHAPTER 1. Introduction 1

1.1 IPMC 1

1.1.1 IPMC discovery and basic morphology 1

1.1.2 Mechanics of IPMC actuation 4

1.1.3 Mechanics of IPMC sensing 7

1.1.4 Integrated actuation and sensing 8

1.2 IPMC Microgrippers 10

1.2.1 Microgripper basic concept 10

1.2.2 Design and manufacture of IPMC fingers 11

1.2.3 Microgripper packaging 13

1.2.4 Control schemes 15

1.3 IPMC Force Scanner 18

1.3.1 IPMC Force Scanner setup 18

1.3.2 Scanner results 21

1.4 Purpose Statement 24

1.5 Contribution 25

CHAPTER 2. Literature Review 27

2.1 Properties of IPMC 27

2.2 Finite Element Analysis (FEA) Modeling of IPMC 35

CHAPTER 3. IPMC Distributed Force Model 38

3.1 Introduction 38

3.2 CAD Modeling of IPMC 41

3.3 Meshing the IPMC 43

3.4 IPMC Distributed Force Model 45

3.4.1 Model overview 45

3.4.2 Comsol/Matlab electrochemical model 46

3.4.3 Electrical model for arbitrary shapes 67

3.4.4 Matlab IPMC force model 77

3.4.5 Comsol/Matlab distributed force simulation 88

3.5 Concluding Remarks 103

CHAPTER 4. Results 105

4.1 Overview 105

4.2 Electrical Model Results 105

4.3 Electrochemical Model Results 108

4.3.1 Cation migration 108

4.4 Force Model Results 112

4.4.1 Deformation 112

4.4.2 Force distribution 114

CHAPTER 5. Conclusion 117

5.1 Results of the Study 117

5.2 Limitations of the Study 119

5.3 Future Research 121

Appendices 123

Appendix A – Programming Examples 123

A1. Function Concentration_table_generation 123

A2. Function Concentration_extract 129

A3. Function Concentration_export 129

A4. Function Concentration_profile_selection 130

A5. Function Force_generation_rectangular 130

A6. Function Triangular_force_scan 133

References 142

TABLE OF FIGURES

Figure 1. IPMC polymer and electrode layers 1

Figure 2. IPMC water and cation migration 2

Figure 3. IPMC actuation 2

Figure 4. IPMC laser workstation 3

Figure 5. IPMC microgripper fingers 3

Figure 6. IPMC integrated sensor actuator 8

Figure 7. IPMC microgripper 10

Figure 8. Signatone laser workstation 11

Figure 9. Parker MX80L 11

Figure 10. IPMC electroded holder 13

Figure 11. Prototype microgripper holder 14

Figure 12. Lab equipment schematic 16

Figure 13. Microgripper robot 17

Figure 14. IPMC Force Scanner schematic 18

Figure 15. IPMC force map 18

Figure 16. IPMC with nodal network 19

Figure 17. IPMC Force Scanner 20

Figure 18. Rectangular IPMC with nodes 20

Figure 19. Average force rectangular IPMC 22

Figure 20. Percent deviation rectangular IPMC 23

Figure 21. Cation redistribution 28

Figure 22. Equivalent circuit diagram 31

Figure 23. Sensor actuator sandwich 33
Figure 24. IPMC distributed force model schematic 46
Figure 25. Creation of a cubic domain 53
Figure 26. Comsol's model navigator GUI 54
Figure 27. Physics menu 54
Figure 28. Conductive Media DC subdomain settings 55
Figure 29. Conductive Media DC boundary settings 55
Figure 30. EK Flow subdomain settings 56
Figure 31. Free mesh parameters 57
Figure 32. Solver parameters GUI 58
Figure 33. Plot parameters 60
Figure 34. Cross-section plot parameters 61
Figure 35. Save-as GUI 62
Figure 36. Comsol/Matlab connection dialog 63
Figure 37. Work-plane settings 71
Figure 38. Create composite object 72
Figure 39. Extrude GUI 73
Figure 40. Electrostatic force between micelle 79
Figure 41. IPMC force model 86
Figure 42. IPMC and transducer domains 93
Figure 43. Add function dialog 94
Figure 44. Solid stress/strain subdomain settings 96
Figure 45. Distributed body load 97

Figure 46. IPMC with force transducer domain 101
Figure 47. Voltage distribution for a rectangular IPMC 105
Figure 48. Voltage distribution for a fin shaped IPMC 106
Figure 49. Voltage distribution for a segmented IPMC 106
Figure 50. Voltage distribution for an IPMC with unactuated portion 107
Figure 51. Predicted voltage as a function of length for a rectangular IPMC 107
Figure 52. Predicted electric field through the thickness of an IPMC 108
Figure 53. Simulation of cation migration in a Nafion cube 109
Figure 54. Chart of cation migration through the thickness of an IPMC 110
Figure 55. Simulated migration of cations through an IPMC box 111
Figure 56. Simulated deformation of a rectangular IPMC 112
Figure 57. Simulated deformation of a segmented IPMC 113
Figure 58. Simulated deformation of an IPMC with an unactuated portion 113
Figure 59. Force distributions for a triangular IPMC 114
Figure 60. Force distributions for an IPMC shark pectoral fin 115

LIST OF TABLES

Table 1. IPMC Force Scanner results 21

CHAPTER 1. **INTRODUCTION**

1.1 **IPMC**

1.1.1 IPMC discovery and basic morphology

Ionic polymer-metal composites are a type of smart material that mimics the action of natural muscles by responding mechanically to electrical stimulation. They are commonly referred to as artificial muscles for their similarity to real muscle and have been the subject of promising research in engineering, bio-engineering, chemistry, and aerospace fields[1][2]. Sub-millimeter thick sheets of IPMCs are created by coating

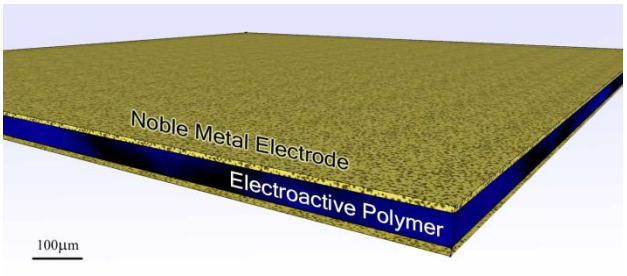


Figure 1. IPMC polymer and electrode layers

electro-active polymer (EAP) films, usually Nafion, on both sides with a noble metal, generally gold or platinum (Figure 1). This is accomplished through a deposition process that creates a 1-5 micron thick electrode with thin dendrites that anchor the electrode firmly to the polymer. The electrode that is created by the deposition process is not continuous but consists of tightly packed metal grains. The function of the noble metal layers is not only to create a conductive surface through which the IPMC can be charged but also to store those opposing charges, much like a parallel plate capacitor [3]. When the plates are charged in this way, mobile ions and solvent in the intervening polymer

migrate and collect on one side (Figure 2). This characteristic is referred to as electro-activity. High electro-activity means greater tendency for ionic motion within the polymer. Electro-activity is very important in IPMC; it is the collective movement of the mobile ions that is largely responsible for the bending of the muscle. For one, the migration depletes the anode and saturates the cathode with positive charge. The resulting imbalance of charge through the thickness results in the expansion of the cathode due to induced electrostatic forces. Additionally, the movement of the ions has an associated "parasitic" movement of larger solvent molecules in the polymer [4]. In effect, the mobile ions carry polarized solvent molecules to one side of the muscle causing it to swell while dehydrating the opposite side causing it to shrink. This is, in essence, like a sponge that is wet on one side and dry on the other, where the wetted side expands and curls around the unexpanded one (Figure 3).

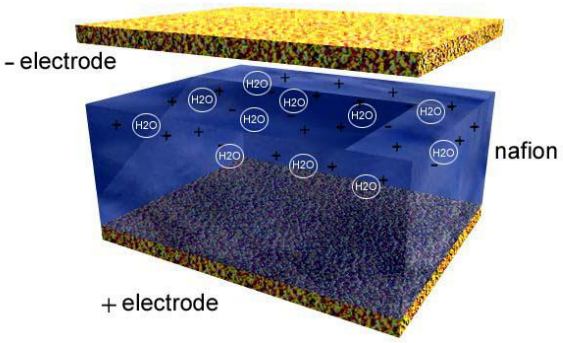


Figure 2. IPMC water and cation migration

IPMCs are manufactured as thin sheets measuring several square centimeters and typically no thicker than several hundred microns. The sheets are, at least visually, not dissimilar from thick tinfoil with dim satiny faces. Tactilely, they resemble a polymer more that a metal, and are barely rigid much like the blade of a thin leaf. It is

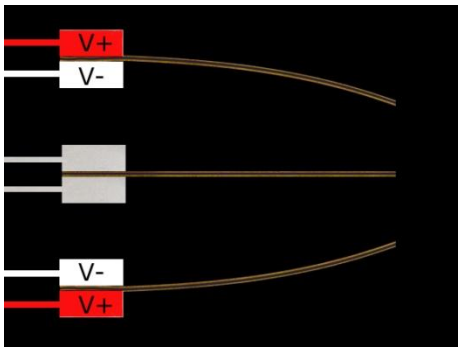


Figure 3. IPMC actuation

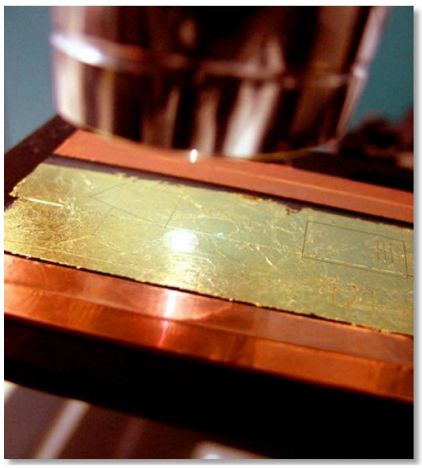


Figure 4. IPMC laser workstation

not surprising to find that they are commonly described as lightweight actuators. Their high electro-activity coupled with low rigidity allows them to demonstrate high degrees of bending when supplied with small voltage potentials [3]. The actuators can be processed into geometries specialized for specific applications. Generally, this means simply using a scalpel or scissor to remove thin slices of material from the main sheet. However, it also possible to process the IPMC into complex shapes using automated laser cutters (Figure 4) operating at a frequency where both platinum and Nafion are ablated. Since IPMC is mechanically simple and consists of a single component, it is completely scalable in the plane of the sheet. This is specifically attractive for the creation of microgrippers, since miniaturization of the actuation mechanism does not require the scaling of individual components but rather simply cutting the IPMC sheet into smaller pieces. This task has become relatively easy since the introduction of CAD programmable laser machining equipment to the lab.

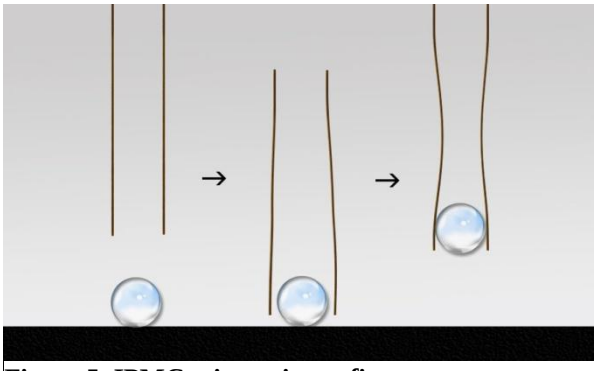


Figure 5. IPMC microgripper fingers

An important characteristic arising out of their low rigidity is compliance. This is an especially desired attribute for manipulation of fragile bio-objects, e.g., oocytes. The IPMC's grip conforms to the surface of the objects in its grasp (Figure 5). This increases the firmness of the grip while decreasing the deformation and stress on the cell.

Additionally, the high strain deformation that IPMC demonstrates does not generate high force (typically < 0.1 N). Coupling high compliance and low force in a single actuator is ideal for the manipulation of biological cells which are delicate and easily damaged. The attributes of IPMCs contrast with that of other cell gripping technologies that are rigid, high force, low strain and risk deformation damage to cells. The two additional attributes of wet operation and low voltage actuation increase the suitability of IPMCs for the task of cell gripping [5]. The former allows the gripper to work with moist cells in aqueous environments. The latter decreases the risk of introducing a voltage difference across the cells and killing them. It is also important to note that the materials that the IPMC are constructed from are relatively non-reactive and are, in any case, sealable. So called "dry IPMC" are polymer encased versions of hydrated IPMCs. They can be used in cell manipulation applications where chemical reactivity with the biological cell is a concern.

1.1.2 Mechanics of IPMC actuation

While it is generally accepted that the migration of ions and solvent due to an induced electric field is responsible for the actuation of IPMC, the exact chemical morphology of the material as well as the physical mechanisms that underlie its actuation are still actively researched. The most widely accepted model for the morphology of Nafion was first described by Hsu in 1981 and was based on wide and small angle x-ray diffraction studies [6]. According to Gierke, perflourinated ionomers (i.e., Nafion) in the hydrated state display phase separation, forming distinct hydrophobic and hydrophilic regions. The hydrophilic regions have the form of 4 nm spherical inverted micellar structures separated at a distance of 5 nm. The micelles are connected to one another by 1 nm diameter micro-channels. Overall, the micellar clusters and the channels form a

cubic grid. This is the so called cluster-network morphology. In this model, the cations and solvent reside in the spherical micelles and the channels. The backbone fluorocarbon chains of Nafion make up the hydrophobic region. The side chains in the polymer backbone structure terminate in sulfonate groups called pendants. These sulfonate pendants form the boundary of the spherical hydrophilic regions. The cations and the sulfonate pendants tend to attract one another and form pseudo-dipoles within the spherical clusters [7].

The actuation response of IPMC depends very heavily on the level of hydration, type of cation, and type of solvent used [8]. During actuation an electric field is set up through the thickness of the Nafion membrane. The induced electric field produces an electrostatic force on the cations (e.g., Na+) which are driven from cluster to cluster through the channels. The magnitude of this migration is dependent on how easily the matrix is traversed by the mobile charges. Smaller cations in a fully hydrated matrix will pass more quickly and readily through the matrix than, say, a larger cation in a solvent depleted matrix. The difference between these two situations may be quite dramatic for the macroscopically observable behavior of the membrane. In fact, it is quite possible that only the former demonstrates any perceptible movement at all. The electrophoretic migration of cations results in a thin boundary layer of high cation concentration at the cathode. According to model proposed by Nemat-Nasser, the increase in concentration of cations results in an initial fast volume expansion of the clusters followed by a slow volume decrease as the cations are redistributed. These volume changes are accompanied by a stiffness increase for the bulk polymer in the cathode region [9]. Anions are fixed to the pendant chains and are immobile during actuation. Consequently, a thicker cation

depleted region forms at the anode. In this region, the backbone polymer matrix experiences relaxation as the remaining anions repel one another. Overall, the redistribution of cations and the resulting stress field bends the Nafion quickly towards the anode. A simultaneous albeit slower migration of water molecules proceeds through the channels as water molecules attached to cations are dragged towards the cathode. The migration of water molecules adds positive hydrostatic pressure to the clusters at the cathode causing them to slowly expand as new molecules arrive. The combined effect of cation and water migration results in an initial quick (~1-10s) actuation towards the anode that eventually gives way to a much stronger actuation in the direction cathode that lasts several minutes as the cations are slowly redistributed in the clusters [4].

The force model in this thesis suggests that the actuation of the IPMC may be explained by considering the electrostatic interaction between the micellar clusters only. This is to say that as cations vacate the clusters in the anode and flood into the clusters at the cathode, the result will be two boundaries layers near each electrode. The anode layer will contain negatively charged clusters which repel each other. Similarly, the cathode will contain positively charged clusters which also repel each other. This theory is perhaps unusual because the cation migration results in charge imbalance in the cathode and anode regions and will translate into positive pressure on both sides of the actuator. However, in Chapter 3 it will be shown that as a consequence of the form of the force equation and the nature of the redistribution of the cations in the actuator, that the force production at the cathode is superior to that of the anode. In other words, despite positive pressure at the anode and cathode, the process overall still results in an imbalance in force which drives the macroscopic actuation toward the anode.

1.1.3 Mechanics of IPMC sensing

The idea that an IPMC also acts as sensor was first presented in a paper that used the material in a smart accelerometer for machinery and structures [10]. The device in that paper was an IPMC sandwiched between two electrodes that transmits a voltage to a detector when the IPMC film was squeezed. It was later discovered that the IPMC generates a voltage both in the hydrated and dry states when subjected to mechanical deformation, and furthermore, that the IPMC performs much better as a sensor in the dry condition rather than the hydrated state [11]. Shahinpoor coined the term flexogelectric effect to describe this property of ionic polymers, particularly when it arises from bending as opposed to compression [12].

Although, the mechanisms underlying the flexogelectric effect have yet to be completely uncovered, it has been suggested that the production of stress in the backbone polymer contributes to the displacement of charges in the clusters during imposed deformation [9]. For the IPMC in the undeformed state, the anions that reside on the boundaries of the clusters are balanced by cations located in the saturated clusters. When the polymer is deformed, the cations are shifted from their equilibrium position proportional to the magnitude of the deformation. It may further be the case that stressing the backbone polymer creates hydrostatic pressure that causes the flow of water molecules and cations from regions of high pressure to low pressure regions. The magnitude of the voltage produced by IPMC is proportional to the rate with which the polymer is deformed. For this reason, the material is really a velocity sensor rather than a positional sensor.

1.1.4 Integrated actuation and sensing

There have been several attempts to combine the sensing and actuation properties of the IPMC into a single self-sensing actuator. One of the first was called the IPMC sandwich. This device coupled a thick (~200 micron) IPMC actuator to a thinner (~60um) IPMC that was to be used as a sensor. The actuator and sensor were cut to the same dimension and glued directly one on top of the other. This was done so that as the thicker IPMC was actuated the thinner one generated a small voltage relative to their coupled velocity. In this way the movement of the entire system was tracked. Though the scheme was relatively straight forward, in practice there were complications. The electromagnetic field generated in the actuated IPMC was being detected as a voltage signal in the sensor. This meant that the voltage being generated through the movement of the sensor IPMC was being drowned out by the voltage input driving the actuator. In an attempt to get rid of the interference, a layer of gold leaf was added between the sensor and actuator. The gold layer was attached to ground. Despite the success using the gold layer to electrically insulate the sandwich, the extra gold layer and the glue required to keep it in place added unfavorably to the rigidity of the stack. As had been mentioned earlier, IPMC actuators produce low actuation force and, therefore, the slight increase in rigidity severely limited the actuation. It was decided that the IPMC sandwich, though promising, was in its current state not practical

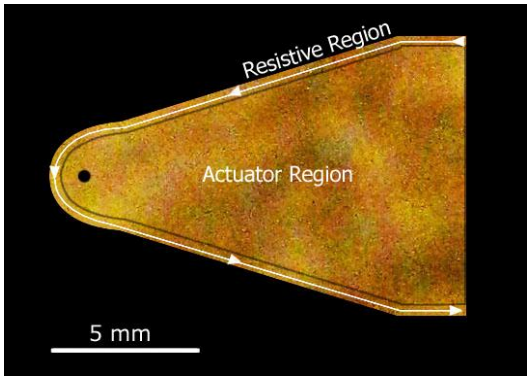


Figure 6. IPMC integrated sensor actuator

for the purpose of microgrippers.

Recently, researchers at the University of New Mexico have been developing IPMC with channels cut into the surface electrodes (Figure 6) that can be used to detect displacement. The channels are created when thin sections of electrode material on boundary of the actuator are removed while preserving the intervening polymer layer. The key to understanding how the channels work is to understand that the metallic surfaces of the electrodes are not solid but consist of small closely packed islands of metal. When the polymer bends, these islands either become more closely packed or begin to separate, resulting in changes in resistance for the surface. By tracking the changes in resistance on the surface of the IPMC as it is actuated, the magnitude of the deflection can be calculated. This actuator sensor combination was superior to the IPMC sandwich for three reasons in particular. For one, the sensor and actuator were integrated, having been created from a single piece of IPMC. This in turn meant less material was needed, since the addition of the sensing capability only require the addition of a thin channel, usually around the perimeter of the actuator, and not an entire new section of IPMC. Also, the addition of the channel, since it required only a small fraction of the entire actuator surface to work, did not contribute significantly to the rigidity of the overall package. Finally, the creation of the sensor channel was facilitated greatly by the discovery that a green (532 nm) laser is capable of ablating the platinum surface electrodes while leaving the polymer intact. Even very small sensing channels can be created on the surface electrodes in an efficient manner. This contrasts quite significantly with time consuming and delicate work of handling and gluing of thin IPMC and gold leafing.

1.2 **IPMC Microgrippers**

1.2.1 Microgripper basic concept

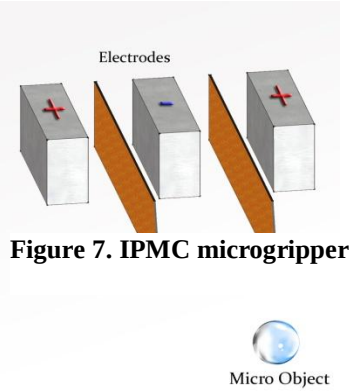


Figure 7. IPMC microgripper

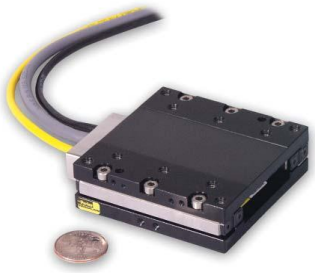
The basic components of a microgripper are two IPMC actuators, a holder with electrodes, and a power supply (Figure 7) [13]. The essential idea is that two IPMC actuators of matching dimension (commonly referred to as fingers for the purpose of manipulation) are fixed in a holder as cantilevered beams parallel to one another with a slight gap between their tips. Their holder has electrodes that meet each IPMC finger's electrode faces so that each of the electrodes can be sent a voltage or current signal. Normally, each finger will be voltage driven and have one face that is set to ground and the other will be receiving a small positive or negative voltage (~2V) depending on the required direction of actuation. This arrangement is such that if the two fingers are actuated towards one another their tips meet. As can be imagined, this device can be used as electrically driven pair of tweezers which is capable of gently capturing small objects between its fingers.

1.2.2 Design and manufacture of IPMC fingers

The creation of IPMC fingers up until relatively recently had been an entirely manual process where thin slices of the material, usually rectangular, were removed from larger sheets using a straight edge and scalpel. At that time, there were really no good methods of creating fingers of complex geometries or channels in the surface electrodes of the IPMC material. However, the addition of an automated CAD driven workstation (Figure 8), the process of creating IPMC fingers has been improved dramatically. The workstation combines a QuikLase Trilite laser with three frequencies (266 nm, 532nm, and 1064nm), a Signatone Probe S-1160 probe station with high powered optics, and a set of Parker MX80L linear programmable stages (Figure 9). Now the process of creating IPMC fingers begins when a profile is created in CAD software such as Pro-E or Solidworks. The profile is stored as a .dxf file for export to NI Motion which is a program capable of accepting CAD designs and converting them into motion profiles that can be executed by stages. Once this has been accomplished, the motion profiles created in NI Motion are stored as Labview executable codes. It is important to convert the final motion profiles into Labview code since Labview allows for the simultaneous control of the laser

Figure 8. Signatone laser workstation

Figure 9. Parker MX80L



workstation and the linear stages. Once the profile is in Labview, the decision is made on how to best accomplish the cuts. For instance, sometimes a finger's design cannot be created with a single continuous cut (i.e., a rectangular finger with a circular hole in its interior). In this case, the laser power can be set to zero over the line that connects the two cuts. It might also be advantageous to increase the velocity of the stages over the uncut line to make the entire process more efficient. Whatever the case, the parameters that control both the stages and the laser can be managed in the Labview environment.

It was discovered by accident that the electroactive Nafion is almost entirely transparent to the green (532 nm) laser while the platinum based electrodes were ablated quite easily by that frequency. While the discovery precluded the usage of the green laser for the purpose of removing IPMC fingers from the base sheets, it did open up the interesting possibility that channels and sections be created on the surface electrodes in a straightforward manner. Now it was possible to not only create the integrated sensor actuator type fingers but also the so-called digitated IPMC. This IPMC, similar to the self sensing IPMC actuator, has its electrode surface divided into multiple sections. However, as opposed to having areas dedicated to sensing, the goal of the digitated IPMC finger is to have multiple areas on the finger that can be actuated individually. This creates more complex actuation schemes. It has, for instance, been suggested that by dividing a rectangular IPMC into two equal sections that a twisting motion can be accomplished by actuating the two sides in opposite directions. Further, by combining both sectioned electrodes and complete cuts it may be possible to create IPMC actuators that look and function like hands with fingers that can be actuated individually.

It is beneficial to be able to simulate, rather than experimentally test, potential designs to see whether they are well suited for a task. This is true for many reasons. The most important is that there are significant costs associated with the creation and testing of each new design. For one, IPMC material is lost. Second, it is time consuming for researchers. Finally, the process of creating and testing prototypes ties up multiple pieces of equipment in the lab. It is easy to see why simulation is a more efficient route given that the model yields reasonable results and is easy to use.

1.2.3 Microgripper packaging

As was briefly discussed in the first section of the chapter, in order to operate an IPMC actuator or receive a sensing signal there must be some way of establishing the electrical contact with the surface of the material. In the case of microgrippers, this requires the design of specialized holders that facilitate communication with the device and hold the fingers securely at fixed distance from one another. Most of the holders that are currently in the lab are specialized devices that were either created using some rapid prototyping process or were manually created from modified electrical components (Figure 10). The design and construction of microgripper holders can be considerably challenging, particularly in cases where the fingers are very small and contain channels. The trouble lies in creating the wiring for several

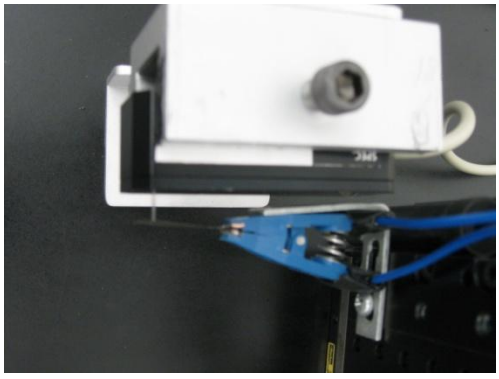


Figure 10. IPMC electroded holder

electrical contacts (>3) in a sub-millimeter span for each face of the IPMC finger. Each contact must contact the surface of the IPMC gently, securely and be sufficiently spaced from other electrodes to avoid electrical shorting.

Originally, the holders were constructed by hand since the fingers used were relatively large, simple, and usually rectangular. These holders were created using modified IC test clips that had copper plates fixed to their mouths using conductive epoxies. The method was useful for the purpose of actuating such simple fingers. However, a significant challenge relating to the newly added ability to create fingers of greater complexity is how to create electrode holders that are capable of actuating them. The ultimate goal of the lab is to create artificial muscle microgrippers that are able to grasp and manipulate micro-objects such as biological cells. This task suggests that the lab will eventually need to create exceedingly smaller microgripper fingers and holders.

The search for better methods of creating increasingly smaller holders has predictably led into rapid prototyping technologies. The microgripper in Figure 11 was designed by UNM students using CAD software and sent off for rapid prototyping. Rapid prototyping though indispensible for creating very small microgrippers can become quite expensive (>$100 per holder). Accordingly, each microgripper holder is designed with consistency and interchangeability of the IPMC in mind. This particular microgripper holder has a

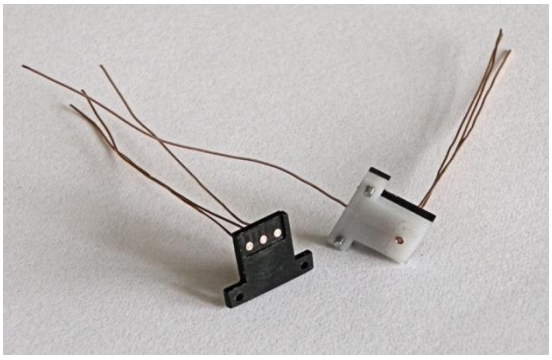


Figure 11. Prototype microgripper holder

specific contact positioning system called a key. The idea is that each actuator will have a key of a fixed shape while the actuator portion of the IPMC will vary. This means that the paths on the IPMC must be cut in a specific manner in order for the material to make proper connection while being able to send and receive signals. The microgripper holder pictured will allow for an IPMC key size of 250 microns by 400 microns to be held in place while the tip can be scaled down to tens of microns in size. The microgripper holder was designed to hold two IPMC's with a maximum 80 microns of separation to allow for the pick and place of a micro sized objects. Smaller separations are accomplished by replacing the middle section of the holder (shown in white).

The amount of effort involved in creating a new microgripper holder accentuates the necessity for approximating the suitability of a finger design using modeling. It hardly seems reasonable to complete the process of designing a specialized holder for fingers that are, for one reason or another, unable to perform the task that they were created for.

1.2.4 Control schemes

The IPMCs in the lab are controlled mainly through Labview, a National Instruments DAQ board, and an electroded holder (Figure 12). The benefit of this system is that it allows for the simultaneous real time processing of output and inputs. The outputs are typically analog voltage signals which are sent through a conditioning amplifier to the IPMC's electrodes. This is accomplished easily using the Data Acquisition Assistant GUI in the Labview back panel. Since Labview supports multiple outputs, an additional benefit is that the digitated IPMC, which require multiple voltage signals, can be controlled by simply adding additional analog voltage channels to the

DAQ Assistant. Similarly, the DAQ Assistant also handles multiple inputs. This can include not only resistance measurements from sensor actuator type IPMC but also inputs from varied devices including force transducers and laser vibrometers. This is useful for testing IPMC devices when actuation occurs simultaneously to force and displacement

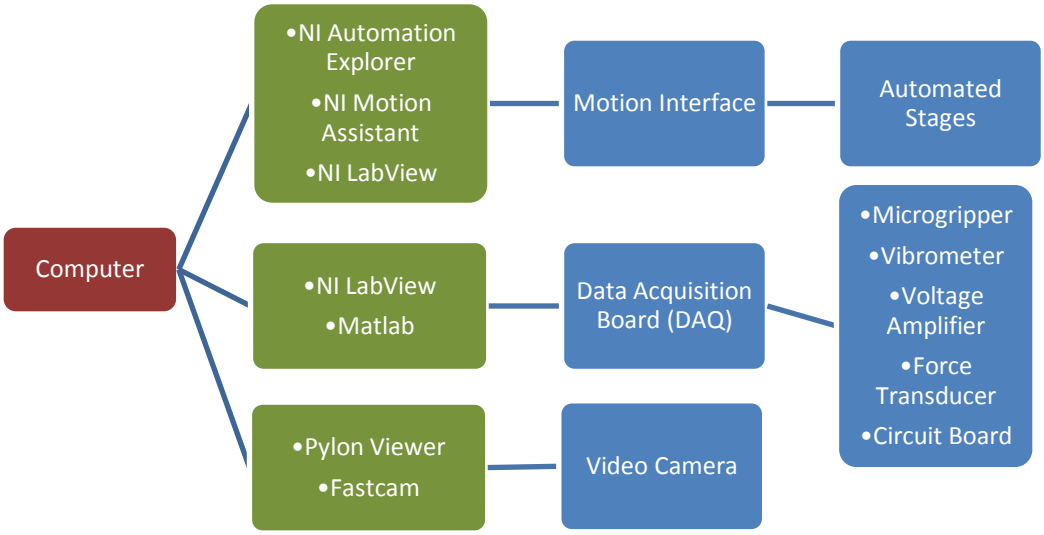


Figure 12. Lab equipment schematic

measurements. It also means that control loops can be built where controlled actuation is achieved by adjusting the control voltage as a function of resistance measurements. Cameras, motion devices, oscilloscopes, and various other lab equipment are also compatible with Labview, meaning a large number of experimental configurations can be achieved.

The latest attempts at more sophisticated actuation of IPMC have centered on the creation of control loops for self sensing IPMC. By tracking a sensor actuator type IPMC's displacement using a force transducer and simultaneously tracking the change in resistance in its sensing channels, a relationship between the displacement and resistance

can be uncovered. This information can be used in a control scheme where the IPMC's resistance is tracked and used to command specific positions. As has been discussed earlier, IPMC's actuation is not constant for a given voltage and eventually the

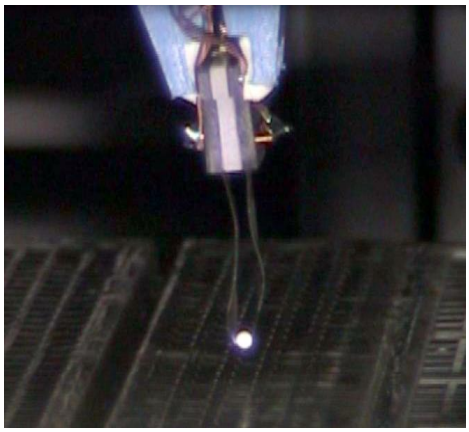


Figure 13. Microgripper robot

IPMC will reverse its course and actuate in the opposite direction. Fortunately, this process is slow and can be held off for a substantial period of time by adjusting the voltage to maintain a surface resistance corresponding to a specific position. Extending this theme of using surface resistance changes to track motion, PID controllers can also be built which use the difference between the current position and the commanded position based on resistance. Labview also has a Matlab interface which allows variables to be sent into Matlab and have variables returned to Labview. This means all the control tools available in Matlab, which are often more familiar to UNM students, can be used.

IPMC based microgrippers can be attached to automated stages. This configuration can serve as an automated microgripper robot, which can grasp and manipulate small objects. Such an autonomous system has already been demonstrated using one millimeter polymer spheres. There are plans to increase the capabilities of this device through the addition of a haptic joystick, which allows users to "feel" microobjects as they are being manipulated. Also planned is the use of machine vision to automatically locate and reposition objects.

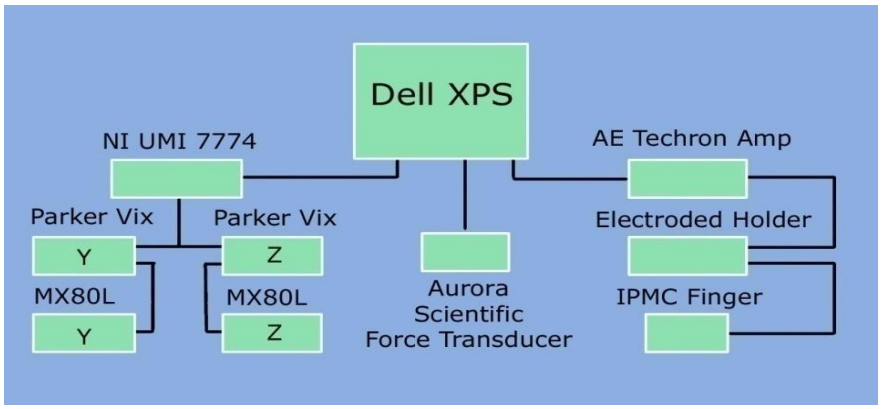


Figure 14. IPMC Force Scanner schematic

1.3 **IPMC Force Scanner**

1.3.1 IPMC Force Scanner setup

In order to facilitate the creation of a distributed force model, it was imperative that a method of experimentally determining the actual force distribution be created. This way any potential force model be built around and compared to actual force measurements. As it turns out, ultimately the creation of the device was important not only for model comparison, but interestingly enough, because no such measurements had ever been taken. The IPMC Force Scanner (Figure 14) allowed for the evaluation and comparison of the gripping strength and force variance of existing fingers. The results of the Force Scanner are topographic force maps that display the relative strength of the forces (Figure 15). Several notable results arising from these measurements are: the force falls exponentially from the point where the voltage is applied, the natural warpage of the IPMC has substantial effects on force output, and a dry IPMC displays low force variance over multiple runs especially at points farthest

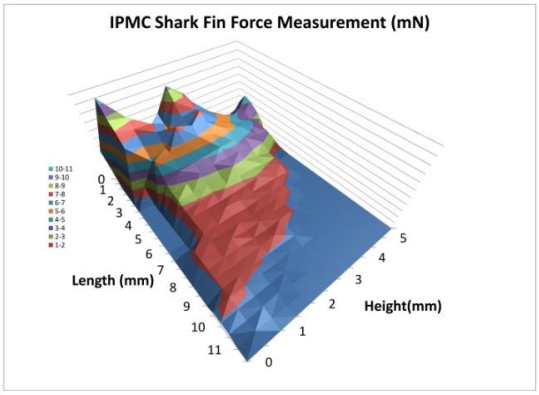


Figure 15. IPMC force map

from the fixed end of a cantilevered actuator. These results will be discussed in detail at the end of this section. For now, the components and methods involved in capturing a force distribution using the IPMC Force Scanner will be discussed.

To measure the force output of an IPMC finger, the finger is placed in a custom holder mounted to stages. The holder is a modified 8 pin IC test clip with positive and negative electrodes on either side. The stages are Parker MX80M and L types. The MX80M is a manual stage that is used to control the distance between the finger and the force transducer. It is used to "touch off" the actuator and the transducer, positioning the force transducers straw so that it is just at the surface of the actuator. Once the proper distance has been established, two electric MX80L linear stages mounted orthogonally to one another move the IPMC in a plane through a preprogrammed trajectory following an imaginary nodal network. The imaginary nodes mark the location of the force measurement. Collectively the force measurements can be rendered as topographic maps where the forces and corresponding locations are easily visualized.

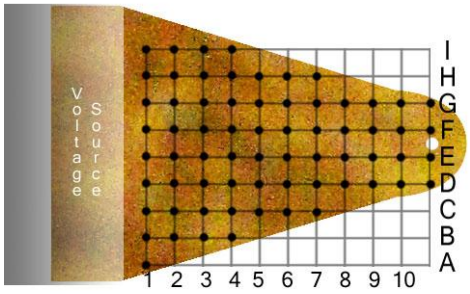


Figure 16. IPMC with nodal network

The coordinated movement, actuation and measurement of the force from an IPMC finger takes several steps to accomplish. The stage trajectories are programmed in National Instruments Motion software. These trajectories consist of individual node to node movements separated by delays when force measurements are taken (Figure 16). The trajectory is converted into NI Labview code using conversion software included in

NI Motion. Once the resulting code is loaded into Labview, sinusoidal voltage outputs are programmed to coincide with the delays in the stage trajectories. These voltages are sent to a voltage amplifier connected to the electrode holder and used to actuate the IPMC finger. Measurements from the force transducer are programmed to occur simultaneously with actuation. In effect, the stages position the IPMC finger node by node in front of the force transducer pausing each time so that the finger can be actuated and the resulting force profile recorded (Figure 17). This profile contains the entire force history at a given point under a given voltage signal. Normally, as regards to force, the interest is in determining how strong a muscle is at a given point on the actuator. For this reason, the maximum force is extracted from the force profile and used to generate the force maps. The measurement cycle is run iteratively so that multiple measurements are taken at each point. This is performed in order to yield statistical information about variations in the actuator's performance. Generally, the mean and standard deviation of the maximum force is of most interest.

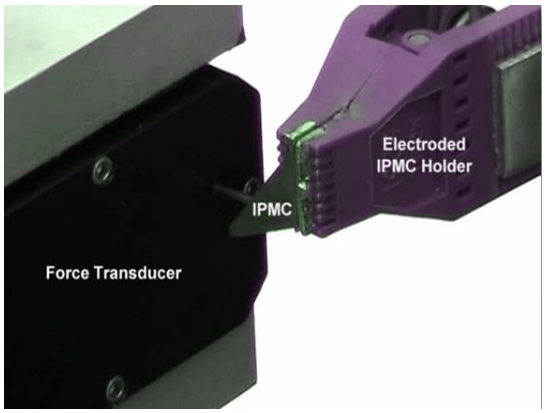


Figure 17. IPMC Force Scanner

A description of the actual process follows. First, a 17mm x 7mm actuator was cut from a sheet of dry platinum-Nafion type IPMC on a Signatone laser workstation. The actuator was loaded into the electroded holder and adjusted so that

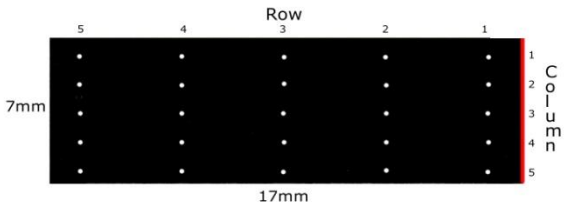


Figure 18. Rectangular IPMC with nodes

the plane of its body lies tangential and nearly in contact with the straw of a force transducer. The red line shows where the driving voltage is supplied by the holder (Figure 18). A 25 node network was programmed in Labview. The nodes were incremented by 1 mm in the rows and 3 mm in the columns. At each of the nodes the actuator was subject to 0.25 Hz sine wave with an amplitude of 2 volts for 4 seconds. The force for each actuation was measured using an Aurora Scientific 403A force transducer, which has a range of 0-5 mN, and recorded into a matrix by Labview. After a force has been recorded for all of the 25 nodes, the matrix was then sent to an Excel file. In the experiment a total of 20 complete runs were recorded. Average and maximum forces for each of the 25 nodes were calculated and graphed.

1.3.2 Scanner results

	Column				
	1	2	3	4	5
	1.20±.39	2.93±.25	3.59±.26	4.72±.23	4.02±.15
	1.31±.17	1.51±.14	1.86±.08	1.85±.05	1.70±.07
	0.74±.09	0.86±.08	0.99±.07	1.03±.05	1.10±.06
	0.69±.06	0.72±.06	0.81±.06	0.93±.16	0.91±.09
	0.55±.05	0.57±.04	0.56±.05	0.54±.07	0.55±.03

Table 1. IPMC Force Scanner results

As can be seen in Table 1, the force output for the IPMC varies significantly along the length. For every column except column 1, a clear exponential decrease in force is evident. The largest exponential decrease is in column 4 where the force falls by 88 percent over 15 mm. This column also contains the highest value of 4.7 mN and the lowest value of 0.54 mN.

In column 1, a lower than expected force was measured in its first row, this can be attributed to residual stresses in the unactuated IPMC and to the importance of alignment of the actuator to the force transducer. Normally IPMC actuators experience residual stresses from manufacturing. These stresses cause them to warp slightly. For the first node in column 1, the IPMC was warped away from the force transducer causing it to exert a force only after actuating some distance rather than immediately. This effect can also be seen in the fourth row of columns 4 and 5, however, in this instance the actuator is warped towards the transducer rather than away from it. It is important to note that this effect diminishes along the columns where the IPMC displays greater displacement. This is evident from Figure 19. The magnitude of the measured force varies most in row 1. In this row the forces are higher than anywhere else on the actuator. However, the magnitude of the displacement the lowest since this row is close to the fixed end. In row 5, however, the IPMC is capable of relatively large displacements but the magnitudes of the forces are the lowest. In this row there is a fairly uniform force distribution.

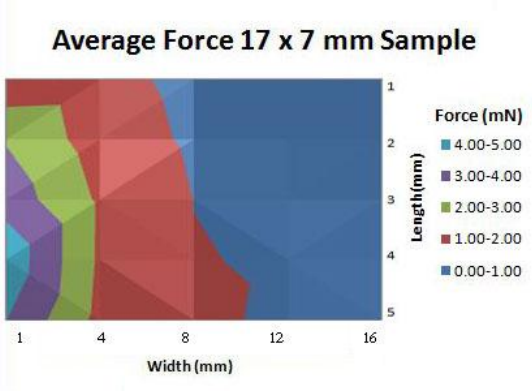


Figure 19. Average force rectangular IPMC

In general, the forces measured at each of the nodes did not vary by more than 10 percent of the average force for that node (Figure 20). Notable exceptions occurred where warping was present, for instance, in the first row in column 1 where the highest deviation of 34 percent occurred.

The IPMC Force Scanner can produce force measurements quickly and efficiently. For the 7 mm x 17mm sample, the force scanner produced data from 20 runs in under an hour. Each run was performed in under two and half minutes. Measurement from each node took just under 5 seconds. The results showed that forces along the length of an IPMC actuator decrease exponentially. A fall of 88 percent in force was reported for the sample, with a maximum of 4.72 mN and a minimum of 0.54 mN.

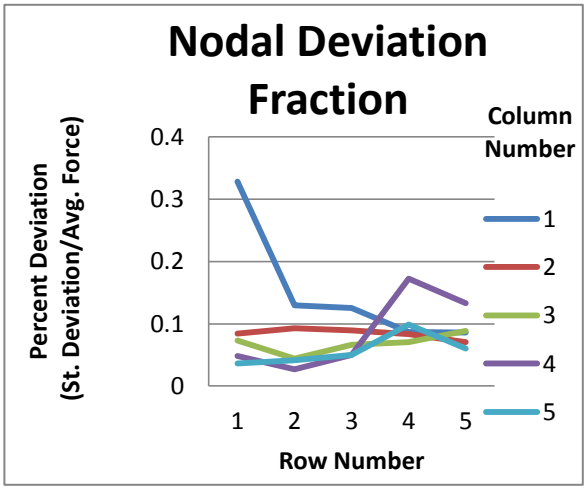


Figure 20. Percent deviation rectangular IPMC

The experiment pointed out the effects of warping due to residual stresses on force measurements. Actuator warping towards the sensor creates unusually high values, while warping away creates unusually low ones. This is corrected if a third stage was added to the scanner allowing it to automatically adjust to the actuator surface. In a similar way, spacing between the actuator and the sensor was shown to have a pronounced effect on the force output. Small tilts in the actuator produced slightly skewed force results. This can be fixed, however, with the creation of new IPMC holders that are precision machined. A holder of this type ensures that the actuator is maximally aligned to the force transducer.

1.4 **Purpose Statement**

It has long been realized that one of the major limitations in the IPMC field is an absence of suitable engineering models [9]. IPMC actuation characteristics are still poorly understood and the models that describe them are largely inadequate to predict finger actuation or force output in an engineering environment. They cannot be used to simulate the force output of arbitrarily shaped fingers nor used to design fingers with particular properties [11]. For the UNM microgripper project in particular, the need for a working engineering model is especially important. The lab has the capability to produce fingers of virtually any dimension and yet, still largely produces only simple rectangular shapes much like others in the field. These rectangular shapes are ubiquitous in the literature and have traditionally been used because they are the easiest to create by hand. Advancements in technology have not, unfortunately, had the effect of changing or improving the overall form or function of IPMC fingers. Only the creation of appropriate engineering models will allow researchers to have purposeful design control and motives for altering current finger designs.

The model that is most helpful to UNM's cell microgripper research is one that describes the force output of a finger in relation to its shape. There does not exist a way to estimate if an amount of material can produce a required force without directly testing it. This has been a major limitation in the application of the material to actuation tasks where force output and not deflection is the primary concern. This was the case when a dust wiper constructed for a NASA space rover failed to produce enough force to perform its task [14]. In failing, the hope was that a thicker muscle, in production at the time, could provide suitable force. Again, in the absence of engineering models, much of the

determination of the suitability of IPMC to a force task is left to ad hoc experimentation and educational guess. The costs of these methods of development are not to be underestimated. IPMC is only inexpensive relative to comparable technologies. The costs incurred by institutions that study the material from producing useless prototypes can be prohibitive, both in time and in money. In addition, it is anticipated that the future of IPMC includes manipulation and sensing tasks at the micro-level. However, it has yet to be determined whether an IPMC can produce suitable force at that dimension [1]. This is also true at larger scales in current IPMC research. How much material is needed to design a finger that can produce enough force to support the embryonic cell of a mouse is simply not known. Questions such as these cannot be answered by current models. These questions will exist as long the need for a suitable model exists.

1.5 **Contribution**

The proposed force model will consist of a finite element model that is implemented using Comsol Multiphysics and Matlab. This model is capable of simulating the force output of IPMC actuators of arbitrary dimension at any point on its surface. Such a model can be used to analyze the suitability of IPMC devices through simulation, eschewing the tedious and wasteful process of creating and testing multiple prototype devices. The model consists of several interlinked sub-models each addressing a particular physical phenomenon: an electrical model, a migration model, and a force model. The primary concern of the electrical model will be to identify the voltage distribution across the face electrodes. This voltage is the driving influence for actuation and is used as an input for the migration model, which predicts the motion of charged

particles in the material. Once the distribution of charges has been predicted it can, in turn, be used to calculate the electrostatic force driving the IPMC's actuation. This accomplished using a novel force equation that suggests a different physical interpretation for the electroactivity seen in IPMC. These forces can be used in a stress-strain model to predict the force output and the motion of the actuator. Using the Matlab interface for Comsol, a routine that yields a topographic map of forces for any IPMC is developed. Ultimately, it is these topographic force maps that can be used to assess the suitability of IPMC actuator with a given geometry to a specific task.

Checking the validity of the force predictions required the development of a new experimental device capable of performing multiple force measurements on actual IPMC actuators. This device, the IPMC Force Scanner, also has topographic force maps as a final output. This makes for easy visual comparison between the output of the model and the experimental actuators. The experimental and simulation force maps for several actuator geometries will be presented at the end of this thesis. As will be seen, there is very close correspondence between the two for each respective actuator.

CHAPTER 2. **LITERATURE REVIEW**

2.1 **Properties of IPMC**

Since the actuation properties of IPMC were first introduced in 1992, the field has steadily progressed in its understanding of the macroscopic properties of the material [4]. The first models to identify and characterize the macroscopic qualities were black box models based on experimentation. The methodology of these researchers was to get as much experimental data on behavioral phenomena as possible and curve-fit equations to the results. These early explorations into IPMC properties gave us a great deal of information regarding its actuation, sensing, and material characteristics [15]. In the first two years the postulation of parasitic movement of water molecules with mobile ions was greatly responsible for the electric response of the material was proposed. This relationship was established in experiments where the diminishing actuation of the material was observed to coincide with dehydration [15]. This was not true for IPMC samples actuated underwater. In fact, submerged samples have been observed to actuate hundreds of thousands of times without noticeable degradation of response [16]. It later was discovered that the backbone of the polymer is essentially fixed but interstitially contains both mobile cations and water. The cations move as an electric field is setup by charging the metal faces of the material[4]. During actuation, water molecules bond ionically to the positive charges and they migrate together toward the anode (Figure 21). Samples of IPMC doped with different cations displayed varying force and displacement characteristics. Nemat-Nasser demonstrated that sodium ions produced greater force output relative to lithium or hydrogen for Nafion based IPMC[17]. As early researchers understood, cation migration was only part of the actuation mechanism, and although this

mechanism proved to be dominant, it was too slow to account for the almost instantaneous movement of the IPMC. In order to account for the faster response, it was postulated that Coulombic forces between the charges on the electrodes caused expansion on the cathode and expansion on the anode. This reaction was not only fast but also positively contributed to the actuation of the IPMC [9].

Another characteristic born out of the early experiments and relating to material

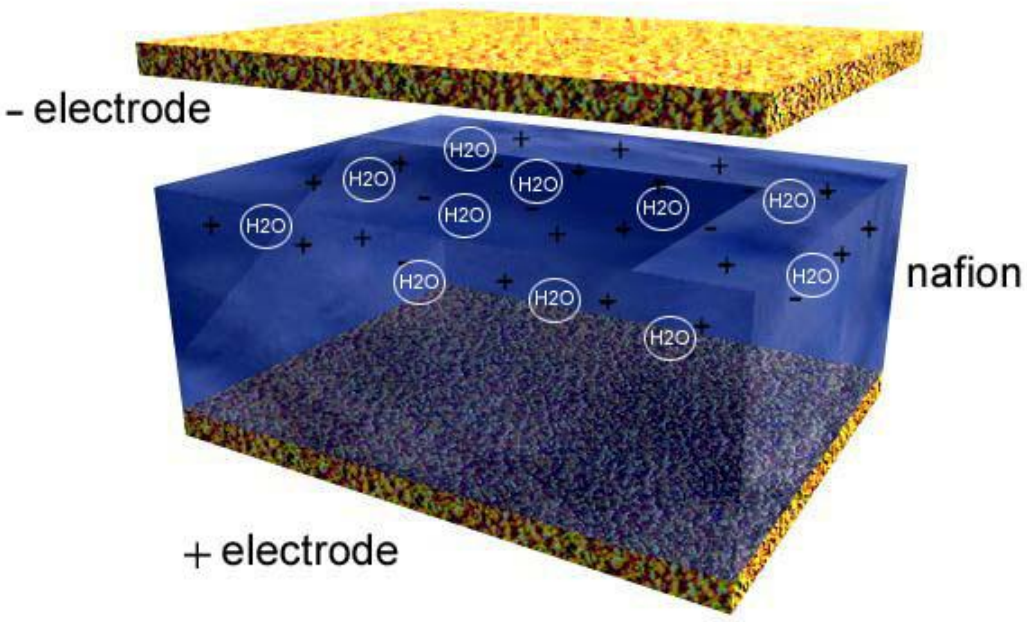


Figure 21. Cation redistribution

hydration was hydrolysis. It is known that electrolysis becomes a factor as the driving voltage increases to more than a couple of volts[15]. If the voltage is high enough, hydrolysis will occur rapidly and the material will desiccate generating hydrogen gas at its electrodes. The response of an IPMC actuated in this fashion will not last longer than a few minutes. Furthermore, hydrated polymer is better shielded from the scorching that can occur if an IPMC is driven with larger voltages. Hydrolysis is one of the factors that

still limits the driving voltage of wet IPMC actuated in air. Interestingly, the hydration requirement and hydrolysis did not prove a great hindrance to the current applications of the material. In fact, especially in the field of biological cell microgrippers, the required driving voltages are generally no larger than a few volts and the ability to work in aqueous environments is a necessity. The voltage requirement is particularly important given that exposures to high voltages can alter or kill cells.

Early research also contributed to the understanding of the physical and chemical morphology of the material. Having understood the importance of ion transport, researchers sought to identify the physical characteristics that contributed to the electroactivity of IPMC. Studies delving into the structure of the noble metal plating acting as the electrodes revealed that their construction played a pivotal role in the success or failure of the material [18]. The early development of the first IPMC was complicated by the non-reactivity of the Nafion; the polymer is closely related to Teflon. Techniques of fixing the gold or platinum electrodes to the polymer securely and consistently posed a complex problem for manufacturers of the material. A solution came when chemical etchants were applied to the faces of the Nafion prior to metal deposition [19]. As a result of this process, the noble metal electrodes formed microscopic dendrites that anchored it to the face of the polymer. A modern IPMC has a metal polymer gradient, the faces of the material being entirely metal and its center being entirely polymer. The metallic layers are necessarily thin compared to the polymer one. The faces of the IPMC viewed microscopically are grainy. This graininess diminishes the conductivity of the electrode, the boundaries of the grains inhibiting the transfer of charge. This contributes to a fairly rapid decrease in the electric field away from the

supply voltage. Although a thicker metal layer contributes to greater conductivity across the face of an IPMC and increase the strength of the electric field translating into greater actuation, it also adds to the stiffness of the material. In this way, the added conductivity is more than negated by the added stiffness of an inherently weak actuator.

Early researchers framed their models in terms of the understood characteristics of the material. Their methods yielded linear models of the material that described its basic actuation characteristics but generally ignored transient behavior. This is because most of the transient behavior results from a complex chemical-mechanical-electrical reaction within the material that even current theories have yet to fully address. These black box models did not increase knowledge of the underlying principles but did contribute to the working knowledge of the material. In 1994, Kanno et al. described the actuation process of IPMC in three distinct phases: electric, stress generation, and mechanical [15]. This model proved to successfully apply control methods to describe the transient current through the polymer in terms of the voltage input, but only for initial actuation. Additionally, via curve fitting, they were able to link the current response to displacement under various voltages. Their methods further included a simple circuit model that demonstrated that the IPMC's electrical response was described successfully by RLC circuits. These simple models later proved to be important to IPMC researchers, as the models were elaborated on and grew to encompass new phenomena as they were uncovered [3]. Simple RLC circuits, like the one in Figure 22, are still instructive when thinking about the initial flow of current through the material. They correctly describe the sudden rise and exponential decrease of the actuation current in the first seconds of response. The models were not, however, in any way predictive in the way current

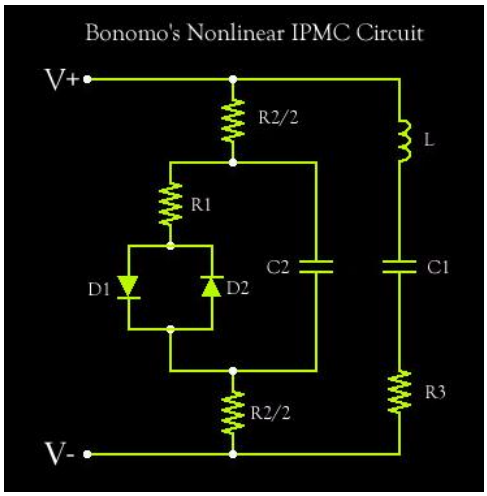


Figure 22. Equivalent circuit diagram

models aim to be. It is fairly clear that at least some of the early researchers were not aware of the important sensing characteristic of IPMC. While it was understood, at least qualitatively, that ionic motion provided the driving mechanism for actuation, it was not determined from the models that the converse was also true.

The discovery that IPMC had the ability to sense motion created a whole new field of IPMC research. A literature search demonstrates that papers dedicated to the subject are almost as numerous as those dedicated to actuation. Tremendous interest was generated as the suitability of the material was proposed for numerous applications in the biomechanical and engineering fields. The interest comes not only from those who benefit from the simplicity of an IPMC sensor, but also those who benefit from the incredible scalability of the material. In addition, the material's light weight and near instantaneous response attracted those who were interested in the material for vibration sensors [10]. The sensing capability of IPMC again lies in the presence of mobile charges within the interstitial space of the negatively charged backbone polymer. Mechanical deformation of an IPMC strip causes one side of the polymer to compress, concentrating the negative charge. Simultaneous expansion on the other side of the strip has the opposite effect, increasing the distance between negatively charge molecules. In this configuration, the mobile ions move toward the expanded side where charge concentration has fallen. This shift of positive ions and water molecules can be detected as a voltage difference at the IPMC electrodes

[1]. The mechanisms that lie beneath the sensing characteristics, being almost entirely the mechanism of actuation run in reverse, do not produce as pronounced a voltage as might first be expected. In fact, experiments have revealed that the voltage produced by mechanical deformation had to be amplified by two orders of magnitude if it were to be used to deform the same piece of material electronically [1]. Given the magnitude of the sensing response of IPMC and the lack of deterministic models, it is little wonder that the existence of the property eluded many early researchers. It is important, however, to note that subsequent research focused not only on characterizing the sensing properties of IPMC, but also to improve the quality of the signal itself [11].

Research carried out at the University of New Mexico tied together the characterization of the materials actuation and sensing in the IPMC sandwich (Figure 23). The sandwich, which consists of two IPMC samples joined by a glue layer, can sense its own deflection. The exploration into the performance of the IPMC sandwich includes research into mechanical resistance to deformation, methods of increasing the sensor signal, as well as the strength of the actuator. Research such as this is indicative of the interconnectivity of the sensing and actuation properties of the material, and also to a general trend in the field, where the distinction between the sensing and actuation research has been waning.

It is not uncommon in the literature for advancements in both sensing and actuation to be discussed in a single paper. Furthermore, it may be anticipated that at some point in the near future that models encompassing both behaviors may be made. It is already the case, that at least qualitatively, we understand the two phenomena to be linked through ionic motion.

Claudia Bonomo, a leading researcher in the field of non-linear IPMC phenomena, acknowledges that a lack of understanding at the molecular level is the main limitation barring a complete model of the material [20]. Most of the leading researchers in the field including Shahinpoor, Newbury, Bonomo, Tadokoro, Nemat-Nasser, etc., have been working actively to identify the underlying principles guiding the behavior of IPMC artificial muscles [8]. The most recent papers present the actuation characteristics of IPMC in terms of non-linear electrodynamics, chemistry, and mechanical properties. Their goal is to identify and mathematically model the many microscopic phenomena involved with IPMC in the hopes that they will yield a model that also describes the material on a macroscopic level. Researchers exploring white box models have been able to uncover some very important mathematical descriptions including ion transport, material strains, columbic forces, and

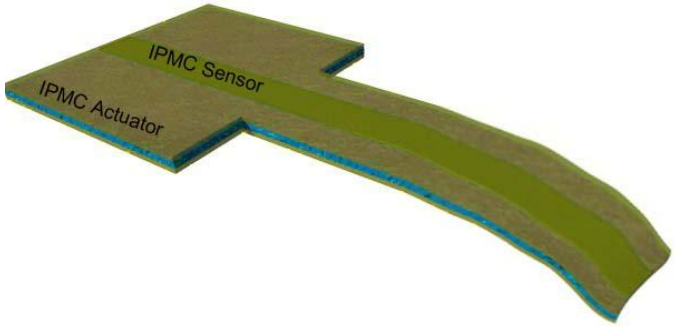


Figure 23. Sensor actuator sandwich

induced currents within the material. They are, however, hindered by the apparent complexity of the interactions of the various forces within the material.

There are, for instance, disagreements about what the main driving force for actuation is. Nemat-Nasser believes that actuation is largely due to the electrostatic forces that exist within the polymer as charges are redistributed. In this state, regions in the polymer that are high in cation density will extend, while cation depleted areas will relax [9]. At first this might sound like the more familiar idea that mobile cations carrying water molecules cause extension at the anode. However, it is important to note that the extension he is speaking of is not swelling due to hydration; it is purely a result of electrostatic interaction. This is not to say that Nasser disregards the mechanism of swelling entirely, but it is true that his theory regards the mechanism as secondary in most cases. This contradicts not only the mainstream thinking in the field but also many other current models. For instance, both Shahinpoor and Tadokoro have proposed continuum models that regard hydration as central to the actuation of IPMC [2]. It might be perhaps interesting to find that the continuum models of Shahinpoor and Tadokoro as well as the micromolecular model of Nasser agree quite well with experiment. Branco, in a paper extending the theories of both Nasser and Shahinpoor, explained that the main limitation of all phenomenological models in the field of IPMC is that they all currently rely on parameter identification [21]. In addition, the relationships between the forces in the models are assumed, i.e., they are reasoned to have the influence they do. The influence of one mechanism as compared to another has not really been determined, leading to the types of contradictions seen in the prominent theories. In addition, because

the models rely on system identification, the results they give are not universal to all IPMC but only to the single IPMC sample that was measured.

Thus far there is no commonly accepted model of IPMC actuation or a complete one. Lacking suitable guidance on non-linear behavior, most of the engineering force models within the field deal with closed loop control of IPMC. Their concern is effectively managing the non-linearity of the material using feedback loops to stabilize force output [22]. Their methods have bypassed the current gap of knowledge concerning force characterization and allow for the material to be applied for some engineering purpose. However, finite element models describing the action of EAP materials have presented new methods of analysis. It now seems reasonable to assume that results of the most current phenomenological models can be used as the basis of a finite element model. For one, the exact microscopic nature of actuation is not a concern as long as the models can match experimental data. Secondly, one of the major shortcomings of the models from a phenomenological standpoint is that they rely on parameterization. However, from an experimenter's standpoint, this means that the models are conformable to the IPMC in their lab.

2.2 **Finite Element Analysis (FEA) Modeling of IPMC**

Electromechanical Characterization of Non-Uniform Charged Ionic Polymer-Metal Composites (IPMC) Devices presents a 3D FEA model for IPMC actuation [23]. The paper is based on the force equations presented in another paper by one of the authors[21]. It presents a continuum model and an equivalent circuit model for IPMC

transduction based on electrostatic force. It suggests that the actuation of the IPMC can be modeled based on the repulsive electrostatic forces that exist between fixed anions in the IPMC. These forces arise as cations are forced from micellar clusters near the anode that the remaining negative charges repel one another. In this model, the clusters near the cathode, which attain a positive charge density, experience no force since these ions are not fixed to the polymer backbone. However, it seems highly unlikely that repulsive electrostatic forces in the anode regions alone can lead to deformation *toward the anode*.

In Multiphysics Modeling of an IPMC Microfluidic Device, the author presents a 2D FEA model for the heating of an IPMC strip [24]. In it he shows that the heating of an IPMC actuator is dependent on the conductivity of the electrodes, the magnitude of the voltage input, and the mechanics of mass transfer through the composite strip. The findings are that the actuation of the IPMC results in small changes in temperature for the IPMC over time. The results also show the electric field distribution through the thickness of the polymer layer is constant except in regions nearest the electrodes.

In Modeling IPMC Material with Surface Characteristics, the preliminary results of an ongoing Comsol FEA model describing electrical phenomena in IPMC actuators using Ramo-Shockley theorem is presented [25]. In particular, the authors discuss the effects of ionic motion considering variable resistance and capacitance in the electrodes for 2D and 3D models. The results show good correspondence to experimental data on the time evolving electrode voltage at a point and electric current through the IPMC. However, the paper states explicitly that the model needs improved meshing techniques before it is applied generally to the problem of deformation. The work expands on an

earlier paper where a 2D simulation of the tip displacement for an oscillating IPMC actuator [26]. The paper presents a useful way to model the electrokinetic migration of ions in Comsol. While the model was able to do a reasonable job of tracking the tip displacement of an IPMC actuator, the force equation that was used to drive the actuation was not tied to any physical phenomena. Instead, a parametric equation based on concentration changes was presented where the parameters were assigned fit experimental data.

CHAPTER 3. **IPMC DISTRIBUTED FORCE MODEL**

3.1 **Introduction**

The following material presents a force model that predicts the maximum force output of an IPMC actuator of arbitrary dimension. It consists of several coupled physics models corresponding to the coupled electrochemomechanical transduction processes that are collectively responsible for the observed macroscopic actuation of cantilevered IPMC. The end result is a computer simulation of the distributed force measurement, similar to the one taken using the IPMC force scanner introduced in Chapter One. Namely, in the simulation there is a solid domain representing the electroactive membrane being actuated into "contact" with a cylindrical domain representing the straw of a force transducer. The solution will be a force prediction that can be compared to the experimentally determined values.

The model is carried out using Comsol Multiphysics and Matlab. Comsol Multiphysics is an FEA software used for the simulation of problems involving coupled physical phenomena. The creation of models in Comsol depends on the addition of predefined interfaces, each dedicated to a single physical phenomenon, being added to a base CAD model. For instance, a problem involving the expansion of a metal due to an imposed current requires the addition of an interface involving structural mechanics and one involving electrical conduction to a geometric domain representing the metal. The physical equations contained in the interfaces can be changed as necessary to suit the needs of the problem. The base CAD model itself can either be imported using a supported filetype or designed directly in the software. Comsol contains a suite of solvers, as well as, many convenient options for graphical representation of final results.

Comsol also has a Matlab interface which allows the users to access all the functions of Comsol from Matlab programmatically, in addition to the functions and toolboxes contained in Matlab itself. For anyone familiar with Matlab, it is easy to see how this translates into a vast expansion of design capability. It additionally makes it convenient to run simulations iteratively while storing multiple results, and also, gives the user greater flexibility and control over how the simulations are carried out.

Sections 3.2 and 3.3 cover general processes for creating and meshing IPMC geometry. As will be seen, the complete IPMC distributed force simulation described in section 3.4 is composed of several smaller models designed to reduce model complexity and computational load. Each of the models has a section dedicated to its theory and implementation. The sections of 3.4 are listed below with a brief description for the reader's reference.

- 3.4.2 Comsol/Matlab electrochemical model

  The first step is the creation of a base model that yields a history of ionic concentration through the thickness of the IPMC as a function of time and input voltage. This model is based on a single easily meshed cubic element that can provide an accurate look at the evolving migration of ions under a given voltage signal. It is created using both Comsol and Matlab via the Comsol/Matlab interface.

- 3.4.3 Electrical model for arbitrary shapes

  The second step is the creation of an electric model that is capable of predicting the distributed electric field for an arbitrarily shaped IPMC

actuator based on what electrical input it is receiving. This chapter outlines the CAD modeling of an IPMC actuator, implementing and solving the model in Comsol, and exporting the solution to Matlab.

- 3.4.4 Matlab IPMC force model

The results of two previous models are sent to Matlab where, based on the distribution of the electric field, the ionic concentration distribution in an arbitrarily shaped IPMC is predicted. Given the distributed concentration, Matlab is used to predict the stress field in the material using a novel force equation. The theory behind the force equation is discussed in this section. Finally, the stress field is processed and stored in a file for later use in the distributed force simulation.

- 3.4.5 Comsol/Matlab distributed force simulation

A stress strain simulation is set up in Comsol that uses the stress field predicted by the IPMC force model. The simulation involves the force experienced by a force transducer in contact with an IPMC actuator. The model is programmed to run iteratively in Matlab. Each successive run returns a force prediction at a different location on the IPMC. Ultimately, these collective simulated measurements produce a distributed force prediction that can be compared to the force maps measured by the IPMC Force Scanner.

3.2 **CAD Modeling of IPMC**

CAD modeling of IPMC requires the creation of three domains corresponding to the Nafion polymer and the two noble metal electrodes. In 3D modeling, the domains that need to be created are solid geometries. It is possible to create the geometry in any traditional CAD software that allows for the creation of .stl files which can be imported directly into Comsol. However, the Comsol environment includes a drawing mode where solid geometries can be created directly. There are two ways that this can be accomplished. For simple geometries, such as rectangular and cylindrical shaped IPMC, the user can access GUIs from the drawing menu that contains fields describing the dimension of the geometry and its position. Once the fields have been completed, the user accepts the geometry and a solid domain is created automatically. To create the entire IPMC, which is a composite sandwich, the user will be required to enter the GUI three separate times defining each layer at the appropriate height.

For more complex geometries, the user enters a 2D drawing mode by first creating a work-plane and defines 2D geometries that are extruded to create the layers. Once a work-plane has been defined, Comsol automatically switches into a 2D work environment. In this mode, the option to create points, lines, simple 2D shapes, and Bezier curves becomes available. Using these tools any of the complex geometries that have been written about in the literature including hands, fish fins, tadpoles, and bird wings can be created [27] [28]. The process of creating IPMC models with complex geometry is simplified because the electrode and polymer layers tend to have matching profiles with regard to a plane bisecting the actuators thickness. This means that for most IPMC, a single profile is created in the work-plane and is extruded three times at varying

heights and thicknesses to create the layers. For instance, to create an IPMC shark pectoral fin, one creates a work-plane and draws the profile of the fin onto it using a Bezier curve. Going into the draw menu and selecting extrude, the user enters the thickness of the first electrode and its height. The user then selects the same profile and again extrudes it using the values of thickness for the polymer layer, selecting the height that places it squarely on the first electrode layer. The final electrode is extruded on top of the first two layers using the same process.

In some specialized IPMC designs it may be desired that the model of an IPMC actuator contain holes, multiple fingers, or electrodes that are segmented. These additional features of the IPMC are used to give a specific actuation profile or add degrees of freedom to the actuator. Segmented electrodes can also be used as sensing channels on IPMC where the deflection can be sensed by monitoring the resistance in the channels. In all these cases, it is desirable to have a method of modeling these designs in CAD software so that they can be simulated. Usually the most efficient way of modeling IPMC with holes or segmented electrodes is to create a base model and then subtract geometry from it. This can be accomplished using "Create Composite Objects" tool in the Comsol draw menu. This tool allows for Boolean manipulation of geometry such as addition or subtraction of domains that are in contact. To create a rectangular IPMC with a hole, for instance, the user first builds the rectangular IPMC using either of the two methods listed above and then creates a solid cylindrical domain that intersects all the layers. Next, the user enters the "Create Composite Objects" tool and lists the names of all the solid domains that make up the IPMC separated by "+" in the formula bar. Finally, the user enters the name of the cylindrical domain following a "-." This signifies

to Comsol that the cylindrical domain is to be subtracted from the domains that make up the IPMC. By selecting "keep internal boundaries," the individual layers that represent the electrodes and the polymer remain intact after the Boolean operation is complete. This same process has been used to model IPMC with segmented electrodes, although the process can be much more involved depending on the complexity of the geometry.

3.3 **Meshing the IPMC**

Meshing the IPMC is probably one of the greatest challenges concerned with the modeling of IPMC. The geometry must be meshed in such a way that the electrical and chemical gradients are being represented faithfully. Unfortunately, these gradients are extremely high and occur through the thickness of the IPMC. The thickness of a typical IPMC might be on the order of two hundred microns, whereas the length and height may be on the order of a cm. An IPMC finger of dimension 5w x 15l x 0.2h mm represents a dimension mismatch of two orders of magnitude for the polymer and three orders of magnitude for the electrodes. Due to the extreme flatness of the solid domains and because it is favorable, and sometimes necessary, to avoid elements of high aspect ratio, capturing the necessary gradients might require that the mesh has an unreasonable number of elements. This results in long solution times or out of memory errors without the aid of a supercomputer. It is therefore important, if possible, to develop a solution or alternative method to the problem of meshing the thin geometries present in IPMC actuators. In section 3.4 of this chapter "IPMC Distributed Force Model," such an alternative method is presented that avoids meshing the extremely disproportional

electrodes. For now, two general methods of meshing thin geometries that have been used successfully to model IPMC will be discussed.

The Comsol Multiphysics online knowledge base contains a solution titled "meshing thin geometries." It discusses a couple of methods to handle the modeling of geometries that are excessively thin such as IPMC actuators are. The first method is a rectangular swept mesh. This type of meshing is accomplished by creating a rectangular grid on one face of the solid geometry of interest. The rectangular grid is swept across the geometry to create a mesh with solid rectangular elements. The swept mesh works well in situations where the IPMC is a simple rectangular prism but not for more complex or segmented geometries. It also does not work well in simulations such as the one that concerns this thesis, where a solid cylindrical domain will be added for the purposes of performing force contact studies. The second method discussed in "Meshing thin geometries" works much better in this case.

In order to mesh IPMC with complicated geometries, the best method is a scaled mesh using the free mesh parameters GUI. This method scales the domain of interest by a multiplier and meshes the domain using tetrahedral or triangular elements before returning the geometry to its original dimension. In the case of the IPMC actuator, it is the thickness of the IPMC that will be scaled by a factor. It is important that the factor not exceed ten as a hard rule not just for IPMC but for any model. The scaling results in elements that are flatter than normal. The goal is to get the largest number of elements through the thickness (>8) while simultaneously limiting the overall number of elements to the tens of thousands. This ensures that the gradients are captured with fidelity and the overall model remains easily solvable. This is accomplished by experimenting with

settings in the free mesh parameters GUI where the user can either specify the values for various parameters or select from several preprogrammed options that produce elements of varying size. The preprogrammed options are selected from a list box that contains options simply named: normal, course, fine, extra fine, etc. In most cases, the user will be able to select one of these options along with a scaling factor and experience no problem.

For the modeling of the transducer straw, scaling the geometry before meshing will be disabled since in that domain it is easy to obtain well shaped elements. For this domain, the user will simply enter the free mesh parameters and select one of the preprogrammed size options in the list box. It is important to consider the size of the elements on the surface of the IPMC when selecting a mesh size for the transducer straw because the faces of these two domains are necessarily in contact. If the user commands that the sizes of the elements for the two domains differ greatly, the meshing process will fail. A good rule of thumb is to keep the meshes within two preset sizes of one another.

3.4 **IPMC Distributed Force Model**

3.4.1 Model overview

The goal of the model is to describe the force output of an IPMC as a function of its chemical, mechanical and electrical properties. Specifically, there is interest in simulating the effects of geometry and scaling on the force response of the IPMC fingers used in microgrippers. As can be seen the entire process is taken in steps (Figure 24). In this section, the parts of the model will each be discussed in turn starting with an

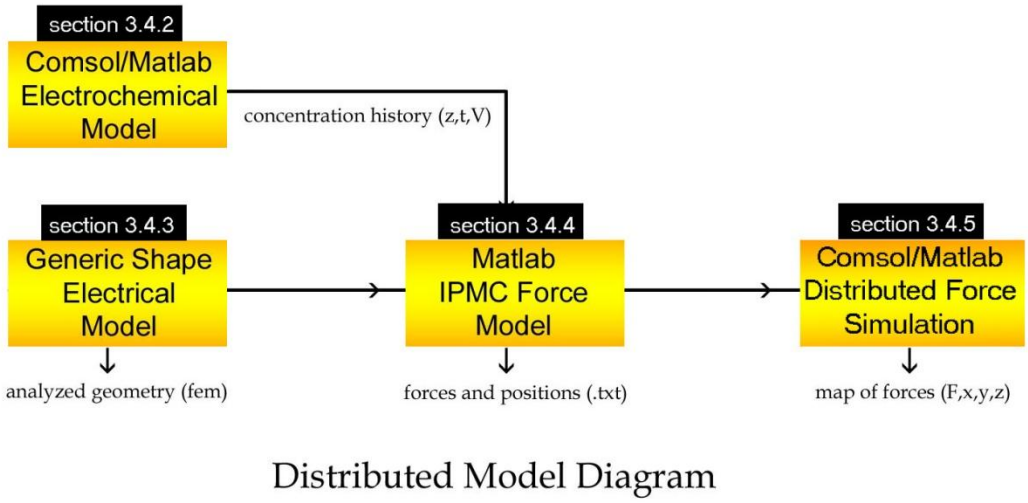


Figure 24. IPMC distributed force model schematic

electrochemical model that describes the induced redistribution of cations due to an imposed electrical field. Next, a model for predicting the electric field's distribution in an arbitrarily shaped IPMC actuator is presented. The result of this model will be passed to Matlab where the stresses in the material will be predicted. The process of passing the electric model's results into Matlab and predicting the stress field will be discussed in the third part of this section. In the final part of this section, Comsol and Matlab will be used for the purposes of simulating a distributed force measurement on an IPMC actuator.

3.4.2 Comsol/Matlab electrochemical model

3.4.2.1 Introduction

In this first part, the process of modeling the time dependent migration of cations under an imposed electric field for the purpose of force modeling will be discussed. Before proceeding with the description of the model, a few words will be said about the form of the model and the relation of its form to its purpose.

The form of the model is a simple 180 micron cube representing the electroactive Nafion layer. At this point the electrode regions have been omitted. The focus is on capturing the distribution of cations in the material as accurately as possible. The shape was chosen because it is easy to mesh with minimum elements. As was discussed before, it is important that the mesh through the thickness has at least eight elements to capture the cation concentration gradient that forms during actuation. It is especially important that the two distinct boundary layers that form near the respective electrodes are captured as accurately as possible. The first is a thin layer very high in cation concentration near the cathode. The second, a relatively thicker boundary layer that is completely depleted of cations near the anode. Since the remaining polymer remains essentially neutral through the actuation cycle, it is those two regions in the polymer that are of the most import, since the changes in chemistry that are responsible for the mechanical motion of the bulk actuator are present only there. It is therefore vitally important that the IPMC block be meshed with quality elements, especially near the cathode and anode regions.

As a general theme in the modeling of IPMC in this thesis, where multiphysics are involved, the geometry is kept simple and conversely, where the geometry is arbitrary, multiphysics are avoided. This keeps the models computationally efficient and easily solvable. In this case multiphysics is involved, DC conduction and electrophoresis, so a simple easily solved geometry was used rather than jumping immediately to a complicated geometry. The goal at this point is to capture the distribution of cations under an arbitrary DC signal (i.e., sine, square, triangle, and constant) as a function of time. The responses will be cataloged for later use in the Matlab Force Model where they will be applied to an actuator of arbitrary dimension. It

is important to note, the assumption that allows the result from this model of simplified geometry to be generalized is that the electrodes are sufficiently conductive that the only significant variance in the local electric field is through the material and not in any other direction. This means that locally the direction of motion for the cations is directly through the material, as if the voltage across the electrodes were constant at every point on the actuator.

Once the electrokinetic model is solved in Comsol for one instance, it will be converted into Matlab code to be run iteratively. This is done so that a catalog of responses is recorded at voltage levels less than the input voltage. They will be used to simulate the response of the IPMC actuator at points that are some distance from the point where the voltage has been applied. At these distant points the voltage signal will have the same phase as the input signal but will have diminished in strength by some amount. So if, for instance, the model involves the simulated response of an IPMC actuator to a 2V sine input, the model will be run iteratively in Matlab to get the responses for the values between 0-2V sine wave inputs. This ensures that the ionic response is captured for every possible input value experienced by points on the IPMC no matter where that point lies. The collection of responses is stored as a matrix in Matlab that can be referenced for use in force calculations.

3.4.2.2 Theory

The interest is in simulating the electromigration of cations under an imposed electric field through a porous medium. This is modeled using Comsol's Electrokinetic Flow application mode. To predict the transport of charged species through ionic solutions, it uses the equation

$$R = \delta_{ts}\frac{\partial c}{\partial t} + \nabla \cdot (-D\nabla c - zu_m Fc\nabla V + c\boldsymbol{u}) \tag{3.1}$$

where R is a reaction term, D is the diffusivity, c is the concentration, z is the charge number, u_m is the mobility, F is Faraday's constant, and u is the initial velocity of species. The bracketed term is the Nernst-Plank equation for ion transport, used to model the flux of cations. It contains terms describing diffusion, electrophoretic migration, and fluid velocity for the surrounding medium. In this case, the *cu* term is zero since the medium containing the cations is not a flowing liquid and is essentially fixed.

The problem involves solving equation 3.1 for the case of an isolated domain subject to an electric field. This means that the system is conservative with respect to the number of cations. As a result the R term is equation 3.1 is zero. In addition, every boundary will have the insulation condition seen in equation 3.2, where no net transport occurs normal to the surface of a boundary.

$$\boldsymbol{n} \cdot (-D\nabla c - zu_m Fc\nabla V + c\boldsymbol{u}) = 0 \tag{3.2}$$

The term ∇V in equation 3.1 is solved for using the Conductive Media DC application mode. The application mode combines Poisson's equation and Ohm's law in the single equation,

$$-\nabla \cdot (\sigma \Delta V - J^e) = Q_j \tag{3.3}$$

where sigma is the conductivity, V is the voltage, $J^e$is the externally generated current density, and $Q_j$is a current source. The term Q_j is zero since the Nafion does not generate any current during actuation.

The top and bottom of the domain representing the actuator have an electric potential boundary condition. These are the boundaries that contact the electrodes in an

$$V = V_0 \tag{3.4}$$

IPMC actuator. This is given simply in equation 3.4. Usually, one of the boundaries will be a ground condition with $V_0 = 0$. All the other boundary conditions are set to electrical insulation using equation 3.5. These are the edge boundaries that either contact the open

$$\boldsymbol{n} \cdot \boldsymbol{J} = 0 \tag{3.5}$$

air or another section of actuator. In the first case, there is no current flow across the boundary in either direction. In the second case, the assumption is that there exists symmetry with respect to the potential on either side of the boundary. This is reasonable since the potential is not expected to vary locally in any appreciable way.

3.4.2.3 Model Overview

In section *3.4.2.4 Modeling* the process of creating an electrochemical simulation for IPMC will be discussed in detail. The following list contains the major activities involved in the creation of the simulation. Each activity is listed in the order it will be discussed.

1. Create geometry using Comsol's CAD tools
    - Create a 180 micron cubic domain. This domain is representative of the Nafion layer an IPMC actuator.
2. Add physics to the base geometry model

- In the Comsol environment physics is added to the base geometry model through the addition of application modes. Each application mode contains equations pertaining to a single physical phenomenon.
- Add the Conductive Media DC application mode to the model. This application mode simulates the electric field in a conductor.
- Add the Electrokinetic Flow application mode. This application mode simulates the redistribution of ions due to an imposed electric field.

3. Set the subdomain settings and boundary conditions
    - The subdomain settings contain the material definitions for Nafion. It is also where the two application modes become coupled through a shared variable. The subdomain settings for each application mode must be populated.
    - The boundary conditions for each application mode need to be defined. The voltage input driving the IPMC actuation is input at this point.
4. Mesh and solve the model
    - The model is meshed using Comsol's free mesh parameters.
    - A solver is selected and the model is solved.
5. Visualize and inspect the results
    - Methods for postprocessing the results of the model and visualizing the results are discussed.

6. Export the model into Matlab
    - The model is saved and exported into Matlab as an m-file. A connection between Comsol and Matlab is created so that the model can be modified in the Matlab environment.
7. Add features to the model in the Matlab environment
    - The model is converted into a Matlab function that can accept inputs and return outputs. (Appendices A1 and A2)
    - The model is set to perform parametric sweeps over many voltages. This is used to create a history of ionic concentrations in an IPMC for a given voltage signal.
8. Export a file containing the ionic concentration history of an IPMC
    - A history of ionic concentration is exported from Matlab as a text file that will be used for force calculations. (Appendix A3)

3.4.2.4 Modeling

The model begins with the creation of a single domain representing the electroactive polymer. To create the geometry, the method described in section 3.2, direct solid modeling, will be used. The method is used to create a single solid box of dimension 180 l x 180 w x 180 h in microns using the drawing interface in Comsol. The result is a cubic region representing a section of an IPMC actuator minus the electrode regions (Figure 25).

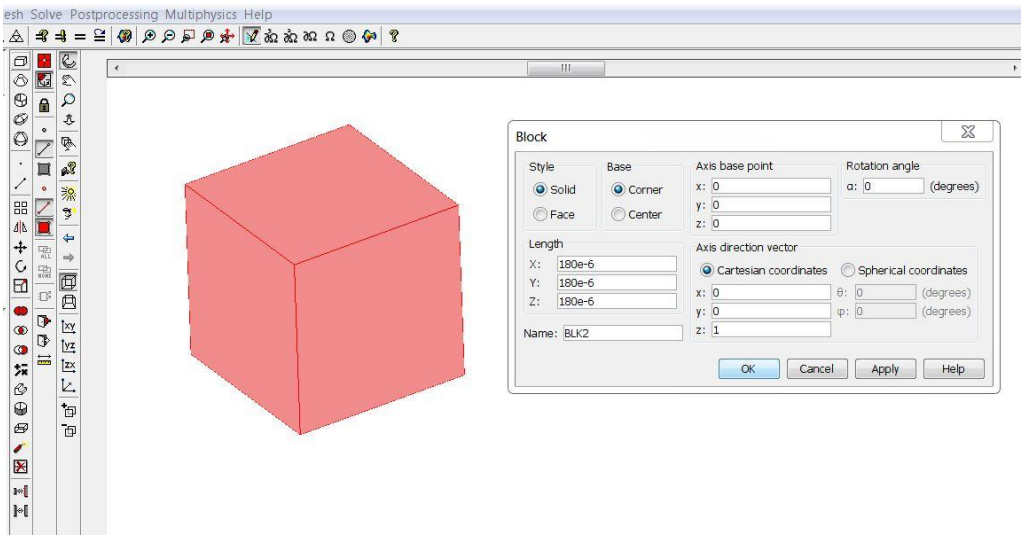


Figure 25. Creation of a cubic domain

Once the geometry has been created, the Conductive Media DC application mode is added to the model by entering the model navigator and adding it to the multiphysics list (Figure 26). This application mode solves for the electric field in a conductive media. Knowledge of the electric field will be necessary for solving the electromigration problem. The Electrokinetic Flow application mode also needs to be added to the model. This application mode predicts the redistribution of cations given an applied electric field. Whenever more than one application mode is being used, the multiphysics mode should be toggled on in the model navigator and it is also important to indicate to Comsol which is the ruling application mode. This helps Comsol decide which solver to suggest for the problem type. In this case, the ruling application mode is the Electrokinetic Flow, so under the "Ruling application mode" list box on the bottom right hand corner of the model navigator window the electrokinetic flow option is selected. With multiphysics enabled and the proper application modes selected, the model navigator is exited so that the main window can be seen. On the left hand side of the main window is the model tree. It shows the geometries that are present in the model with their associated physics

listed as collapsible sub-branches. For this example, there is only one geometry of interest and two physics modes present in the model tree. By single clicking the Conductive Media DC mode it is highlighted and now becomes the active mode. This means that all the options that are available in the main menu bar now apply to it. As will be seen, in general, modeling will proceed from the left to the right along the main menu bar. Namely, the geometry will be drawn, the physics defined, the geometry meshed, the problem solved, and then the information from the model visually represented.

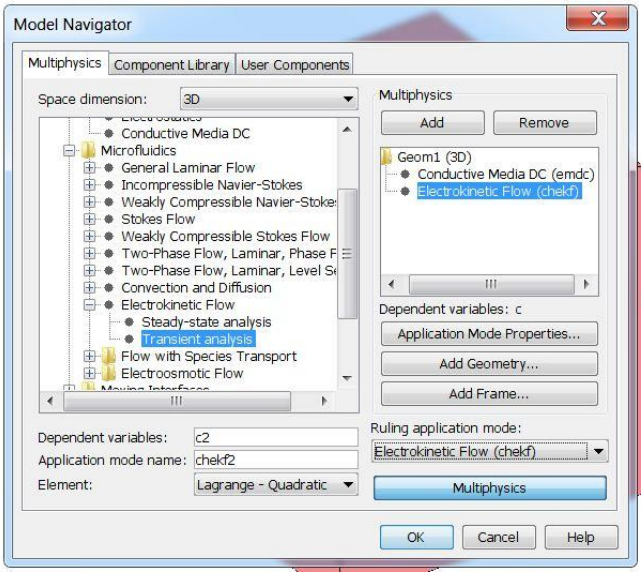


Figure 26. Comsol's model navigator GUI

Since the geometry has already been defined, the next step is to define the physics. Under physics in the main menu there are two options named subdomain settings and boundary settings (Figure 27). First, enter subdomain settings. The settings will be associated with Conductive Media DC since that is the active mode. In this GUI, the user defines values that are associated with the electrical properties of the domains of interest (Figure 28). In this case, only the electrical conductivity of the Nafion needs to be defined since there is only one domain

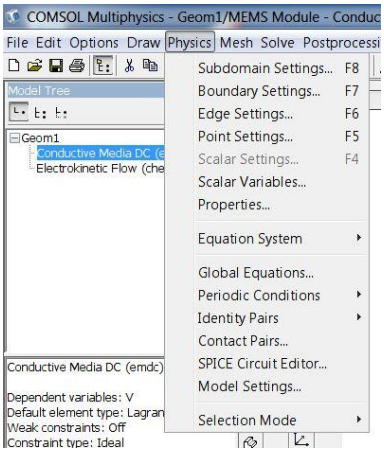


Figure 27. Physics menu

and no current sources. The equation that is being used to solve the problem is shown in the top of the subdomain settings GUI. In this case it is Ohm's law.

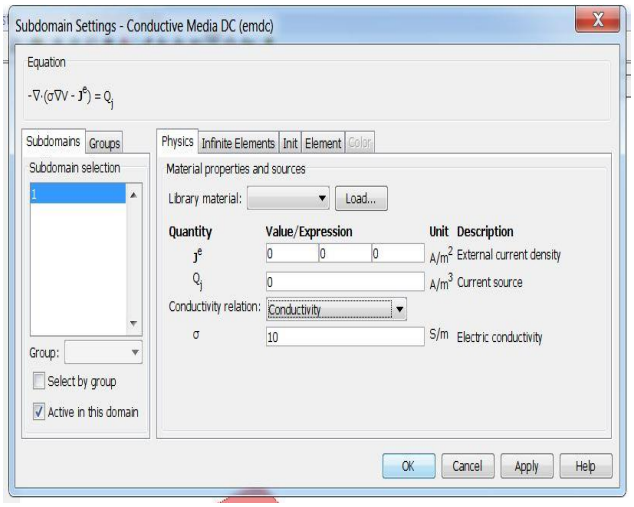


Figure 28. Conductive Media DC subdomain settings

Once the material properties have been successfully defined, the user enters the boundary settings GUI (Figure 29). On the right side of this GUI is a list of boundaries that can be selected by single click. Once selected, the name of the boundary becomes highlighted in blue indicating that the modifications being made apply only to it. The user proceeds by visiting the settings of each of the boundaries and selecting a condition that applies to it. In this case, there will be two voltage conditions on the top and lower boundaries of the cubic domain and electric insulation on the four sides. The sides can be selected all at

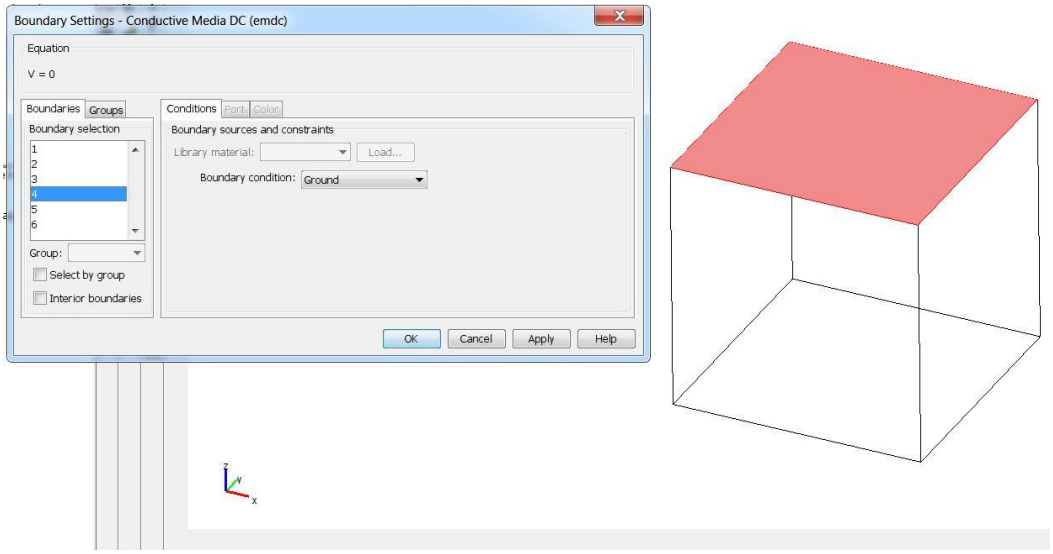


Figure 29. Conductive Media DC boundary settings

once by holding the control button and clicking each respectively. As the side boundaries are selected in turn, they become highlighted in pink in the visualization in the main window indicating the selection. Directly to the right of the list of boundaries is a drop-down list box entitled boundary conditions that has all the available conditions listed under it. To define the conditions at the side wall the "electric insulation" option is chosen from the list box. Next, the name of the top surface is selected from the boundary list and the boundary condition "ground" is applied to it by selecting that option from the list box. Finally, the bottom surface is selected and the "electric potential" boundary condition is selected. For this option, a box appears where a numeric value for the voltage can be entered. For IPMC this will be a value between 0-3V. The domain is now configured as a dielectric where the single domain is the insulator.

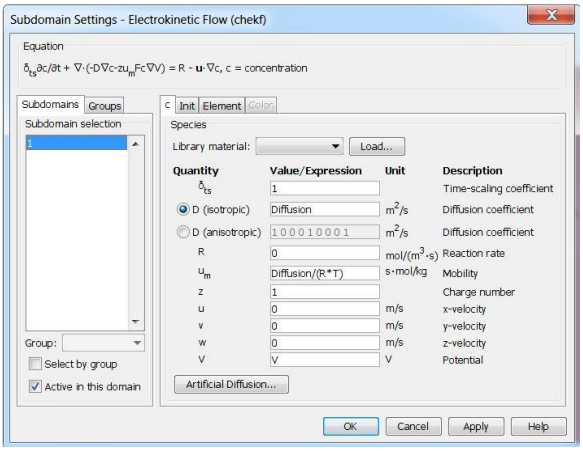


Figure 30. EK Flow subdomain settings

Having defined all the necessary material constants and boundary conditions for the Conductive Media DC application mode, the user selects the Electrokinetic Flow application mode from the model tree. This time when the subdomain settings dialog box is opened, the constants that need to be defined relate to charged species that are present in the material (Figure 30). These values include: mobility, diffusion coefficient, initial velocities, initial concentrations, and voltage potential. The two application modes, Electrokinetic Flow and Conductive Media DC, are coupled through the entry in the voltage potential box. The value entered is the

variable “V” which stands for voltage and is a global variable produced by the Conductive Media DC application mode. Again the equation is visible at the top of the dialog box. The equation that is used for Electrokinetic Flow contains terms for electrophoresis and diffusion. Since there is only one domain, once the correct values have been entered in the dialog box, the dialog can be closed. The boundary settings dialog can now be opened, however, since the default boundary condition for Electrokinetic Flow is insulation and this is the correct value, there is no need to do this. The model is now ready for meshing.

To mesh the model the scaled mesh described in section 3.3 will be used. From the main menu bar under the mesh heading is free mesh parameters. The free mesh parameters dialog box has navigation tabs (Figure 31). The three

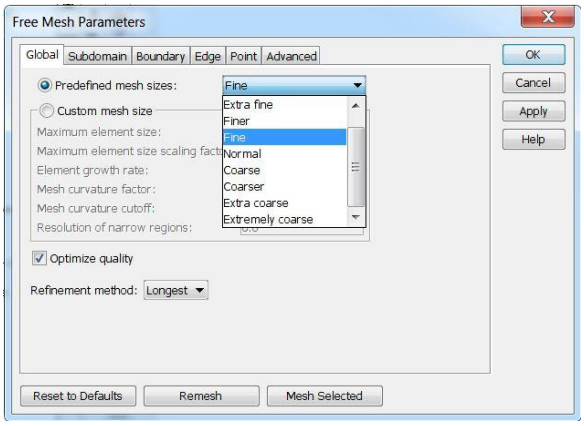


Figure 31. Free mesh parameters

tabs that are of interest are the global, subdomain, and advanced tabs. The dialog opens by default in the global tab. This tab contains a list box labeled “Predefined mesh sizes” where the user can select meshes with preset values. Although, for most cases, including this one, the preset sizes will be adequate, there is also a custom mesh size option where the user can create a custom mesh. For now, the user simply selects a value from the predefined mesh sizes or leaves the value at normal. Next, clicking into the subdomain tab, the user sees a list of the subdomains that are present in the model. Normally, the highlighted subdomain in this tab is the active one which the settings in all the other tabs

are applied to. However, since there is only one subdomain in this model, all the changes apply only to that subdomain automatically. After highlighting the lone domain in the model by clicking on it, the user proceeds to the advanced tab where the option to scale geometry is listed. Each orthogonal direction in the model has its own input box so in the "z-direction scale factor" input box the user enters a multiplier between one and ten. This is done to increase the number of elements through the thickness of the Nafion layer where all the important physics occurs. It is important to note that even though the input box will accept values above ten, such values are not to be used; Using scaling factors above ten results in unreliable results. Now that all the relevant tabs have been visited and the proper values entered the user can now select "mesh selected" at the bottom of the free mesh parameters dialog box. This meshes the subdomains indicated in the subdomains tab. Comsol will indicate when the meshing has completed.

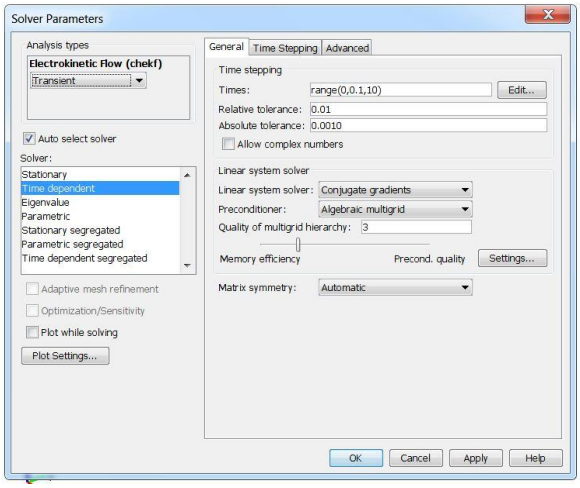


Figure 32. Solver parameters GUI

The next step is to select a solver. Under the "Solve" item in the main menu bar is "solver parameters." The solver parameters dialog contains information about the type of analysis, the solver being used, and the solver settings (Figure 32). When the dialog is first opened the "auto select solver" checkbox is automatically selected. This option selects a solver based on the type of analysis being undertaken, as well as the ruling application mode. In this case the type of analysis is transient and the ruling application

mode is Electrokinetic Flow. Comsol automatically selects solvers based on problem type, a time dependent solver with conjugate gradients as the linear solver. If desired, the user has the option to change the linear solver at any time by selecting a new solver from the "linear system solver" drop down list box. After selecting a solver the user sets a time range for the simulation. This is set in the "times" dialog where the user enters the length of the simulation in the format "range(start, increment, end)." So for a four second simulation with half second increments, the user enters "range(0, 0.5, 4)." One other important feature in this dialog is the tolerance that can be increased to help ease convergence, if necessary. After the user accepts all the settings the solver parameters dialog closes and the user selects the equal sign in the toolbar at the top of the screen. This starts the solver and automatically opens a progress dialog. In this dialog, there is progress bar that indicates how far along the solution is. In addition, there is a convergence tab that when opened shows a graph of the convergence behavior of the solver, so that the user can judge how well the solver is handling the problem. Assuming all the models parameters have been correctly set, Comsol will indicate that the solution is finished after several seconds and the solution will be displayed graphically in the main window.

For the purposes of this section, there is not a lot of concern with the many methods of visualizing the solution to the problem in Comsol beyond what is necessary to verify that the solution is valid. As will be seen shortly, the real interest is in moving this basic model into Matlab where programmatically we can do parametric sweeps. However, it may be worthwhile to give a brief summary of the visualization functions so that any potential user can be aware of them. All plotting options are available through

the postprocessing item in the main menu. The first item in the postprocessing drop down menu is plot parameters (Figure 33). This dialog controls the visualizations that occur in the main window in Comsol. Plot types available for the main window include: slice, subdomain, boundary, arrow, and deformed shape plots. Each plot type has its controls located in a separate tab. Clicking the tab, the user finds controls relating to the variable that is being plotted, the general appearance of the plot, and the units. Most of the options for these controls are in drop down menus where the user selects the option from a list of available options. Near the very top of every tab is a checkbox which activates the plot when checked. Though multiple plots may be activated at a single time, some may not be visible when others are present. For instance, if a slice plot and a subdomain plot are both active, only the subdomain plot will be visible. Other times plots may be used simultaneously for greater visual effect. Such is the case for the subdomain and deformation plots. This is especially true in the case when the user wants to see the actual displacement of the IPMC actuator during simulation and the concentration of ions at the same time.

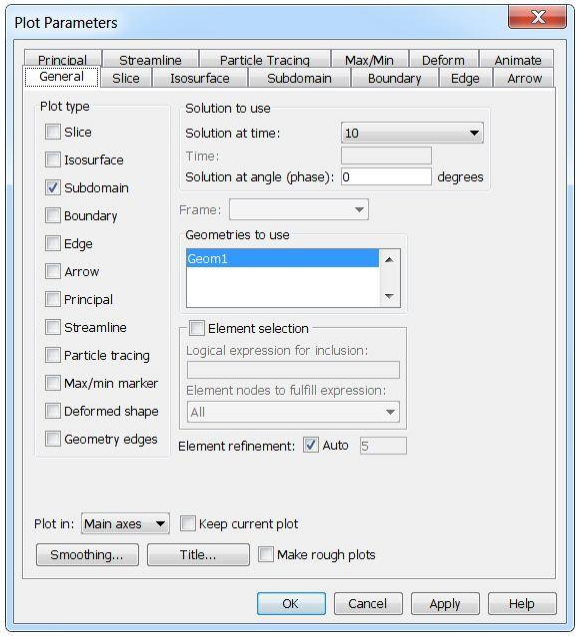


Figure 33. Plot parameters

Two additional types of plots under the postprocessing menu that are commonly used are the cross section plot and domain plot (Figure 34). Both of these dialogs can be used to create line plots for variables. As compared to the plot parameters, rather than the plot appearing in the main window, each time a plot is created a new window appears where the plot is displayed. These plots have menus at the top where the data can be exported, modified, or saved in many common picture formats. Since it is possible for multiple lines to be produced for a single plot, these types of plots are good for imaging

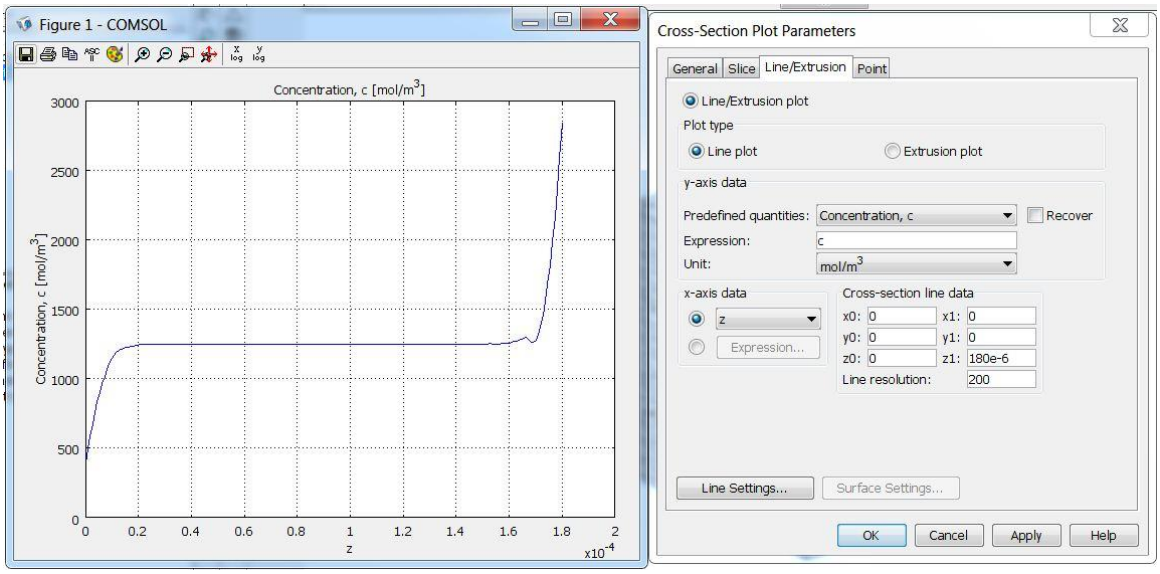


Figure 34. Cross-section plot parameters

the evolving state of cation distribution as the simulation progresses. They are a simple way to verify that the simulation is producing realistic results and that nothing unusual has occurred.

Now that setting up, solving, and imaging the model have been discussed, the model will be exported as a Matlab file. Note that *Comsol stores everything that has occurred from the time the model has opened to the time when the file is exported*. This means any changes applied to the model, even the erroneous ones that are later removed,

are stored in the history of the model by Comsol. So if, for instance, after successfully running the model, the user modifies the plotting options and creates 20 different plots, when the user exports the model, there will be code for 20 different plots. So it is important when designing the model for the purposes of export to be mindful of the process because otherwise the code can become quite convoluted and have large sections that need to be erased manually in Matlab.

Now, having said all that, there is a way to clear the history of the model, keeping all the geometry and parameters in place (losing meshes and solutions). There is a reset option under the file menu that serves this purpose. However, this option changes the format of the exported file. Namely, the definitions for the geometry contained in the Comsol model will be moved into a separate file. In some cases, this will not matter. For instance, the model now being discussed, where the parametric sweeps will not involve changes to the geometry. However, in a coming section, modifications to the geometry will be automated programmatically. This is much easier if the definitions for the geometry are contained directly in the Matlab code because additional code will not need to be written to call an entirely separate file.

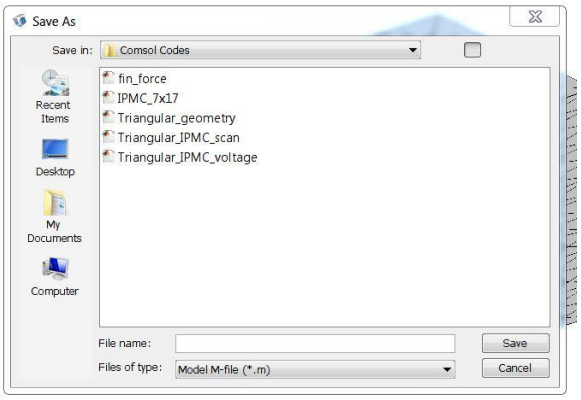


Figure 35. Save-as GUI

Exporting the Comsol model into Matlab is accomplished simply through the "save as" command. When the save as dialog appears (Figure 35), the model needs to be given a name, a storage location, and the file type option changed to "Model M-file." This will produce a file that is editable

and able to be run by Matlab. There is one caveat, however. The functions contained in the exported file are not native to Matlab and require that a connection be set up between the two programs. So trying to run the file in Matlab alone will immediately return an error. To set up the connection, the user needs to go to File>Client/Server/Matlab>Connect to Matlab (Figure 36). This will open a new instance of Matlab that is in communication with Comsol. The new instance of Matlab will open even if Matlab is currently open. When Comsol and Matlab have connected, the command window in Matlab will indicate the success.

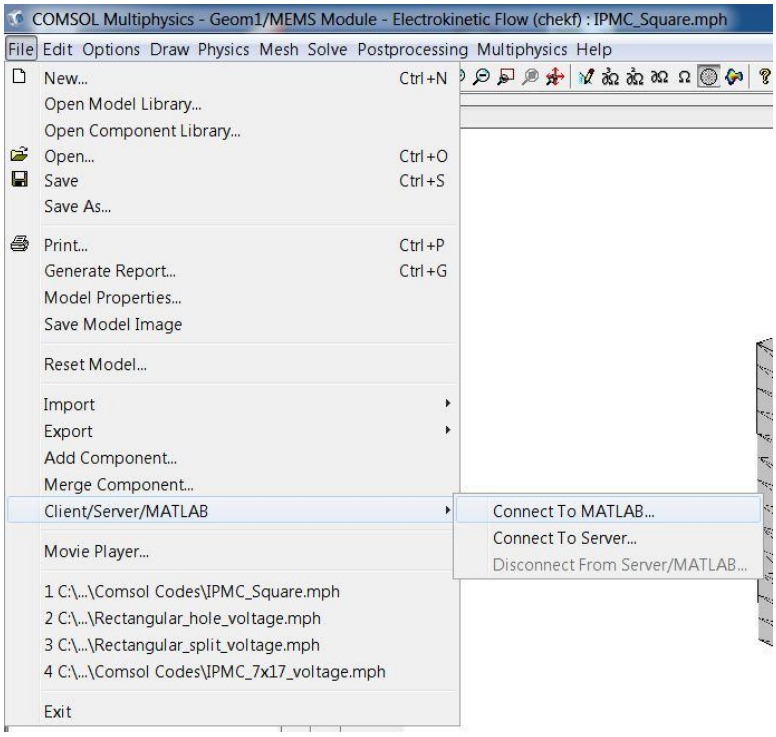


Figure 36. Comsol/Matlab connection dialog

When the m-file of the Comsol model is opened it has several blocks of code under headings that indicate whether the block contains data, parameters, or some operation. For instance, there is a block where the constants used in the model are located. Another block listing information about the Comsol version used to create the file and when the model was created. Processes such as creating geometry, meshing, and solving the model all have dedicated blocks. Initially, the code is not in the form of a function that can be called nor accept input variables from

the command window. Rather the file is a history of sorts, describing major events that have occurred since the model was created or last reset. Now say, for instance, that the user went through the process of solving the model once before realizing that something was incorrect. If the user makes a correction and modifies the model, that process will be evident in the m-file as excess code that needs to be eliminated from the file. There are other cases where there is excess code in the sense that a particular block is not necessary for the functioning of the program directly but has some use during coding. Such is the case when the user has code for multiple plots that are used to verify a solution. In this case, the plot's codes can simply be commented out when not needed.

Once Matlab and Comsol are in communication and there is an m-file version of the model, a program that will yield concentration histories for specific voltage signals can be created. As a first step, it is important to review the m-file and determine if any unwanted code is present and, if necessary, delete or disable it. Next, a variable containing a vector of solution times is created. This variable is used to indicate to the solver when solutions are returned, and also to create signals that evolve over time. This is useful for the creation of sine signals which are a very commonly used to drive IPMC actuators. A variable for voltage will also be created. This variable is also a vector and contains an evenly spaced series of numbers from zero to the maximum voltage. These voltage values will be used as the parameters for the parametric sweep. The final variable is an empty array that is used as a preallocated space for the storage of concentration values. The form of this array is Concentration(z, time, voltage).

Once these variables have been created, the remainder of the code can be placed inside a "for loop" whose number of iterations matches the number of elements in the

voltage vector. Next, the user changes the voltage signal. The block of code containing the description of the input signal is located under one of the headings "Application mode." As can be imagined, there are two application mode headings corresponding to the Conductive Media DC and Electrokinetic application modes that were programmed in Comsol. The two can be distinguished by referring to a line in the block containing the code "appl.mode.class =." The following string will either be "EmConductiveMediaDC" or "ElectroKF." The Conductive Media block contains two lines in succession with the variables "bnd.VO" and "bnd.type." The first, "bnd.VO," contains a series of comma separated numerical values for boundary conditions. "bnd.type" contains the types of boundary conditions each corresponding value in "bnd.VO" refers to. For instance, bnd.VO = {0,1,2} and bnd.type = {'V0','V','nJ0'} indicates that condition type V0 has a value of 0 ('V0'=0), condition type V has a value of 1 ('V'=1), and condition type nJ0 has a value of 2 ('nJ0'=2). In Matlab, 'V0' indicates a ground condition, 'nJ0' indicates electrical insulation, and 'V' indicates a voltage condition.

The variable controlling the voltage signal will be the value corresponding to 'V', a voltage condition, in "bnd.VO." There will only be one such condition given how the model was set up in Comsol. Once the value has been located it can be replaced with a variable string or a time dependent formula. In the instance where the user simply wants to increment through the voltage vector the variable "bnd.V0" will be modified to look like "{0, num2string(voltage(i)), 0}." In the case where a time dependent voltage signal is required the variable can have the form "{0, strcat(num2string(voltage(i)), 'sin(c*t)'), 0}" where the voltage input will change with each iteration and also evolve over time.

The next step is to add the vector of solution times to the solver's block of code. This block is located under the "Solve problem" header. The solution times for the solver are listed in a line containing the variable "tlist" followed directly by a comma. After this comma the name of the variable containing the solution times is added. For instance, if the solution vector is called "time," the final line will look like "'tlist', time, …."

Finally, there needs to be a way to extract and store the solutions. This can be accomplished using the function postinterp. The solution to the electrokinetic problem at the end of each iteration is stored in a structure called FEM. The function postinterp can reference the fem structure for a particular solution at a given point in space and in time. The format of the function is postinterp(fem, 'solution variable', [x y z], 'T', t). In this case, the solution variable of interest is the concentration which is reference through the variable 'c'. By making repeated calls using postinterp, a concentration profile for cations through the thickness of the IPMC can be collected. This can be accomplished by placing postinterp into two "for" loops. The outer loop increments by time steps returning the concentration profile for every solution time. The inner loop increments by z height returning the concentration values at points through the thickness. The results are stored in the concentration array according to the voltage, solution time, and z-height. The final array will be called by the IPMC Force Model to solve for the forces produced by IPMC actuators of varying geometry.

3.4.3 Electrical model for arbitrary shapes

3.4.3.1 Introduction

In the last section, the creation of a model that was able to predict the evolving distribution of cations through an IPMC actuator given a particular voltage signal was presented. Now the focus is shifted toward discovering how the voltage distributes itself across the electrodes of arbitrarily shaped IPMC. As was briefly mentioned in the previous chapter, the surface electrodes are resistive in such a way that the input voltage decreases in strength as the distance from the input point is increased. This must mean that the muscle's actuation must also be diminished by some amount, since the strength of the actuation is proportional to the input voltage. It is therefore important that any proposed force model account for this variance.

There is an additional reason for the interest in modeling the voltage distribution. In Chapter 1, segmented IPMC were discussed. These included deflection sensing IPMC which are actuators with small sensing channels scored around their perimeters and IPMC actuators with patterned electrodes. The former had areas on their surfaces that were entirely dedicated to resistance measurement and did not receive any voltage signal. The latter was capable of being driven by multiple independent voltage sources. These two cases cannot be modeled without accounting for the variation of voltage on the surfaces of the IPMC.

This section will cover the solution of an arbitrarily shaped IPMC with patterned electrodes in Comsol. First, the creation of the three domains representing the IPMC layers will be demonstrated using Comsol's drawing tool. Patterning of the electrodes

using Boolean geometry manipulation will also be shown. Second, the process of setting up, meshing, and solving the model will be discussed. Finally, the model will be exported to Matlab as a FEM structure. There the solution to the model can be extracted and incorporated into the force model.

3.4.3.2 Theory

The Conductive Media DC application mode that was used in earlier in section *3.4.2 Comsol/Matlab electrochemical model* is again applied. In that section, the application mode was used to simulate the electric field in a cubic domain representing Nafion and also as an input to the Electrokinetic Flow application mode. This time it will be used instead to simulate the potential distribution in the electrodes of an arbitrarily shaped IPMC actuator. The equations used by the application mode were already discussed in section *3.4.2.2 Theory* and they will not be repeated here. The geometry and boundary conditions, however, have changed somewhat. In this model there are three domains. A 180 micron thick domain representing the polymer Nafion sandwiched between two 10 micron thick domains representing electrodes. There are two voltage potential boundary conditions (equation 3.4) applied at the boundaries where electrical contact would be maintained. Usually, one of the boundaries will be some small voltage (1-3V) and the other will be a ground condition (0V). All other external boundaries will be set to electrical insulation (equation 3.5). This is where the IPMC meets the open air. The internal boundaries existing between the domains representing the Nafion and the electrodes have the continuity boundary condition.

$$-\boldsymbol{n} \cdot (\boldsymbol{J}_1 - \boldsymbol{J}_2) = 0 \tag{3.6}$$

This equation states the current is continuous normal to the plane of the interior boundary.

3.4.3.3 Model Overview

The model predicts the electric potential in an arbitrarily shaped IPMC actuator. The following list contains the major activities involved in creating the simulation. Each activity is listed in the order it will be discussed.

1. Create geometry using Comsol's CAD tools
    - Create three domains corresponding to the layers in an arbitrarily shaped IPMC actuator. The middle domain is 180 microns thick and represents the Nafion layer. It is sandwiched between two 10 micron layers representing the electrodes.
    - The layers are created using 2D geometry contained in work planes that get extruded to create 3D geometry.
    - Channels are created using the Boolean logic to manipulate geometry.
2. Add physics to the base geometry model
    - In the Comsol environment physics is added to the base geometry model through the addition of application modes. Each application mode contains equations pertaining to a single physical phenomenon.
    - Add the Conductive Media DC application mode to the model. This application mode simulates the electric field in a conductor.
3. Set the subdomain settings and boundary conditions

- The subdomain settings contain the material definitions for Nafion. The subdomain settings for the Conductive Media DC application mode are populated.
- The boundary conditions for the Conductive Media DC application mode need to be defined. The voltage driving the IPMC's actuation is input.

4. Mesh and solve the model
    - The model is meshed using Comsol's free mesh parameters or a swept mesh.
    - A solver is selected and the model is solved.
5. Visualize and inspect the results
    - Methods for postprocessing the results of the model and visualizing the results are discussed.
6. Export the model's solution into Matlab
    - The model's solution is exported to Matlab's workspace as a FEM structure. The solution contains the electric potential distribution for both electrodes.

3.4.3.4 Modeling

The first step in creating the Electrical Model is to create the geometry to be analyzed. First, a new 3D model is opened in Comsol. Although, the model starts in a 3D environment, the user will soon be switching a 2D workplane where a profile of the three layers will be drawn and extruded to create solid geometry. This method was

introduced in section 3.2, as 2D to 3D modeling is an efficient method for modeling of IPMC actuators of arbitrary dimension. The method is well suited to IPMC actuators because each of the layers (two electrodes and one polymer) has the same cross sectional profile. This translates into the creation of one profile that needs to be extruded at different heights to create each of the respective layers.

Once in a new 3D Comsol model, the user goes to Draw>Work-Plane settings. The work-plane settings dialog is used to create arbitrarily oriented planes that traverse the three dimensional space. The 2D profiles that are drawn on the planes can be extruded to create geometry. The work-plane settings has multiple tabs each containing a different method for creating the work-planes (Figure 37). For instance, the work-planes can be created parallel to the face of existing geometry, tangent to existing edges, or defined arbitrarily by the user. In this instance, the goal is to create the profile of an IPMC actuator in the xy plane and extrude it into the z direction. The first tab in the dialog is the "Quick" settings. Here the user simply selects from three of the coordinate planes and inputs an offset distance from the plane. So the user simply selects the "x-y" plane and inputs an offset of "0" meters. This closes the work-plane settings dialog window and

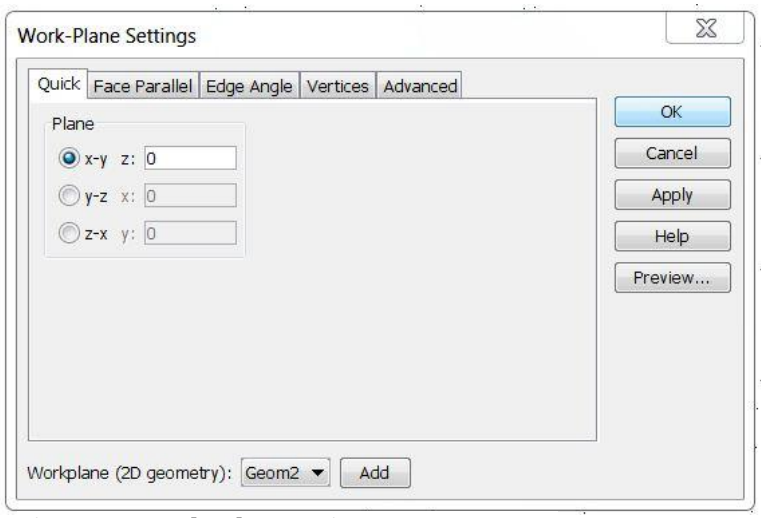


Figure 37. Work -plane settings

creates a work-plane coincident with the xy coordinate plane. The work-plane appears in the main Comsol window and geometry can be drawn.

As was discussed in section 3.2, several options are available in 2D drawing mode for creating and manipulating profiles in the work-planes. Four of the available options are GUIs for creating circles and squares. To create geometry using these options the user simply enters dimension and position parameters. The user additionally has the option to create profiles using points, lines, and Bezier curves. The remainder of the geometry creation tools in the 2D drawing mode relate to the manipulation of geometry in the work-plane. These are tools such as mirror, scale, move, rotate, and array. Finally, there is a "create composite object" GUI where Boolean operators can be applied to geometry (Figure 38). This tool when used in conjunction with the scaling tool is very useful for creating segmented type IPMC.

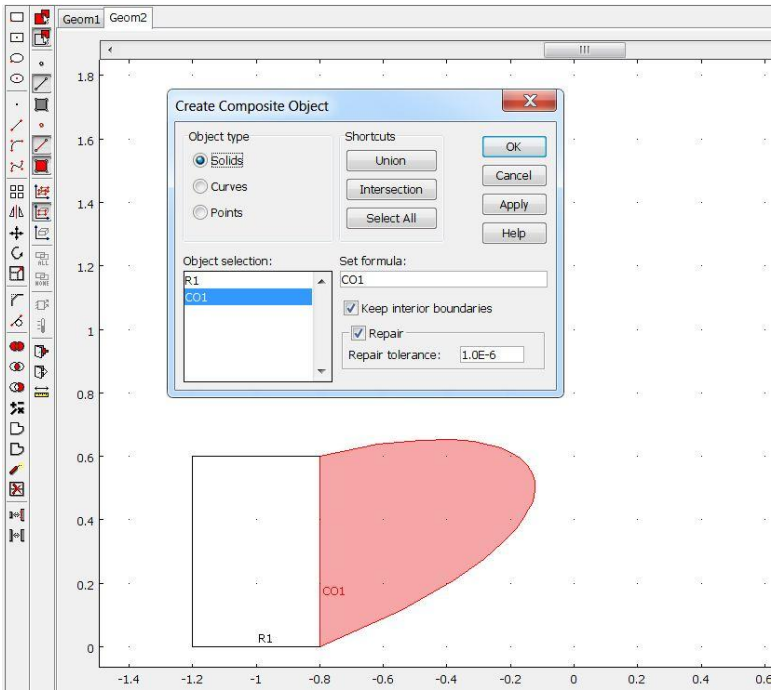


Figure 38. Create composite object

The first item to draw is the cross sectional profile of the IPMC. In the case that the IPMC has segments, they are ignored for now and will be added later. Once the

profile has been created, the user goes to Draw>Extrude. The extrude dialog will open prompting the user to select the profile to be extruded and to what depth (Figure 39). The user also has the option at this point to displace, rotate, and scale the object in the xy plane before extrusion. Selecting the actuator's profile, the user enters the thickness of the electrode in the distance parameter and clicks okay. This closes the dialog and displays the newly created 3D geometry in the main window. Now that the first layer has been take care of, the work-plane must be translated upward so that it rests on the top surface of the existing extrusion. The user again enters the work-plane settings dialog by clicking Draw>Work-Plane settings. This time when the dialog opens the user enters the dialog and selects the xy plane, the depth of the electrode is entered as the z offset. When the user clicks okay, the work-plane is displayed in the main window. Fortunately, Comsol retains profiles when new work-planes are created.

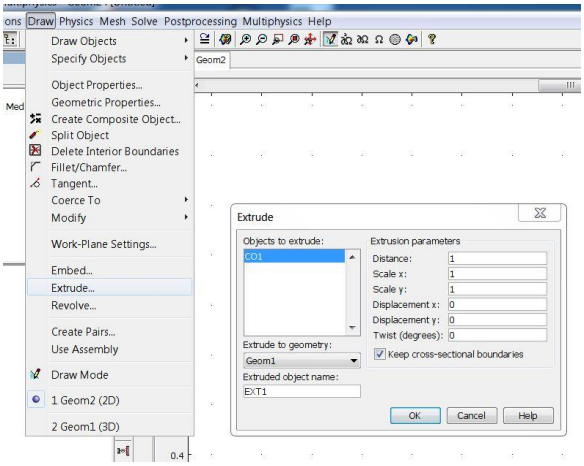


Figure 39. Extrude GUI

So the previously created geometry will not need to be redrawn in the new plane. The user simply extrudes the profile using the depth of the polymer layer as the extrusion distance. This creates a second thicker layer on top of the first, representing the polymer. The process of creating a new work-plane and extruding the profile is repeated to create a third layer representing the second electrode. The height of the work-plane will be the combined thickness of the polymer layer and one electrode layer. The extrusion distance

will be the thickness of the electrode. Having completed this process, the basic IPMC actuator is geometry is complete.

If the goal is to create an IPMC with segmented electrodes, the user can now do so by adding channels to the electrodes. To accomplish this, the work-plane settings dialog is revisited and the profile of the channels is added to a work-plane at the bottom surface of one of the electrodes. For simple geometries, like rectangles, the channel's profile can either be drawn in directly using lines or by subtracting similar shapes of different sizes so that the remaining overlap forms the channel. For more complicated geometries, the channel can be created using Bezier curves. Another option is to first duplicate the profile of the entire actuator to create a congruent profile and then scale the profile down so that its perimeter is coincident with the outer boundary of the channel. The user can then duplicate the scaled profile and shrink it so that its perimeter is coincident with the inner boundary of the channel. The two scaled profiles can then be subtracted from one another so that their overlap forms the channel. In any case, the channel's profile, once created, is extruded through one of the electrodes. The extrusion then gets subtracted from the electrode, segmenting it. The work-plane is then translated so that it lies on the lower surface of the second electrode. The process of extruding the channel's profile and then subtracting the resulting solid geometry is then repeated to create the second segmented electrode.

The general process of setting up the electrical model proceeds very similarly to the electrochemical model described in the last section. Namely, the geometry is created, the material definitions are entered, the boundary conditions defined, the geometry is meshed, and a solver is chosen. Consequently, this time the process of setting up the

model will be discussed in less detail with respect to the form of the Comsol user interfaces.

Now that the geometry has been defined, the user visits Multiphysics>Model navigator and adds the Conductive Media DC application mode to the 3D geometry. There will be no need to select a ruling application mode in the navigator this time because there is only one application mode in the model. By going to Physics>Subdomain settings in the main menu, material definitions can be added to the model. The relevant parameters that need to be defined are the conductivities. This can be accomplished by selecting, in turn, each of the subdomain names in the model from the subdomain selection list and entering the respective conductivity values in the "electric conductivity" dialog box. The user then exits the Subdomain settings and enters Physics>Boundary settings. The boundary settings can take on many forms depending on the interest of the user. In most cases, however, there will be two voltage conditions, two continuity conditions, and the remainder will be electric insulation. The two voltage conditions will be at the points where physical contact is maintained between the IPMC actuator and the device supplying an electrical signal. Usually, these conditions are applied along the thin boundaries at the edges of the electrodes. The continuity conditions are located at the interfaces between the electrodes and the Nafion. These conditions are the default in Comsol and do not need to be changed. The remainder of the IPMC actuator's surface has electrical insulation as its boundary condition. This is where the IPMC meets the open air.

The scaled meshing procedure described in section 3.3 and implemented in the last section, will again be used to mesh the three solid subdomains. The user enters the

"free mesh parameters" dialog and selects a domain, a predefined mesh size, and a scaling factor for the z dimension. Again, it is important that the scaling factor not exceed 10. Once the parameters have been set, the user clicks "mesh selected" to mesh the currently selected domain. The user repeats this process for each of the remaining domains. Normally the two electrode domains can be selected and meshed simultaneously since there dimensions usually match. Unlike the geometry in the previous section there are two extremely thin geometries representing the electrodes present. The user must be careful about checking the quality of the elements in this region to make sure that they are acceptable. There are two helpful tools for checking the mesh quality in Comsol both listed under the "Mesh" item in the main window. The first is the mesh statistics. This window lists information that can be used to judge the quality of the mesh including, the total number of elements of different types, as well as, the minimum element quality. The second is mesh visualization parameters. This dialog can be used to display element quality graphically in the main window with elements in different quality ranges appearing in different colors.

Once a quality mesh has been accomplished, the model is ready to be solved. The user enters the "solver parameters" dialog by clicking F11 or through the "Solve" item in the main menu. Comsol automatically selects a "conjugate gradients" solver with an "algebraic multigrid" presolver. The user can keep these settings or if necessary select another solver. The Comsol users' manual has a list of solvers arranged by problem type that can be of some help when selecting an alternate solver. The second tab in the dialog is labeled "stationary." Here, the user can adjust the tolerances and set the maximum number of iterations. This dialog also has a checkbox labeled "highly nonlinear

problem" which when clicked lowers the damping parameters and helps the problem solve. The user should try to first solve the problem without this box since it slows down the solver considerably. Once the solver parameters have been set, the user clicks okay and the dialog exits to the main window. Clicking on the equal symbol in main menu bar starts the solver. Assuming that the problem completes successfully and looks reasonable, it is always a good idea to go back and change the mesh slightly and resolve. This is a good way to check that the solution does not change significantly because of the mesh.

Having reached a solution, it can now be exported for future use in Matlab. This is accomplished by directly exporting the FEM structure that contains the solution. Once a link between Comsol and Matlab have been the user can export the file by pressing Control+F or by entering the File>export>FEM structure. The user will be prompted to name the structure before it is sent to the Matlab workspace.

3.4.4 Matlab IPMC force model

3.4.4.1 Introduction

With a model capable of predicting the expected concentration distributions through the thickness of the IPMC given a voltage input and one capable of predicting the variance of the voltage across the surfaces of an arbitrarily shaped IPMC actuator, the process of constructing a simulation of distributed forces can begin. Namely, at points on the surface of an arbitrary IPMC actuator subject to a particular voltage input, the voltage distribution can be determined and used, via the concentration tables, to predict the local concentration values through the IPMC. In this section, it will be discussed how this is

accomplished programmatically in Matlab. The results will then be converted into distributed force which will be used to drive the IPMC during actuation simulations.

3.4.4.2 Theory

In section 1.1 *IPMC*, the general characteristics and structure of Nafion based IPMC were discussed. In that section, the cluster morphology, first proposed by Gierke, was introduced. That model held that the basic structure of IPMC consists of two distinct phases, one hydrophobic and one hydrophilic. The bulk polymer constitutes the hydrophobic regions. Embedded in the bulk polymer are spherical inverted micellar structures arranged in a square lattice configuration. The micelles are connected to one another by cylindrical channels. Collectively, the structure of micelle and channels form a hydrophilic substructure inside the hydrophobic bulk polymer. When the IPMC is hydrated, the water collects in the micelle and channels. Cations reside within the inverted micelle tending toward the fixed ions at the boundaries of the micelle. Initially the number of mobile cations and fixed anions within the micelle are equal. However, upon actuation the cations are driven through the hydrophobic regions on their way toward the cathode. Eventually, this process results in the creation of two thin boundary layers, a thin high concentration layer at the cathode and a thicker layer completely depleted of cations at the anode.

This model proposes that the actuation response of the IPMC depends on the electrostatic interaction between neighboring micelle as cations are removed from or added to them. Near the cathode, cations are added to the micelle so that each becomes positively charged. Since all the micelle in this region will have acquired a positive charge also, there is a net electrostatic force which has the action of forcing the micelle

apart. Similarly, as the region near the anode becomes depleted of its cations, the result is micelle with net negative charge that similarly repels their neighbors. The contention that the charge imbalance that occurs during actuation results in positive forces at both the anode and the cathode might seem odd at first glance given that the net effect is actuation toward the cathode. However, as will be seen shortly, the force production at the cathode enjoys a slight advantage given that the force is proportional to the concentration squared. Essentially, the micelle in the anode experience force when cations migrate resulting in unpaired anions. However, there are a fixed number of anions in every micelle. This limits the amount of force that can be produced in the

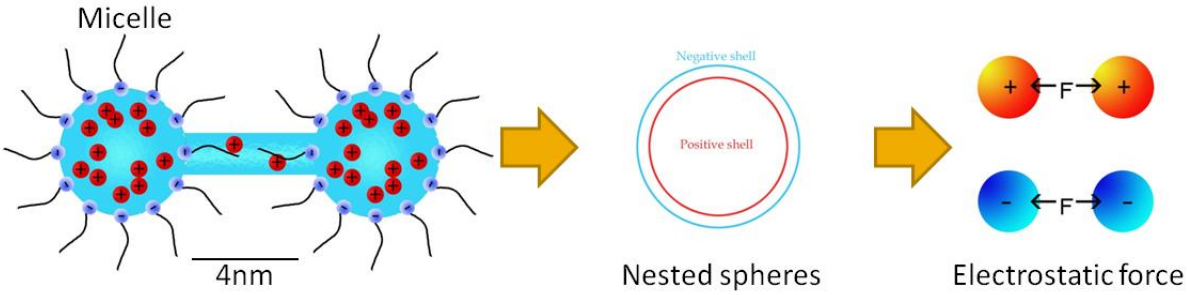


Figure 40. Electrostatic force between micelle

anode. A much larger force can be produced at the cathode where the number of cations in the micelle can become much larger than the number of anions.

In this model, the micellar clusters are modeled as nested spheres. The outer sphere represents the fixed anions on the periphery of a micelle. The slightly smaller inner sphere is made up of the mobile cations residing inside a micelle and tending toward the outer anions. Since the micelles are being represented as nested spheres and the interest is in calculating the force between two such structures, it is helpful to use an equivalent representation of two points centered within the spheres whose charge

depends on the differing number of cations and anions. The force between these two points is then calculated using Coulomb's law.

$$F_{cluster-cluster} = \frac{k_e q_1 q_2}{d^2} = \frac{k_e\left(e(c_+ - c_-)\right)^2}{d^2} = \frac{k_e(e\Delta c_+)^2}{d^2} \text{ where}$$
$$k_e = \frac{1}{4\pi\epsilon} \tag{3.7}$$

The first expression in (3.7) simply states that the force (F) between two points is proportional to their charge (q) times a constant and inversely proportional to the square of the distance (d) between them, Coulomb's law for point charges. The second expression is specific to the case of two nested spheres with equivalent charge. This approximation holds for IPMC, since locally the concentration does not vary largely. It states that the charge in each sphere is equal to the difference between the number of cations and anions multiplied by the elementary charge. The third expression says that the difference between the number of cations and anions is equivalent to the change in concentration since their numbers are initially equal. However, the number of anions within a micelle is fixed so that the change in concentration is entirely dependent on the changing number of cations. Notice that this equation assumes that there is no electrostatic force between the micelle initially.

The concentration of cations output from the electrochemical model will be given in moles per cubic meter, however (3.7) concerns the change in concentration for an individual micelle. Therefore, an equation is needed to convert the molar cation concentration into the micellar concentration. The necessary equation can be seen in

(3.8), where $\Delta\overline{c_+}$ is the molar concentration of cations, N_A is Avogadro's number, and $N_{cluster}$ the number of clusters per cubic

$$\Delta c_+ = \frac{\Delta\overline{c_+} * N_A}{N_{cluster}} \tag{3.8}$$

meter. The number of clusters is calculated using (3.9). Equation 3.9 is simply the total number of cations

$$N_{cluster} = (\frac{1m}{d})^3 \tag{3.9}$$

divided by the number of clusters per cubic meter. It assumes that the cations are initially distributed equally. Substituting (3.8) and (3.9) back into (3.7) gives the final equation for the force

$$F_{cluster-cluster} = \frac{k_e(e\Delta\overline{c_+}N_A)^2}{(N_{cluster}\, d)^2} \tag{3.10}$$

between two micelles embedded in the IPMC.

Now the force in (3.10) must be related to the overall problem of micelles arranged in a lattice structure. In particular, the interest is in discovering the force experienced by a single layer of micelle within the 3D lattice. The reason this is important is because it is the repulsion of one layer from another that causes the macroscopic actuation of the IPMC. So looking at cubic section of material of even concentration, it is seen that the force in a particular direction is equivalent to the force exerted on a single plane normal to that direction. In other words, the overall force simply depends on the force that two layers of micelle exert on one another. This may

seem surprising at first glance, one may expect that the force depends on the volume. However, the volume is made up of a series of planes and each plane is pushed no more strongly than the next. Meaning, the force cannot be increased by adding layers given that the layers are of equal distance. This leads to the simple result.

$$F_{layer-layer} = F_{cluster-cluster} * N_{cluster}^{2/3} \tag{3.11}$$

In this equation the layer on layer force is equivalent to the force on a single micellar cluster multiplied by the number of clusters in the layer, where $N_{cluster}$ is in moles per cubic meter. Now (3.11) pertains only to a double layer system. This equation has to be adjusted slightly to account for the presence of multiple layers. Given even separation, the force contributed by each additional layer is smaller by a factor of $\frac{1}{(d(n+1))^2}$. Assuming a large number of layers, the solution can be approximated as the infinite sum of $\frac{1}{x^2}$ whose solution is $\frac{\pi^2}{6}$. Multiplying this factor and equation (3.11), the IPMC force is finally given in equation (3.12). This final equation includes a variable *k* that experiment has shown to have a value between 1

$$F_{IPMC} = \frac{\pi^2}{k6} F_{cluster-cluster} N_{cluster}^{2/3} \tag{3.12}$$

and 5. This variable accounts for the usual and random variance in strength in newly manufactured IPMC sheets.

The limitation of this formulation of force is that it does not capture the so-called back relaxation effect seen in Nafion with sodium counterions. This effect is a slow but strong actuation that occurs over several minutes *in the direction of the anode*. For

instance, when an IPMC is subject to a uniform voltage input, there is a quick actuation toward the cathode lasting tens of seconds which is followed by a slower and more powerful actuation lasting minutes in the opposite direction. In most cases, the back relaxation effect will result in greater displacement that the original quick actuation. So, it may be fruitful at this point to suggest that the model be improved through the addition of a term accounting for the effect. However, as concerns the goal of this model, to predict the force distribution of an arbitrarily shaped IPMC for the purpose of designing microgrippers, this equation suffices. Again, however, it is only valid for forward actuation.

3.4.4.3 Model Overview

The following list contains the major activities involved in creating the model. Each activity is listed in the order it will be discussed.

1. Extract the voltage distributions from the FEM structure
    - The FEM structure contains the voltage potential distributions for the electrodes of an arbitrarily shaped IPMC. It was produced by the electrical model.
    - The voltage potential is extracted from the FEM using the Matlab function postinterp. The function returns the voltage at a particular position when passed xyz coordinates. Voltages are sampled from the anode and cathode.
2. Populate a matrix defining the ionic concentration in an IPMC

- Knowledge of the voltage potentials in the anode and cathode is used to populate a concentration matrix defining the distribution of cations in an arbitrarily shaped IPMC during actuation.
- The concentration values are extracted from the file produced by the electrochemical model in section 3.4.2. (Appendix A4)

3. Populate a force matrix
    - The concentration matrix, which lists concentration values, is converted into a matrix of forces. (Appendix A5)
    - The force matrix is stored for later use in IPMC force simulations.

3.4.4.4 Modeling

In section *3.4.3 Electrical model for arbitrary shapes* a FEM structure was exported into Matlab. The structure contained the solution to the problem of how a particular voltage input distributed itself across the electrodes of an IPMC actuator of arbitrary dimension. Now that solution will be mined from the FEM structure and stored in matrix form in Matlab. To do this the Matlab command postinterp (discussed in section 3.4.2.4) will again be used. This time, however, instead of recursively calling the postinterp function by placing it in a "for" loop, the function will be passed a matrix. The matrix contains the coordinates of spatial points in the model and for each point in the matrix a voltage will be returned. The form of the matrix is p = [x1 y1 z1; x2 y2 z2:…]. It returns a column vector listing the voltage corresponding to each of the respective spatial points (V = [V1; V2;...]). Horizontal concatenation of the V and p matrices using the function horzcat puts the result in a convenient form, [x1 y1 z1 V1; x2 y2 z2 V2: …].

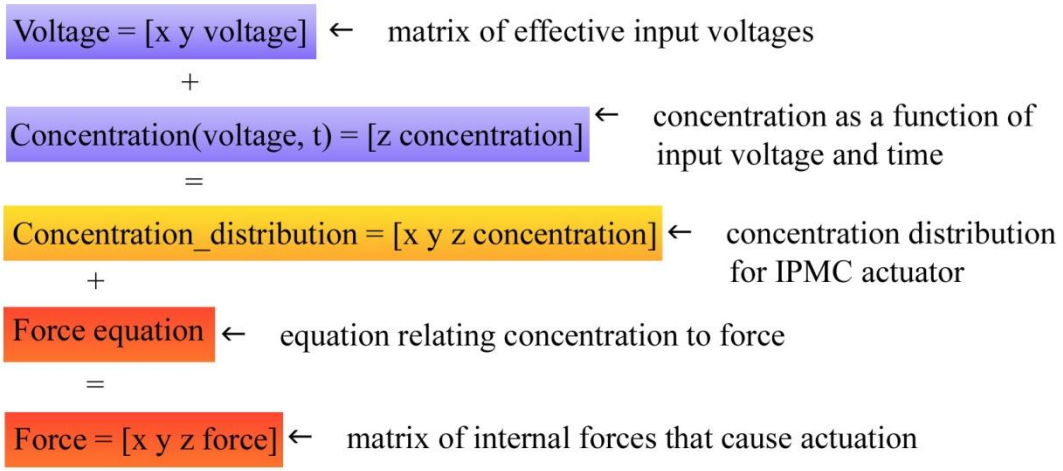
Now an important distinction needs to be made between the point voltages that can be collected using postinterp and the effective voltage driving the actuation. The magnitudes of the voltages that are collected at the points are relative to some global ground. However, an important assumption in the model is that the electric field is everywhere oriented through the thickness of the IPMC. Consequently, the ions travel almost entirely in this direction also. So the voltage that is really of concern here is the difference between a point on the anode and the one on the cathode that lies directly above it because they alone determine the local electric field. This is the effective voltage difference. To collect samples of the effective voltage requires the creation of a matrix of points located in an electrode. This matrix can then be used to sample voltages using postinterp. The same matrix can then be translated in the z-direction so that it lies entirely in the second electrode and used again to sample voltages. This method yields two matrices that when paired give voltage information about a series of points in the cathode and anode that have corresponding x-y positions. Subtracting the cathode's voltages from the electrode's voltages on a point-by-point basis yields the effective voltage for the entire actuator.

Perhaps the easiest way to create a sample matrix is to create an array of spatial points using the meshgrid function in Matlab. This command is typically used to create a 2D grid for 3D plotting purposes but works nicely in this case. The form of the command is [x,y] = meshgrid(X,Y) where X and Y are domains broken into arbitrarily sized segments. This allows the user to create a grid that encloses the entire cross-sectional surface of the IPMC and additionally, specify the sampling intervals in the two independent directions. Once the grid has been created, it is simply a matter of inputting

the returned vectors x and y specifying the sampling grid into postinterp along with a z height (i.e., postinterp(fem,'V',[x;y;z])). In this case, the postinterp function in this form will be called twice, once at the z height of the cathode and a second time at the z height of the anode.

To note quickly, the user may recall that Matlab only accepts rectangular matrices. This means that for irregularly shaped IPMC, the sample matrix might have points that are not valid for sampling. In this case, postinterp function simply inputs

Figure 41. IPMC force model



NAN (not-a-number) in the place of a voltage value. This value is acceptable for Matlab operations like matrix addition and subtraction. However, it is not a value that Comsol can accept. So any time data gets returned or exported to Comsol, it is important to replace a NAN entry with a numerical value (i.e., 0). This is easy to accomplish using the command M(isnan(M)) = n, where M is a matrix and n is an integer. Using zero for n replaces any NAN elements with zero values.

What has been accomplished so far in this section is the creation of a single matrix that contains information about the location and magnitude of the effective voltage

distribution for an entire actuator of arbitrarily shape. Now, recall that in the first section of this chapter the concentration history for a given effective voltage was solved for and stored in Matlab. These two results will now be combined for the purpose of creating an array of concentration values for the entire actuator for a given time (Figure 41). To be specific, there is an array of the form Concentration (voltage, time) = [z concentration], where z is the height of the sampled concentration, time is the elapsed time, and voltage is the effective voltage. This array, given an effective voltage and elapsed time, can return a vector describing the distribution of cations through the IPMC at a point. For instance, referencing Concentration(0.5,2) returns a vector containing concentration values through the IPMC after 2 seconds of actuation given a 0.5V signal. So, given that there is also a matrix of the form Voltage =[x y voltage] which describes the voltage distribution for an arbitrarily shaped IPMC at a number of points, the elements of Voltage can be used to populate a matrix of the form Concentration_3D =[x y z concentration] which defines completely the state of cations in the IPMC. For entries in Voltage that have NAN values, which do not correspond to the data in Concentration, Matlab is simply set to return a vector of zeros for concentration values.

Before progressing to force calculations, it may be helpful to quickly reflect on the process up to this point. A model which simulated ionic concentrations in IPMC during actuation was made. The results of this model were exported into Matlab and processed into an array. A second model was used to simulate the voltage distribution for an IPMC of arbitrary dimension given a voltage input. This model was similarly exported into Matlab. The solution to this model was, through a series of steps, converted into a matrix of positions and effective voltages. Programmatically in Matlab,

the first two models were combined to populate a matrix describing the distribution of cations in an IPMC of arbitrary dimension. Now, the focus is on developing a model that describes the relationship between the changes in ionic concentration to stress in the material.

Implementation of the force equation is relatively straightforward. All the variables except the concentration can be determined ahead of time. The concentration values come from the concentration matrix that was created earlier in this section. That matrix contains the cation concentration values for the entire IPMC. However, since the interest is in knowing the change in cation concentration, the initial concentration needs to be subtracted from every value in the array. The result is an array dCon = [x y z concentration_change]. Each value in this array can be processed using the force equation (3.12) to produce another matrix listing the force at each point. This matrix will have the form Force = [x y z force]. Once the force matrix has been determined, it needs to be stored as a text file so that it can be returned to Comsol. Matlab function dlmwrite can perform this function. Once the file has been created, a line identifying the variables of the form "% x y z force" may be added. This is merely a convenience, however. Comsol will read the variables either way, but in the case when the line is not added, the variables will automatically be assigned non-descriptive names.

3.4.5 Comsol/Matlab distributed force simulation

3.4.5.1 Introduction

The final model is a series of simulated force measurements on an IPMC actuator. Beginning in Comsol, a stress/strain model will be built, which uses the force values that

were created in the last section. This model will include a domain representing the arbitrarily shaped IPMC finger and second domain representing the straw of a force transducer. A single force measurement simulation will be set up in Comsol. That model will then be exported into Matlab. Its code will be modified so that the "straw" will be translated to multiple points over the surface of the IPMC. At each point a simulation will be run and a force measurement recorded. Ultimately, the collection of simulated measurements will be compared to actual measurements produced by the IPMC force scanner.

3.4.5.2 Theory

In this section, there are two domains. The first is a solid domain representing the IPMC actuator. The second is a cylindrical solid domain representing the straw of a force transducer. The interest is in simulating the reaction force experienced by the transducer domain when the IPMC domain is actuated into contact with it. Comsol's Stress/Strain application mode is used for this simulation. It uses the weak formulation of the equilibrium equation 3.13.

$$-\nabla \cdot \sigma = \boldsymbol{F} \tag{3.13}$$

In this equation, sigma is the stress tensor and *F* contains the body forces. The values in F were calculated using the IPMC force model and stored in a text file. Expressed in 3D, this equation has the form in equation 3.14.

$$-\frac{\partial \sigma_x}{\partial x} - \frac{\partial \tau_{xy}}{\partial y} - \frac{\partial \tau_{xz}}{\partial z} = F_x \tag{3.14}$$

$$-\frac{\partial \tau_{xy}}{\partial x}-\frac{\partial \sigma_y}{\partial y}-\frac{\partial \tau_{yz}}{\partial z}=F_y$$

$$-\frac{\partial \tau_{xy}}{\partial x}-\frac{\partial \tau_{yz}}{\partial y}-\frac{\partial \sigma_z}{\partial z}=F_z$$

This equation is used in conjunction with the linear stress strain relationship in equation 3.15.

$$\sigma = D\varepsilon \tag{3.15}$$

In this equation D is an elasticity matrix, σ is a 1x6 vector containing the stress components, and ε is a 1x6 vector containing the strain components. The strain displacement relationships are given in equation 3.16.

$$\varepsilon_x = \frac{\partial u}{\partial x}, \varepsilon_y = \frac{\partial v}{\partial y}, \varepsilon_z = \frac{\partial w}{\partial z}$$
$$\varepsilon_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right), \varepsilon_{yz} = \frac{1}{2}\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right), \varepsilon_{xz} = \frac{1}{2}\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right) \tag{3.16}$$

The IPMC is configured as a cantilevered beam. The fixed end has a zero displacement boundary condition. This is the point where the IPMC would be held by an electrode holder. All other boundary conditions for this domain are free. The second domain is cylindrical and represents the force transducer straw. Again there is one fixed boundary with a zero displacement boundary condition. This condition is applied to the flat boundary on the side opposite where contact is made between the IPMC and the

transducer straw. All other conditions are free. This is done so that the IPMC domain compresses the straw's domain during actuation.

3.4.5.3 Model Overview

1. Create geometry using Comsol's CAD tools
    - A 3D domain representing an actuator's geometry is created.
    - A second domain representing a force transducer's straw is added to the model.
2. Import force values into model
    - The force matrix containing the magnitude and position of forces driving the actuation of the IPMC are imported from a text file.
3. Add physics to the base geometry model
    - The Solid Stress/Strain application mode is added to the model. This application mode can simulate the deformation of a material given an applied load.
4. Set the subdomain settings and boundary conditions
    - The material definitions are added to the model. Composite values are used for the IPMC domain. The transducer straw is defined to have the mechanical properties of glass.
    - The IPMC is configured as a cantilevered beam with one boundary fixed and all others free. The transducer domain is fixed on one end so that it is compressed by the actuator during simulation.
5. Mesh and solve the model

- The model is meshed using free mesh parameters. The IPMC domain is meshed using a scaling factor.
- A solver is selected and the model is solved.

6. Extract the solution
    - The value of the blocked force output as measured by a force transducer is calculated using subdomain integration.
7. Add features to the model in Matlab
    - The model is exported into Matlab as an m-file.
    - A connection is setup between Comsol and Matlab.
    - The model is set to run iteratively, each time returning a blocked force value at a different point on the actuator.
8. Run a distributed force simulation
    - The model is applied to an IPMC of arbitrary geometry. Its output is a collection of positions and forces, a distributed force simulation. (Appendix A6)
    - The distributed forces are plotted in Matlab, producing a force map that can be compared to experimental values.

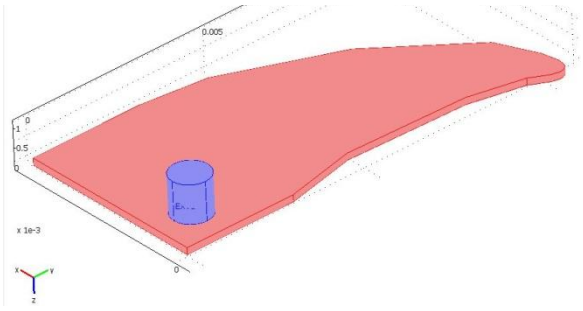
3.4.5.4 Modeling

In section *3.4.4 Matlab IPMC force model*, a matrix containing force values for an IPMC of a particular geometry was created. This matrix defined the internal stress driving that IPMC's actuation. The goal now is to incorporate the values contained in the matrix into a stress/strain model that can predict deflection and force output of the actuator. This requires creating a new model using the same geometry but a different

application mode. The first task, therefore, is to create geometry that corresponds to the force matrix. This can occur two ways depending on the shape of the actuator. For simple geometries like rectangles, the geometry can simply be redrawn. However, when the geometry is complex or contains curves, it is easier to save the geometry when it is drawn the first time while creating the electrical model in section 3.4.3. Recall that in that section, IPMC with complex geometry were created as 2D profiles which were then extruded to create layers. In anticipation of the need to recreate the geometry, the user can save the model at the point when it contains only geometry. So, ultimately, the user will either open a new 3D model and redraw geometry or open an existing 3D model containing geometry.

So, the first step is to somehow recreate the shape of the domain of interest. This time, however, the model will not have separate domains representing the electrodes and polymer. Instead a single domain representing both will be made and composite values for the mechanical properties will be used. This is done so as to avoid having to mesh the extremely thin geometry representing the cathode and electrode. At this point, nothing of interest with respect to deformation is happening in these domains. Bear in mind that this comes with a warning. The conductivity of the electrodes does change slightly with deformation. The electrode consists of closely packed islands of metal which are either forced together, increasing conductivity, or apart, decreasing conductivity, depending on the direction of actuation. Forcing the

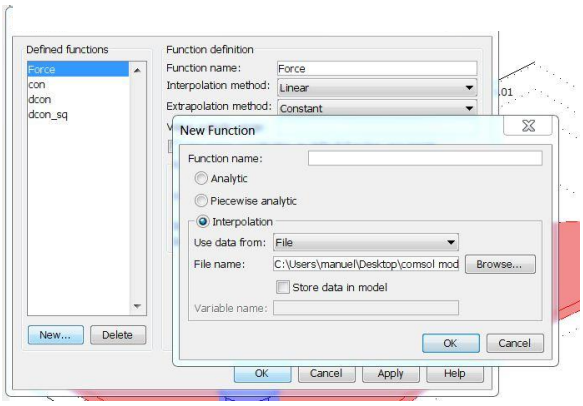
Figure 42. IPMC and transducer domains



islands apart has a larger effect on changing the conductivity than does forcing them together. For this model, however, the interest is in simulating the blocked force. The blocked force is the force produced by the actuator when the transducer is placed just at its surface. Essentially, this is the force measurement that is produced while preventing the IPMC from displacing at a point. The reason why this measurement is important is because it represents the largest force that can be produced by the actuator for a given input. So in measuring the blocked force, the assumption is that the IPMC under consideration will not be allowed to deflect sufficiently to significantly change the conductivity of its surface electrodes.

Once the single domain representing the actuator has been created, a second domain representing the force transducer is added (Figure 42). In the measurements taken with the IPMC Force Scanner, the force transducer was an Aurora Scientific model 403a whose straw is cylindrical and has a diameter of one millimeter. So the domain in this case has the form of a cylinder with the diameter and height of one millimeter. Comsol has a tool for creating solid cylinders so the geometry can be created simply by entering the values for size and position. The IPMC will bend in the direction of the anode, so the cylinder is placed so that its top surface meets this boundary. It is not particularly important what position on the surface of the IPMC the cylinder is placed, as long as it is placed no closer than half a millimeter away from any edge. During the meshing process, Comsol

Figure 43. Add function dialog



will automatically assign smaller elements in the regions around the force transducer in recognition of the relatively high gradients in this area. Placing the cylinder to close to an edge causes Comsol to increasingly create smaller elements. This can be fixed manually but really there is no particular force information at the edge of the IPMC to justify the extra effort.

Before the Solid Stress/Strain application mode gets added to the model, it is a good idea to import the text file containing the force values so that the variables contained in the file will be available when entering the physics settings. In the main menu the user clicks Options>Functions>New. This brings up the dialog for adding a new function (Figure 43). In this dialog there are three radio buttons, the last of which is interpolation. Clicking on it, a drop down menu entitled "Use data from" becomes available. The user sets the menu to file, and selects the file by clicking the browse button on the right hand side of the file name dialog.

With the geometry set, the multiphysics mode can be added. Entering the model navigator the user selects the Solid Stress/Strain application mode from the structural mechanics folder. The user then clicks "add" to add the mode to the model. The user can then click okay to exit back to the main window. The model tree on upper right hand side of the main window lists the application mode just added. The subdomain settings now can be set.

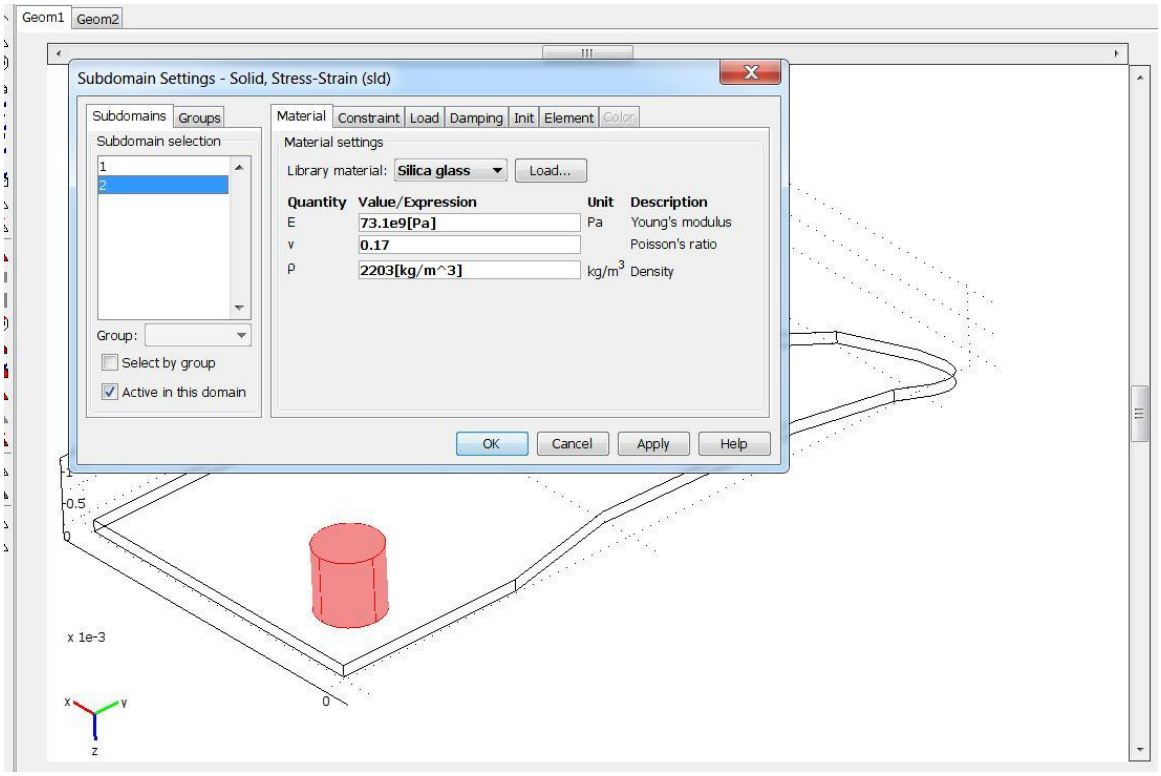


Figure 44. Solid stress/strain subdomain settings

In both domains the material definitions will have to be entered. In the domain representing the IPMC, a distributed body load will be added. Under the physics item in the main window or by clicking F8, the user can enter the subdomain settings (Figure 44). In the subdomain settings dialog, there will be two subdomains present in the model, the first corresponding to the straw and the second corresponding to the IPMC. The first tab in this dialog allows input of the material properties: Library material, Young's

modulus, Poisson's ratio, thermal expansion coefficient, and density. These values can either be entered manually or be automatically populated using settings from the material library. In the case of the domain representing the straw, the values will be automatically set by selecting a material from Comsol's library. First, the user selects the straw's domain. It will become highlighted in the image in the main window once selected. The user then clicks the "Load" button next to the library material input box. A window that lists all the available preset materials appears. Highlighting any of the items in the

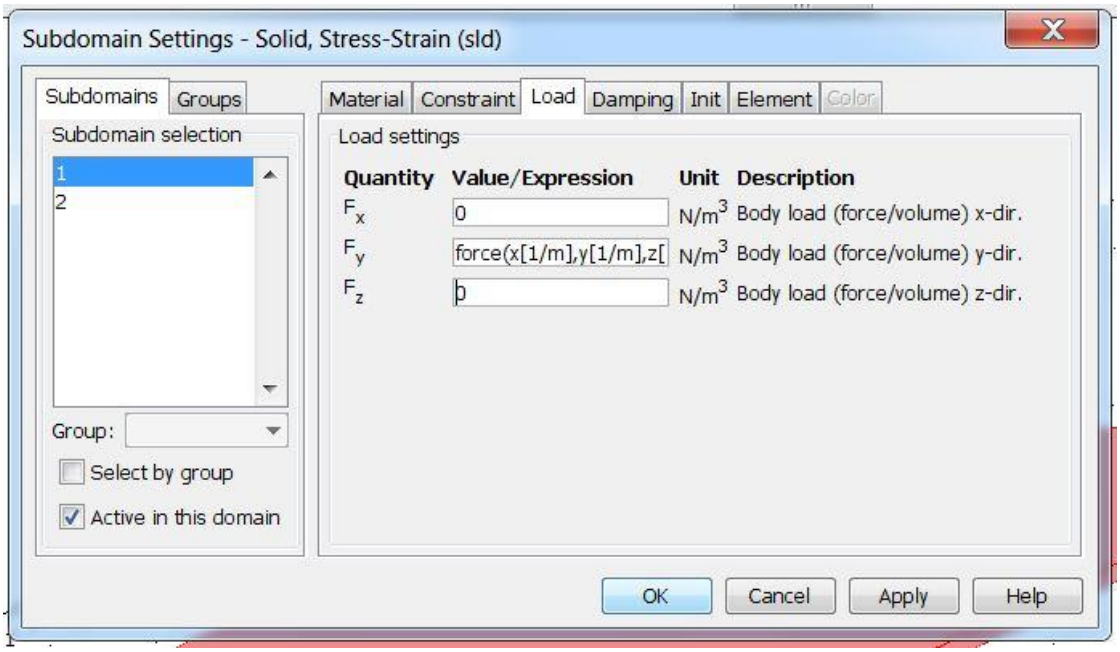


Figure 45. Distributed body load

material list causes the material's properties to be displayed. Highlighting and clicking "okay" populates the material settings. In this case the straw is made of glass, so the user selects "silica glass" from the materials list and all the relevant parameters are automatically populated.

Comsol does not contain a library material for IPMC so its material properties have to be entered manually. In addition, since the IPMC generates force, a body load

will need to be defined. The third tab in the subdomain settings is entitled "Load." In it there are three dialog boxes where body loads can be defined for each independent direction (Figure 45). Here is where the force values which were calculated in the last section are added. Assuming that a file header of the form "% x y z force" was added to the text file containing the forces, the body load in the x direction is simply entered as "force(x[1/m],y[1/m],z[1/m])." This defines the force as a body load in the IPMC domain. Since the inputs must be unitless, [1/m] term is needed. If the units in any dialog box are incorrect, Comsol will flag the error by displaying the correct units in red. Once the material properties for both the IPMC and transducer straw have been entered and the force has been defined, the dialog can be exited by clicking "okay."

The next step is to define the boundary conditions for the model. Ultimately, the domain representing the IPMC will be fixed at one end like a cantilevered beam. The other domain will be fixed at the surface opposite where it contacts the IPMC. This way, during simulation, when the IPMC beam deflects it pushes against the other domain. The boundary settings dialog can be opened either through the physics item in the main menu or by pressing F7. All the boundaries in the mode will be listed in the boundary selection box at the left hand side of the dialog. For the IPMC domain, every boundary will have its constraint condition set to "free" except the one boundary whose condition will be set to "fixed." This boundary is located at the end of the IPMC where it is held and feed an electrical input by an electrode holder. Similarly, for the transducer domain, all the boundaries except one will be set to "free." For this domain, the fixed end is the flat boundary opposite the boundary contacting the IPMC. This end is fixed so that the cylindrical domain is in compression as the IPMC bends towards it. After two

boundaries have been fixed and the rest set to free, the dialog is exited by clicking "okay." The model is now ready for meshing.

To mesh the domains, the free mesher will again be used. This time only the IPMC domain will have a z-direction scaling factor between 2 and 10. The free mesh parameters dialog is entered by pressing F9 or through the main menu under the item "mesh." As a reminder, there are three tabs of interest in this dialog: global, subdomain, and advanced. In the global tab the user sets one of the preset mesh sizes. The subdomain that is to be meshed is selected in the subdomain tab. The advanced tab contains the scaling factors for each independent direction. For the IPMC domain, a mesh size and a scaling factor will be used since it is a thin domain. The transducer domain will not need a scaling factor because it is well shaped and will not contain any steep gradients.

After meshing, the problem is ready to be solved. Comsol will automatically select what it judges to be the best solver for the problem type. If the user decides to change the solver, the solver settings are accessible through the solver parameters located in the solve menu item or by pressing F11. In any case, once the solver settings have been decided on the problem can be solved by pressing the equality sign in the toolbar below the main menu. This opens up a progress bar in the main window that details the solvers progress. Under the convergence tab, a chart showing how well the solution is converging can be seen. After a solution has been reached, its appearance can be modified by visiting the plot parameters settings in the postprocessing menu. Normally, the quantities that are most informative are those showing the displacement, stress, and reaction force. Deformed plot shapes are particularly good because they can be used in

conjunction with other plots. The deformed plot can be enabled by clicking on the deform tab in the plot settings and then checking the "deformed shaped plot" and choosing "displacement" as the quantity.

Assuming that the solution is acceptable, the final step in Comsol is to measure the force produced by the muscle. In the postprocessing menu there is an item entitled "subdomain integration." By selecting the transducers subdomain and setting the quantity to "reaction force z-dir," the force produced by the simulation is displayed at the bottom of the window. This represents the compressive force that is recorded by a force transducer in a real experiment. At this point the model can be saved as a Matlab file, since the entire process has been complete. This is a particularly good idea given that any future actions will add unwanted lines to the Matlab code.

Having created a Matlab version of the force model, the goal is to now transform the single force measurement into a distributed force measurement by altering the Matlab code. In order to make this process clear, the major components of the Matlab code will be discussed. The first and perhaps most important of these is the geometry definitions listed in the first section of the code after the version information. The geometry definitions will be listed under the heading "% Geometry n" where n is 1,2,3,… Each domain in the model will have its own geometry header. In this model, there are two such sections labeled "% Geometry 1" and "% Geometry 2." The first of these describes the cylindrical geometry representing the force transducer straw. It has the form "g = cylinder3('radius','height','position',{'x','y'},'0'). The second describes the domain corresponding to the IPMC. The primary interest is in the position attribute of the cylinder's definition. By changing the x and y values the cylinder can be translated to a

number of positions on the surface of the IPMC actuator. To do this two vectors of x and y positions need to be created. The values contained in each of these vectors are such that when their i^{th} elements are paired they define an xy position for the transducer straw. By iterating over these vectors, allowing each time for a simulation to occur and a solution to be recorded, a matrix describing the distributed force for an IPMC can be made. So given two vectors x_pos and y_pos the cylinder definition is modified as "g = cylinder3('radius','height','position',{num2str(x_pos(i)), num2str(y_pos(i))},'0')." This definition substitutes the constant strings x and y for the element i in the position vectors. The command num2string is included because the attribute position accepts string values and not numeric ones. Now the line "for i= 1:n" can be added at the top of the code before the existing line "flclear xfem." The flclear function clears any existing solutions and meshes, so it is important that the loop is started before this command line. The tag "end" is also added to the last line of the Matlab file.

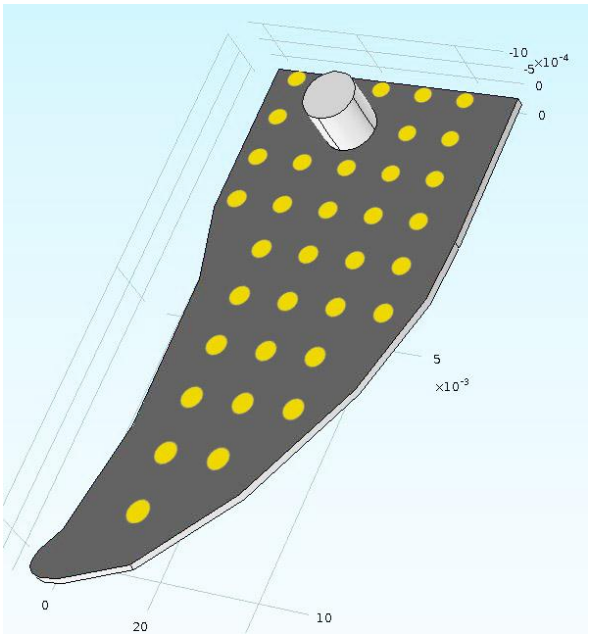


Figure 46. IPMC with force transducer domain

The next section in the code list the constant used in the definition of the model. It is listed under the heading "% Constants." The definitions are in easily identifiable pairs, such as "'Poissons', '.4' or 'Density', '2000'. They can easily be changed by simply exchanging one value for another. After the constants, the mesh settings are listed

under the heading "% Initialize mesh for geometry." The function meshinit contains the settings for the mesher. Two important parameters of the function are the 'hauto' and 'zscale' settings. The parameter 'hauto' refers to the automatic mesh sizes. The number directly following the parameter sets the mesh size. For instance, "'hauto',1" indicates the smallest available mesh and "'hauto',7" indicates the largest mesh. The 'zscale' parameter sets the scaling factor for the z direction, the number directly following this parameter is the multiplier. The initialize mesh section can be useful for performing convergence studies if instead of sweeping over position values the model is swept over mesh sizes and z scaling factors.

The next three sections will not necessarily need to be modified but are still important to be understood. The code below the heading "% Application mode" contains the settings pertaining to the Solid Stress Strain Application Mode being used for the analysis. The line beginning with "equ.Fy =" defines the force function being used to drive the IPMC's deflection. The function itself is found under the "% Functions" heading. The line "fcns{1}.name =" contains the names of all the functions present in the model and the line "fcns{1}.filename =" gives the filepath to the text file containing those functions. Directly below this section, any material definitions sourced from the material library are listed. The heading for this section is entitled "% Library materials."

The final two important sections in the Matlab file relate to displaying and returning a solution. They are the "% Plot solution" and "% Summation" sections. In the plot section the code for any plot that was created in Comsol before saving the model as an m-file will be listed. This section will be commented out when running the force model in an iterative fashion. The summation section contains the code for the

integration of the domain representing the transducer straw. On each iteration, the line "S1 = postsum(fem,'RFz_smsld','dl',2)" will return the total force experienced by the transducer's domain in the z direction. The term "fem" indicates which structure contains the solution. "RFz_smsld" refers to the reaction force in the z direction solution as calculated by the Solid Stress Strain Application Mode. The parameter "'dl', 2" specifies domain number two or the transducer domain. Since the postsum command will return a new solution on each successive iteration, the form of the function must be change so that all the solutions will be stored. For instance, the line could be changed to read "Force(i) = postsum(fem,'RFz_smsld','dl',2)," where the solution variable "S1" was changed into a vector of solutions, Force(i). By concatenating the x and y position vectors with this force vector, a single matrix that contains the location and magnitude of all the measured forces can be produced. This matrix represents a distributed force measurement. The matrix when plotted as a surface with the magnitude of the force as the height produces a graphic similar to the one produced by the IPMC Force Scanner.

3.5 **Concluding Remarks**

In this chapter, several models culminating in a simulated distributed force measurement for arbitrarily shape IPMC were discussed. The first of these was an electrochemical model that described the redistribution of mobile cations through the saturated channels of a Nafion membrane in the presence of an electric field. The second was an electric model for arbitrarily shaped IPMC. These two models were then combined in the IPMC force model, which predicted the cation concentration in an arbitrarily shaped IPMC under a specific voltage input. The concentration predicted was then subsequently

transformed into a distributed body load in the Nafion via an electrostatic force equation. The equation presupposed a repulsive electrostatic force existing between micellar clusters of unbalance charge. Finally, programmatically in Matlab, the IPMC force model was converted into a distributed force measurement.

CHAPTER 4. **RESULTS**

4.1 **Overview**

In this chapter, results from each of the individual models that make up the IPMC Force Distribution Model will be given. In section 4.2 the electric field through the material and the voltage distributions for multiple shapes that were predicted by the electrical model will be shown and discussed. Section 4.3 will overview the results of the electrochemical model including the evolving distribution of cations under different input signals. Finally, in section 4.4 the simulated deformation for various actuators will be shown along with the force distribution of two arbitrarily shaped IPMC actuators as predicted by the distributed force model.

4.2 **Electrical Model Results**

The electrical model produces reasonable results for all of the IPMC electrode types that were discussed in the thesis. The first is a simple rectangular IPMC actuated by a 2 volt input (Figure 47). The length of the actuator is 17 millimeters and the width is 5 mm. The results show an exponentially decreasing voltage for the anode from 2 volts at the input to around 1.4 volts toward the tip. There is no detectable change in voltage in the x direction, as seems reasonable.

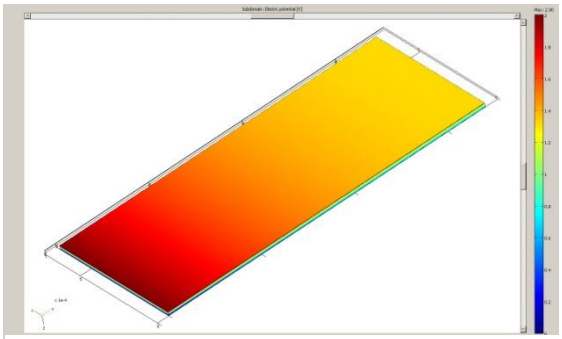


Figure 47. Voltage distribution for a rectangular IPMC

The second is an arbitrarily shaped IPMC based on a shark's pectoral fin (Figure 48). This actuator's largest dimensions are 6 millimeters in width and 12 millimeters in length. Again the voltage decrease on the cathode in the y dimension is exponential with a maximum of 2 volts and a minimum of 1.8 volts.

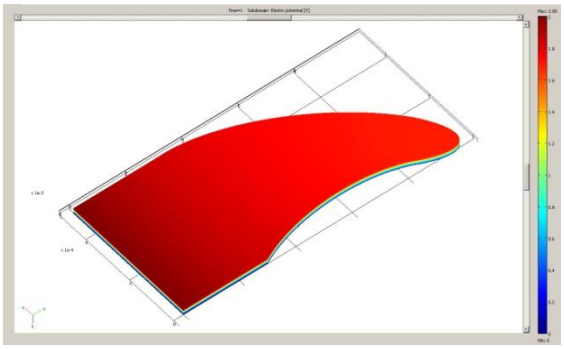


Figure 48. Voltage distribution for a fin shaped IPMC

The third IPMC is an example of a segmented IPMC (Figure 49). Its face electrodes have been split down the middle so that each side can be operated independently. In this case, there is an anode and cathode on each face. Each anode has been fed a 2V input in such a way that the two sides are actuated in opposite directions of one another, resulting in a twisting motion. From the image it is apparent that the two sides remain almost entirely isolated. In addition, comparing its voltage decrease to that for the rectangular IPMC, the voltage falls off more steeply. This is a reasonable result, given that splitting the electrode into two segments results in two thinner electrodes with double the resistance.

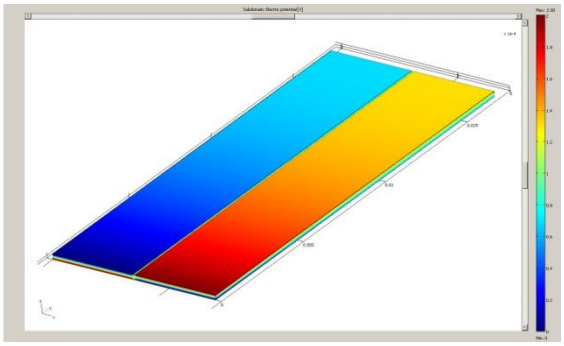


Figure 49. Voltage distribution for a segmented IPMC

The fourth design is an actuator that has a square "island" cut into its electrodes (Figure 50). Again, the design is intended to induce a twisting motion in the actuator.

The idea is that one side will be actuated less strongly than the other because a portion of its surface is blocked from receiving an electrical signal. As can be seen clearly from the picture, scoring the electrode in this fashion indeed isolates the "island" from electrical input. Again, the voltage falls off more quickly than the comparable rectangular IPMC due to increased resistance.

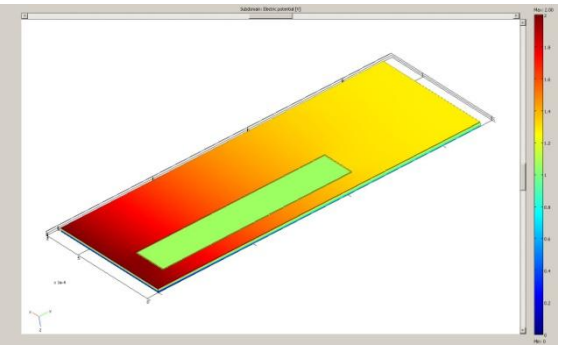


Figure 50. Voltage distribution for an IPMC with unactuated portion

In Figure 51 are two images showing the exponential voltage decrease from the input to the tip along both the anode and cathode. These results are for the simple rectangular IPMC. For the anode, the voltage starts at two volts and then falls rapidly over the first ten millimeters and then gradually approaches a steady state voltage of about 1.3 volts. Similarly, the cathode begins at zero and then after a rapid climb begins

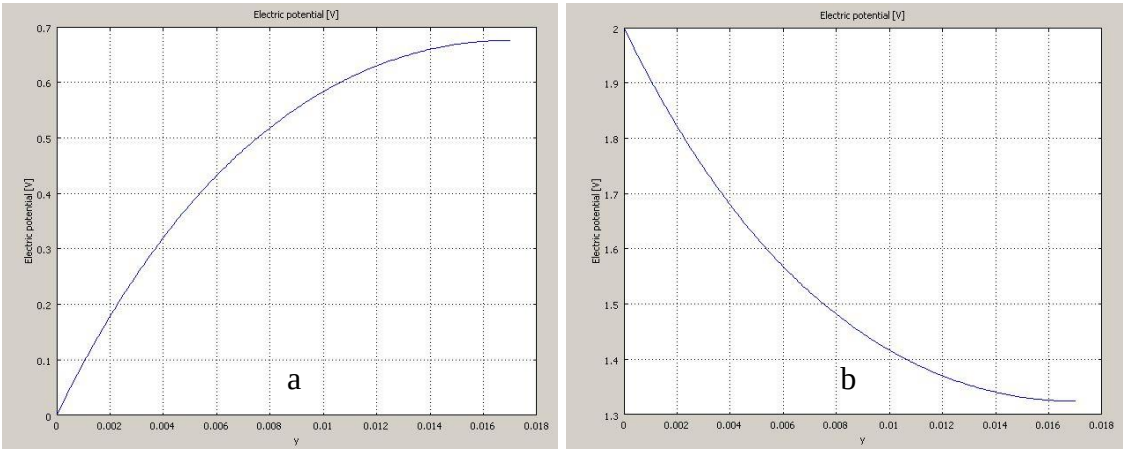


Figure 51. Predicted voltage as a function of length for a rectangular IPMC. From left to right, a) cathode and b) anode

to even out at 0.7 volts. The difference between these two voltages is referred to as the effective voltage difference because it is what drives the cation migration. As is evident from the images, the effective voltage difference is actually proportional to the square of two exponentials.

a b c

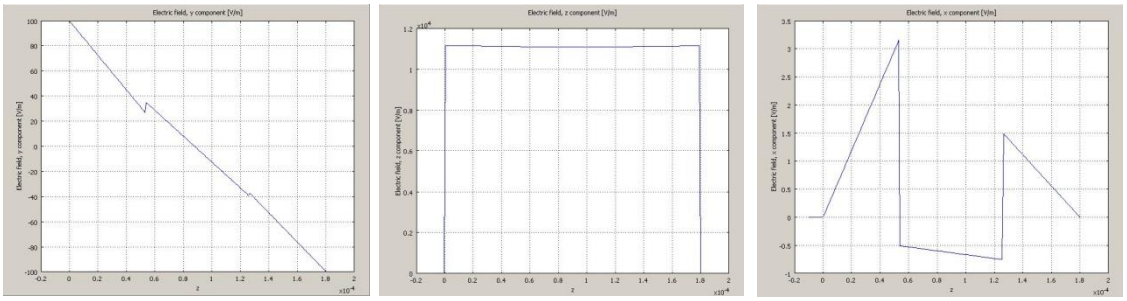


Figure 52. Predicted electric field through the thickness of an IPMC. From left to right, a) x direction component of the E field, b) y direction x direction component of the E field, c) z direction x direction component of the E field

The images in Figure 52 are of the electric field for every direction for line through the thickness of a rectangular IPMC. They seem to confirm the assumption that the electric field is oriented mostly through the IPMC rather than in any other direction. The magnitude of the electric field in the z direction hovers around 1.2E4 (V/m) while the corresponding value for the y direction is below 100 (V/m). The electric field in the x direction looks like noise and peaks at only around 4 (V/m).

4.3 **Electrochemical Model Results**

4.3.1 Cation migration

The images in Figure 53 show the migration of cations in a 180 micron cubic IPMC under a voltage input of two volts for five seconds. The first image (Figure 53 a) was taken a few microseconds after the simulation began. It shows how quickly cations

begin to pile up at the cathode and leave the anode once a voltage difference has been introduced. In the next image (Figure 53 b), taken after a second of actuation, a boundary layer almost completely devoid of cations begins to form near the anode. At the cathode, cations continue to pile up and the formation of a double boundary layer has started.

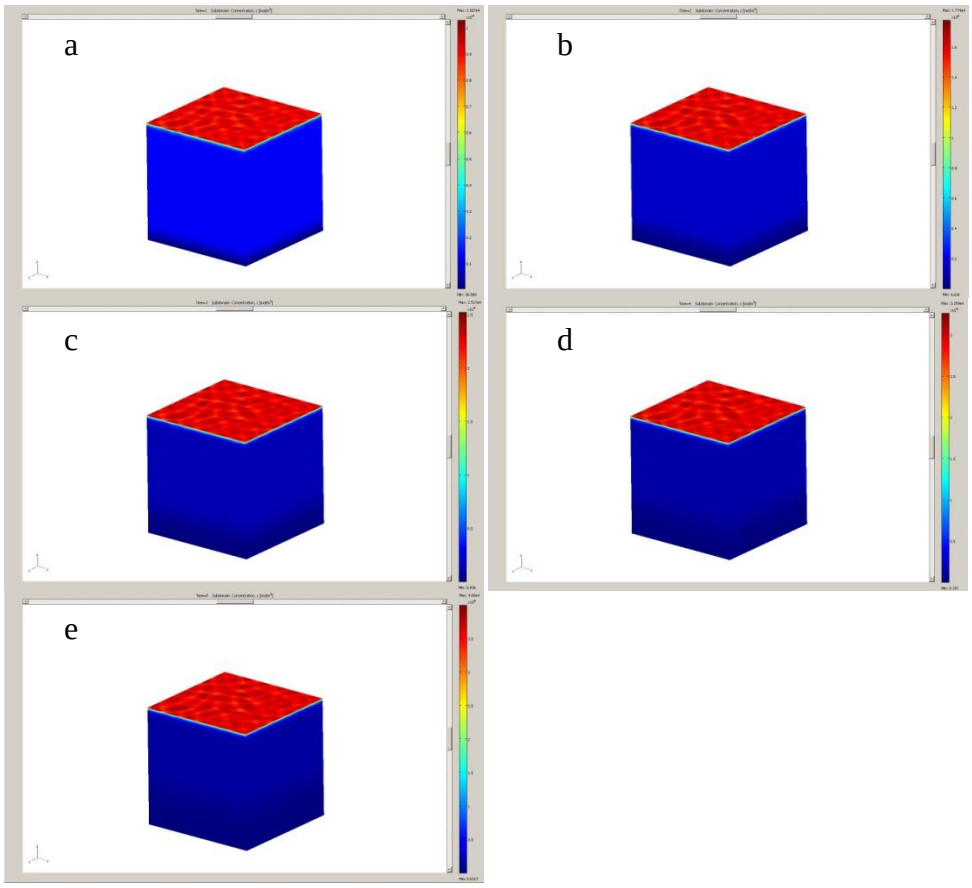


Figure 53. Simulation of cation migration in a Nafion cube. From left to right, top to bottom, at a) 0 seconds, b) 1 seconds, c) 2 seconds, d) 3 seconds, e) 4 seconds, f) 5 seconds

Over the next second, the anode boundary layer becomes thicker and less sharply defined (Figure 53 c). Meanwhile the cathode boundary layer continues to collect higher numbers of cations, as both it and the double boundary layer are become more pronounced. The trend over the next few images is that the cathode and double boundary

layers continue to increase in concentration, while the anode grows thicker and becomes less well defined. This same trend can be seen in Figure 54.

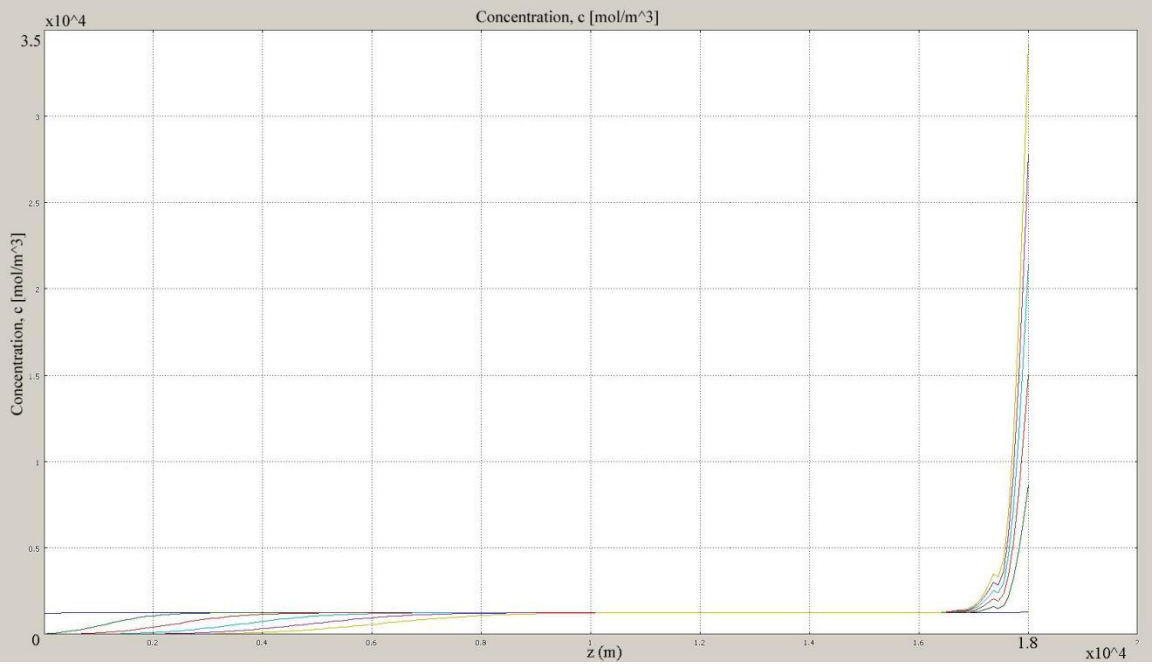


Figure 54. Chart of cation migration through the thickness of an IPMC

Figure 54 shows the evolving migration of cations to the cathode. The interior of the IPMC remains neutrally charged over the five second simulation. The anode becomes increasingly depleted of its cations. However, the gradient between it and the neutral area becomes smaller as time increases. The cathode thickness remains virtually the same but the number of cations it contains increases rapidly. As the cathode's cation concentration increases a small double boundary layer also grows steadily.

The series in Figure 55 shows the migration of cations when the IPMC is actuated with a two volt sine wave having a period of four seconds. In the first image (Figure 55 a) the IPMC is almost entirely neutral with a faint hint of a cation layer forming at the

cathode. There is no discernible activity at the anode yet. This is reasonable given that there is really no voltage yet. After an elapse time of one second, the voltage has attained

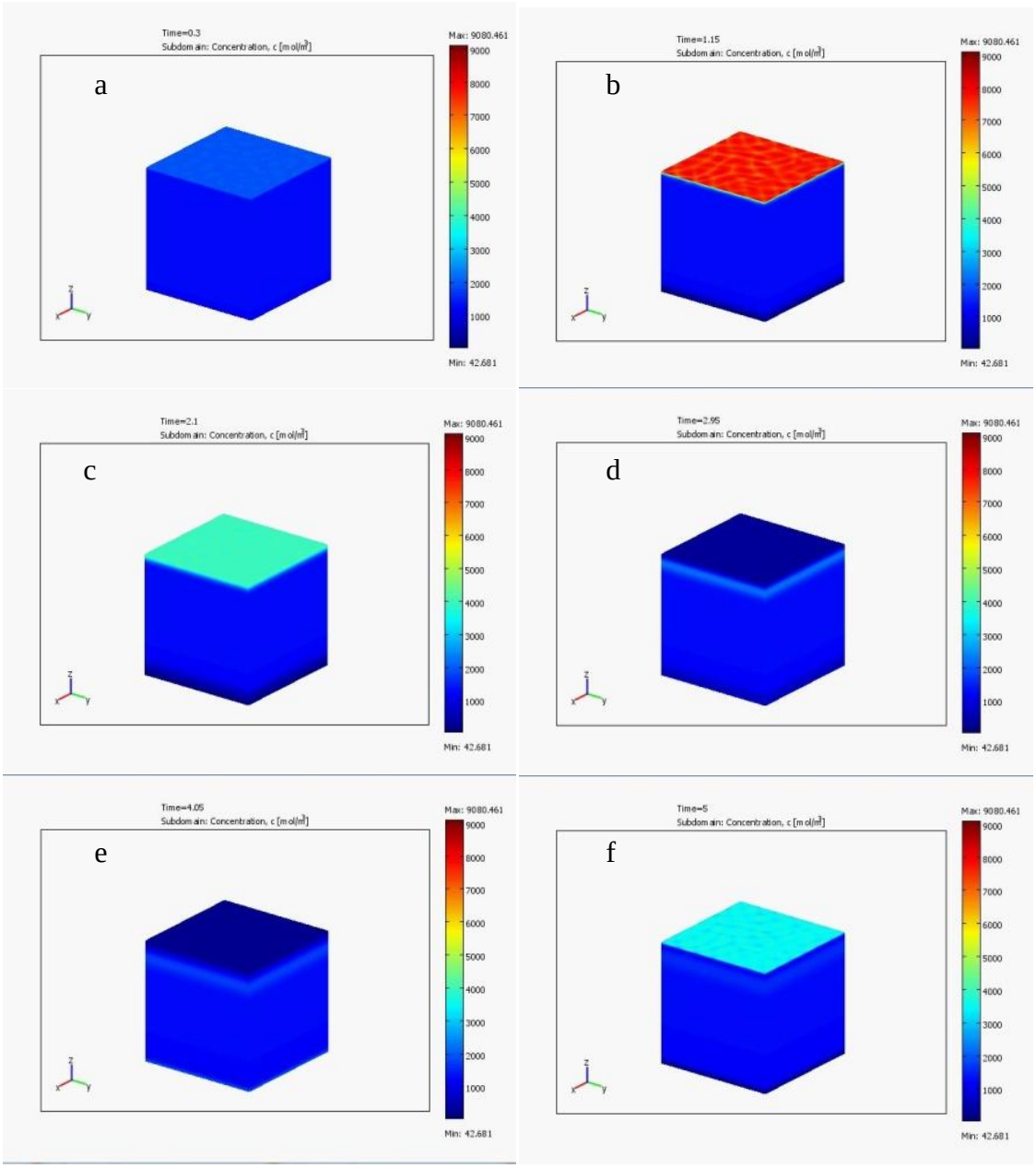


Figure 55. Simulated migration of cations through an IPMC box under a sine input. From left to right, top to bottom, at a) 0 seconds, b) 1 seconds, c) 2 seconds, d) 3 seconds, e) 4 seconds, f) 5 seconds

its highest value of two volts (Figure 55 b). The cathode and anode layer are both clearly visible along with the hint of a double boundary layer at the cathode. The cation

concentration peaks here at around 10,000 moles per cubic meter. In the third image (Figure 55 c), the voltage has again returned to zero. The cathode and anode boundary layers have not disappeared but are much less apparent and more diffuse than in the last image. The double layer has almost completely disappeared at this point. Figure 55 d shows the cation concentration after 3 seconds have elapsed. The voltage on the lower face is now at negative one while the upper face remains at ground. This forces the concentrated layer of cations on the upper face down toward the lower face. As this layer of cations migrates it leaves a region of low cationic concentration in its wake. After another second, the voltage has again fallen to zero (Figure 55 e). The concentrated layer of cations that is now located just off the upper surface of the actuator becomes more diffuse. There is still an area of low cation concentration right on the upper surface. However, the concentration in this region is much lower than the second before. The final image (Figure 55 f) shows the process begin to repeat itself as the voltage once again approaches 1V.

4.4 **Force Model Results**

4.4.1 Deformation

The image in Figure 56 shows the deformation of a rectangular IPMC of dimension 7x17x.18 mm. The IPMC has been actuated with a sine wave having a magnitude of 2V and a period of 4 seconds. The model shows a tip deflection of 1.28 mm for the beam at 1 second elapsed time.

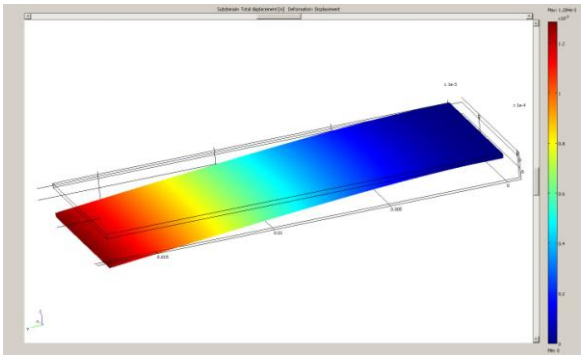


Figure 56. Simulated deformation of a rectangular IPMC

The deformation of the segmented IPMC (Figure 49) which has had each side actuated in a different direction is shown in Figure 57. The image shows that the model predicts very little displacement for an IPMC actuated in this way. Though the IPMC does indeed twist, surprisingly the peak displacement is only 18 micron. This must mean that the IPMC has sufficient stiffness that it does not to deflect much in this direction under the force it produces.

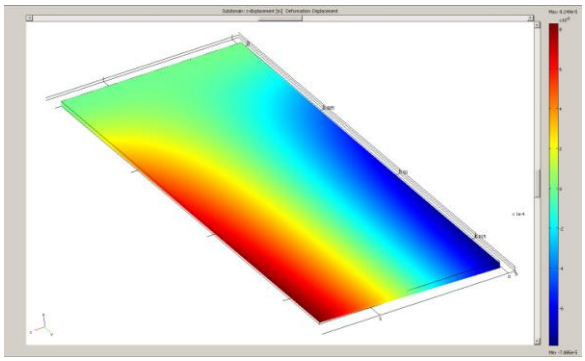


Figure 57. Simulated deformation of a segmented IPMC

The IPMC in Figure 58 corresponds to the one shown Figure 50. This IPMC has a rectangular section removed from its electrodes so that the section is electrically isolated. It has been imagined that such a design might result in a twisting actuation. As can be seen from the figure, the segmenting of the surface does result in a slight amount of twist. The side of the actuator where the section is located does not deform as much as the other side which is fully actuated. The most noticeable effect in the image, however, is a decrease in total deflection. The same beam without the segment displaced almost 1.3 mm at the tip whereas this beam's tip deflected 0.93 mm.

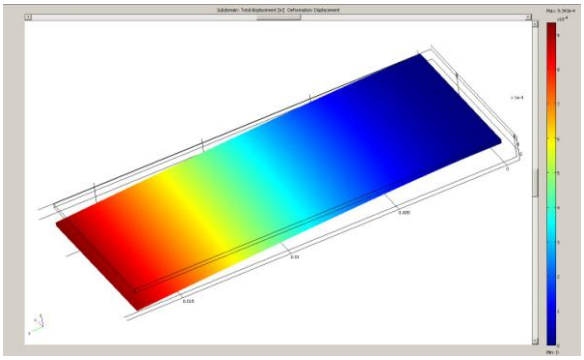


Figure 58. Simulated deformation of an IPMC with an unactuated portion

4.4.2 Force distribution

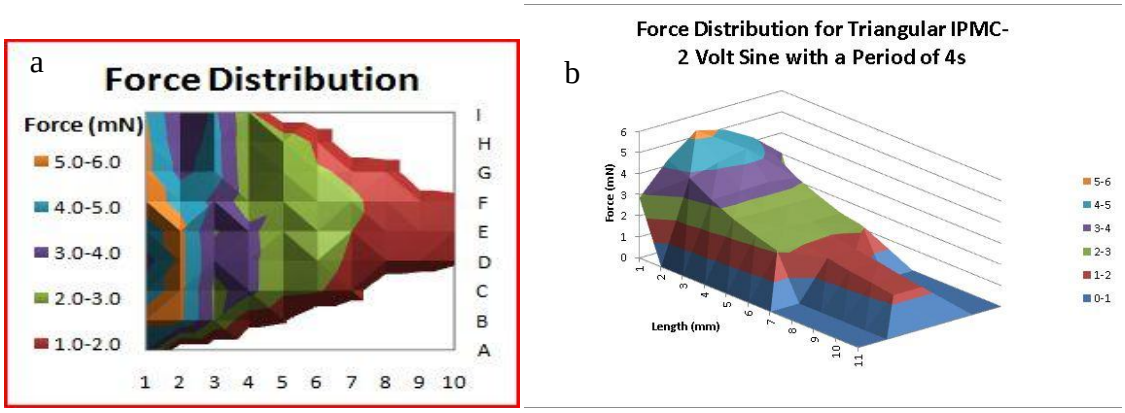


Figure 59. Force distributions for a triangular IPMC. From left to right, a) experimental and b) force simulation

The images in Figure 59 show the results for a triangular IPMC with a 180 micron thickness. Both were produced using a 2 volt sine wave actuation voltage with a period of 4 seconds. The first image is the result of an experimental measurement using the IPMC Force Scanner. It shows an exponential decrease in blocked force from a peak of 7 mN at the base to just less than 1 mN at the tip. The second image was produced using a force simulation using a *k* of 1.5 in the force equation. As can be seen from the figure, the simulation comes very close to the experimentally determined values. The only really noticeable difference is that the simulated force distribution is cleaner and more even. Also the simulation peaked out at 5 mN that is somewhat less than the actual experiment. This may be due to using a k value that was slightly too high. However, it may very well be the case that the experimentally determined value is high. Actual IPMC are slightly warped rather than perfectly flat, as is assumed in the model. Points on the surface of the IPMC may curve outward (or inward) by several micron. This small curvature translates into a force difference as measured by a force transducer. If the IPMC curves away from

the force transducer the measured force will be lower than expected. The converse is true when the IPMC is curved toward the transducer. In the second case, the IPMC will contact the transducer slightly prior to being actuated. Consequently, the measured force will be elevated slightly. As it turns out, toward the fixed end of the IPMC, the spacing between the force transducer and the actuator become increasingly important. The force output is the highest and the displacement is the lowest in that region, meaning that force falls rapidly with small displacements.

Figure 60 a shows the experimentally determined force distribution for an

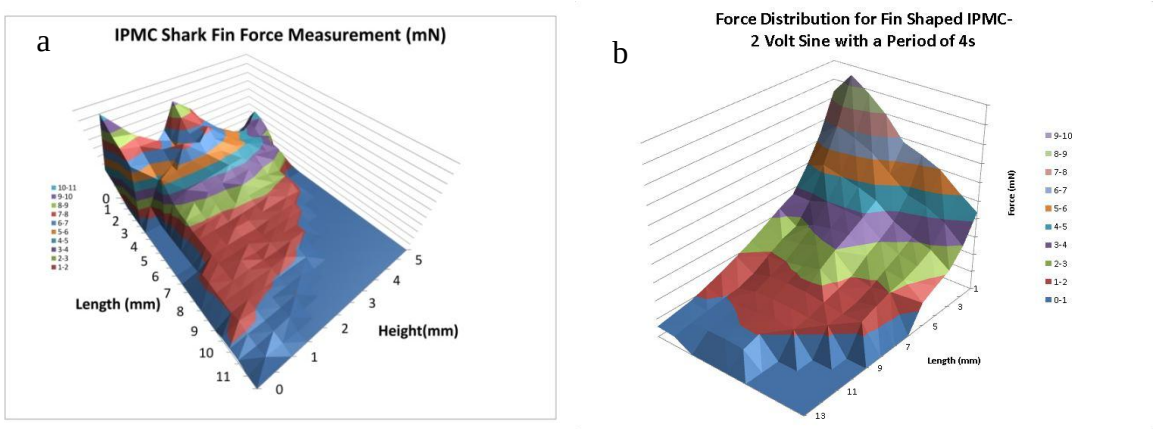


Figure 60. Force distributions for an IPMC shark pectoral fin. From left to right, a) experimental and b) force simulation

artificial shark fin. Similar to the triangular force scan, the force distribution for the fin shows an exponential decrease. This time the force peaks at 11 mN at the base and decreases to 0.6 mN at the tip. The simulated force distribution in Figure 60 b was produced using a k value of 1, meaning that the force equation was not adjusted. The results show good correspondence. However, the exponential decrease is somewhat stronger in the simulation. Also, again there is a difference that becomes more apparent toward the fixed end of the IPMC between the experiment and the simulation. The

experiment displays peaks near the fixed end. This time, however, it is almost certainly warpage in the actuator that accounts for these experimental force values. There is no reason that the force varies this dramatically on that boundary otherwise. In this case, the simulation suggests a more reasonable force distribution in this region, because in the simulation the IPMC is taken to be perfectly flat.

CHAPTER 5. **CONCLUSION**

5.1 **Results of the Study**

The IPMC Force scanner and the IPMC force model have respectively presented a novel way of characterizing and modeling the force output of IPMC actuators. The first has provided an in depth look at how the magnitude of the force output varies for IPMC actuators. The IPMC Force scanner is used to collect multiple measurements at points across the surface of an actuator. Measurements from the IPMC Force Scanner have shown that there is significant and exponential decrease in force output from the fixed end of the actuator to the tip. While it is widely known that the force output of an IPMC actuator varies over multiple actuations. The IPMC force scanner demonstrated that this deviation in force output is not constant over the surface of an IPMC actuator but is more significant toward the fixed end. Ultimately, the measurements are used to create topographic maps of force, called IPMC force maps. These maps provide a simple way to assess the force capability of an actuator. The forces illustrated in this way can be used as a tool to suggest to researchers how designs may be modified to suit specific applications. Perhaps more importantly, however, the force maps provide experimental data for the creation of for the creation of force models.

The IPMC distributed force model is such a model and is the first to use the distributed force measurements provided by the IPMC force scanner. The model is composed of several coupled sub-models that correspond to electrical, electrokinetic, and mechanical phenomena. The output of the model is a simulated force map that can be compared to a corresponding experimental force map measured by the IPMC Force Scanner. As was demonstrated in this thesis, the model provides reasonable simulation

results that are highly comparable to experimental measurements. Given the close correspondence between the experimental and simulated results, the model can be used for the simulation and analysis of the force output for IPMC actuator designs of arbitrary dimension. This type of analysis was not possible before. Prior to the creation of this model, IPMC actuators had to be created and experimentally tested to determine their actuation strength. The practical implication of the model is that the tedious process of manually creating and testing new actuators can be eschewed for a much simpler, faster, and more efficient design process.

The model extends the state of the art in IPMC modeling and simulation. An earlier model by Pugal demonstrated that the electrokinetic migration of ions could be successfully model in Comsol[26]. However, the force equations in that paper were parametric and not tied to any physical phenomena. In this thesis, a novel force equation based on the electrostatic attraction and repulsion experienced by neighboring micelle as they gain or lose charge is presented. This force equation is attractive because it is intuitive, easy to implement, and yields results that match experiment.

The model is the first to address the force output of three dimensional actuators of arbitrary dimension. It is very clear that even for the mot simple IPMC actuator shapes (i.e., rectangular) that the force varies significantly across the face of the actuator in the x and y directions. It is, therefore, important that the model be able to account for this force variance. In addition, there are IPMC with complex geometries (i.e., shark fin, digitated, segmented) that are of interest to researchers. It is beneficial to be able to simulate their behavior as well. The results presented in Chapter 4 demonstrated that the IPMC force model is capable of simulating such actuators.

5.2 **Limitations of the Study**

The biggest single limitation of the model is that it is only valid for the initial quick actuation and does not account for the back relaxation. In this thesis, the maximum force was considered to be the blocked force produced as the actuator bends toward the anode. However, a stronger blocked force can be recorded using the long term actuation toward the cathode. The model is missing a term accounting for back relaxation and does not suggest any new reasons why the phenomenon occurs. The introduction of such a term and theory can be an important addition to the model, because it extends the amount of time the solution is valid. Further, since the goal of the model was to find the maximum possible force output for a finger of arbitrary geometry under a given input, it was not used to perform any transient force studies. Such studies are important and data collected using the IPMC Force Scanner can be used to validate such simulations. However, because it was not necessary to perform such studies to determine the maximum force output, they were not included.

Error estimates are hard to assess. Newly manufactured IPMC sheets, though they appear flat to the eye, under closer examination are slightly wavy. This can lead to experimental force measurements that are predictably high or low depending on whether the defect is concave or convex toward the transducer. For the experimental force distribution maps, it can lead to unusual looking peaks and troughs. This is especially true for the regions closest to the fixed point where the voltage is applied. This was evident in the experimental force distribution for the IPMC shark fin. This makes an assessment of error somewhat tricky. Although, the difference between the measured values and the simulated ones for these regions might be characterized as simulation or

measurement error, this does not make much sense, since the warpage is really just another material property of the IPMC. The warpage does become less of an issue as the transducer is moved toward the tip. Fortunately, this is the most important region on the actuator because a majority of the microgripper applications concern the tip force. It is apparent that the force maps are really showing what the distribution is like if the actuator was completely flat. This result is interesting and important in itself. However, it will require either a more accurate representation of an actuator's geometry or that manufactured IPMC become more flat before the simulation error for an entire actuator can be assessed accurately.

There are several simulation values that have not been accounted for experimentally. For instance, no experimental cationic concentration values have been published. Also, material variances arise from the difficulty in manufacturing the material consistently. For instance, the conductivity of the electrodes varies from sheet to sheet, as does the strength of the muscle. Adding the variable k to the force equation is an attempt to account for the variance in the last stage of the model. However, the real problem is that there are multiple variables for which a reasonable value is assumed but just how much the values vary from one IPMC sheet to another is not known. Since all the variables confound one another it is difficult to nail down any specific value for those variables. Therefore, an important limitation of the model is that it is informed by incomplete experimental data in some areas.

In relation to the last problem, there are addition terms that might reasonably affect the behavior of the material. These terms such as temperature change during actuation, electroosmosis, variable resistance, and other terms, have not been included in

this model. Again, because of the complexity of the problem it is difficult to assess the relative importance of these terms to the force output, especially when their parameters are only reasoned to have the values that they are assigned. In other words, the model is missing several known phenomena for IPMC that may or may not influence the force output of the actuators.

The model is not a sensing model. It does not include the voltage generation that occurs when the IPMC deflected. Nor does it account for the changes in conductivity in the electrodes which are necessary to simulate the output of a sensor actuator type IPMC.

5.3 **Future Research**

Many of the improvements that can be made to the model have already been implicitly suggested in the last section. The addition of terms relating to the back relaxation and the sensing properties of the material can improve the model greatly. Also, using the model to perform time dependent solutions is a worthwhile undertaking. As had been mentioned before, the method needed to validate transient force solution does exist and is readily available in the IPMC Force Scanner. The model has not yet been used in parametric studies involving changing geometry, nor has a sensitivity study involving variables in the physics model been undertaken. However, such studies can be performed relatively easily. A particularly interesting simulation is to see how small an actuator can become while still returning useful force values.

Finally, in retrospect, the IPMC distributed force model can be made more efficient if the electrochemical model and the electrical model are approached in reverse

order. In the current version, the electrochemical model is run first, so that a collection of ionic responses to voltage inputs can be recorded in a concentration table. Then, the electrical model is run for an IPMC of a particular geometry. Voltage predictions are sampled from the IPMC electrodes and their values used to determine the ionic response via the concentration tables. Instead, the electric model should be run first. The result can then be sampled to determine the voltage inputs at various points on the actuator. These voltages then become the input for the electrochemical model. This way no excess electrochemical simulations are run for voltage values that are not used.

In addition, the electrochemical model is based on a cubic element representing the Nafion layer. However, this model can be more efficiently run as a 1D model, since the interest is on how the ionic concentration varies in the z direction only.

APPENDICES

APPENDIX A – PROGRAMMING EXAMPLES

A1. Function Concentration_table_generation

```
%This M-file generates an array containing concentration values for a
%1e-6m IPMC cube as a function of input voltage, time, and z height.
%The results can be used to calculate the force response of IPMC


function [Concentration] = Concentration_table_generation(Vmax,tmax)


% If some geometry objects are stored in a separate file,

% the name of this file is given by the variable 'flbinaryfile'.




%Create a vector with input voltages

Voltage = (0:.1:Vmax);

[~,n] = size(Voltage);

t = (0:.05:tmax);

[~,m]=size(t);


%Preallocate space for concentration values


Concentration = zeros(181,m,n);


%Create loop that simulates the electrochemical response of IPMC at
each
```

```
%voltage.  Store the results of each iteration in an array of the form
%C[z,t,V].

%Begin loop
for i = 1:n

flclear fem

% COMSOL version
clear vrsn
vrsn.name = 'COMSOL 3.5';
vrsn.ext = 'a';
vrsn.major = 0;
vrsn.build = 603;
vrsn.rcs = '$Name:  $';
vrsn.date = '$Date: 2008/12/03 17:02:19 $';
fem.version = vrsn;

flbinaryfile='IPMC_square.mphm';

% Constants
fem.const = {'R','8.314', ...
  'T','293', ...
  'Diffusion','2.8e-11', ...
  'F','96485', ...
  'Eps','120*8.85e-12', ...
  'Youngs','5e8', ...
  'Poissons','.487', ...
  'Density','2000', ...
  'IPMC_x','15e-3', ...
```

```
  'IPMC_y','5e-3', ...
  'IPMC_z','180e-6'};

% Geometry
clear draw
g1=flbinary('g1','draw',flbinaryfile);
draw.s.objs = {g1};
draw.s.name = {'BLK1'};
draw.s.tags = {'g1'};
fem.draw = draw;
fem.geom = geomcsg(fem);

% Initialize mesh
fem.mesh=meshinit(fem, ...
                  'hauto',2, ...
                  'zscale',5, ...
                  'point',[], ...
                  'edge',[], ...
                  'face',[], ...
                  'subdomain',1);

%Call current voltage element and convert it into a string.
V = Voltage(i);
Vstring = num2str(V);

% Application mode 1
clear appl
appl.mode.class = 'EmConductiveMediaDC';
appl.module = 'MEMS';
appl.assignsuffix = '_emdc';
```

```
clear bnd
%Simulate a 0.25 hz sine voltage input
bnd.V0 = {0,strcat(Vstring,'*sin(pi*.5[1/s]*t)'),0};
bnd.type = {'nJ0','V','V0'};
bnd.ind = [1,1,2,3,1,1];
appl.bnd = bnd;
clear equ
equ.sigma = 10;
equ.ind = 1;
appl.equ = equ;
fem.appl{1} = appl;

% Application mode 2
clear appl
appl.mode.class = 'ElectroKF';
appl.module = 'MEMS';
appl.assignsuffix = '_chekf';
clear prop
clear weakconstr
weakconstr.value = 'off';
weakconstr.dim = {'lm2'};
prop.weakconstr = weakconstr;
appl.prop = prop;
clear bnd
bnd.type = 'N0';
bnd.ind = [1,1,1,1,1,1];
appl.bnd = bnd;
clear equ
equ.D = 'Diffusion';
equ.V = 'V';
```

```
equ.init = 1250;
equ.um = 'Diffusion/(R*T)';
equ.ind = 1;
appl.equ = equ;
fem.appl{2} = appl;
fem.frame = {'ref'};
fem.border = 1;
fem.outform = 'general';
clear units;
units.basesystem = 'SI';
fem.units = units;

% ODE Settings
clear ode
clear units;
units.basesystem = 'SI';
ode.units = units;
fem.ode=ode;

% Multiphysics
fem=multiphysics(fem);

% Extend mesh
fem.xmesh=meshextend(fem);

% Solve problem
fem.sol=femtime(fem, ...
                'solcomp',{'V','c'}, ...
                'outcomp',{'V','c'}, ...
                'blocksize','auto', ...
```

```
                'tlist',t, ...
                'tout','tlist', ...
                'linsolver','cg', ...
                'prefun','amg');

% Save current fem structure for restart purposes
fem0=fem;

% Plot solution
% postplot(fem, ...
%           'tetdata',{'c','cont','internal','unit','mol/m^3'}, ...
%           'tetmap','Rainbow', ...
%           'tetkeep',1, ...
%           'tetkeeptype','random', ...
%           'solnum','end', ...
%           'title','Time=10    Subdomain: Concentration, c [mol/m^3]',
...
%           'geom','off', ...
%           'grid','on', ...
%           'campos',[-3.7947777755789635E-4,-2.0450555732208687E-
4,2.2665713497876716E-4], ...
%           'camtarget',[3.433503873772565E-5,4.4746293660489787E-
5,9.769020956275869E-5], ...
%
'camup',[0.21988089724932888,0.1348564394061384,0.966160510358154], ...
%           'camva',39.59775270904893);

Concentration(:,:,i) = Concentration_extract(fem,m);

end
end
```

A2. Function Concentration_extract

```
function [Concentration] = Concentration_extract(fem,m)
Concentration = zeros(181,m);

%for times [0:.1:5]s

for i = 1:m

%get the concentration values through the thickness z=[0:180]um

    for j = 1:181

      Concentration(j,i)= postinterp(fem,'c',[90e-6;90e-6;(j-1)*1e-
6],'T',.1*(i-1));

    end

end
```

A3. Function Concentration_export

```
function [] = Concentration_export(IPMC_concentration)

%prepares IPMC concentration data for export to Comsol
```

```
dlmwrite('Rectangular_7x17_concentration.txt',IPMC_concentration','deli
miter','\t','precision','%2.6f','newline','pc');
```

A4. Function Concentration_profile_selection

```
function [con]=concentration_profile_selection(concentration, num_z,
V_unrounded, s)


V = roundn(V_unrounded,-1);


if (isnan(V)==1)

    con = zeros(1,180/num_z+1);

elseif (0<=V)

    con = concentration(1:num_z:181,s*20+1,(abs(V)*10)+1)';

elseif (0>V)

     con = concentration(181:-num_z:1,s*20+1,(abs(V)*10)+1)';

else

    con = zeros(1,180/num_z+1);

end

end
```

A5. Function Force_generation_rectangular

```
%Function Force_generation_rectangular generates a text file containing
a concentration matrix [x y z con Force] given a comsol fem structure,
a concentration lookup table of the form C(V,z,t), the time of
solution, and a multiplier k.
```

```
function []= Force_generation_rectangular(fem,concentration,s,k)

%Dimension of bounding box in mm (should be large enough to enclose the
%entire IPMC in the xy plane. It is okay if regions in the box don't
%contain geometry, these regions will be given NaN values.)

IPMC_width = 7;
IPMC_length = 17;

%Number of samples per mm in x and y
num = 2;
%Number of microns per sample in z
num_z = 5;

%Create an empty square matrix at z=0 (the anode) that contains the
points
%in the xy plane where the voltage will be sampled.

[x,y] = meshgrid(0:IPMC_width*num, 0:IPMC_length*num);
pz = zeros(1,(IPMC_width*num+1)*(IPMC_length*num+1));
p = [x(:)'; y(:)'; pz]*(1e-3/num);
p_cat = [x(:)'; y(:)'; pz+(num*190e-3)]*(1e-3/num);



%Extract voltage information from electrical model

IPMC_anode_voltage = postinterp(fem,'V',p);
IPMC_cathode_voltage = postinterp(fem,'V',p_cat);
%Calculate effective voltage difference, the difference between the
%voltages of the cathode and anode
```

```
IPMC_voltage = IPMC_anode_voltage-IPMC_cathode_voltage;


%Create a table with voltages and positions,
IPMC_voltage_table=[x;y;z;V]


IPMC_voltage_table = [p;IPMC_voltage];


%Create table with concentrations and positions, IPMC_concentration =
[x;y;z;c]


%Create vector of z heights

[~,n]= size(p);

z = (0:num_z:180)*1e-6;

one = ones(1,180/num_z+1);

IPMC_concentration = [];


%Select a concentration profile for a position xy given the effective
voltage at that point


for i = 1:n


   V = IPMC_voltage_table(4,i);


   %Choose this concentration profile if the voltage at xy has these
values


    [con] = concentration_profile_selection(concentration,num_z,V,s);


    %create a matrix of x and y positions, xy_pos = [x;y]

    xy_pos = [p(1,i)*one; p(2,i)*one];
```

```
    %horizonatally concatenate the xy position matrix to the z and
concentration matrices

    new_con = [xy_pos;z;con];

    %horizontally concatenate the concentration values for each xy
position

    IPMC_concentration = [IPMC_concentration new_con];



end



%Write the final xyz concentration data to a text file where the
filename

%is specified by the user.  The form of this file is [x y z
concentration Force],

%the file must be accessed and the header "%x y z con Force" added in

%the first row in order to make the file readable to Comsol.



dcon = IPMC_concentration(4,:)-1250;

IPMC_c = [IPMC_concentration; dcon.^2*9e2*F_multiplier];



filename = input('enter a file name or number = ','s');%prompt user for
input

dlmwrite(filename,IPMC_c','delimiter','\t','precision','%2.6f','newline
','pc');%write the text file



end
```

A6. Function Triangular_force_scan

```
function [Force,xPos,yPos] = Triangular_force_scan()

%This function simulates a distributed force measurement for a
triangular IPMC.  It returns three vectors, a vector of x positions, a
vector of y positions, and a force vector.  Together they define the
force distribution for the actuator.
```

```
%Transducer measurement positions

xPos = [1*ones(1,11) 2*ones(1,11) 3*ones(1,11) 4*ones(1,11)
5*ones(1,11) 6*ones(1,11)]*1e-3;

yPos= [1 NaN(1,10) 1:6 NaN(1,5) 1:10 NaN 1:10 NaN 1:6 NaN(1,5) 1
NaN(1,10)]*1e-3;

[~,n] = size(xPos);

%Preallocation for recording force measurements

Force = zeros(1,n);


%Create loop that measures at each position

for i = 1:n


    if isnan(yPos(i))==1

        Force(i) = NaN;

    else

flclear xfem


% COMSOL version

clear vrsn

vrsn.name = 'COMSOL 3.5';

vrsn.ext = 'a';

vrsn.major = 0;

vrsn.build = 603;

vrsn.rcs = '$Name:  $';

vrsn.date = '$Date: 2008/12/03 17:02:19 $';

xfem.version = vrsn;


% Geometry 2

carr={curve2([0,0.0070],[0,0]), ...

  curve2([0.0070,0.0050],[0,0.01]), ...
```

```
  curve2([0.0050,0.0020],[0.01,0.01]), ...

  curve2([0.0020,0]),[0.01,0])};

g1=geomcoerce('curve',carr);

carr={curve2([0.0020,0.0035,0.0050],[0.01,0.015,0.01],[1,0.707106781186
5475,1]), ...

  curve2([0.0050,0.0020],[0.01,0.01],[1,1])};

g2=geomcoerce('solid',carr);

g3=geomcoerce('solid',{g1});

g4=geomcomp({g2,g3},'ns',{'CO1','CO2'},'sf','CO1+CO2','edge','none');

g5=geomdel(g4);

g6=extrude(g5,'distance',[180e-
6],'scale',[1;1],'displ',[0;0],'twist',[0],'face','none','wrkpln',[0 1
0;0 0 1;0 0 0]);



% Geometry 1

g7=cylinder3('5e-4','1e-
3','pos',{num2str(xPos(i)),num2str(yPos(i)),'0'},'axis',{'0','0','-
1'},'rot','0');

flclear fem



% Analyzed geometry

clear s

s.objs={g6,g7};

s.name={'EXT1','CYL1'};

s.tags={'g6','g7'};



fem.draw=struct('s',s);

fem.geom=geomcsg(fem);



% Constants

xfem.const = {'R','8.314', ...

  'T','293', ...
```

```
  'Diffusion','1e-10', ...
  'F','96485', ...
  'Eps','20*8.85e-12', ...
  'Youngs','5e8', ...
  'Poissons','.487', ...
  'Density','2000', ...
  'IPMC_x','15e-3', ...
  'IPMC_y','5e-3', ...
  'IPMC_z','180e-6'};

% Initialize mesh for geometry 2
fem.mesh=meshinit(fem, ...
                  'hauto',4, ...
                  'zscale',1, ...
                  'point',[], ...
                  'edge',[], ...
                  'face',[], ...
                  'subdomain',[2]);

% Initialize mesh for geometry 1

fem.mesh=meshinit(fem, ...
                  'hauto',2, ...
                  'zscale',5, ...
                  'point',[], ...
                  'edge',[], ...
                  'face',[], ...
                  'subdomain',[1], ...
                  'meshstart',fem.mesh);
```

```
xfem.fem{1}=fem;

% (Default values are not included)

fem=xfem.fem{1};

% Application mode 1
clear appl
appl.mode.class = 'SmeSolid3';
appl.module = 'MEMS';
appl.gporder = 4;
appl.cporder = 2;
appl.sshape = 2;
appl.assignsuffix = '_smsld';
clear prop
prop.largedef='off';
appl.prop = prop;
clear bnd
bnd.constrcond = {'free','fixed'};
bnd.ind = [1,2,1,1,1,1,2,1,1,1,1,1];
appl.bnd = bnd;
clear equ
equ.nu = {'Poissons','mat1_nu'};
equ.rho = {'Density','mat1_rho'};
equ.betadK = {1.5,0};
equ.dampingtype = {'Rayleigh','nodamping'};
equ.E = {'Youngs','mat1_E'};
equ.n = {1,'mat1_n'};
equ.Fy = {'.25*Force(x[1/m],y[1/m],z[1/m])',0};
```

```
equ.alpha = {0,'mat1_alpha'};

equ.ind = [1,2];

appl.equ = equ;

fem.appl{1} = appl;

fem.frame = {'ref'};

fem.border = 1;

clear units;

units.basesystem = 'SI';

fem.units = units;

xfem.fem{1} = fem;


flclear fem

fem.sdim = {'x','y'};

fem.border = 1;

clear units;

units.basesystem = 'SI';

fem.units = units;

xfem.fem{2} = fem;


% Functions

clear fcns

fcns{1}.type='interp';

fcns{1}.name={'con','Force'};

fcns{1}.filename='C:\Users\M Martinez\Desktop\comsol models\Matlab
Codes\Concentration_triangular_2V_1s.txt';

fcns{1}.fileindex={'1','2'};

fcns{1}.method={'linear','linear'};

fcns{1}.extmethod={'const','const'};

xfem.functions = fcns;


% Library materials
```

```
clear lib

lib.mat{1}.name='Silica glass';

lib.mat{1}.varname='mat1';

lib.mat{1}.variables.nu='0.17';

lib.mat{1}.variables.E='73.1e9[Pa]';

lib.mat{1}.variables.mur='1';

lib.mat{1}.variables.sigma='1e-14[S/m]';

lib.mat{1}.variables.epsilonr='2.09';

lib.mat{1}.variables.alpha='0.55e-6[1/K]';

lib.mat{1}.variables.C='703[J/(kg*K)]';

lib.mat{1}.variables.n='1.45';

lib.mat{1}.variables.rho='2203[kg/m^3]';

lib.mat{1}.variables.k='1.38[W/(m*K)]';

lib.matgroups{1}.name='Resistivity';

lib.matgroups{1}.variables={'alphares','T0','res0'};

lib.matgroups{1}.descr={'Temperature coefficient','Reference
temperature','Resistivity at reference temperature'};



xfem.lib = lib;


% ODE Settings

clear ode

clear units;

units.basesystem = 'SI';

ode.units = units;

xfem.ode=ode;


% Multiphysics

xfem=multiphysics(xfem);
```

```
% Extend mesh
xfem.xmesh=meshextend(xfem, ...
                      'geoms',[1]);


% Solve problem
xfem.sol=femstatic(xfem, ...
                   'solcomp',{'w','v','u'}, ...
                   'outcomp',{'w','v','u'}, ...
                   'blocksize','auto', ...
                   'linsolver','spooles');


% Save current fem structure for restart purposes
fem0=xfem;


% Plot solution
% postplot(xfem, ...
%          'tetdata',{'disp_smsld','cont','internal','unit','m'}, ...
%          'tetmap','Rainbow', ...
%          'tetkeep',1, ...
%          'tetkeeptype','random', ...
%          'title','Subdomain: Total displacement [m]', ...
%          'grid','on', ...
%          'campos',[0.0498865950826033,-
0.0427809490344632,0.018360512787978265], ...
%
'camtarget',[0.0035000001080334187,0.006035533733665943,9.0000001364387
57E-5], ...
%          'camup',[-
0.09935968063307937,0.26459539618241923,0.9592272567976585], ...
%          'camva',6.881909873964115);
```

```
% Summation
Force(i)=postsum(xfem,'RFz_smsld', ...
            'unit','N', ...
            'recover','off', ...
            'dl',2);


% Geometry 2
fem=xfem.fem{2};


% Geometry objects
clear s
s.objs={g5};
s.name={'CO1'};
s.tags={'g5'};


fem.draw=struct('s',s);
xfem.fem{2}=fem;
    end


end
Force = reshape(Force,11,6);


end
```

REFERENCES

[1] M. Shahinpoor, Y. Bar-Cohen, J. Simpson and J. Smith, "Ionic polymer-metal composites (IPMCs) as biomimetic sensors, actuators and artificial muscles- a review," *Smart Materials and Structures,* pp. 7(6): p.R15-30, 1998.

[2] S. Tadokoro, S. Yamagami, T. Takamori and K. Oguro, "Modeling of nafion-Pt composite actuators (ICPF) by ionic motion," in *Smart Structures and Materials 2000: Electroactive Polymer Actuators and Devices* , Newport Beach, 2000.

[3] C. Bonomo, L. Fortuna, P. Giannone and S. Graziani, "A circuit to model the electrical behavior of an ionic polymer-metal composite," *Circuits and Systems I: Regular Papers, IEEE Transactions on,* pp. vol.53, no.2, pp. 338- 350, 2006.

[4] M. Shahinpoor and K. Kim, "Ionic polymer-metal composites - I. fundamentals," *Smart Materials and Structures,* pp. 10(819): p.819-33, 2001.

[5] W. Zhou and J. Mai, "MEMS-fabricated ICPF microactuators for biological manipulation," *Smart Materials and Structures,* p. 332, 2003.

[6] T. Gierke, G. Munn and F. Wilson, "The morphology in nafion perfluorinated membrane products, as determined by wide- and small-angle x-ray studies," *Journal of Polymer Science,* p. 19(11): 1687, 1981.

[7] S. Nemat-Nasser, "Micromechanics of actuation of ionic polymer-metal composites," *Journal of Applied Physics,* pp. 92(5): p.2899-915, 2002.

[8] S. Nemat-Nasser and Z. Shahram, "Modeling of the electrochemomechanical response of ionic polymer-metal composites with various solvents," *Journal of*

Applied Physics, p. 064310, 2006.

[9] S. Nemat-Nasser and J. Li, "Electromechanical response of ionic polymer-metal composites," *Journal of Applied Physics,* pp. 87(7): p. 3321-31, 2000.

[10] K. S. R. Sadeghipour and S. Neogi, "Development of a novel electromechanically active membrane and 'smart' material based vibration sensor/damper," *Smart Materials and Structures* , pp. 1(1992): p.172-79, 1992.

[11] C. Bonomo, L. Fortuna, P. Giannone, S. Graziani and S. Strazzeri, "A model for ionic polymer-metal composites as sensors," *Smart Materials and Structures,* pp. p. 749-58, 2006.

[12] M. Shahinpoor, "A new effect in ionic polymeric gels: the ionic flexogelectric effect," *Smart Materials and Structures,* pp. p. 42-53, 1995.

[13] U. Deole, "Artificial muscle microgrippers," *M.of Science, Dept. of Mechanical Engineering, University of New Mexico,* 2005.

[14] Y. Bar-Cohen, "Challenges to the application of IPMC as actuators of planetary mechanisms," *Proc. of SPIE's 7th annual Int. Symp. of Smart Stuctures and Materials,* pp. 1-6, 2000.

[15] R. Kanno, A. Kurata, M. Hattori, S. Tadokoro, T. Takamori and K. Oguro, "Characteristics and modeling of ICPF actuator," in *Proceedings of Japan-USA Symposium on Flexible Automation*, Kobe, 1995.

[16] Y. Bar-Cohen, "Electroactive polymers as artificial muscles - realities and challenges," in *Proceedings of the 42nd AIAA*, 2001.

[17] S. Nemat-Nasser, S. Zamani and Y. Tor, "Effects of solvents on the chemical

and physical properties of ionic polymer-metal composites," *Journal of Applied Physics,* pp. 99(10): p.104902-1-17, 2006.

[18] M. Shahinpoor and K. Kim, "The effect of surface-electrode resistance on the performance of ionic polymer-metal composite (IPMC) artificial muscles," *Smart Materials and Structures,* pp. 9(4): p. 543-51, 2000.

[19] K. Kim and M. Shahinpoor, "Ionic polymer-composites: II. manufacturing techniques," *Smart Materials and Structures,* pp. p.65-79, 2003.

[20] C. Bonomo, L. Fortuna, P. Giannone, S. Graziani and S. Strazzeri, "A nonlinear model for ionic polymer metal composites as actuators," *Smart Materials and Structures,* pp. 16(2007): p.1-12, 2007.

[21] P. Branco and J. Dente, "Derivation of a continuum model and its electric equivalent-circuit representaion for ionic polymer-metal composite (IPMC) electromechanics," *Smart Materials and Structures,* pp. 15(2006): p. 378-392, 2006.

[22] N. Bhat and K. Kim, "Precision force and position control of an ionic polymer-metal composites," *Proceedings of the IMechE,* pp. Vol. 218: p.421-31, 2004.

[23] B. Lopes and P. Branco, "Electromechanical characterization of non-uniform charged IPMC devices," *Journal of Physics: Conference Series 127,* 2008.

[24] T. ,. A. F. Johnson, "Multiphysics modeling of an IPMC microfluidic device," *Microsyst. Technology,* pp. 14:871-879, 2008.

[25] D. Pugal, A. Aabloo and K. Kim, "Modeling IPMC material with dynamic surface characteristics," *Proceedings of Smart Materials, Adaptive Structures and*

Intelligent Systems, 2009.

[26] D. Pugal, "Model of self-oscillating ionic polymer-metal composite bending actuator," *Thesis. Tartu University,* 2009.

[27] J. Jung, B. Kim, Y. Tak and J. Park, "Undulatory tadpole robot using ionic polymer-composite actuator," *Intelligent Robots and Systems,* pp. vol 3: p 2133-38, 2003.

[28] X. Tan, D. Kim, U. N., D. Laboy and J. J., "An autonomous robotic fish for mobile sensing," *International Conference on Intelligent Robots and Systems,* pp. 5424-5429, 2006.