Walking Silicon: Actuators and Legs for Small-Scale Terrestrial Robots

by

Daniel Savino Contreras

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Kristofer S. J. Pister, Chair
Professor Michel Maharbiz
Professor Liwei Lin

Summer 2018

Walking Silicon: Actuators and Legs for Small-Scale Terrestrial Robots

Abstract

Walking Silicon: Actuators and Legs for Small-Scale Terrestrial Robots

by

Daniel Savino Contreras

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Kristofer S. J. Pister, Chair

This dissertation presents work on components and actuators for silicon-based walking centimeter-scale robots. The focus on this work was on the actuators used to drive these robots and the linkages that make the basic structure of the robot leg.

Pin-joints are used as the basic unit of the leg linkages. The pin-joints were tested in terms of robustness and demonstrated high maximum tensile loads of over 5mN and compressive loads in excess of 100mN.

Electrostatic inchworm motors were the actuator of choice for these robots. These motors demonstrated over 1mN of output force and shuttle speeds up to 0.4m/s. Newer high force motors have demonstrated 5mN output force at 100V, the highest from an electrostatic inchworm motor to date.

These components were combined for a single-legged walker which demonstrated walking under power and support from external wires. After the single legged walker a hexapod robot using multichip assembly was designed, fabricated, assembled, and tested. The robot used three separate chips to route signals from planar legs to a central hub chip that is tethered to an external control circuit by 9 wires. The robot demonstrated taking steps.

This work also presents a future vision for robots based on this same silicon technology. Using these components a new generation of walking robots can be developed, pushing the path forward toward autonomous operation.

Contents

List of Figures

List of Tables

Acknowledgments

I owe a great deal of thanks to the staff and other members of the Marvell Nanofabrication Laboratory. I would never have gotten far in my work without their unending support and wide breadth of knowledge. Thank you to the staff members of BSAC as well as the financial support from BSAC members. Thank you to EECS staff for all their help and guidance.

I would also like to thank the members of the Pister Lab who have been with me through my entire time at Berkeley. Thanks to David Burnett and Joey Greenspun for being excellent mentors and my introduction to graduate school at UC Berkeley, Brad Wheeler and Daniel Drew for making an excellent cohort, Fil Maksimovic for being a helpful ear, Hani Gomez, Craig Schindler, and Brian Kilberg for their refreshing enthusiasm, and Travis Massey for his infinite generosity with materials and time. And a special thanks to Daniel Drew and Joey Greenspun for being the best cleanroom buddies I could have asked for.

Thanks to Daniel Drew, Joey Greenspun, Fil Maksimovic, Claire Lochner, and Balthazar Lechene for our many adventures across Varisia.

Thank you to Kris Pister for being a fantastic advisor, patient mentor, and for giving me the opportunity to work on something so crazy, difficult, and fascinating.

Thanks to Emily Chang who has given me happiness, structure, and warmth in my life. Especially in the form of two adorable cats that have destroyed any previously held notion that I would never be a cat person.

Lastly I would like to thank my family. Caminaron muchas millas y trabajaron con todas sus fuerzas para darnos esta oportunidad. Gracias por todo tu apoyo y ayuda. Me dan coraje seguir adelante.

Chapter 1

Introduction

When most people think of robots they imagine a wide variety of machines. Robotic toy dogs for entertainment, large industrial factory machines, and service robots that perform functions like vacuuming and cleaning are first and foremost in the public imagination. There are also autonomous drones, which are prevalent in the commercial and military space. Self driving cars are also shaping the public perception of robots and machine learning. Even households are being outfitted with some degree of automation with smart speakers and a constant barrage of household sensors on the market.

Most of these robots range from tens of centimeters to meters in size. The incredible advances in mechanical and electrical design at the human-scale have made all this possible. Yet trying to make robots smaller leads to a handful of critical issues that are much easier to overcome at these larger scales. Power at smaller size scales is costly in terms of mass, since every milligram is a critical loss of force output from an actuator. Also, integrating the necessary high force actuators with these power sources is by no means a trivial task. Assembly becomes difficult when the precision necessary enters the limits of human capability.

The focus of this work is to demonstrate a mobility platform for millimeter to centimeter scale terrestrial robotics, targeting autonomous operation with on-board power and computation. This platform relies on a combination of silicon processing and hand assembly.

1.1 The World of Tiny Robots

The field of millimeter to centimeter-scale robotics has made significant strides in the last few decades. Numerous small-scale robots that are able to walk, fly, and swim have been demonstrated on a variety of platforms with a myriad of clever actuation schemes.

There are a number of centimeter scale robots that have achieved autonomous operation. HAMR, a biologically-inspired robot that uses piezoelectric actuators and measures 4.5cm in length and weighs 2.8g has recently demonstrated walking with its own on-board power supply [1, 2]. Kilobot, another centimeter-scale robot, is fully autonomous and is externally programmable on a large scale by light [3].

Moving on to smaller scales, on-board power and computation quickly become an issue. Robot control falls into three main categories; external field control, tethered control, and untethered autonomous operation.

Going over external field control, levitating diamagnetically controlled robots from SRI have demonstrated large scale control of numerous robots and can assemble macroscale structures using specialized tools [4, 5]. There are also electric field actuated microrobots that can perform complex tasks with multiple robots [6], as well as magnetic field actuated legged robots that can move at many body lengths per second [7, 8].

Tethered robots rely on external wires being connected to the body of the robot, providing power and control from an external source. A robot using flexible polyimide joints and thermal actuators demonstrated that it was capable of lifting a load of 2.5g [9]. Another robot used shape-memory-alloy actuators to drive hand assembled silicon linkages [10, 11].

Fully autonomous operation by a robot is still a difficult problem. A robot using an electrostatically actuated resonator demonstrated motion without external wires for approximately 10 seconds [12]. Another robot used a stack of multiple chips, which included a chip of microelectromechanical systems (MEMS) actuators and mechanisms, a custom digital CMOS brain, and an array of high voltage solar cells for a power source [13]. This robot was ultimately unable to walk forward due to issues with the locomotion mechanism.

1.2 Little Robots, Big Problems

The breadth of microrobotics literature shows the many innovations in actuation and locomotion schemes but points to the central issue of autonomy. Power sources that are light enough for robots to lift and provide enough power are hard to come by and even with these power sources the robots are not long lived, moving slowly and with great effort.

The tethered robots described earlier suffer from high power draw. The robot based on shape-memory-alloy actuators consumes approximately 0.4W to operate [14] and can move at speeds of 1-2mm/s. The polyimide joint based robot consumes up to 1.1W at its maximum speed of 6mm/s with 4 times its body weight as a load. Recent battery technology has demonstrated energy densities of 0.36J/mg [15]. At the load capacity of this robot, it could operate for 100 seconds and move 60cm, not counting the weight and power of on board computation.

The electrostatic resonant actuator robot can operate by itself and move relatively quickly at 33mm/s while tethered, but when untethered it moves at 2mm/s, for 10s with 2.5kV stored on a 10nF capacitor [12].

The previous 10mg MEMS Hollar microrobot came close to moving autonomously. The robot was able to actuate by itself while under light for extended periods of time. It has even demonstrated functioning nearly 15 years later. However, it failed to take a step forward with its locomotion scheme. The robot was meant to crawl forward while dragging its rear half. The arms could not overcome the friction in the rear half to move forward.

This robot used high force, low power electrostatic motors to drive a silicon linkage fabricated alongside the motors. The motors were approximately 4% of the total power draw, with high voltage buffers consuming a majority of the power. This was the key to its autonomous operation and what is improved upon in this work.

1.3 This Work

The work presented in this dissertation is as follows. First work on each of the unit cells of the leg is presented in Chapter 2 and Chapter 3. These include the silicon pin-joint, which makes up the basic mechanical structure of the leg, and the electrostatic inchworm motor, used to drive the linkages and actuate the leg. These chapters focus on characterization of these components in terms of durability for the joints, and force output of the motors.

Chapter 4 goes into detail on the dynamics of the motors. The details on the MATLAB dynamics model are shown. It also presents an optimization scheme for finding an ideal motor device layer thickness.

Chapter 5 discusses the design, assembly, and testing of the single legged robot. This robot was a test for the concept of planar robots. The robot was made from a combination of electrostatic inchworm motors and pin-joints from the previous chapters.

Chapter 6 goes over the next stage for the robotics platform, the hexapod silicon robot. After successful demonstration of the single legged walker, the next step was to implement more legs for a more statically stable and faster design.

Chapter 7 concludes this work, presenting a vision for the future of the platform and laying a series of next steps to achieve a truly autonomous walking robot system.

Chapter 2

Silicon Pin-Joints

The basic mechanism of the robot leg is the silicon pin-joint. Ideally these structures have a single rotational degree of freedom in the plane of the patterned silicon layer. By joining these unit cells together we are able to form complex linkages with rotational and translational degrees of freedom in the plane of the wafer, all using a single mask.

2.1 Basic Design

Fig. 2.1 shows a cartoon diagram of a silicon pin-joint. A rotating element sits inside of a joint holder and the two pieces are held together by a weak rotary spring. The mechanism allows for motion in six axes with constraints. Since the pin-joint is fabricated in a single layer of silicon and does not have a capping structure to hold the rotating element in plane,

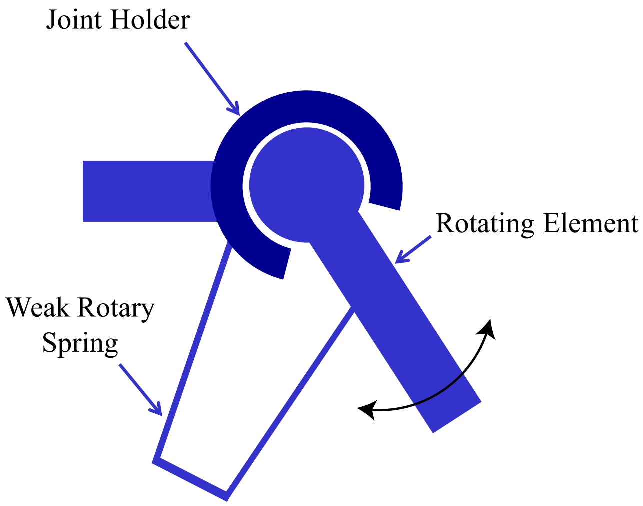


Figure 2.1: A diagram showing the basic pin-joint unit cell

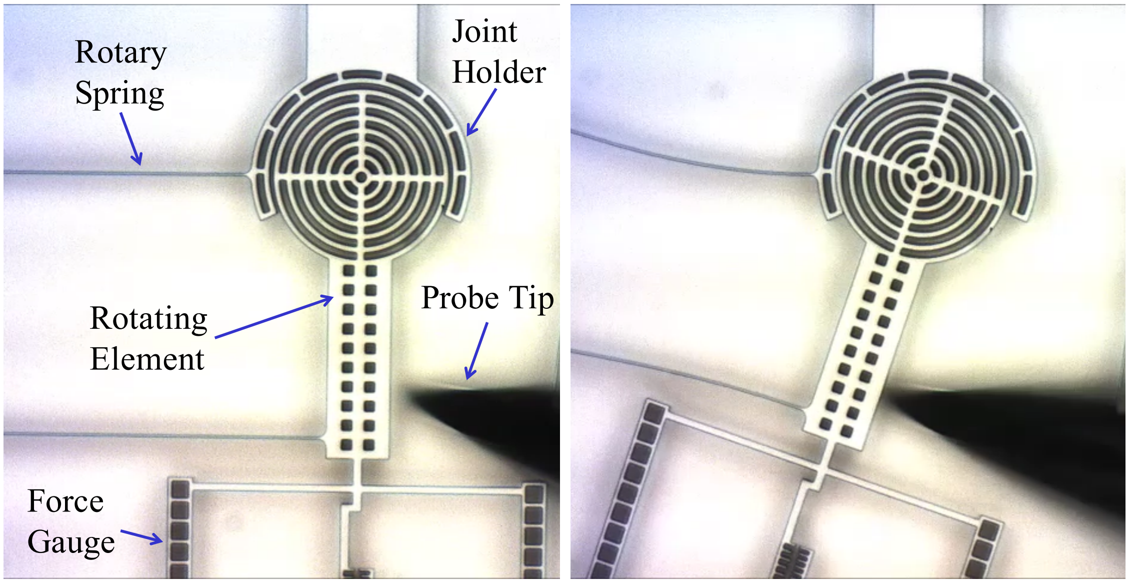


Figure 2.2: A fabricated pin-joint. The pin-joint is fabricated with etch holes to ensure its release during fabrication. The static pin-joint with parts labeled is shown on the left. The frame on the right shows the pin-joint manually rotated using a probe tip

a weak rotary spring is used to hold the joint together after the structure is released. The spring is stiff enough to hold the structure together but weak enough to allow the joint to fully rotate without breaking. A fabricated pin-joint test structure is shown in Fig. 2.2. In the image sequence, the structure is manually rotated using a tungsten probe tip on a micromanipulator. The force gauge partially seen in the figure is used to measure the durability of the joints during tensile strength experiments.

2.1.1 Limits on Angular Travel

This simple pin-joint design is fundamentally limited in its angular travel by the radius of the rotating element, the size of the opening, and the size of the link extending out of the rotating element. As the opening approaches the size of the rotating element, the rotating element can slip out of the holder with no resistance, and would only be held together by the rotary spring. This is shown in Fig. 2.3. On the other hand, if we confine the rotating element too much within the joint holder we reduce its angular travel.

The limitation in travel is influenced by a few key geometries in the joint. The radius of the inner joint holder is $r_{jh=}d/2+g$. Looking at the chord length of the circle, w_o, we can implement some degree of tolerance, t, such that $w_o = d - t$ to keep the rotating element from slipping out. This gives the angle of the opening as

$$\theta_o = 2\arcsin\frac{\frac{d-t}{2}}{d/2+g} = 2\arcsin\frac{d-t}{d+2g} \tag{2.1}$$

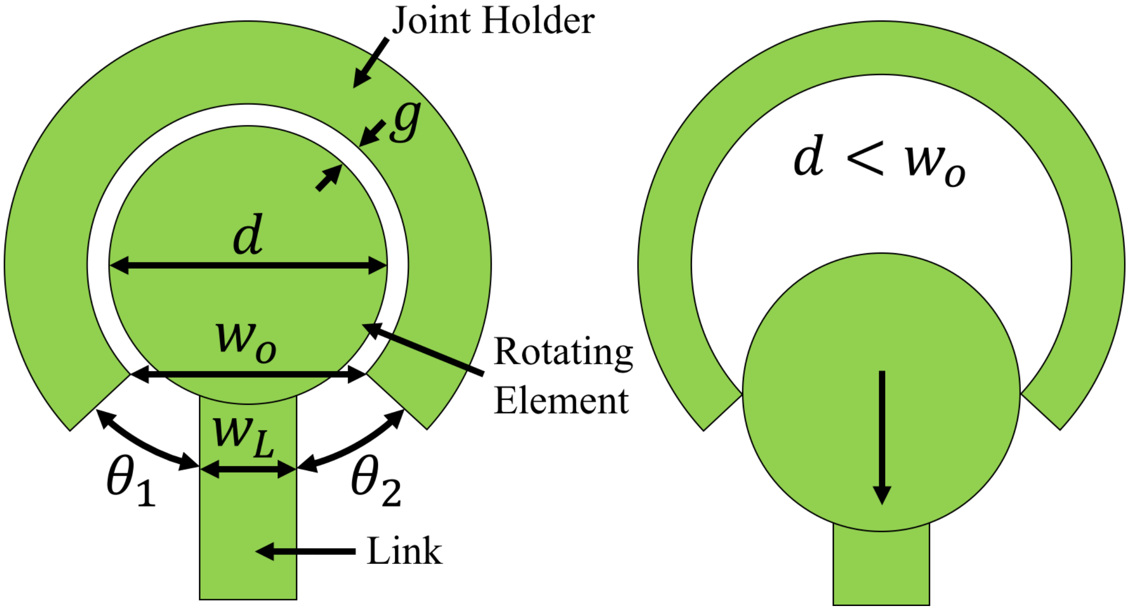


Figure 2.3: A diagram of the dimensions of the pin-joint. There are three critical dimensions when designing the pin-joint, the diameter of the central portion of the rotating element, d, the gap between the rotating element and the joint holder, g, and the width of the opening, w_o. w_o determines the angular size of the opening, θ

Another fundamental limit is the width of the beam that the rotating element is connected to, W_L. This further limits the range of angular travel by the angular size of the beam from the center of the rotating element. Subtracting this from the angular opening we can find the actual angular travel of the joint

$$\theta_{travel} = \theta_1 + \theta_2 = 2\arcsin\frac{d-t}{d+2g} - 2\arcsin\frac{W_L}{d} \tag{2.2}$$

It should be noted that θ_1 and θ_2 are usually not equal, since the distribution of the angular travel around the joint depends on the desired initial position after fabrication and final position at end of travel of the joint.

2.1.2 Chosen Leg Joint Geometries

Eq. 2.1 and Eq. 2.2 are plotted for various radii and gap sizes in Fig. 2.4 with $t = 1\mu$m. With this tolerance the joint slowly approaches an opening of 160°. This does not factor in the size of the link extending out of the rotating element

Fig. 2.4 also shows the curves corresponding to the parameters of the joints of the robot legs. The minimum feature size in the silicon fabrication process used to create these joints

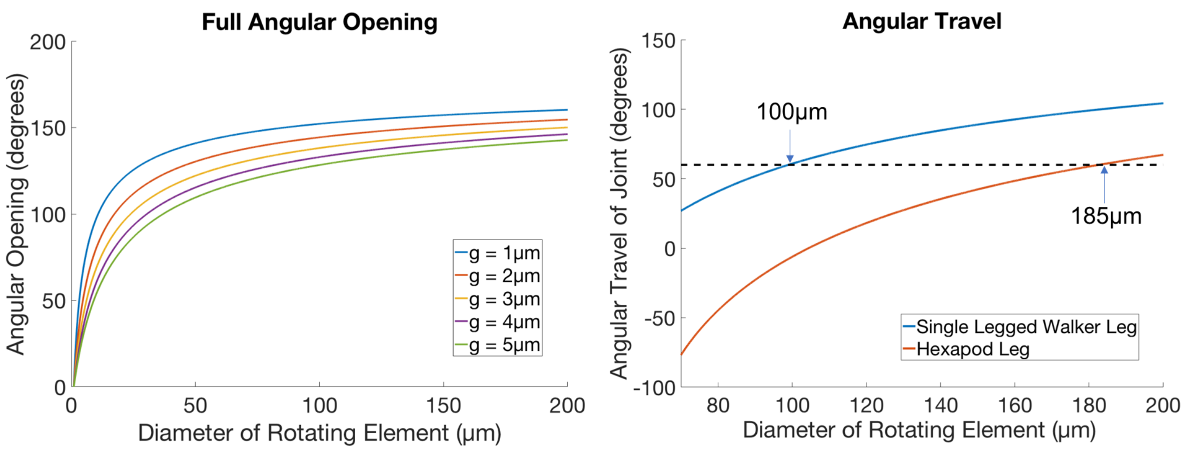


Figure 2.4: A plot of the angular opening of the joint as a function of the diameter of the rotating element with a tolerance of 1μm and varying gap sizes. The angular travel approaches over 160° for a 1μm gap. The angular limit is lower for larger gap sizes (left) A plot of the design space used for the robot legs of the single-legged walker and the hexapod (right)

will determine g. For our process this is 2μm line and space. Additionally, a higher tolerance than 1μm is also preferred for the sake of durability of the joint as it undergoes tensile loading.

The single-legged walker was designed with joints having a tolerance of 20μm and link width of 34μm, while the hexapod joints had a tolerance of 40μm and a link width of 62μm. These dimensions were overdesigned for the sake of leg durability and safety while handling the robots. Additionally, for each robot an angular travel of 60° was desired. This was the maximum joint travel for each linkage's designed sweep patterns. This is marked by the black dashed line, also marking the radii of the joint's rotating elements.

2.1.3 Rotary Spring Design

Rotational Stiffness

The pin-joints are fragile structures since they are only held together by weak silicon beams. Initial spring designs used a basic rectangular spring that allowed for motion in the plane of the joint. We can approximate the rotational stiffness of the rotary spring by assuming the spring acts as a linear spring, k_s, for a small rotational displacement, θ. The moment the spring imparts on the links is given by

$$M = Fr = \kappa\theta \tag{2.3}$$

where F is the force at a moment arm r, the distance from the center of the joint to the spring's attachment point. F can be approximated for a displacement, x, as

$$F = k_s x = k_s r\theta \tag{2.4}$$

where we have translated linear displacement x into an angular displacement $r\theta$. Substituting this F into Eq. 2.3 we get a rotational stiffness of

$$\kappa = k_s r^2 = \frac{ETw_s^3}{NL_s^3} r^2 \tag{2.5}$$

where, as a first order assumption, we can express k_s as the linear serpentine spring constant with fixed-guided boundary conditions at the ends of the beam. This is expressed as

$$k_s = \frac{ETw_s^3}{NL_s^3} \tag{2.6}$$

where E is the Young's Modulus, T is the device thickness, w_s is the spring width, N is the number of beams along the length of the spring, L_s is the beam length.

For the dimensions of the spring used, $T = 40\mu\text{m}$, $w_s = 2\mu\text{m}$, $N = 2$, $L_s = 290\mu\text{m}$ and $r = 75\mu\text{m}$, we get an analytical rotational stiffness of $6 \times 10^{-9}\text{Nm}$.

Fig. 2.5 shows a screenshot of the simulation of the rotary spring. The spring was rotated from $-45°$ to $-45°$. Fig. 2.6 shows a plot of the simulation data. There are nonlinearities for large angular deflections. For angles in the range of travel (-0.5 to 0.5 radians) the simulated rotational stiffness is approximately $5.6 \times 10^{-9}\text{Nm}$ indicating good agreement between the analytical expression and our simulation.

The maximum stress was 800MPa in the simulation. This corresponds to 0.47% maximum strain, below the safety limit of 0.5%. It should be noted that the fabricated device will be subject to an approximate 1μm undercut during the DRIE step and will be approximately 8 times less stiff due to losing half its drawn width.

Out-of-Plane Stiffness

Out-of-plane stiffness was also important to characterize to see how weak these springs are to out-of-plane forces. Fig. 2.7 shows a simulation of the out-of-plane motion of a joint. The long sections of the beam are deformed torsionally. The stiffness will come from a combination of this torsional deflection and the linear deflection of these torsionally deformed beams. The torsional spring constant of these beams is given by

$$\kappa_{beam} = \frac{JG}{L_s} = \frac{Tw_s^3 G}{3L_s} \tag{2.7}$$

where J is the torsional constant dependent on the cross-section ($\frac{Tw_s^3}{3}$ for a rectangular cross-section) and G is the shear modulus of the material (80GPa for silicon). The displacement, z, is applied to the two long section beams across the shorter middle beam, L_m. We are also applying the moment at a distance L_m. The out-of-plane stiffness is given by

$$k_z = \frac{Tw_s^3 G}{3L_s L_m^2} \tag{2.8}$$

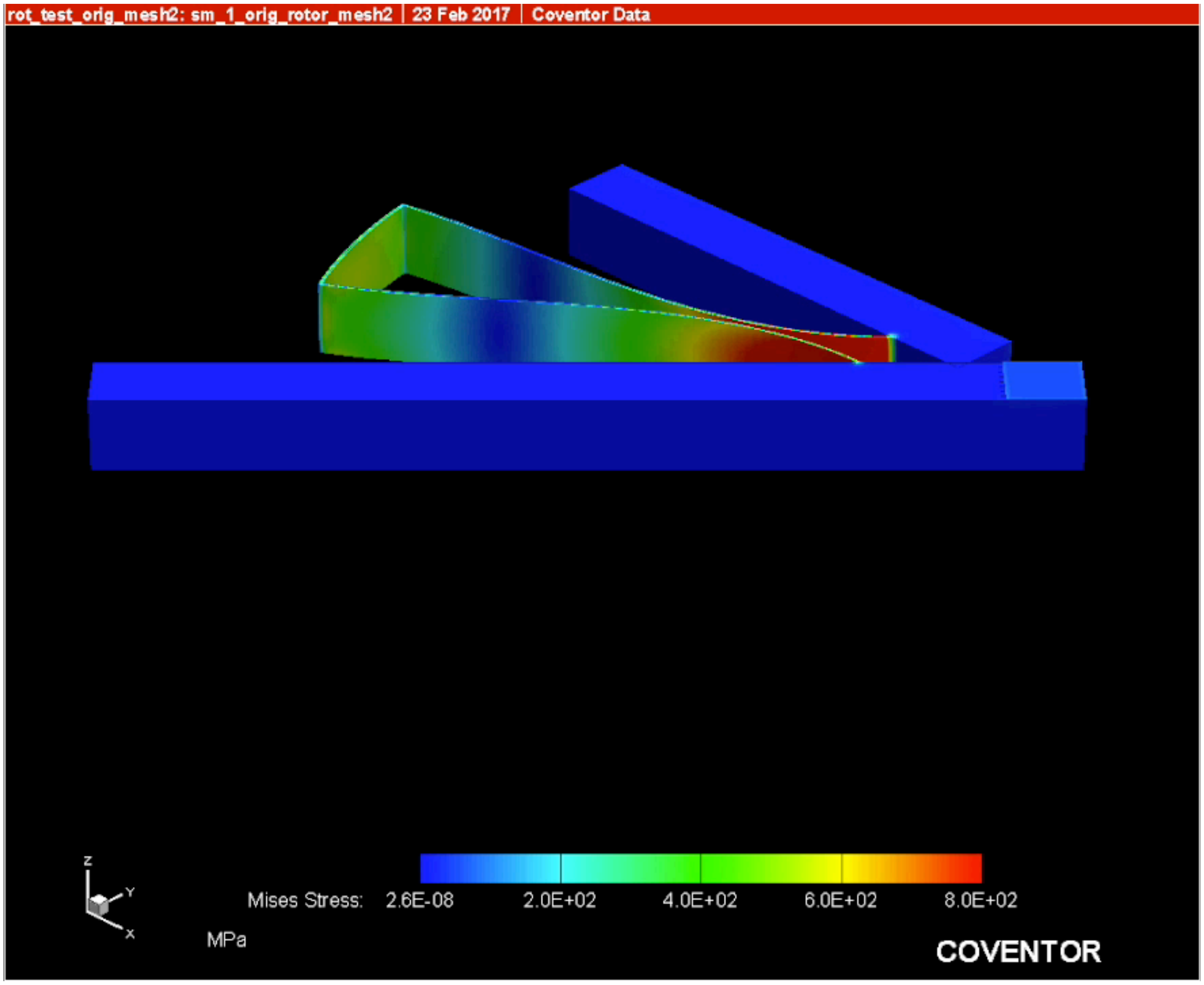

rot_test_orig_mesh2: sm_1_orig_rotor_mesh2 | 23 Feb 2017 | Coventor Data
Z
Y
X
Mises Stress: 2.6E-08 2.0E+02 4.0E+02 6.0E+02 8.0E+02
MPa
COVENTOR

Figure 2.5: A screenshot of the CoventorWare FEA environment used to simulate the rotary springs

For the dimension of our beam we get a stiffness of 2.9N/m. Simulation results, shown in Fig. 2.8, give a stiffness of 0.55N/m. Our analytical result does not factor in the linear bending of the beam which could reduce the spring constant further, matching simulation. According to simulation, the rotary spring would fail under a 20μN out-of-plane load, since the joint would leave its device layer plane at this point.

Improving the Spring Design

A different spring design was used for the hexapod robot. This design used curved springs with multiple meanders to improve the survivability during fabrication. These curved springs were also designed to survive higher stress during operation. The design for these springs is

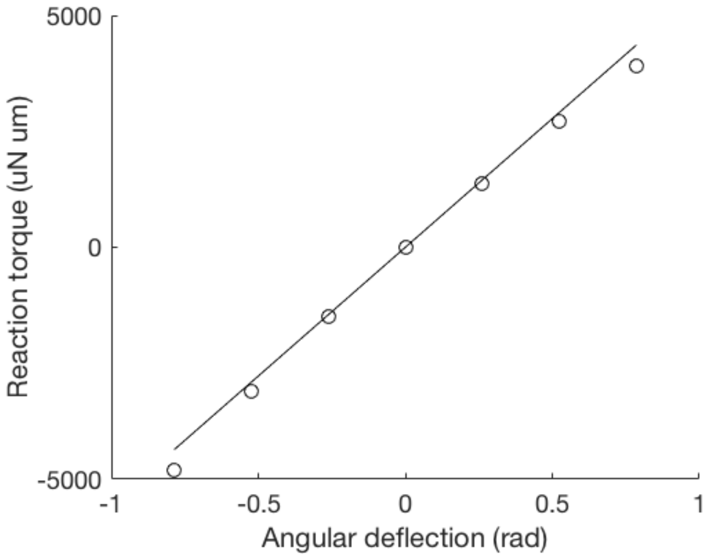


Figure 2.6: Plotted result of the rotational moment caused by the rotary spring on the joint from the CoventorWare simulation. The solid line corresponds to a rotational stiffness of 5.6×10^{-9}Nm

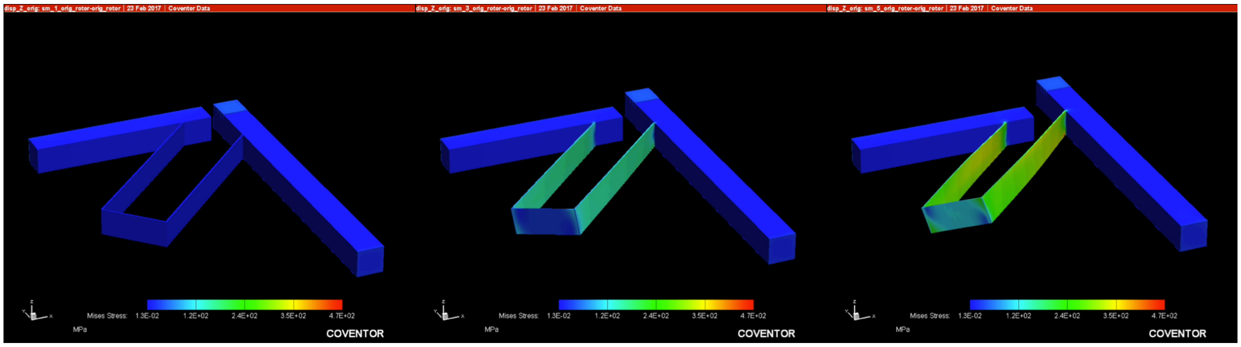

disp_Z_orig: sm_1_orig_rotor-orig_rotor | 23 Feb 2017 | Coventor Data
disp_Z_orig: sm_3_orig_rotor-orig_rotor | 23 Feb 2017 | Coventor Data
disp_Z_orig: sm_5_orig_rotor-orig_rotor | 23 Feb 2017 | Coventor Data
Mises Stress: 1.3E-02 1.2E+02 2.4E+02 3.5E+02 4.7E+02
MPa
COVENTOR

Figure 2.7: Simulation of the out-of-plane motion of the rotary spring to measure the out-of-plane stiffness

detailed in Section 6.2.1.

2.2 Pull-Out Force Characterization

2.2.1 Pin-Joints Under Tension

The impact of the pin-joint opening on the angle of travel was discussed earlier. Some degree of tolerance will ensure that the rotating element is enclosed by the joint holder so

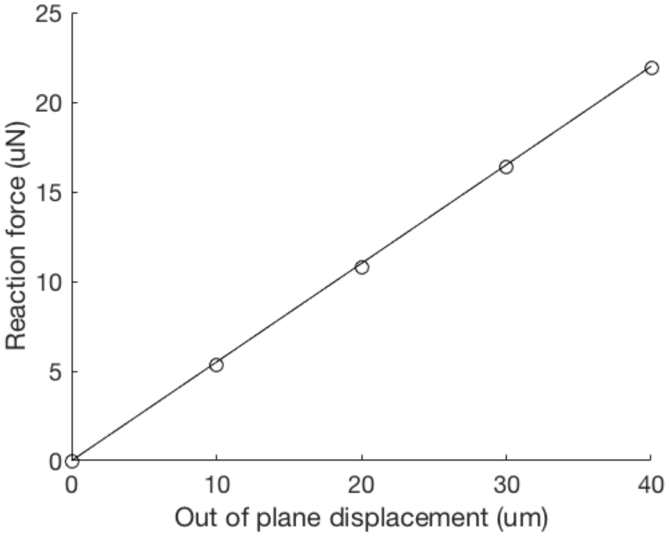


Figure 2.8: Simulation results from the CoventorWare simulation of the rotary spring being linearly displaced out-of-plane

that a tensile load on the joint in the plane of the wafer will not immediately dislocate the joint.

However, even with some degree of tolerance, as designed, the joint holder will bend if enough force is applied. This further limits the tensile loading on the joint. In order to ensure a reliable design for the joints to deal with the forces they will experience in a leg, it was important to analyze bending and friction model for the joints and verify this analysis experimentally.

2.2.2 Curved Beam Bending

The work in this section is adapted from Roark's Formulas for Stress and Strain, specifically the chapter on curved beams [16].

Fig. 2.9 shows a free body diagram of the pin-joint while under tension. The rotating element is pulled while the joint holder is held static. Once the rotating element makes contact with the holder, exerting a force P' on the joint holder. A force $P = \frac{P'}{2}$ is applied to the bending of one side of the joint holder, assuming the rotating element does not deform.

The joint holder has a width W. The radius from the center of the joint to the centroidal axis of the joint holder is given by R. The positional angle, starting from the point of application of the force, is given by x. Force P is broken down into two components, the transverse force with respect to the beam V and the axial force with respect to the beam H. From these component forces we can use Castigliano's Theorem of beam bending.

Castigliano's theorem, or the theorem of least work, states that in a static system under stress the system will achieve the minimum energy possible given its boundary conditions.

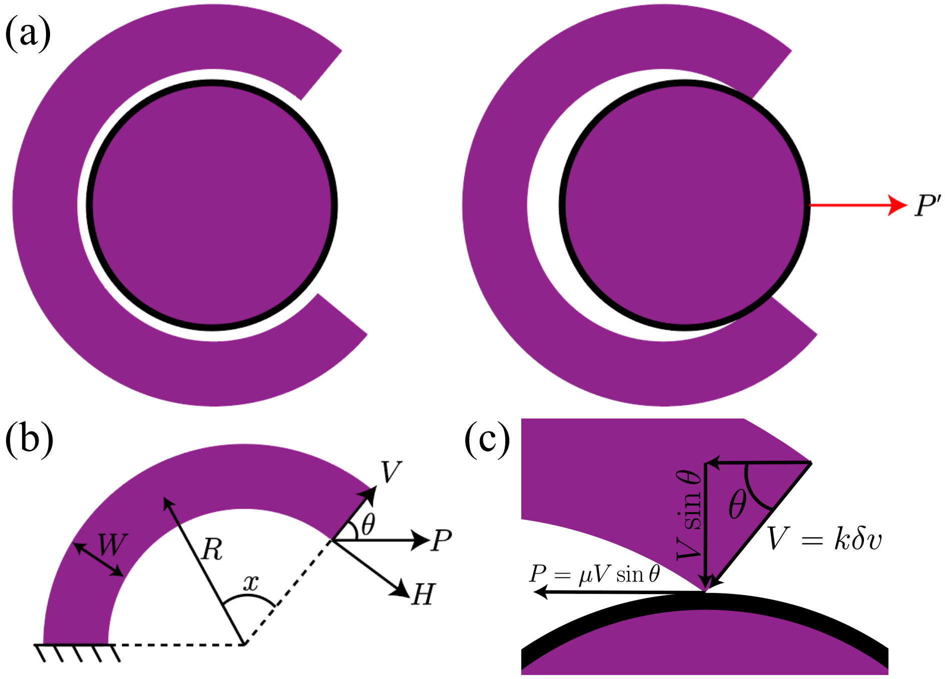


Figure 2.9: The free body diagram of a pin-joint under a tensile load (a) A force P is applied to the rotating element and it makes contact with the joint holder according to the degree of tolerance in the holder (b) This load condition is broken down into a symmetric half model of the joint holder under load $P = P'/2$ (c) The components of the force are broken down further

Thus the total strain energy must equal the energy of the deformation of the beam due to a displacement. In this way, if we find an expression for the strain energy, U_i in terms of the loading, F_i, the resulting displacement x_i can be expressed as a partial derivative of the energy with respect to the load

$$x_i = \frac{\partial U_i}{\partial F_i} \tag{2.9}$$

This will give the force/displacement relationship for a statically indeterminate system. In order to make use of this relationship, we must find the total strain energy in the deflected beam as a function of the loads.

First we find the moment, shear, and axial forces on the beam as a function of the angular position x. These expressions will be

$$M_x = VR\sin x + HR(1 - \cos x) + M_o \tag{2.10}$$

$$V_x = V\cos x + H\sin x \tag{2.11}$$

$$N_x = -H\cos x + V\sin x \tag{2.12}$$

Here M_o is a pure bending moment The total complementary strain energy of the beam will be given by the integral of the moment over the length of the beam

$$U = \int \frac{M_x^2}{2AEe}dx + \int \frac{FV_x^2R}{2AG}dx + \int \frac{N_x^2R}{2AE}dx - \int \frac{M_xN_x}{AE} \tag{2.13}$$

Here, A is the cross-sectional area of the beam and E is the Young's modulus of the material. e is the distance from the centroidal axis to the neutral bending axis. This separation occurs in curved beams since the curvature requires that strain on the outside curve of the beam be different from the strain on the inside curve for a unit segment of the beam because of the difference in lengths. F is the shape factor of the cross-section, required for short thick beams which is what we are dealing with. This is 1.2 for rectangular cross-sections. G is the material's shear modulus.

For single crystal silicon, E and G can vary greatly depending on the orientation of the wafer[17]. For the purposes of this model, since the bending of the of the curved beam is not in a purely cartesian direction, an average Young's Modulus of $160GPa$ and a shear modulus of $65GPa$ were used.

A simplifying assumption we will make is that the deflection of the beam in the direction of V dominates in terms of the pull-out force and the axial deflection is negligible. Obeying the principle of minimum energy, we can equate this strain energy to the energy stored in the deflection of the beam in the direction of V, the relationship between the deflection and the load will be given by

$$\delta v = \frac{\partial U}{\partial V} \tag{2.14}$$

We can measure the expected deflection at the pull-out limit from the tolerance in the layout drawings. We can then express this in the form of Hooke's law to find the force by multiplying this deflection by the constant of proportionality, k, we find in solving Equation 2.14. Thus we get the relationship

$$V = k\delta v \tag{2.15}$$

Fig. 2.9 (c) shows the components of this reaction force. The component acting in parallel to the pull-out force does not have an influence at the moment of pull-out, because this force does not have a surface to act on at the point of max deflection. The component that acts normal to the surface of the joint is taken and multiplied by a friction constant, μ

$$P = \mu V \sin\theta = \mu k \delta v \sin\theta \tag{2.16}$$

This is the value used for P. P' is double this value. For the experimental array, the width of the joint holder and the size of the opening were varied.

2.2.3 Pull-Out Force Test Structure Design and Measurements

Fig. 2.10 shows the layout of the pull-out force test structure. The joint is attached to a force gauge and handle for a probe tip to pull on. The force gauge uses a set of silicon

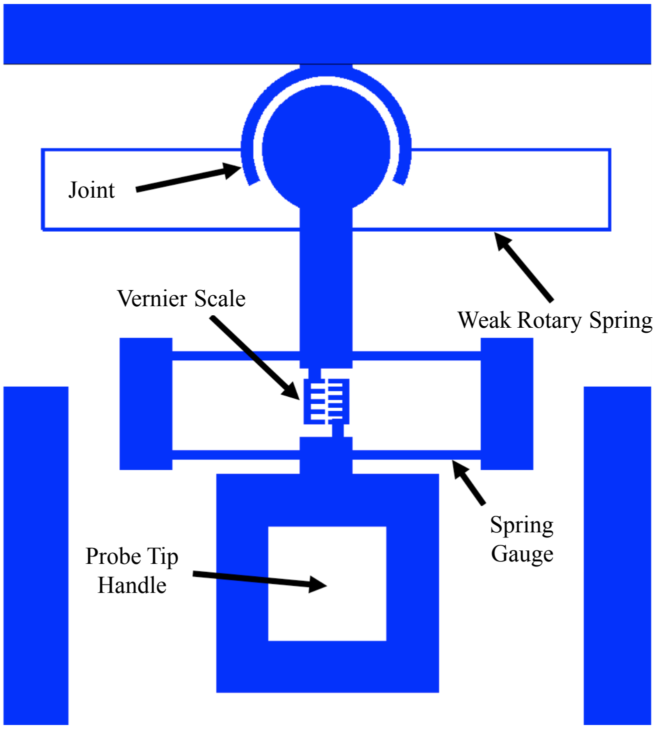


Figure 2.10: Simplified layout of the pull-out force test structure. A spring-based force gauge is attached the the joint along with a handle for a probe tip. The spring gauge has a vernier scale to optically measure the displacement

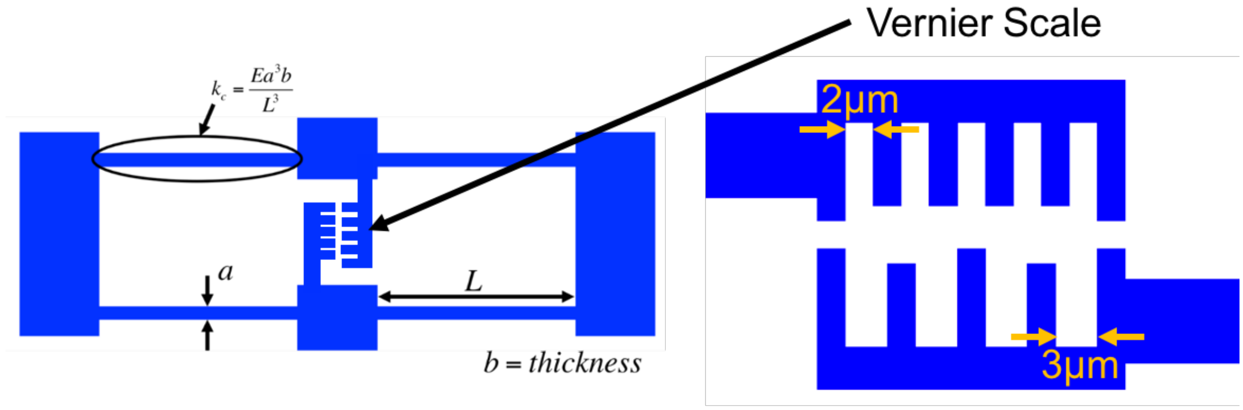


Figure 2.11: Diagram of the force gauge used to measure the pull-out force (left) A close up of the vernier scale used to measure displacement (right)

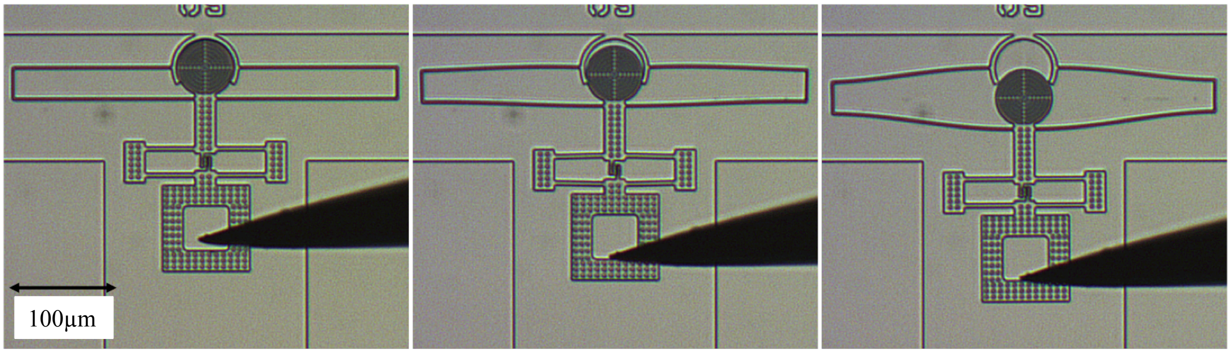


Figure 2.12: A fabricated device under test. The probe tip is shown in the probe handle. The series shows the joint being pulled out of its joint holder

beams that act as springs. The dimensions of the spring are shown in Fig 2.11 as well as the dimensions of the vernier scale used to read displacement. The gauge used has a resolution of 1µm.

As the device is pulled, the springs will deflect and the force will be given by the spring constant and the measured displacement. The spring is a set of series of beams with the fixed-guided boundary condition. This spring constant is given by

$$k = \frac{Ea^3b}{L^3} \tag{2.17}$$

where a is the width of the beam, b is the thickness of the silicon layer, and L is the length of the beams. The beam width will be reduced by approximately 1µm due to the DRIE undercut. Gauges with widths of 6.5µm, 10.5µm and 15.5µm were used on the gauges. For each gauge $L = 100$µm and $b = 40$µm. According to. [17], since the orientation of the gauge is aligned to the cartesian grid of the wafer, $E = 169$GPa.

Measurements were made using recorded videos. For each test structure, video was recorded showing the pull-out sequence. Each captured video was processed using feature tracking software that outputs feature displacement with respect to another feature. software tracked the displacement of the vernier scale to measure the ultimate pull-out force. In every video, this maximum value occurred right at tensile failure of the joint.

2.2.4 Spring Gauge Calibration

The spring constants of the force gauges were calibrated using a Dage 4000 in-plane shear force sensor. This tool is used to measure the shear strength of wirebonds and it was capable of measuring the spring constant of the force gauges with slight layout modifications.

Fig. 2.13 shows the test structure used to measure the force vs. displacement of the gauge. The test structure is made of a push mass that the probe head of the Dage pushes against as indicated. The anchor is attached to a set of trusses through springs, the same

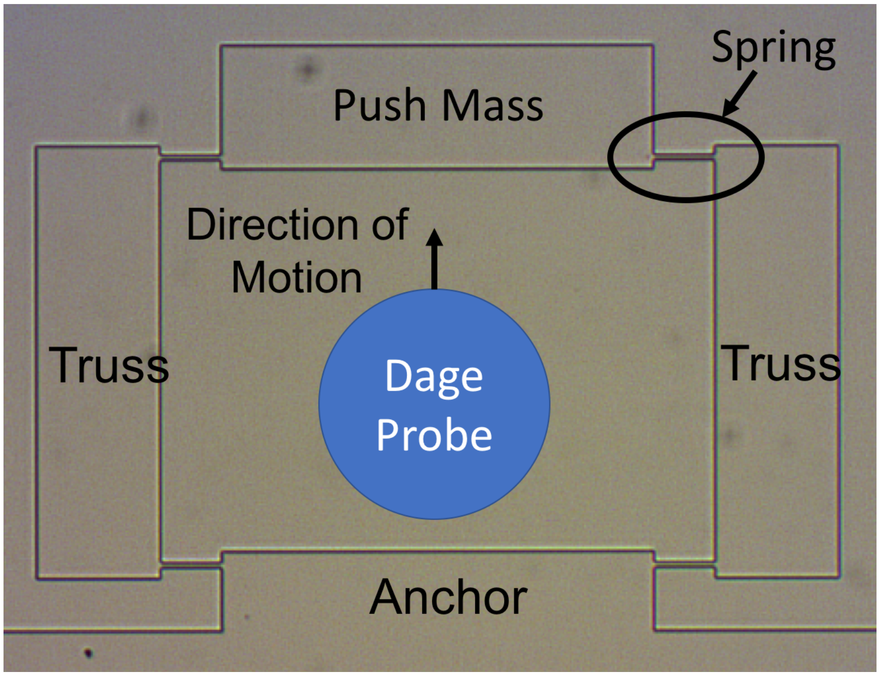


Figure 2.13: Test structure for measuring the spring constant of the gauges used on the pull-out force test structures

springs used on the pull-out test structures. The trusses are large and bulky to prevent them from bending and influencing the experiment. The trusses are tied to an anchor through another set of springs, exactly matching the suspension on the pull-out structures on the push mass. The space between the push mass and trusses is wide to allow the Dage probe head to land with enough tolerance, which can be over 500µm in size. Fig. 2.14 shows the

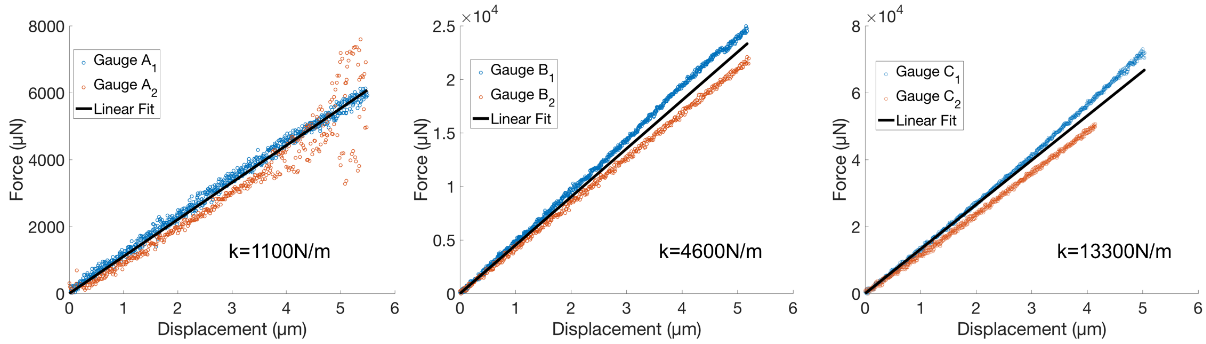


Figure 2.14: Data from the test structures using the Dage force tester. The average slope of the data used as the spring constant for each gauge is shown in each plot

data from the Dage for the three gauges used in the experiment. In addition to the analytical expression, simulations in CoventorWare were also performed by displacing the spring and finding the force. The CoventorWare simulations factor in the reduction in the beam width after fabrication. All three values for each gauge are compared in Table 2.1. It should be noted that the analytical expression factors in the device undercut during DRIE that reduces beam width while the CoventorWare simulation does not and used layout at the full beam width.

Table 2.1: Analytical, simulated, and measured spring constants for the pull-out force gauges with respect to beam width

Drawn Beam Width (μm)	Analytical Spring Constant (N/m)	Simulated Spring Constant (N/m)	Measured Spring Constant (N/m)
6.5	1120	1080	1100
10.5	6000	4960	4600
15.5	20600	14580	13300

There is good agreement between the analytical model, simulation, and the measurement for the 6.5μm beam width gauge. The simulated and measured values for the thicker beams show noticeable deviation from the analytical model. For the thickest beam the measured value is lower than the analytical value by over 35%. Simulation results showed large stress concentrations around the attachment points of the beams to the trusses/anchors. Bending at the anchors could indicate a combination of a rigid and pinned boundary condition rather than perfectly rigid.

2.2.5 Results

The pull-out force is plotted against the analytical normal force on the rotating element in Fig. 2.15 as a function of the width of the joint holder for the various angular openings. The ratio between these values is taken as the friction coefficient between the two surfaces. The angular travel of these joints depends on the width of the link extending out of the joint. For a link of width 38μm, the angular travel is given by the angular opening minus approximately 40°.

Measured pull-out force vs the width of the joint opening is shown in Fig. 2.16. The analytical lines are plotted using Eq. 2.16 and the friction coefficients from Fig 2.15.

An important thing to note in Fig. 2.16 is the weakest joint parameters. The black line for the 110° opening defines the widest angular opening and a limit in our design space. In order to design a joint that will survive tensile loads in the force range for the motors, which is approximately 1mN, we should use joints that are above this line. From this plot, we can stipulate a safe design space for our joints. The joints demonstrated surviving compressive loads outside of the range of the designed gauges, about 100mN.

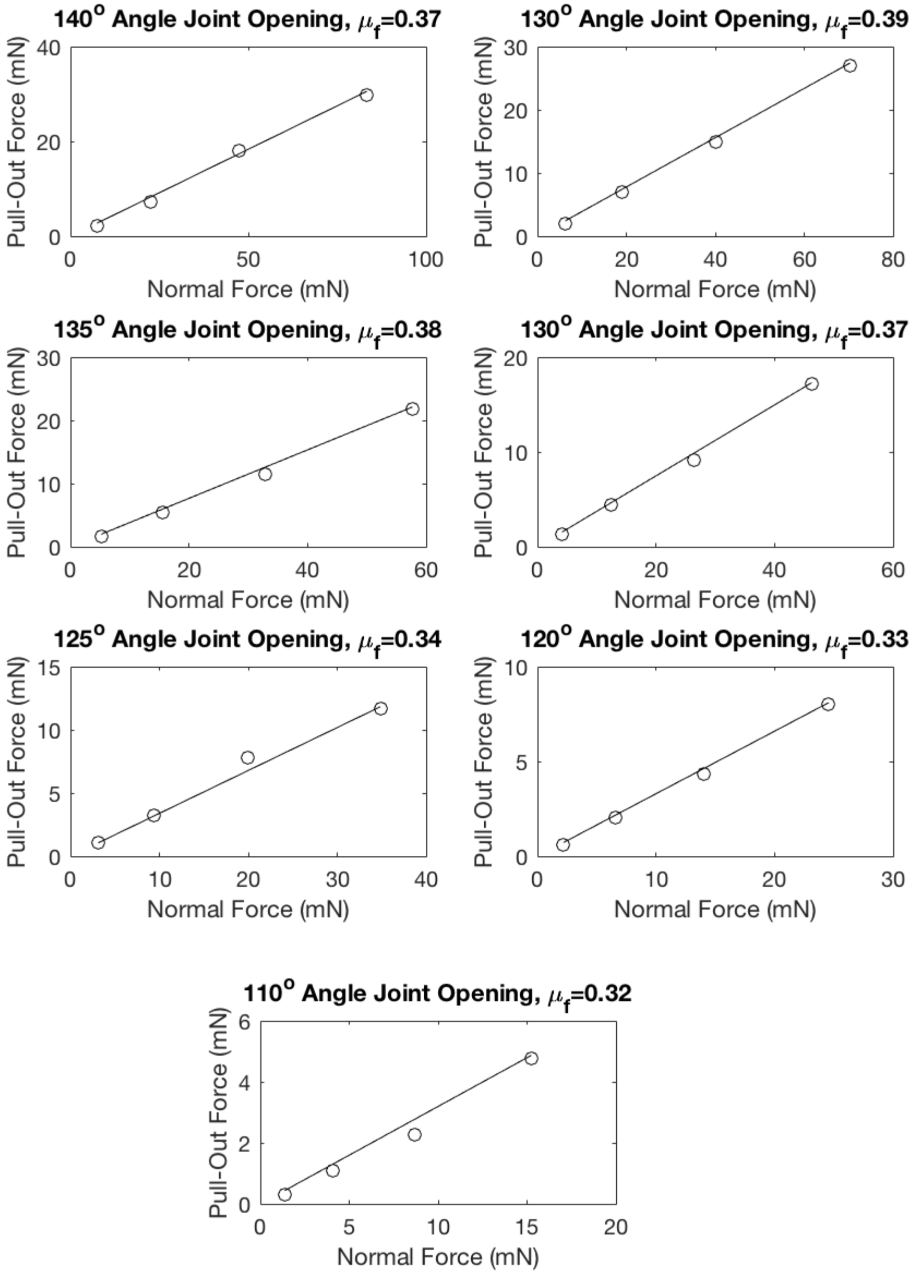


Figure 2.15: Measured pull-out force plotted against the analytically derived normal force. The slope of this line is taken as the friction coefficient

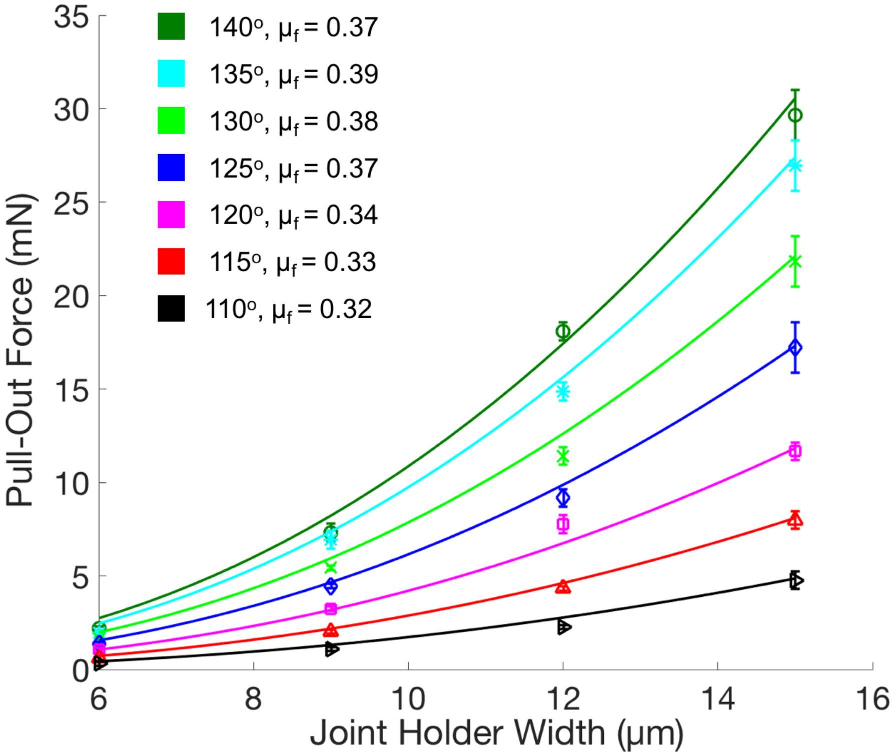


Figure 2.16: Measured pull-out force vs the width of the joint holder. The frictional for each curve is shown in the legend. Analytical predictions from Eq. 2.16 with friction coefficients from Fig. 2.15 are also shown

Chapter 3

Electrostatic Inchworm Actuators

Inchworm motors based on electrostatic gap closing actuator (GCA) arrays were chosen in this robot design for their high force densities and low power draw. This chapter focuses on the adaptation of angled-arm inchworm actuators for use in the microrobot leg. This chapter focuses on characterization of GCA arrays in terms of force and dynamics, application of angled-arm electrostatic inchworm motors on robot legs, and other applications of these motors.

3.1 Gap Closing Actuators

3.1.1 GCA Unit Cell

Fig 3.1 shows a diagram of a gap close unit cell in a GCA array. An array of N capacitive gaps in air, where one set of plates is held stationary and the other set of plates is movable

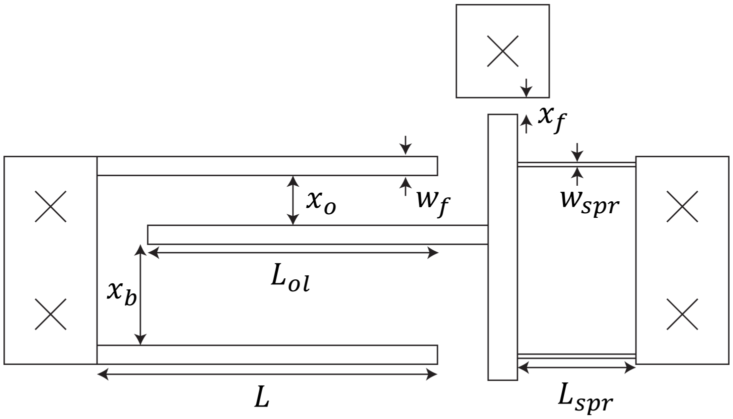


Figure 3.1: A unit cell of a set of capacitively actuated gap closing fingers. N of these fingers are arrayed to increase the force output

has a capacitance defined by

$$C(x) = \epsilon_o N L_{ol} T \left(\frac{1}{x_o - x} + \frac{1}{x_b + x} \right) \tag{3.1}$$

where ϵ_o is the vacuum permittivity of 8.85×10^{-12}F/m, L_{ol} is the overlap length of the gap, T is the thickness of the gap closer into the page, x_o is the initial forward gap size (in the direction of actuation), x_b is the rear gap size, and x is the position of the movable finger. To keep the gap from fully closing and shorting the array, a gap stop of size x_f is used to stop the moving fingers at a gap size of $x_o - x_f$. We are neglecting fringing field effects. Assuming the fringing field is proportional to the size of the plate gap, the fringing field would be approximately 10% of the total capacitance for an open gap and 2% for a closed gap.

The movable fingers are supported by a set of springs, tied to the substrate. The springs have width w_{spr} and length L_{spr}. Treating these beams as a set of two fixed-guided beams in parallel, the spring constant is given by

$$k = \frac{2ETw_{spr}^3}{L_{spr}^3} \tag{3.2}$$

where E is the modulus of elasticity of silicon. GCA arrays are usually designed and fabricated in the cartesian grid of the (100) wafer. Thus for these design cases $E = 169GPa$ is used [17].

3.1.2 Voltage Application, Energy, and Force Output

Typical operation involves applying a voltage to the anchored set of fingers and keeping the movable fingers grounded. This prevents the fingers from snapping down to the substrate, since the substrate is typically held at ground. With an applied voltage, the energy required from the source to charge the capacitor is given by

$$U_{tot} = C(x)V^2 \tag{3.3}$$

This is the energy required per cycle to charge up the GCA. The final energy stored on the capacitor is given by

$$U_s = \frac{1}{2}C(x)V^2 \tag{3.4}$$

where half of the energy from the source dissipates due to resistive loss.

The full system energy is given by the energy from the source and the energy stored in the capacitor. As the capacitor is charged to its final value, the system energy can be defined as the initial source energy minus this stored capacitor energy

$$U_{sys} = U_{source,0} - U_s \tag{3.5}$$

Since the grounded fingers are movable, the electric field between the plates is able to move the fingers. Taking the negative derivative of the system energy in equation 3.5, we can find the force on the finger

$$F_{es} = -\frac{dU_{sys}}{dx} = \frac{dU_s}{dx} = \frac{1}{2}V^2\frac{dC}{dx} = \frac{1}{2}\epsilon_o V^2 N L_{ol} T\left(\frac{1}{(x_o - x)^2} - \frac{1}{(x_b + x)^2}\right) \tag{3.6}$$

3.2 Electrostatic GCA Array Inchworm Motors

Inchworm motor is a generic term for a motor system that uses actuators that take a series of small steps to generate large displacements, over hundreds of times the actuator step size, on a shuttle through some form of mechanical advantage. The shuttle provides mechanical advantage to the smaller actuators, that usually have high force output in the milliNewtons, but small travel of a handful of micrometers. In addition to the electrostatic actuators described in this chapter and focused on in this work, inchworm motors relying on thermoelectric and piezoelectric actuators have also been designed and tested [18, 19, 20].

Electrostatic GCA array-based inchworm motors for microrobotic applications were first introduced in [21]. This first motor design could output 6.5μN of force with a travel of 40μm. This original design, fabricated in a 2-poly process, was improved on in [22]. For these thick SOI-based motors, the force output reached 260μN with 80μm of displacement. The 10mg silicon robot [13] used these actuators for its leg.

These designs used a clutch-drive based system of mechanical advantage. A small GCA array, called the clutch, was used to engage onto the inchworm shuttle. This GCA array required only enough force to keep the shuttle from slipping with a combination of friction and gear teeth. After this engagement another, higher-force, GCA array called the drive pulled on the entire clutch/shuttle assembly providing force output and the displacement on the shuttle.

Following this work, a different inchworm design was introduced in [23]. Rather than using a separate clutch and drive to engage onto and move the shuttle, these motors used an angled arm to provide the mechanical advantage directly from a single GCA array onto the shuttle. Due to removal of the clutch actuator, the electrical signals for these motors have simpler signal routing and the motors are able to operate faster than the clutch-drive inchworm motors. The following sections will focus on the adaptation and design of the angled-arm inchworm motors for the purpose of walking microrobots. Hereafter, references to inchworm motors will refer specifically to angled-arm inchworm motors.

3.2.1 Basic Design

The inchworm motors in this work are based on the design methodology shown in [23].

Fig. 3.2 shows a cartoon diagram of an inchworm motor. An array of four separate GCA arrays sit around the central shuttle. The stator fingers are anchored to the substrate while the linear rotor fingers are held by a spring. An angled arm is attached to the ends of the

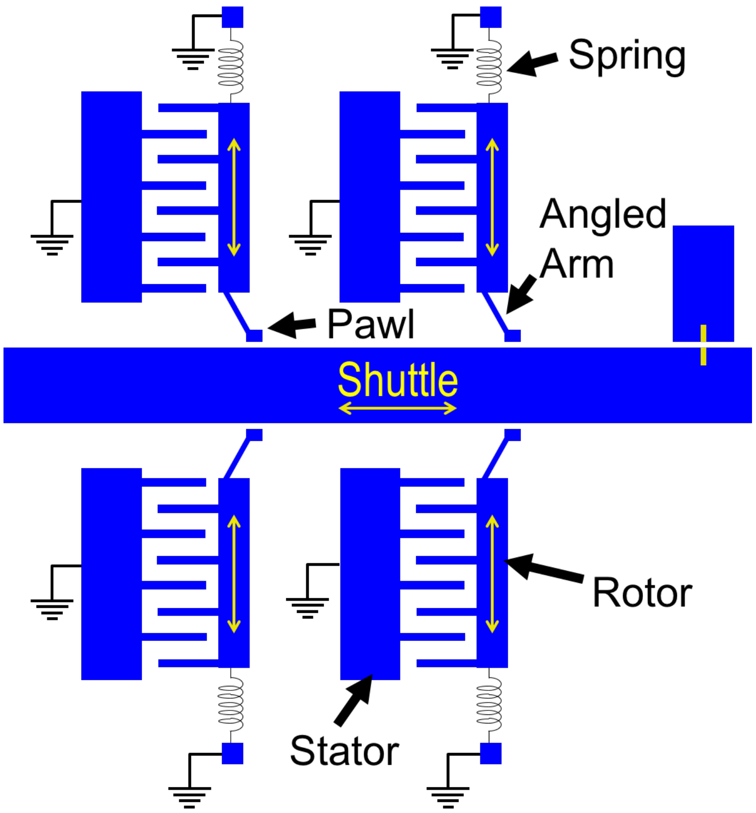


Figure 3.2: Diagram of an electrostatic GCA array-based angled-arm inchworm motor

GCA array rotors. The end of the angled arm has a pawl that is meant to impact and move the shuttle. An anchored gapstop, not shown, keeps the fingers from pulling in completely and shorting the device. The GCA arrays on opposite sides of the shuttle are tied to the same high voltage signal.

Fig. 3.3 shows how the inchworm motor functions. The motor follows the sequence:

1. The motor begins deenergized

2. One set of GCA arrays are energized to voltage V_D. The GCA arrays come in and impact the central shuttle and cause horizontal movement of the shuttle, indicated by the yellow mark

3. The rear set of GCA arrays is energized, the rear pawl come into contact with the shuttle

4. The front set of GCA arrays is deenergized allowing the rear set to fully come in and move the shuttle. The front and rear sets work in tandem, taking incremental displacements to produce hundreds of micrometers of displacement

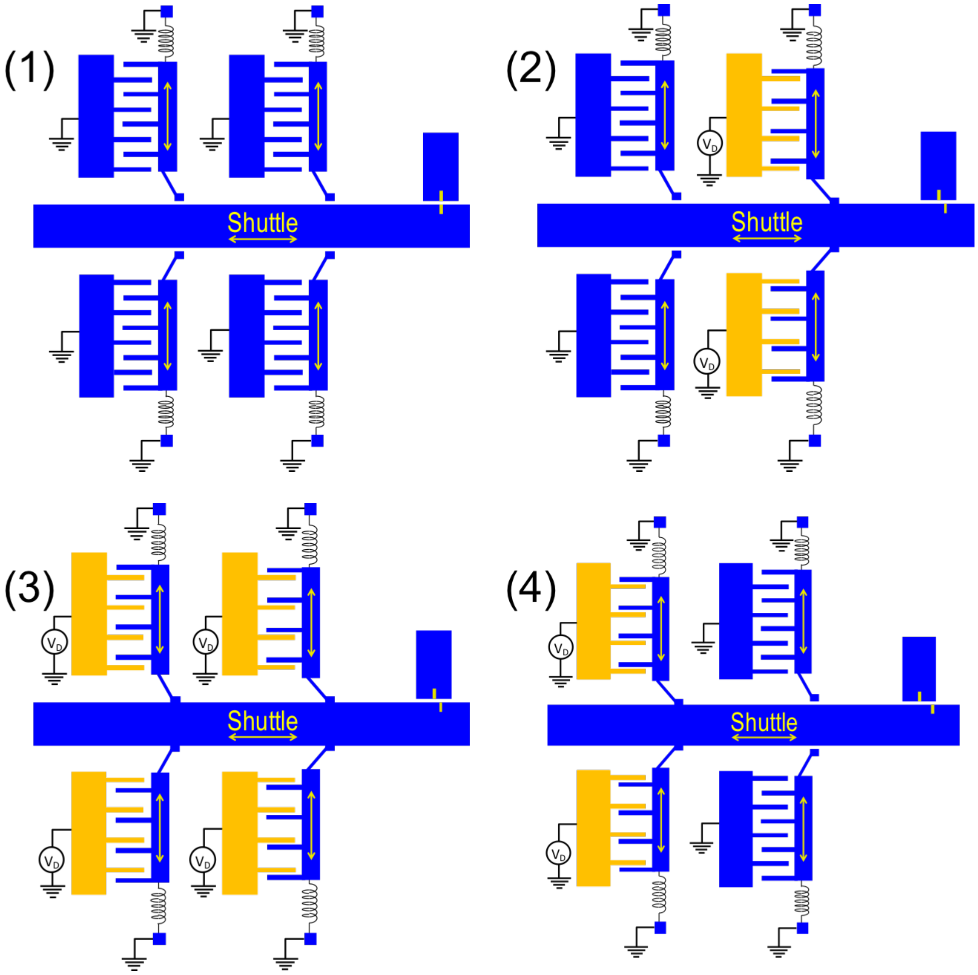


Figure 3.3: Sequence of the angled arm inchworm motor actuating

The motor is unidirectional in the direction of the pawls. When all the GCA arrays let go, the shuttle is retracted passively by the spring support it is attached to.

3.2.2 Pawl-Shuttle Interface Design

The design of the interface between the pawl and the shuttle is critical to the force output of the motor. This interface determines the mechanical advantage between the GCA array and the shuttle.

Fig. 3.4 shows the dimensions of the pawl-shuttle interface. The pawl is at an angle α with respect to the shuttle. The pawl is separated from engagement with the shuttle by a distance Y_I, determined by the gear dimensions of the teeth.

The pawl and the shuttle are lined with teeth of lateral size d_t and protruding from the base a distance h_t. Square teeth were found to cause the motor to jam due to the corners

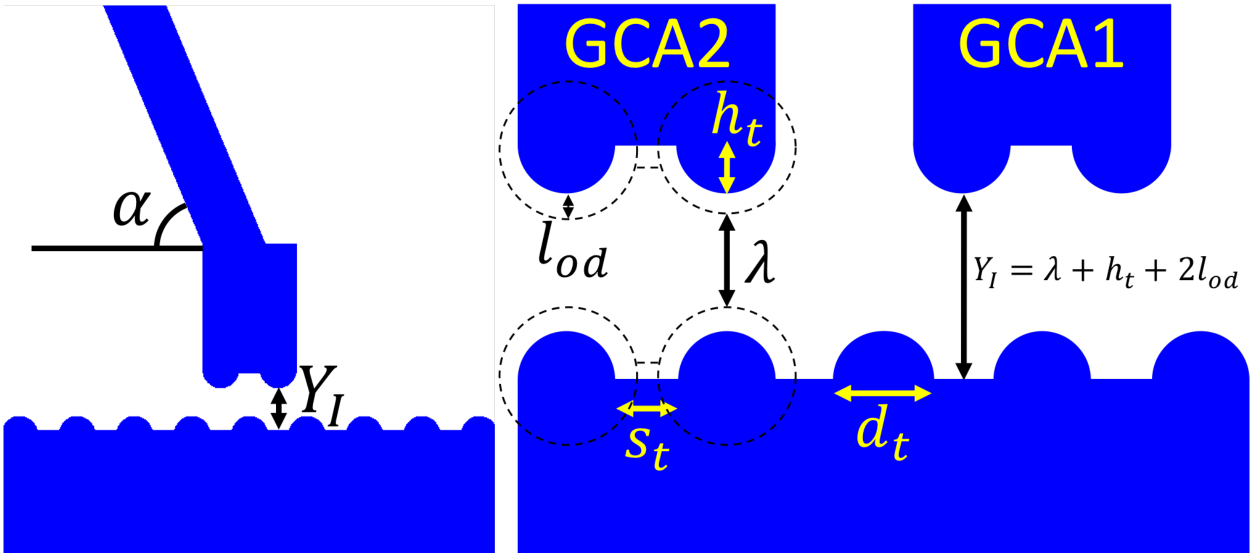


Figure 3.4: Diagram for the dimensions of the pawl-shuttle interface

catching on the tooth edges. The teeth are rounded to prevent them from interlocking during shuttle retraction. The diagram shows a dimension l_{od} known as the overdraw length. This distance is added onto layout and mask generation to account for the undercut during the DRIE.

The two GCA arrays are offset from each other by a set number of teeth. One set of teeth must be lined up to correctly engage with the shuttle after fabrication and the other set of teeth must be offset such that once the motor takes a step, the teeth line up for the next engagement. The step size of the motor will be a distance $\frac{1}{2}(d_t + s_t)$.

The pawl teeth that are lined up after fabrication are separated from the shuttle by the minimum feature size of the lithography system, λ. This is desirable for increasing the areal force density of the motor due to electrostatic force scaling inversely with the square of the gap between the fingers, shown in Eq. 3.5. The engagement length is given by $Y_I = \lambda + h_t + 2l_{od}$. This value is critical for determining the force output of the motor.

3.2.3 Analytical Expressions for Displacement and Force

Fig. 3.5 shows the static state of the motor when the pawl comes into contact with the shuttle and at the end of the GCA arrays travel. A final gapstop that keeps the fingers from pulling in completely is not shown.

The mechanical advantage of the arm can be related from the slope of the arm to the tangent of the angle as

$$\frac{\Delta x}{\Delta y} = \tan \alpha \tag{3.7}$$

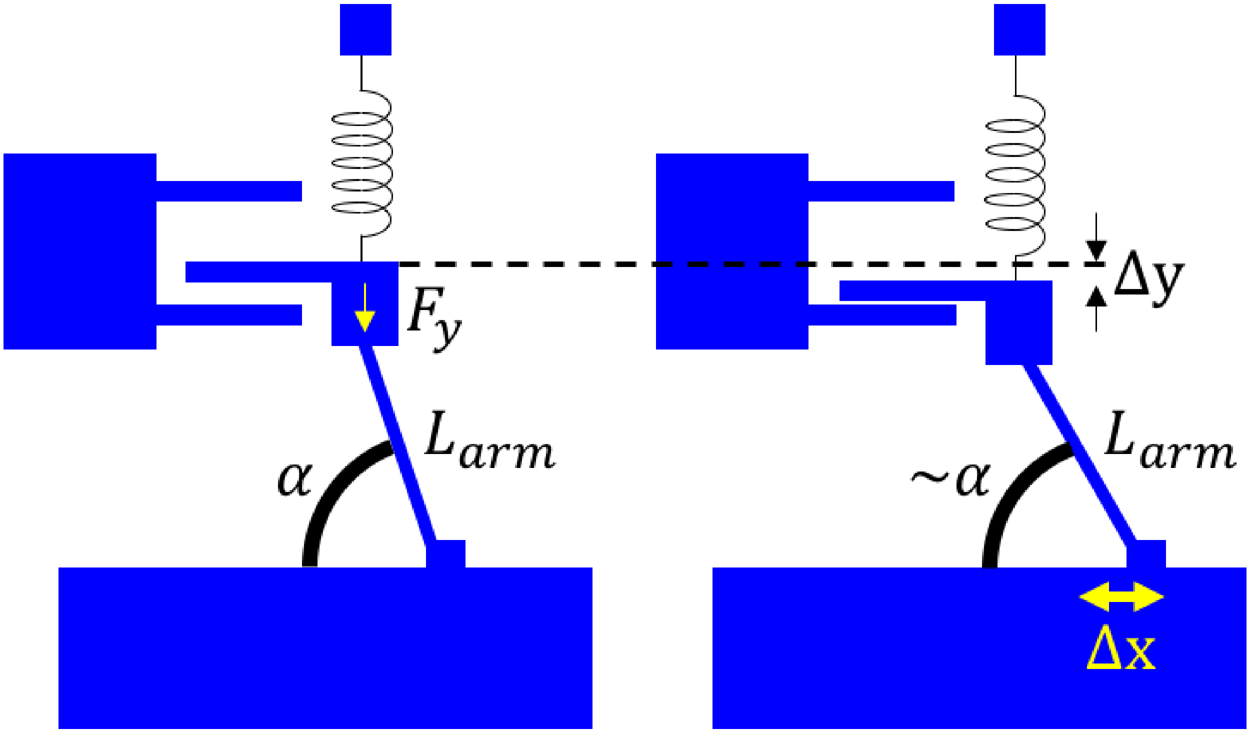


Figure 3.5: Statics diagram of the pawl and shuttle at the point of engagement and at the end of travel

Due to the dependence of the GCA array on the gap size, the maximum force on the shuttle load will be at the point of engagement. As the gap decreases the force output increases quadratically, which is not useful when pushing on a static load over this limit. However this force increase goes into the acceleration of the shuttle which has a strong effect on motor speed, discussed later.

Looking at the statics of the system, the force out of the GCA array at the point of engagement will be the electrostatic force minus the spring force of the GCA at the engagement distance Y_I

$$F_y = \frac{1}{2}\epsilon_o V^2 N L_{ol} T \left(\frac{1}{(x_o - Y_I)^2} - \frac{1}{(x_b + Y_I)^2} \right) - kY_I \tag{3.8}$$

Balancing the moments from F_y and F_x around the arm we get that

$$F_x L_{arm} \sin\alpha = F_y L_{arm} \cos\alpha \tag{3.9}$$

$$F_x = \frac{F_y}{\tan\alpha} \tag{3.10}$$

As the arm continues pushing into the shuttle, the arm will flex as the shuttle moves. This deflection is shown in Fig. 3.6. We assume that the final angle of the beam is still

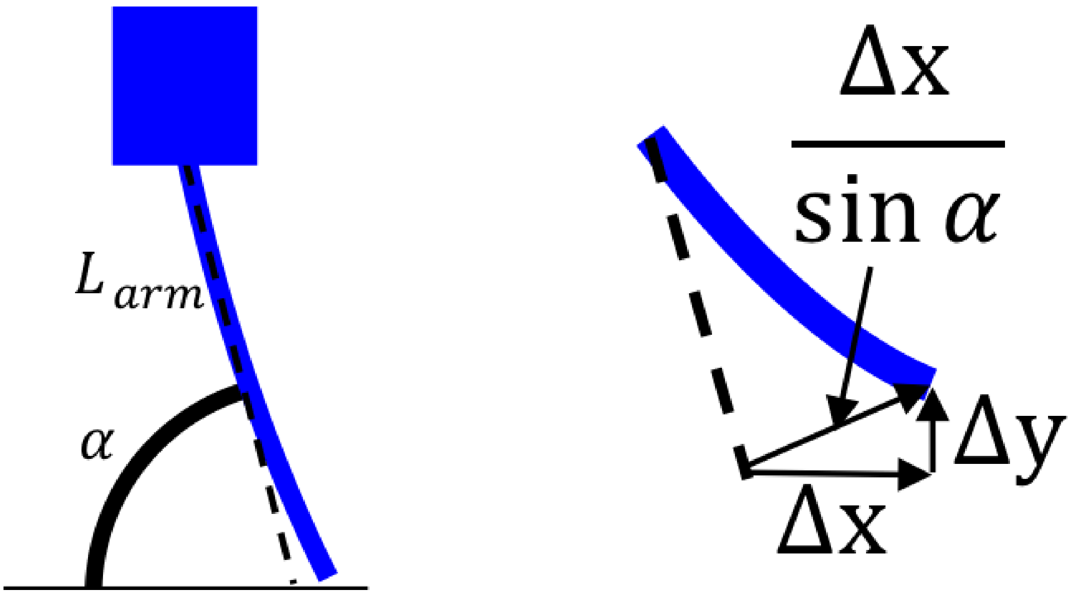


Figure 3.6: Geometry of the angled arm deflection. The final angle of the arm is approximately α (left) Exaggerated deflection of the end of the angled arm (right)

approximately α. Looking at the arm as a cantilever being deflected on its free end, the end that contacts the shuttle, the force lost to this spring is given by

$$F_{arm} = k_{arm}\frac{\Delta x}{\sin \alpha} = \frac{3EI_{arm}}{L_{arm}^3}\frac{\Delta x}{\sin \alpha} \tag{3.11}$$

where I_{arm} is the area moment of inertia of the angled arm. Balancing moments while accounting for the arm, we can find an analytical expression for mechanical advantage of the arm throughout its deflection

$$F_x L_{arm} \sin \alpha = F_y L_{arm} \cos \alpha - F_{arm} L_{arm} \tag{3.12}$$

This gives us an expression for the balancing load force as a function of the shuttle displacement

$$F_x(\Delta x) = \frac{F_y}{\tan \alpha} - \frac{3EI_{arm}}{L_{arm}^3}\frac{\Delta x}{\sin^2 \alpha} \tag{3.13}$$

This derivation assumed that the pawl end of the angled-arm has rotational freedom. Another model assumes that the pawl end of the arm is fixed to the shuttle rotationally as well as positionally, leading to a fixed-guided boundary condition on the beam. Looking at a simplified half model for the beam as done with folded flexures, the stiffness loss term could be as much as 4 times larger.

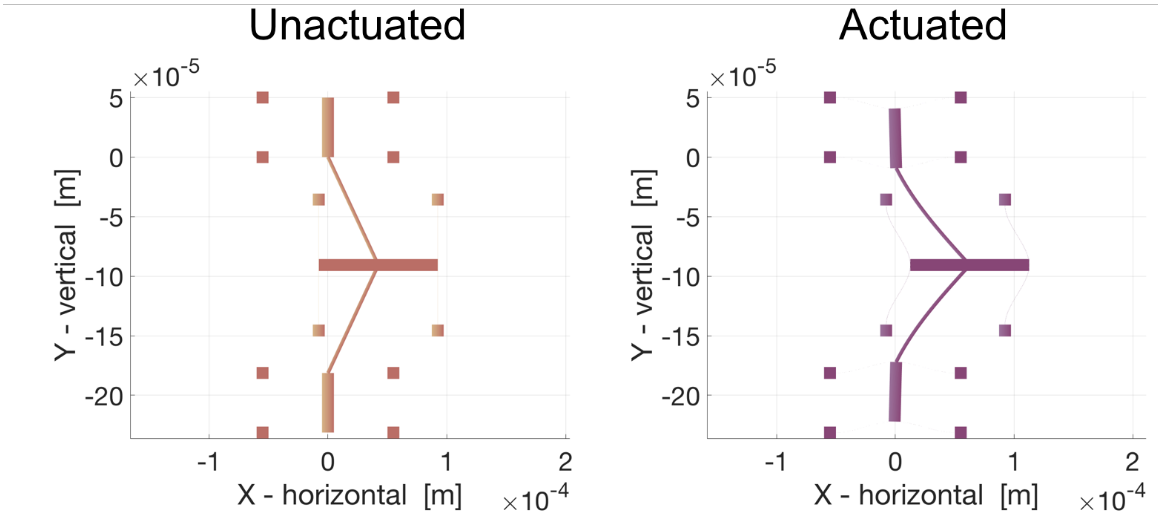


Figure 3.7: SUGAR simulation of the angled arms

3.2.4 Motor Optimization Scheme

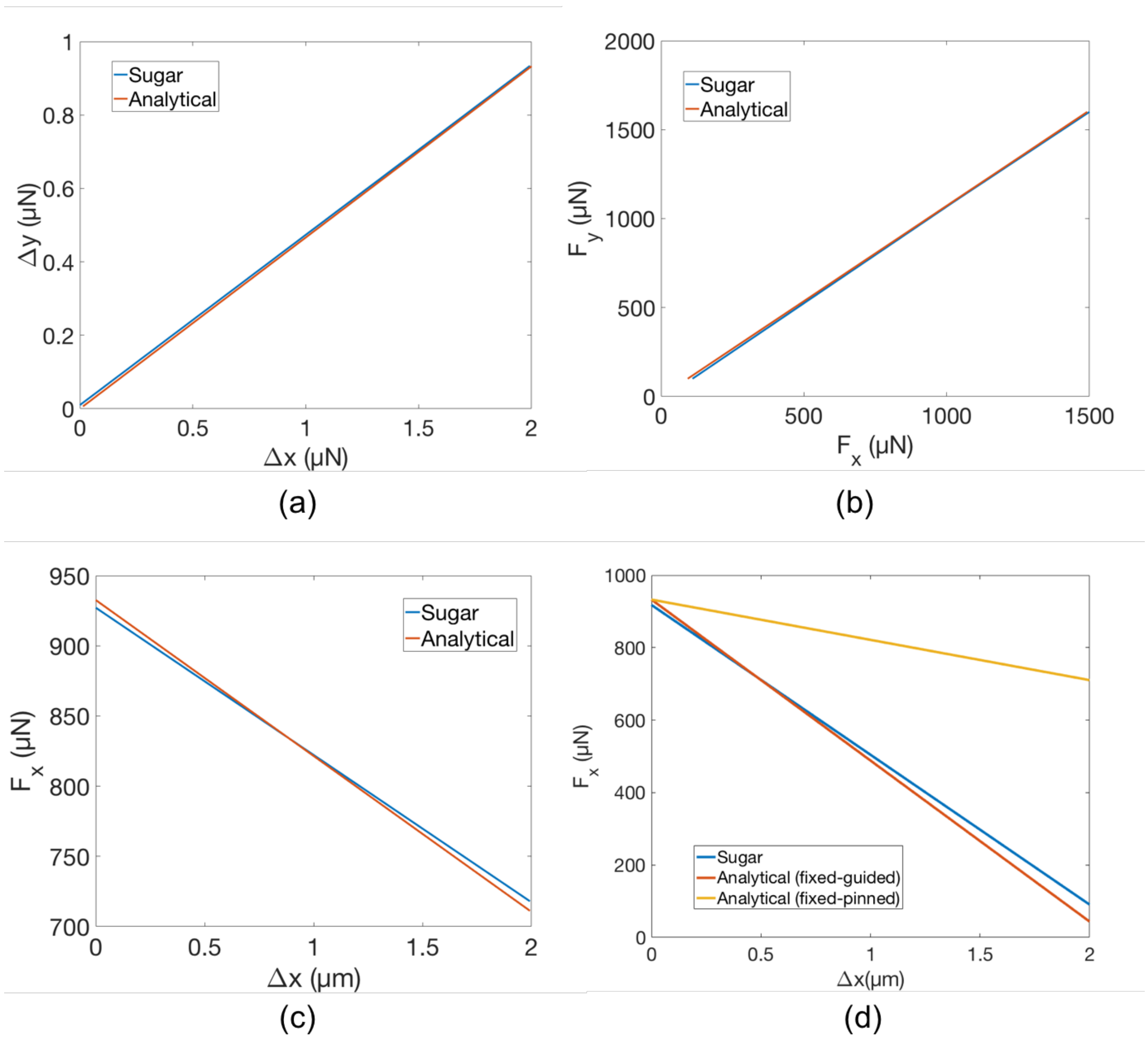
The method for motor generation used in [23] uses the equations derived in the previous section. The force output in Eq. 3.10 is taken and divided by the area for a variety of geometries, including the additional area needed for the angled arm. This value is maximized for the parameters given, including but not limited to minimum feature size, device layer thickness, and maximum finger length before catastrophic pull-in.

3.2.5 SUGAR Displacement and Mechanical Advantage

SUGAR is a nodal simulation tool designed for MEMS. A simulation environment for the angled-arms was created to verify the analytical expressions used in the designs. The simulation environment does not factor in contact due to the lack of an adequate contact model.

Fig. 3.7 shows the simulation environment. Angled arms are attached to weak linear suspensions. The suspensions are designed to be weak enough to not impact simulation results. The ends of the angled arms attached to the shuttle are defined as rotary joints, free to rotate but locked positionally to the shuttle flexure. The simulation is only for the exact moment after and following impact. This simplification is done due to the lack of an adequate fast performing contact model for the MEMS devices. The files used in this simulation are given in Appendix. A.

Fig. 3.8 shows the comparison between the sugar simulation and the analytical expressions. Fig. 3.8(a) compares the displacement of the GCA array to the shuttle displacement given in Eq. 3.7. Fig. 3.8(b) plots the engagement force found in Eq. 3.10. Fig. 3.8(c) plots



(a)

(b)

(c)

(d)

Figure 3.8: Plot comparing analytical and simulation results for the statics of the angled-arm system

the force as a function of the shuttle displacement found in Eq. 3.13. Fig. 3.8(d) shows the comparison between SUGAR and the analytical expression assuming fixed-guided boundary conditions. The analytical expressions includes a 4 times increase in the stiffness loss term from Eq. 3.13. The analytical expression assuming fixed-pinned boundary conditions is also shown. In all four cases we see good agreement between the simulation and the analytical model, indicating that our linear approximations in the analytical expressions are correct according to SUGAR.

3.2.6 Angled Arm FEM Simulation

In order to form a better model for the angled arm boundary conditions during engagement, a simulation was performed in CoventorWare's FEM simulation environment. Fig. 3.9 shows stills from the simulation environment for the angled arms. The arms are displaced in the +y direction into the shuttle. The shuttle is anchored such that it can only move in the x-axis. The shuttle section is connected to an anchor through a beam. This is done to

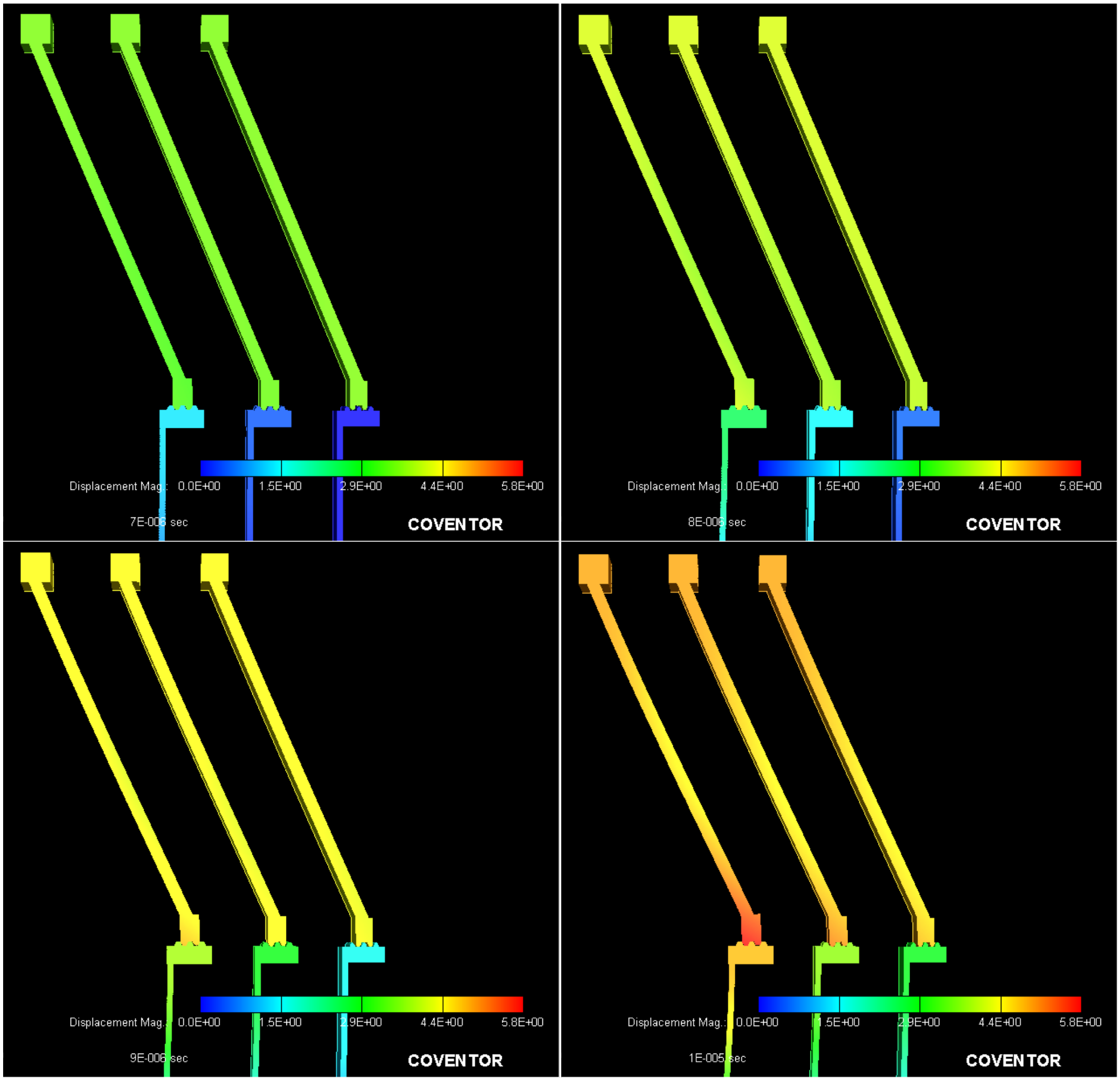

Displacement Mag.: 0.0E+00 1.5E+00 2.9E+00 4.4E+00 5.8E+00
7E-006 sec
COVENTOR
Displacement Mag.: 0.0E+00 1.5E+00 2.9E+00 4.4E+00 5.8E+00
8E-006 sec
COVENTOR
Displacement Mag.: 0.0E+00 1.5E+00 2.9E+00 4.4E+00 5.8E+00
9E-006 sec
COVENTOR
Displacement Mag.: 0.0E+00 1.5E+00 2.9E+00 4.4E+00 5.8E+00
1E-005 sec
COVENTOR

Figure 3.9: Stills from the CoventorWare simulation of the pawl-shuttle interaction

measure the reaction force on the shuttle section on the anchored face. The three different arms have various degrees of undercut at the pawl/shuttle teeth, from no undercut to 0.4µm of undercut.

The toothed surfaces of the pawl and the shuttle have contact boundary conditions applied. As the pawl moves into the shuttle, the contact boundary condition keeps the surfaces from moving past each other and the mechanics of the contact can be analyzed. A friction coefficient of 0.3 is applied between the surfaces. Rather than being rounded surfaces, the teeth are simplified to hexagons in order to speed up simulation time. This is due to having fewer surfaces for the contact analysis to deal with. The simulation can take between 5 and 10 hours for the number of elements shown depending on the fineness of the meshing.

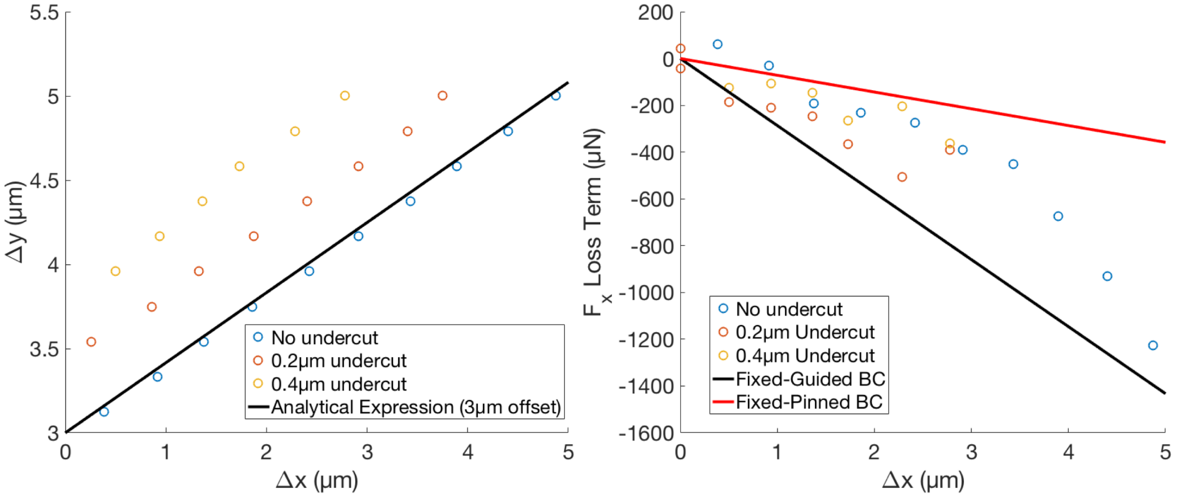


Figure 3.10: Results from the CoventorWare simulation looking at the displacement of the shuttle with respect to the angled arm (left) Results for the CoventorWare simulation of force loss for the angled arm plotted alongside the fixed-pinned and fixed-guided analytical model expressions (right)

Fig. 3.10 shows the results from CoventorWare alongside Eq. 3.7 and Eq. 3.13 and the fixed-guided beam boundary condition.

The displacement follows the expected trend from Eq. 3.7 and SUGAR. The different y-offsets are because of the different undercut on each arm increasing the initial distance between the teeth before contact.

The force loss plots the second term in Eq. 3.13 alongside the data from the force on the anchored face in the simulation after contact. For small deflections and smaller undercut the simulation tends to agree more with the pinned condition. The simulation diverges from this line as the displacement increases, being bounded between the guided and the pinned conditions, implying some sort of combined boundary condition model is more correct. With our degree of undercut, the fixed-guided boundary condition would be more appropriate according to simulation. It should be noted some points in the loss term are positive. This is because upon contact simulation indicated that there is a slight deflection in the negative x-axis on the shuttle section leading to a positive force vector on the anchor.

Eq. 3.10 was not simulated. Precisely positioning the arm at the exact engagement point is difficult in simulation and the simulation would have to be run multiple times while changing the spring constant of the beam anchoring the shuttle section. Results from this experiment indicate a $\tan\alpha$ relationship between F_x and F_y near the engagement point.

3.2.7 Force Output Measurement

Spring Calibration

To verify the force output experimentally, inchworm motors with stiff springs attached to the shuttle were fabricated. The inchworm motors were driven until the springs stopped the shuttle. This displacement is read on a vernier scale and taken as the maximum motor output force. The max force is taken for varying applied voltages.

Springs were calibrated using an array of folded flexure comb drive resonators to determine the fabricated beam width. Beams in a DRIE process are subject to undercut during the DRIE step. The actual beam width will be the drawn beam width a and some δa. The square of the resonant frequency can be related to the spring constant of the suspension by

$$\omega^2 = \frac{k}{m} = \frac{k}{\rho T A} \tag{3.14}$$

where ρ is the density of silicon, b is the device thickness, and A_S is the area of the shuttle mass. For a folded flexure with beam thickness b, fabricated beam width $a - \delta a$, length L, the spring constant is given by

$$k = \frac{2ET(a-\delta a)^3}{L^3} \tag{3.15}$$

Plugging this into Eq. 3.14 we can solve for a

$$a = \beta\omega^{2/3} + \delta a \tag{3.16}$$

where $\beta = \sqrt[3]{\frac{L^3\rho A}{2E}}$. For an array of comb drives with different beam widths, we can plot the measured resonant frequency, ω_n, to the two-thirds power against the drawn beam width, a. The y-intercept of this line will give us the negative of the beam undercut, δa. We can use this δa to modify our analytical spring constant.

The comb drive parameters for these experiments are given in Table. 3.1. Fig. 3.11 shows the measured data from three different arrays of comb drives from separate wafer runs with the drawn beam width shown. The average device undercut was 0.93µm with a variation of ±0.12µm.

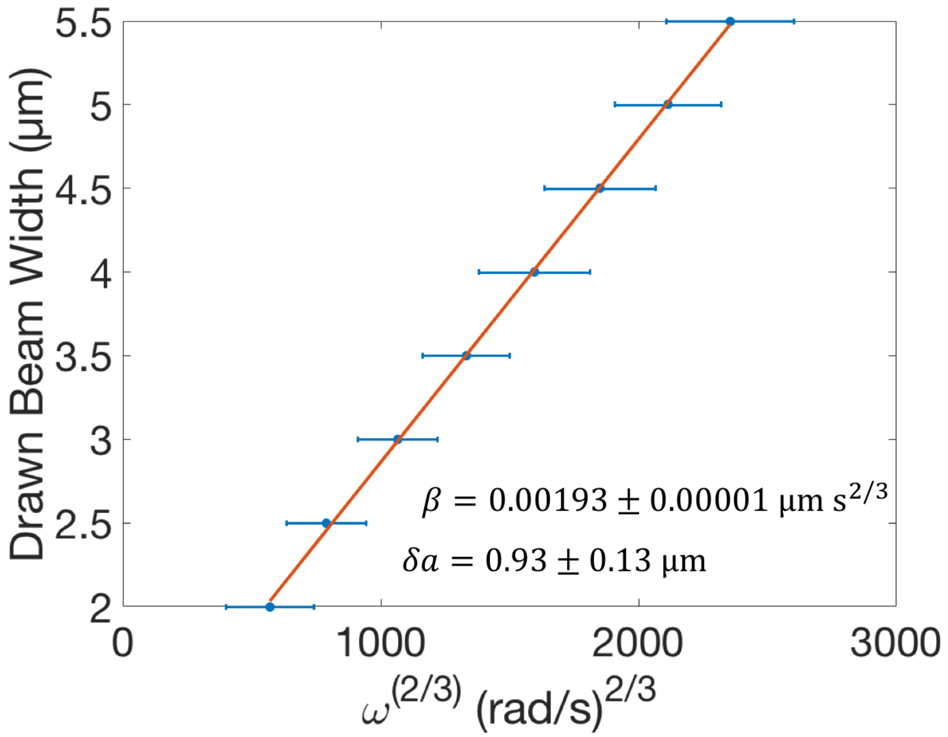


Figure 3.11: Data for an array of comb drives used to determine the process undercut during the DRIE step

Table 3.1: Dimension of the comb drives used to measure device undercut. T is device thickness, L is spring length, N_g is the number of gaps, w_f is the finger width, w_g is the gap size between fingers, L_f is the finger length, A_S is the shuttle area, A_T is the truss area. Dimensions are all as drawn, and do not factor device undercut

T(μm)	L(μm)	N_g	w_f(μm)	w_g(μm)	L_f(μm)	A_S(μm)^2	A_T(μm)^2
40	300	50	2	2	50	36000	2000

The slope of the line, β, is related to the ratio of the Young's Modulus to the density. For a Young's Modulus of 169GPa and a density of 2.3g/cm^3, we get a a literature value of 7.35×10^7m^2/s^2. The average value of the slope from Fig. 3.11 gives a value of 6.74×10^7m^2/s^2 with a variation of $\pm 0.76 \times 10^7$m^2/s^2. Assuming a density of $\rho = 2300$kg/m^3, this would give a Young's Modulus of $E = 155 \pm 18$GPa. The value of 169GPa falls within the uncertainty of the measured values, but the mean value is low and the variation is large according to the results from [17].

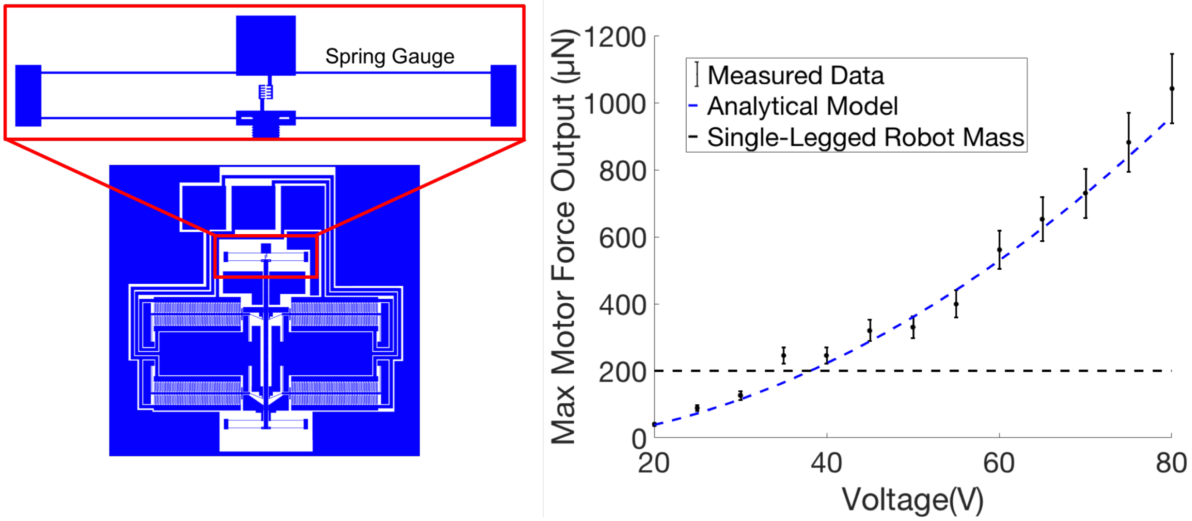


Figure 3.12: Spring gauge used to measure the motor force output (left) The raw motor force output of an inchworm motor (right)

Measurements

The force output of the inchworm motor is shown in Fig. 3.12. The motor exerts over 1mN of force at 80V. This is about 5x the amount of force necessary to lift the single-legged robots measured mass of approximately 20mg. Uncertainty values are taken from the average of three devices as well as propagation from the variation in spring constants from device undercut variation.

The analytical model was corrected to better match the data. The typical undercut of about 0.45µm measured from Fig. 3.11 lead to overestimates of the force output. The total gap size of $Y_I = \lambda + h_t + 2l_{od}$ is smaller in the fabricated devices than predicted by the model. To better match the data, a gap fudge factor between the pawl and shuttle teeth was introduced into the model such that $Y_I = \lambda + h_t + 2l_{od} - g_{fudge}$. This accounts for the variations in the patterning of the interlocking teeth and the fact that narrower spaces (<4µm gap sizes) tend to have smaller undercuts during the etch. The correction factor, g_{fudge} was approximately 0.2µm less undercut between the pawl and the shuttle.

Another interesting note, because the motor takes fixed approximately 2µm steps, points for different voltages often have the same force output according to the gauge. This can be seen at 35V and 40V as well as 45V and 50V. A weaker gauge would have better granularity in measurement values.

Chapter 4

Motor Dynamics and Thickness Optimization

The force output of GCA array-based inchworm motors has been well characterized in previous literature with excellent experimental verification [22, 23]. The dynamics of similar GCA actuators has been modeled and analyzed in the realm of MEMS relays [24, 25]. The speed of the GCAs relates directly to the motor shuttle speed. By achieving faster shuttle speeds, these motors could drive microrobot legs to rival the speeds of insects.

In order to fully formulate a dynamic model for these actuators, viscous damping has to be taken into consideration. Damping models have been investigated for various configurations of GCA. Bao et al. shows work on modeling squeeze-film damping in various geometries of MEMS structures [26]. Li et al. further improved on this model to account for laterally resonating structures with a small substrate gap [27]. Squeeze-film damping has also been investigated for vertically actuated RF MEMS switches [24].

With a model for the gap closer and motor dynamics a more robust model for inchworm optimization can be developed.

4.1 GCA Dynamics

Work on the dynamics of GCA arrays used in inchworm motor configurations has been presented previously and forms the basis for the work presented here [28]. This section will focus on further formalizing this model and will show a wider breadth of data to verify this model according to device geometry.

4.1.1 Basic Dynamics of an Electrically Actuated Gap Closer Array

Fig. 4.1 shows a unit cell of the GCA array during pull-in and release, highlighting the components of the dynamics equation. The forcing function for an electrostatically actuated

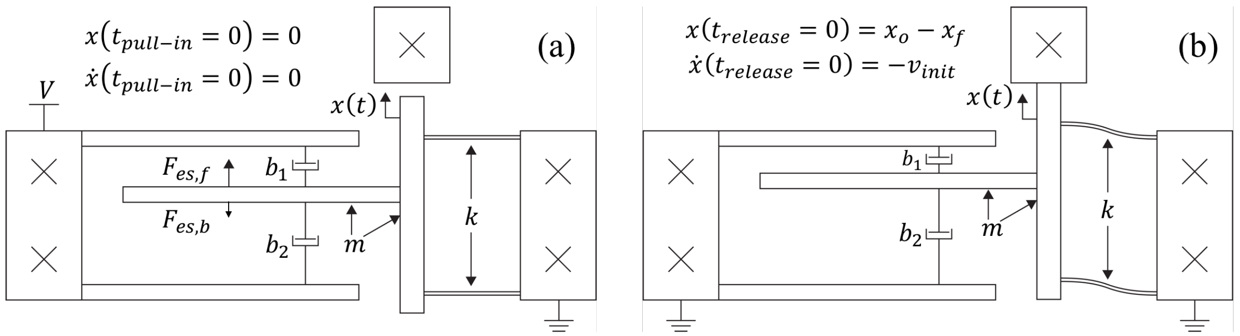

$x(t_{pull-in} = 0) = 0$
$\dot{x}(t_{pull-in} = 0) = 0$
(a)
V
$x(t)$
$F_{es,f}$
b_1
$F_{es,b}$
b_2
m
k
$x(t_{release} = 0) = x_o - x_f$
$\dot{x}(t_{release} = 0) = -v_{init}$
(b)
$x(t)$
b_1
b_2
m
k

Figure 4.1: A unit cell of the GCA array, highlighting the components of the dynamics equation for the pull-in time (left) and the release time (right)

GCA is given by

$$m\ddot{x} + b\dot{x} + kx = F_{es} \tag{4.1}$$

where x is the displacement of the movable finger array from its nominal position, m is the mass of the movable finger array, b_1 and b_2 are the damping coefficients, k is the spring constant of the support spring, and $F_{es,f}$ and $F_{es,b}$ are the electrostatic actuation forces to each corresponding finger.

Once the fingers are pulled in they make contact with the gap stop. Release of the fingers is passive and relies on the spring to reset the position of the fingers. This forcing function is given by

$$m\ddot{x} + b\dot{x} + kx = 0 \tag{4.2}$$

where the initial conditions are nonzero.

4.1.2 Gap Closer Dynamics in Air

Fig. 4.1(a) shows the initial state of the system during the active pull-in phase. The damping terms b_1 and b_2 make up the squeeze film damping term, b_{sf}. The Couette flow (shear film) damping term, b_{cf}, beneath the fingers, not shown, is also accounted for in our dynamics model. Fig. 4.1(b) shows the initial state of the system during the release phase.

The mass in Eqs. 4.1 and 4.2 will be the mass of the full GCA array including the central shuttle. This mass can be found from

$$m_{GCA} = \rho_{Si} T A_{eff} \tag{4.3}$$

where ρ_{Si} is the density of silicon, T is the device silicon thickness, and A_{eff} is the effective layout area of the movable finger array. The effective layout area is taken from the layout area of the device but accounting for the lateral etch of the silicon during the DRIE step. This undercut results in a significant reduction of the mass.

For damping, we take into account two forms of damping; squeeze-film between the fingers themselves and shear-film between the movable fingers and the substrate.

Between approximately rectangular plates, the damping coefficient is given by [26]

$$b_{sf} = \mu_{air} N S_1 S_2^3 \beta(\eta) \left(\frac{1}{(x_o - x)^3} + \frac{1}{(x_b + x)^3} \right) \tag{4.4}$$

where μ_{air} is the viscosity of air (1.85×10^{-5}Ns/m), N is the number of fingers in the array, $S_1 = \max(L_{ol}, T)$ and $S_2 = \min(L_{ol}, T)$, $\eta = \frac{S_2}{S_1}$ is the aspect ratio of the finger, and $\beta(\eta)$ is a correction factor that depends on η.

For shear-film damping, we use the expression for Couette flow which assumes a linear velocity profile and no slip boundary condition between the gas boundary at the surface of the movable fingers and the gas boundary at the surface of the substrate. This expression is given by

$$b_{cf} = \frac{\mu_{air} A_{eff}}{d} \tag{4.5}$$

where d is the separation between the movable finger array and the substrate.

The movable fingers are supported by two parallel fixed-guided beams. The spring constant for this suspension was given in Eq. 3.2. The electrostatic force is found from the changing capacitance between the plates and is given by Eq 3.6

We can find the time for the gap to close by solving Eq. 4.1 with the initial conditions of $x(t_{pull-in} = 0) = 0$ and $\dot{x}(t_{pull-in} = 0) = 0$. We also have to factor that the gap does not close completely, since there is a gapstop in our structures that keeps the gap from moving below x_f.

In the case of Eq. 4.2, the spring will restore the structure to its nominal position and will be acting counter to the damping and the inertial terms. We can solve this differential equation with the same coefficients given above and the initial conditions of $x(t_{release} = 0) = x_o - x_f$ and $\dot{x}(t_{release} = 0) = -v_{init}$. The initial velocity will be described in detail in a following section.

4.1.3 Gap Closer Dynamics including Finger and Shuttle Deflection

In preliminary experiments it was observed that during release of the GCA arrays, there was a noticeable dependence on the voltage used to close the gap, even though the voltage is set to zero during release. Here we formulate a model on a dynamics model that includes an initial velocity imparted from the electrostatic deflection of the individual fingers in the array.

The diagram for this finger deflection model is shown in Fig 4.2. For a simple cantilever we can solve the governing equation of elasticity for the beam's deformation and come to the radial resonant frequency of [29]

$$\omega_n = K_n \sqrt{\frac{EI}{\rho_{Si} A L^4}} \tag{4.6}$$

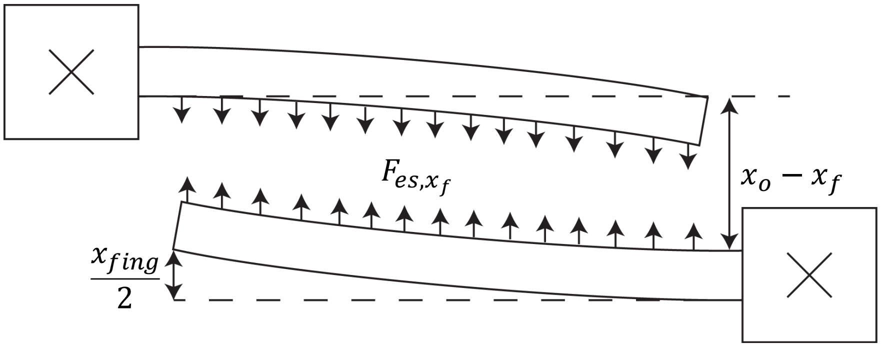


Figure 4.2: Model of the finger deflection for a pair of fingers in the array. It is assumed that each finger deflects by half of the total deflection of a single finger under the same load

where K_n is a constant dependent on the nth mode shape, I is the moment of inertia of the cross-sectional area of the beam, A is the cross-sectional area of the beam, and L is the full beam length. For the case of the first modes shape $K_1 = 3.52$.

When the finger is fully pulled in, it is under a distributed load from the electrostatic force. Assuming a uniform distributed load, the spring constant of the beam is given by

$$k_{fing} = \frac{8EI}{L^3} \tag{4.7}$$

This is a major simplification. The actual load would not be uniform since the electrostatic force has an inverse square relationship with the gap size. Any variation in the gap of as little as 100nm, 10% of the closed gap, can have a change of 100 times on the force at a given position. To develop a more accurate model a finite element method tool can be used to analyze the electrostatic force and displacement relationship of the fingers. The simplification presented here is done as a first hand approximation. An FEM model to compare with our analytical expression is shown in the following section.

The approximate deflection of the free end of the finger at pull-in is then given by the electrostatic force at the final position divided by Eq. 4.7. This force is given by

$$F_{es,x_f} = \frac{1}{2}\frac{\epsilon_o N L_{ol} T V^2}{(x_o - x_f)^2} \tag{4.8}$$

Again, this is assumed to be a uniformly distributed load, which is not the case in actuality but is useful in an initial highly simplified model. It should also be noted that both fingers are free to deflect but this model assumes one set of fingers is perfectly rigid. The displacement will then be

$$x_{fing} = \frac{F_{es,x_f}}{k_{fing}} \tag{4.9}$$

For a the nominal finger dimensions, with finger width 5µm, device thickness 40µm, and length 76.5µm, Eq. 4.6 gives a resonant frequency of 1.2MHz. For comparison, the nominal

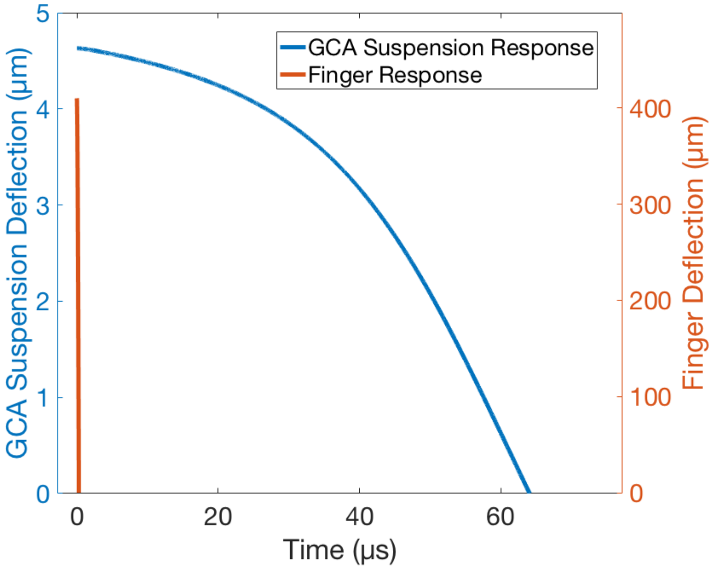


Figure 4.3: Comparing the response of the finger model and the base GCA model for a prescribed initial deflection

GCA with a suspension of spring constant 10.3N/m and mass of 3.7µg has a resonant frequency of 8.4kHz. Thus the response time of the fingers can be over two orders of magnitude shorter than for the suspension.

Fig. 4.3 shows the response of our simplified finger model and the GCA model to an initial displacement. The finger passes through its zero point at approximately 0.3µs. Here we make the assumption that once the beam is released from the electrostatic force, it quickly returns to its nominal position with when compared to the GCA array suspension. The maximum velocity of the effective mass of the cantilever is given by

$$v_{fing} = \omega_1 x \tag{4.10}$$

In the case of the shuttle, the force from the GCA array is placing axial strain on the shuttle. Fig 4.4 shows the diagram for this model. This causes a deflection that also stores energy, imparted onto the array once the fingers are released.

For the shuttle, the axial spring constant is given by

$$k_{shut} = \frac{EA_{shut}}{L_{shut}} \tag{4.11}$$

where A_{shut} is the cross-sectional area of the shuttle, and L_{shut} is the length of the shuttle.

The axial deflection of the shuttle is given by N times the electrostatic force at the final position, divided by Eq. 4.11

$$x_{shut} = \frac{NF_{es,x_f}}{k_{shut}} \tag{4.12}$$

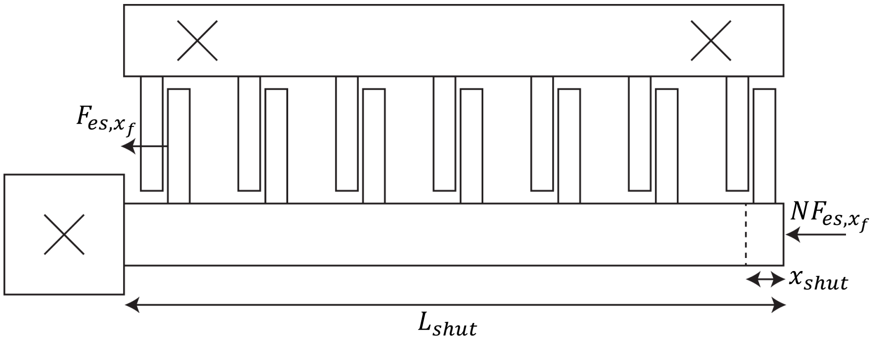


Figure 4.4: Diagram of the model for the axial deflection of the shuttle The shuttle has a cross-sectional area A_{shut}. The dotted line represents the exaggerated axial deflection of the shuttle

The velocity at the time of release will then be

$$v_{shut} = \sqrt{\frac{k_{shut}}{m_{shut}}} x_{shut} = \omega_{shut} x_{shut} \tag{4.13}$$

where ω_{shut} is the natural frequency of the shuttle.

We assume that the fingers and the shuttle relax to their zero position immediately after the voltage is removed from the array. In order to evaluate some sort of initial velocity on the GCA array assembly, we will assume lossless momentum transfer between the effective mass of the fingers and shuttle and the mass of the entire GCA array assembly.

$$v_o = \frac{N m_{fing} v_{fing} + m_{shut} v_{shut}}{m_{GCA}} \tag{4.14}$$

4.1.4 Finger Deflection in FEA

An FEA simulation of the electrostatic force on the fingers at their final position was performed in CoventorWare.

Fig. 4.5 shows the simulation environment. The fingers are suspended from anchors. A range of voltages is applied to the rightmost finger and the leftmost finger is held at ground. The simulation is run to detect pull-in. As the voltage range approaches the pull-in voltage, the analyzer detects convergence of the electromechanical simulation. If the simulation converges the system has reached a static mechanical state. If the simulation diverges the fingers have pulled in to each other indicating catastrophic pull-in failure. The voltage is changed in decreasing increments around this unstable point until the desired tolerance for calculating the pull-in voltage is met.

For these fingers, pull-in was detected in simulation at approximately 75V. Devices typically showed shorting between the movable finger array and the drive fingers above 70V. The finger lengths were chosen based on an expression for catastrophic failure given in [30].

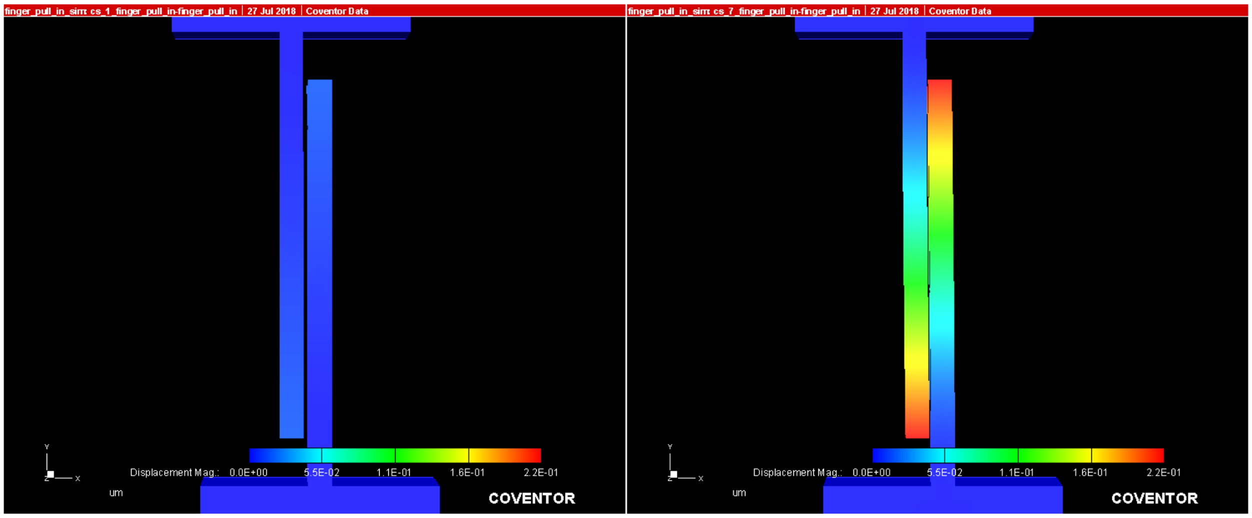

finger_pull_in_sim: cs_1_finger_pull_in-finger_pull_in | 27 Jul 2018 | Coventor Data
finger_pull_in_sim: cs_7_finger_pull_in-finger_pull_in | 27 Jul 2018 | Coventor Data
Displacement Mag.: 0.0E+00 5.5E-02 1.1E-01 1.6E-01 2.2E-01
um
COVENTOR
Y
X
Z

Figure 4.5: Stills from the CoventorWare simulation of the fingers before deflection (left) and at 90V (right)

The derivation assumed a maximum voltage of 100V. The actual maximum voltage is approximately 70-75% of this value.

Fig. 4.6 shows the deflection of the finger as the voltage is increased plotted against the electrostatic force on the conductor. The simulation indicates a softer spring than the analytical model. This was expected due to the electrostatic force's dependence on the gap size. The linear fit indicates a spring constant that is 14% less stiff than the one given in Eq. 4.7. This simulated spring constant is used in the dynamics model.

4.1.5 MATLAB Model

A dynamics model was developed in MATLAB. The model takes the equations and initial conditions in the previous section and uses an ordinary differential equation (ODE) solver (ode45) to solve the displacement over time for the GCA array. In order to solve a second order differential equation, the equation must be presented as a set of first order differential equations. Defining a state vector $X = [x,\ \dot{x}]$, the ODE solver will solve the derivative of this vector

$$\frac{dX}{dt} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \frac{F_{es} - F_b - Fk}{m} \end{bmatrix} \tag{4.15}$$

for the initial conditions $X(t_{pull-in} = 0) = [0,\ 0]$ in the case of the actuated array. For the release phase, $F_{es} = 0$ and $X(t_{release} = 0) = [x_o - x_f,\ v_o]$. The ODE solver is set to complete and measure the time when $x = x_o - x_f$ for the pull-in time and when $x = 0$ for the release time. This time is what is used for the pull-in and release time.

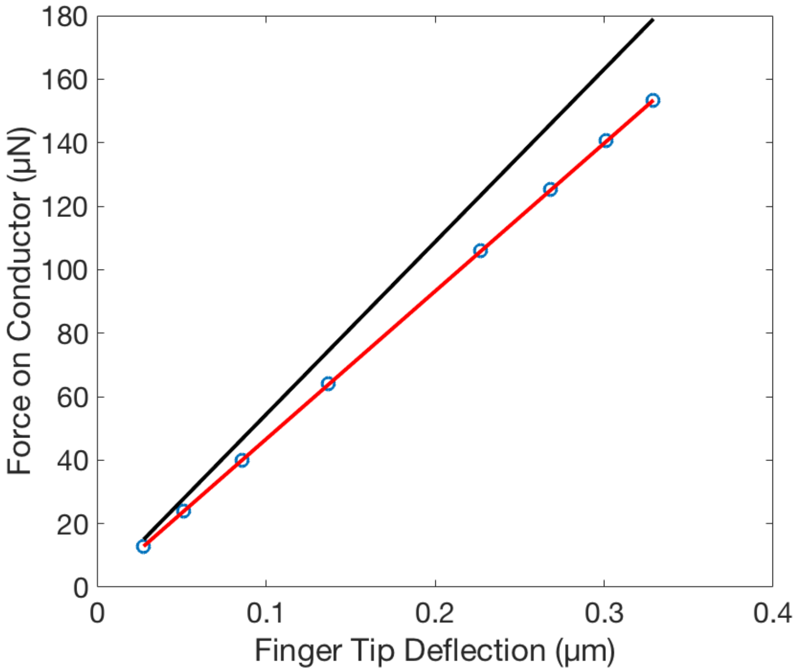


Figure 4.6: A plot of the deflection of the finger tip vs electrostatic force on the finger from the CoventorWare simulation

Before looking at experimental data, we looked at the behavior of our MATLAB model to understand how the model was influenced across a range of variables for each of the components of the forcing function. The full MATLAB model is included in Appendix B.

4.1.6 Size of Gap Over Time

Fig. 4.7 shows the size of the gap over time during actuation and release. The initial size of the gap is 5.8μm and the gap closes to 1μm, defined by the gap stop. The pull-in and release times for the nominal dimensions of the structure are shown in blue. This curve is compared to a 50% increase and a 50% reduction in either the mass, spring constant, or damping coefficient.

While the mass of the structure has a noticeable effect on the pull-in time, a 48% time increase for a 50% mass increase, the spring and damping forces are negligible, causing up to a 1% change for a 50% increase in the spring constant and a 2% change for a 50% increase in the damping coefficient.

In the case of the release times, the three parameters have effects on roughly the same order of magnitude on the time. Fig. 4.8 shows an extended time plot of the release phase for the nominal case. The figure shows oscillation of the GCA array at about 9kHz that die out on the order of hundreds of microseconds. The return phase brings the fingers about

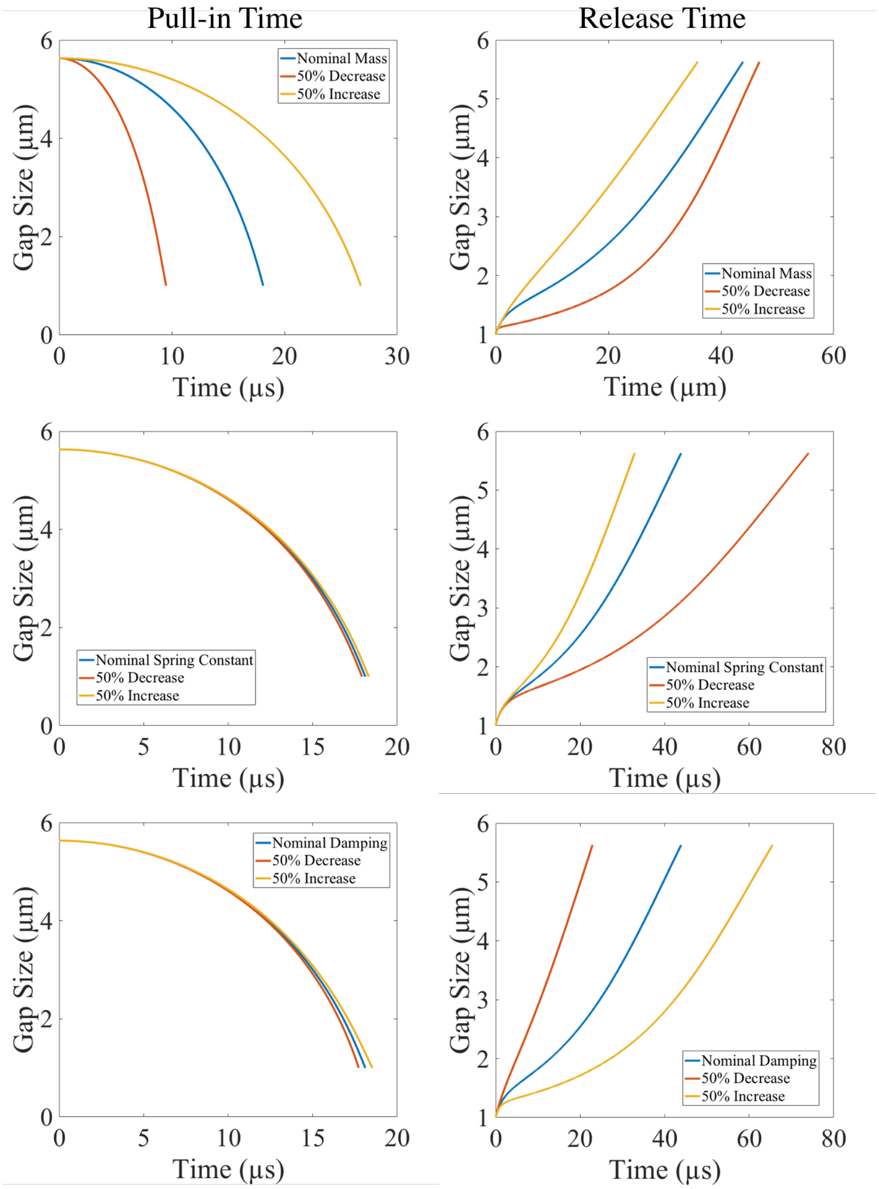


Figure 4.7: The size of the gap is shown during actuation and release. These curves were found using the MATLAB model

1μm from the initial gap stop position.

4.1.7 Pull-in and Release Times

The behavior of the pull-in and release time across ranges of variation for mass, damping coefficient, spring constant, and actuation force are shown in Fig. 4.9. Each of these

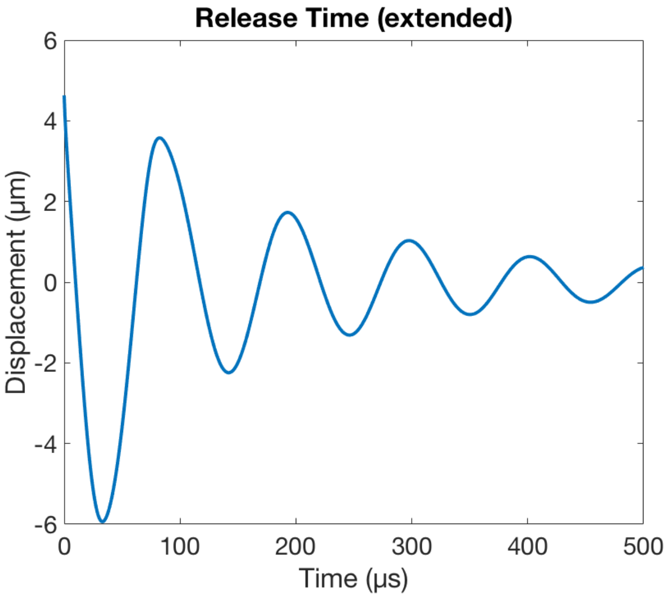


Figure 4.8: Release time simulation extended to 500µs

parameters was varied by an absolute factor ranging from 0.1 to 2. The nominal value of each parameter is shown in the graphs. As anticipated from Fig. 4.7, the damping coefficient and spring constant have a large impact on the release time but a negligible impact on the pull-in time, while mass and actuation force have large effects on the pull-in time.

We can also see the influence of finger deflection and axial deflection of the shuttle in the release time. What is expected to be a flat line for the release time as force is varied is actually a significant linear trend towards faster release times for larger actuation forces.

4.1.8 Detecting Gap Closing in Air

Fig. 4.10 shows a simplified layout model of the device used for experimental verification and a photo of a fabricated device.

For these experiments, the sense pads were tied to a low voltage signal through resistors. Then the grounded fingers make contact with these pads, the low voltage signal drops, signaling the time of gap closure. For pull-in time, the gapstop is used as the sense node. For release time, a movable sense structure must be used since detection must occur at the nominal position of the device. The movable sense structure is positioned with a probe tip to rest against the movable fingers at their nominal position.

Fig. 4.11 and Fig. 4.12 show samples of the pull-in and release signals used for detection respectively. In the pull-in signal, the time is measured from the pull-in voltage on the actuation signal to the beginning of the signal drop on the sense signal. On these signals, switch bounce was noticeable upon contact. Capacitive coupling also caused a slight rise in the sense signal at the time of actuation signal rise.

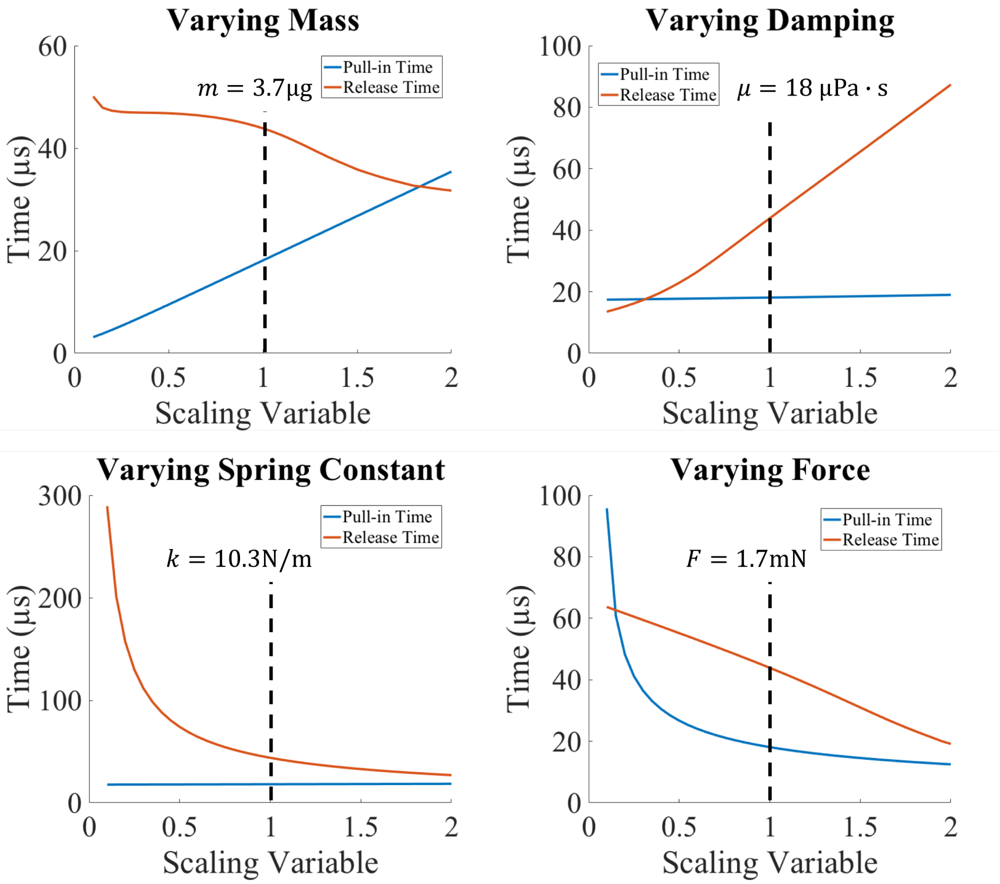


Figure 4.9: Pull-in and release are plotted while varying the mass, damping coefficient, spring constant, and force by a factor between 0.1 to 2. The nominal value of each value is shown above the dotted line, marking where this nominal value occurs

A similar method is used for the release signal measurement. The time from the release voltage of the actuation signal to the signal drop on the sense signal is measured for the release time. On the release signal trace, ringing was noticeable at the time of actuation signal drop. This is attributed to coupling in the measurement setup. Switch bounce was also present, shown in the zoomed out figure.

The structures are metalized so that sidewall contacts are low resistivity. The silicon used for these devices was typically around 10Ωcm. The resisitivity between sidewalls was high, typically over 10MΩ. Sputtered AuPd and TiN were used to provide high sidewall conductivity for these experiments.

The experimental array varied the overlap length of the fingers and the spring constant of the GCA array, as well as voltage for each GCA array. The parameters of the nominal GCA array are shown in Table. 4.1. The variable array is given in Table. 4.2. L_{ol} was varied from $0.2L_{ol}$ to the nominal dimension of L_{ol}, 76.5µm. k_{spr} was varied by changing the nominal

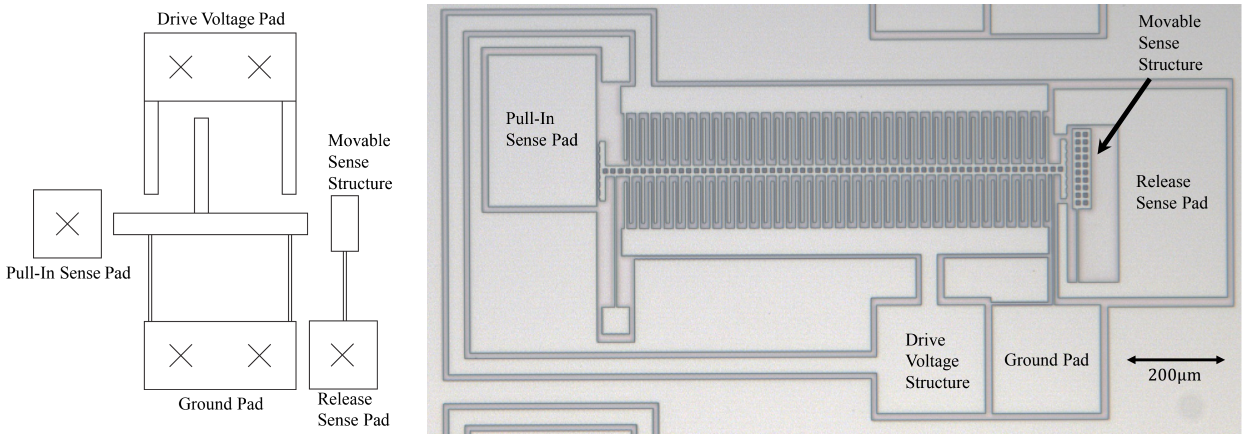


Figure 4.10: Layout for the device, labeling each of the components of the structure (left) A photo of the fabricated device (right)

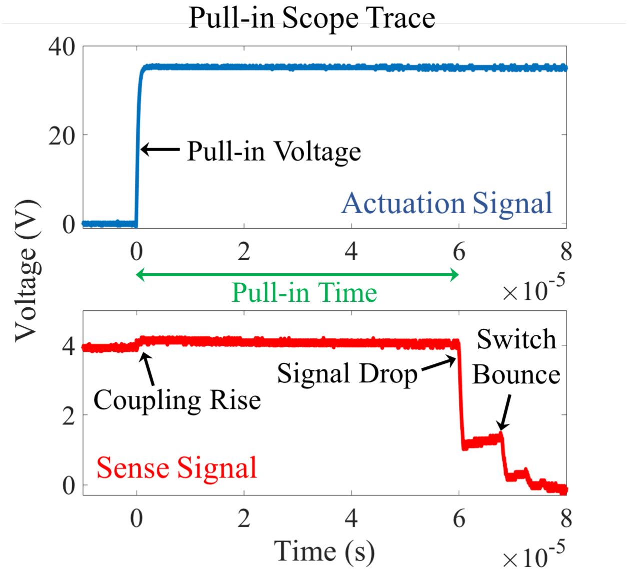


Figure 4.11: A sample scope trace of the signals used to measure the pull-in time

Table 4.1: Nominal dimensions of the GCA array used in these experiments and for the motor. The star ($*$) indicates the nominal dimension of the parameter in the array that was varied for the experiments

$L_{ol}*$(µm)	x_o(µm)	w_f(µm)	x_b(µm)	L(µm)	$w_{spr}*$(µm)	L_{spr}(µm)	N
76.5	4.8	5	7.75	86.5	3	240.8	70

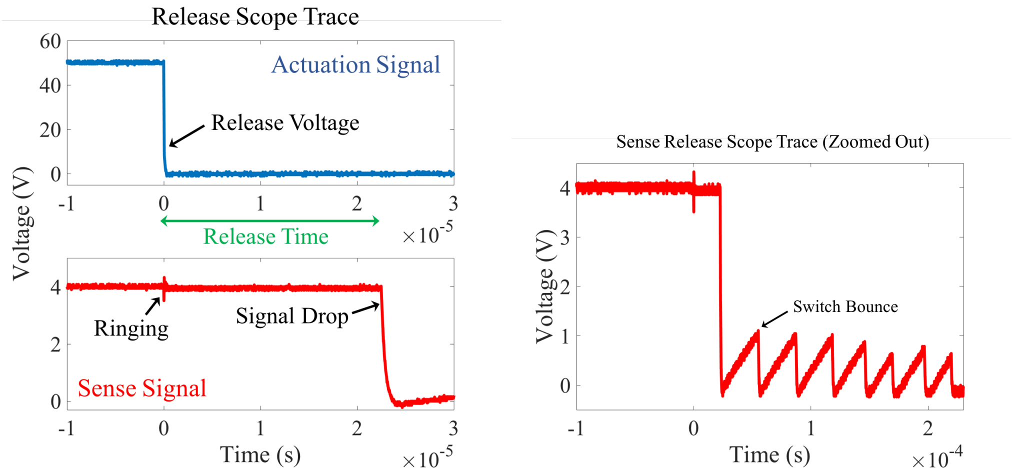


Figure 4.12: A sample scope trace of the signals used to measure the pull-in time (top) A zoomed out image of the same sense signal trace, showing obvious switch bounce (bottom)

Table 4.2: Variable array used for GCA speed experiments

L_{ol}(µm)	15.3	23	30.6	38.3	45.9	53.6	61.2	68.9	76.5
k_{spr}(µm)	1.7	4.7	10.3	19	31.7	49	71.7	100.4	-

dimension of w_{spr} by increments of 0.5µm around the nominal value of 3µm. Additionally, the patterned dimension of w_{spr} will be subject to the approximate 0.5µm undercut during the DRIE.

4.1.9 Experimental Results

Pull-in and Release Voltages

For a GCA array, the pull-in voltage will be found by finding the position at which the spring force equals the electrostatic force. Because the fingers in the array are bounded on both sides, the traditional expression for pull-in voltage is not accurate.

To find our pull-in voltage the electrostatic and spring forces are added to find the full system force. This is set to zero since the electrostatic and spring forces cancel out at pull-in and the expression is solved for the voltage

$$\frac{1}{2}\epsilon_o V^2 N L_{ol} T \left(\frac{1}{(x_o - x)^2} - \frac{1}{(x_b + x)^2} \right) = kx \tag{4.16}$$

$$V_{pi} = \sqrt{\frac{2kx_{pi}}{\epsilon_o N L_{ol} T} \left(\frac{1}{(x_o - x_{pi})^2} - \frac{1}{(x_b + x_{pi})^2} \right)^{-1}} \tag{4.17}$$

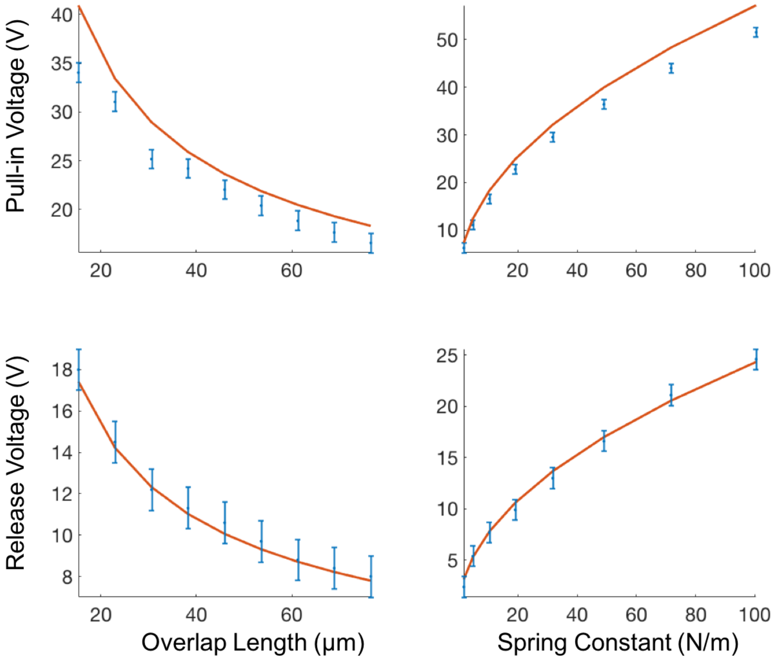


Figure 4.13: Pull-in voltage (top) and release voltage (bottom) for each array. The voltages are plotted as functions of overlap length (left) and spring constant (right). Analytical results from Eq. 4.17 and 4.19 are shown in red

This expression is solved for each value of x within the gap and the maximum voltage, V_{pi} (the point when the spring force becomes smaller than the electrostatic force for increasing gap or when $\frac{dF}{dx} = 0$) is taken as the pull-in voltage and the position at that point, x_{pi}, as the pull-in displacement. For the GCA arrays with $x_b = 1.5x_o$ the pull-in displacement is $0.28x_o$ as opposed to $\frac{1}{3}x_o$ for a single parallel plate. As g_b is made larger, the pull-in displacement approaches $\frac{1}{3}x_o$ as expected.

The release voltage is found by equating the electrostatic force to the spring force at $x = x_o - x_f$ and solving for the voltage

$$\frac{1}{2}\epsilon_o V^2 N L_{ol} T \left(\frac{1}{(x_f)^2} - \frac{1}{(x_b + x_o - x_f)^2} \right) = k(x_o - x_f) \tag{4.18}$$

$$V_r = \sqrt{\frac{2k(x_o - x_f)}{\epsilon_o N L_{ol} T} \left(\frac{1}{(x_f)^2} - \frac{1}{(x_b + x_o - x_f)^2} \right)^{-1}} \tag{4.19}$$

Fig. 4.13 shows the measured pull-in and release voltages for each GCA array. Multiple measurements were performed for each data point. Measurements were repeatable to within 0.1V for the same device.

There is an overestimate in the model for the pull-in voltage. One reason for this could be fringing field capacitance. The error in the force term from neglecting fringing fields is largest when the gap is open, as much as 10%. Measured values at higher voltages are off by less than 10%.

From this data the spring constant in the MATLAB model was calibrated to better match experimental results. An increase in the expected spring constant of 20% for the release voltage data was implemented in the pull-in/release voltage model. This was attributed to finger deflection and deformation of the GCA shuttle, as well as deformation of the contact contributing to the spring constant during pull-in.

Additionally, these voltages are important for measuring the exact moment of pull-in and release. A more accurate measurement of the full pull-in and release times can be made this way by finding the exact time that the pull-in or release process is beginning.

Pull-in Time

Fig. 4.14 shows the pull-in time as a function of voltage for varying overlap length. We see good agreement between the measured data and theoretical line. Most measured values fall within uncertainty of the measured values. Measured points for $L_{ol} = 15.3 upmu$m and $L_{ol} = 15.3$µm fall below the theoretical line. An underestimation of the force from neglecting fringing field capacitance could explain this discrepancy.

The uncertainty of the measured values was also very low, often around 1µs. Values were highly repeatable between different devices and different wafers.

Fig. 4.15 shows the pull-in time as a function of voltage for varying spring constant. Again we see good agreement between data and theory and good repeatability on the same device and across devices. Values were limited to 65V due to catastrophic finger pull-in on the metalized devices at and above 70V. The lowest voltage for each plot is above the pull-in voltage of the device.

Fig. 4.16 shows the same data as Fig. 4.14 and Fig. 4.15 but plotted against the varied overlap length and spring constant for different voltages.

Release Time

Fig. 4.17 shows measured data for release time plotted alongside the theoretical model. We see agreement with the data in a few plots. The dependence of the release time on the voltage is obvious in the data, and the model follows that trend to a certain degree. The uncertainty values for the release time are larger than for the pull-in time. The positioning of the sense node for the release time measurement contributed to these larger values since repeatably placing the movable sense block at the exact same position was difficult.

Fig. 4.18 shows the release time data for varying spring constants across a range of voltages. Once again the maximum voltage was limited by catastrophic pull-in. There is a discrepancy for the data for k=1.7N/m. The measured time is smaller than the expected theoretical time by as much as 40%. An underestimation of the spring constant by 10%

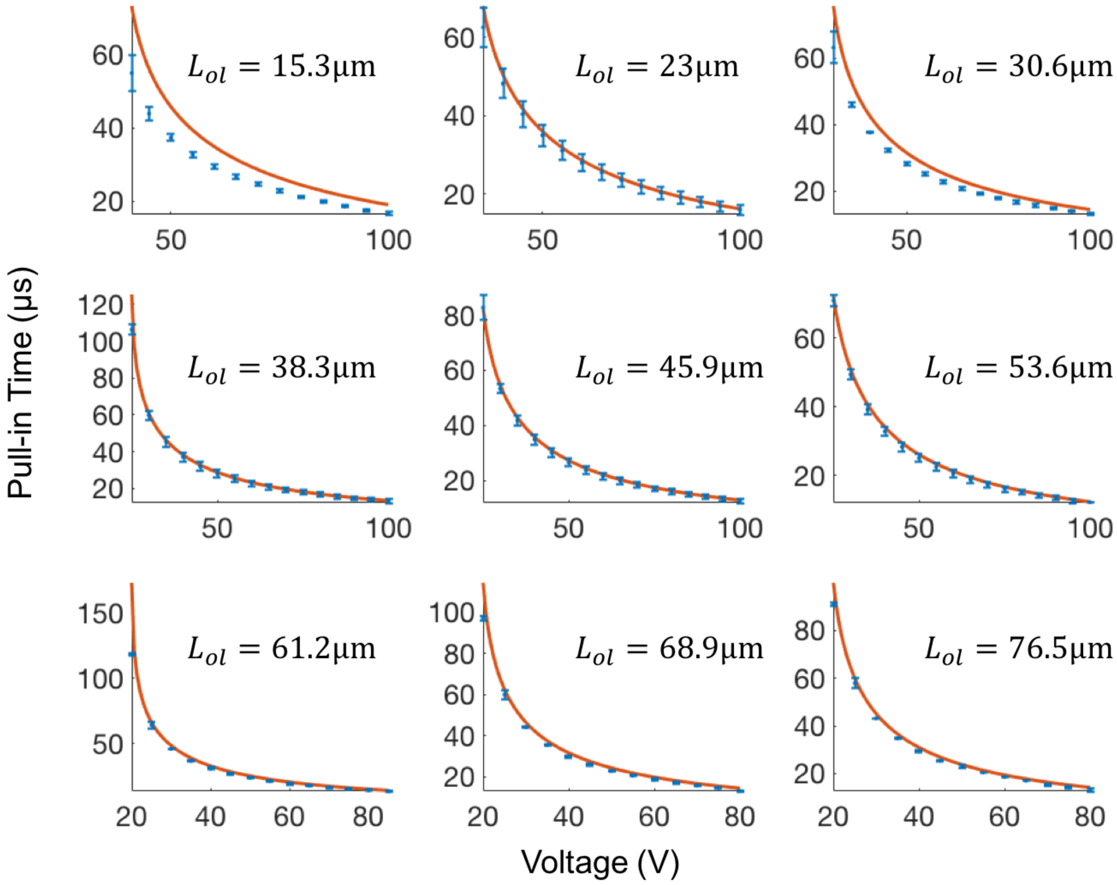


Figure 4.14: Pull-in time measured while varying voltage for varying overlap lengths. The MATLAB model is plotted in red. For this measurement array the spring constant is the nominal value of 10.3N/m

or overestimation of the damping by 20% could explain this result. For the other spring constants the theory underestimates the release. The discrepancy trends smaller for larger spring constants.

Fig. 4.17 shows the same data plotted against the overlap length and spring constant for four different voltages. We see good agreement for varying overlap length and spring constant.

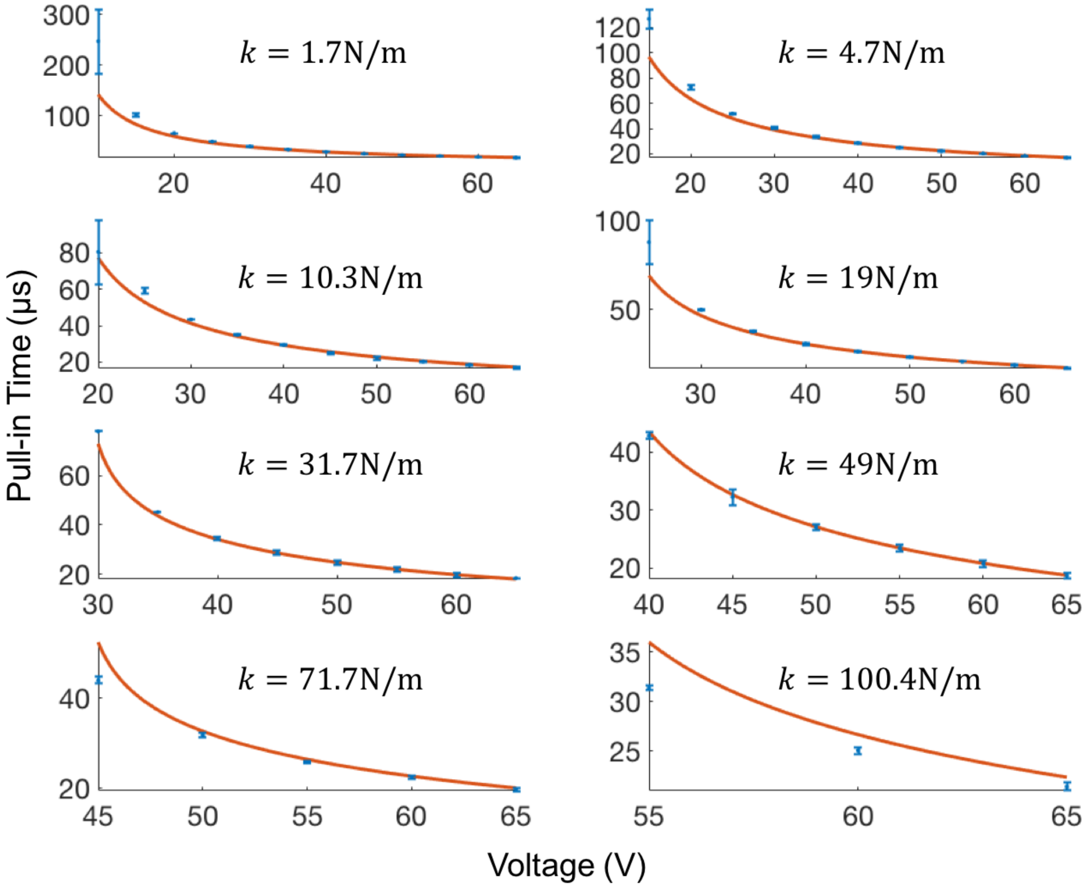


Figure 4.15: Pull-in time measured while varying voltage for varying spring constants. Results from the MATLAB simulation are also plotted in red. For this measurement array the overlap length is the nominal value of 76.5µm

4.2 Inchworm Motor Speed

Gap closer speed is important for modeling the speed of the inchworm motor as a system. A variety of measurements were made to find the speed of the inchworm motor.

4.2.1 Endstop Detection Measurements

Fig. 4.20 shows the end stop detection implemented on the inchworm motors. This system uses a contact at the end of travel of the motor. A low voltage signal is placed on the sense node and when the grounded shuttle makes contact with the sense node, the signal is fed back to the microcontroller controlling the motor. This triggers the feedback response and

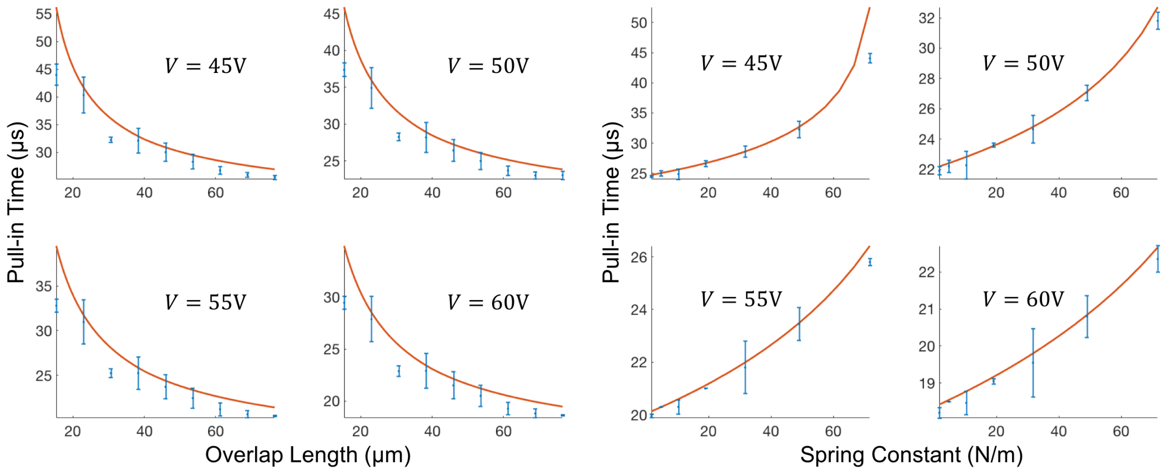


Figure 4.16: Pull-in time measured across the array of varying overlap lengths for different voltages. The spring constant is 10.3N/m (left) Pull-in time measured across the array of varying spring constants for different voltages. The overlap length is 76.5µm (right). The MATLAB model is shown in red

the motor stops.

This is also used to measure the time between the beginning of actuation and the end of actuation. Once the signal drop is sensed, the time between the start of actuation and end of actuation is measured. Then the distance traveled is divided by this time to find the velocity. In order to detect the signal drop, devices were sputtered with AuPd to increase conductivity. TiN was used instead of AuPd in later iterations due to poor adhesion between the silicon and AuPd.

Fig. 4.21 shows the measured velocity using this method. The solid line indicates the expected result according to the step size (2µm per step) and frequency. Most of the points trend superlinear with respect to the expected 4µm/s/Hz velocity line. This could be because of variations in the motor's stepsize as a function of frequency.

4.2.2 Distance Traveled Measurements

Another method used for measuring the speed was to program the motor to take a certain number of steps at a certain frequency. The measured distance traveled was divided by the time the driving waveform takes to complete for the velocity. It should be noted that these devices were sputtered with TiN to test the durability for motor longevity. The highest lifetime of a motor observed was 28 hours without TiN. TiN did not offer a noticeable improvement to device lifetime.

Fig. 4.22 shows the data from these devices. Again, the expected velocity curve was a slope of 4µm/s/Hz. The linear fits trend at least twice as much of a slope. For low velocities

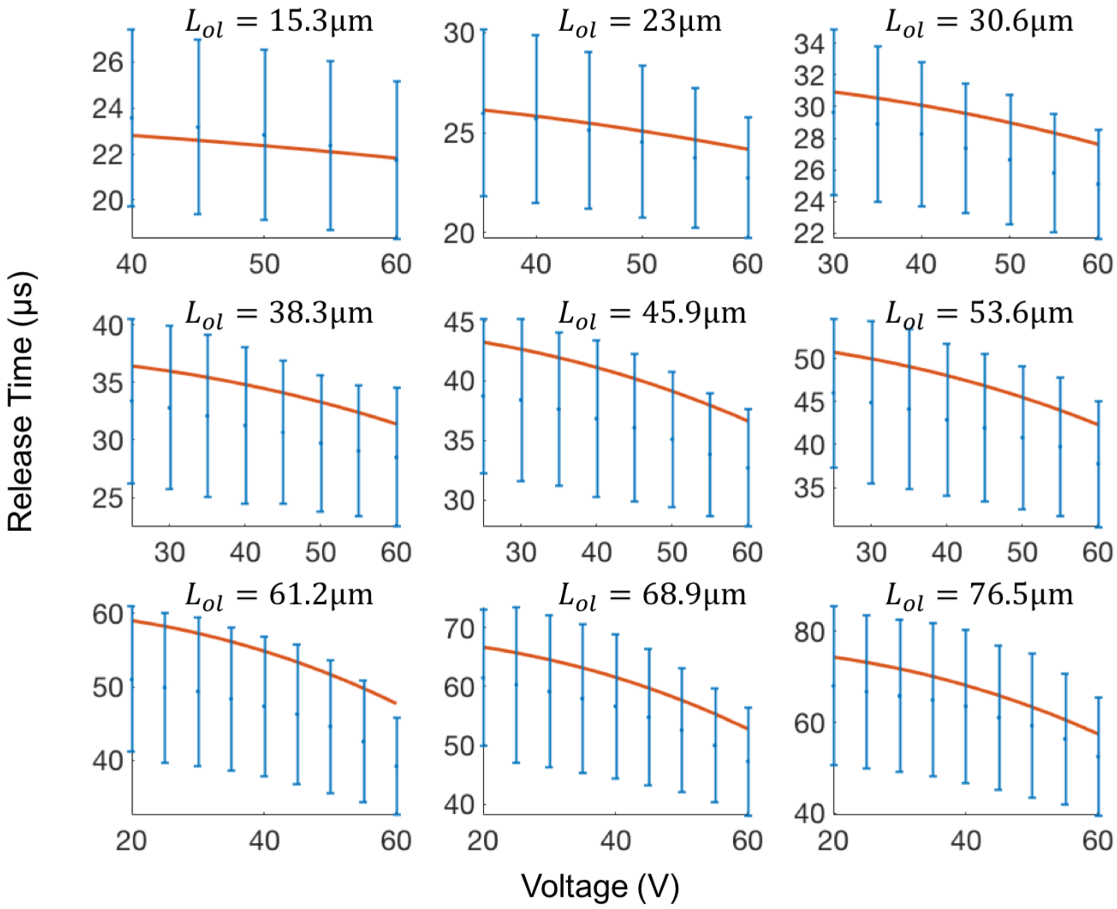


Figure 4.17: Release time measured while varying voltage across the array of varying overlap lengths. The MATLAB model is shown in red. Spring constant is the nominal value of 10.3N/m

it was observed that pawls will often push past the currently engaged pawls due to inertia, leading to larger step sizes.

4.2.3 Work Output

Looking at the work output for these devices we can take data from Fig. 4.23. This plot shows data for 65V for a motor driven to take 10 steps and another motor driven to take 50 steps. We can see for the 50 step case, the motor does not achieve as high a maximum velocity. The spring constant of the suspension for the motors used to take this data was 0.3N/m. The total displacement of the spring for these measurements was 350μm in a time of 1.2ms. Analyzing the work the motor outputs onto this spring, we get 15.3μW. For a

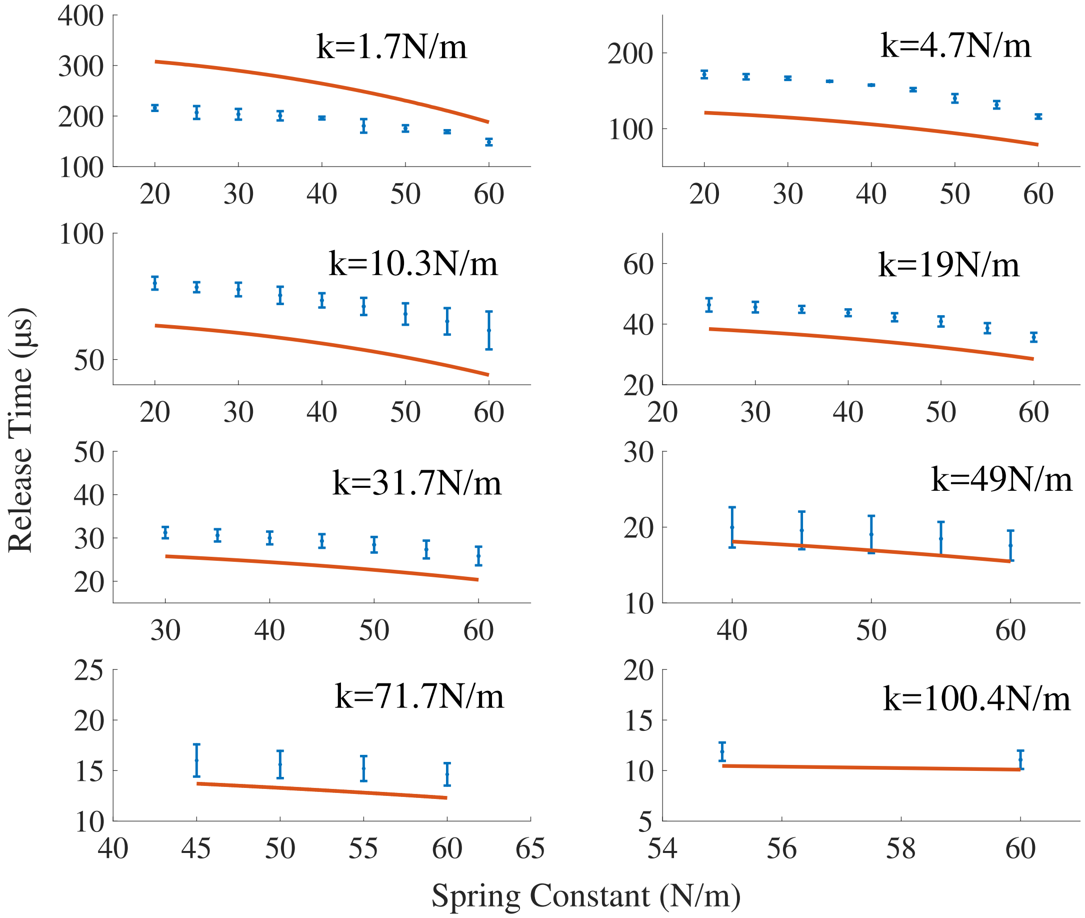


Figure 4.18: Release time measured while varying voltage across the array of varying spring constants. The MATLAB model is shown in red. Overlap length is the nominal value of 76.5µm

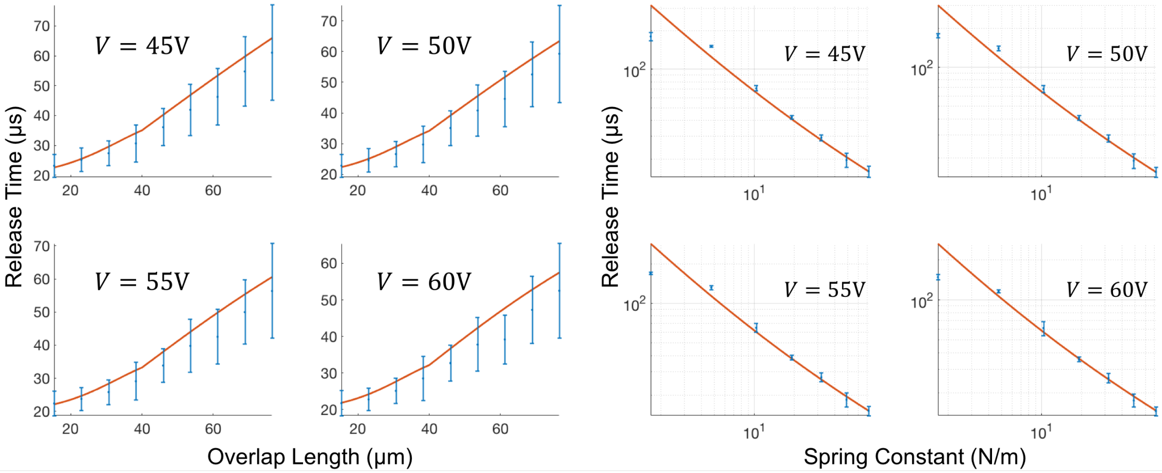


Figure 4.19: Release time measured across the array of varying overlap lengths for different voltages. Spring constant is the nominal value of 10.3N/m (left) Release time measured across the array of varying spring constants for different voltages. This data is plotted on a log-log scale. Overlap length is the nominal value of 76.5μm (right) The MATLAB model is shown in red

reduced motor weight of 0.3mg this corresponds to a work-to-weight ratio of 50W/kg. The substrate would have to be ground down to 100μm thick and 30% of the area etched away, abiding fill requirements for etching. Additionally, the force-to-weight ratio for a reduced weight motor at this voltage is approximately 200.

4.3 Motor Optimization

This optimization scheme seeks out the best thickness to use for an electrostatic inchworm motor in terms of maximum actuation frequency and velocity compared with electrical-to-mechanical efficiency. The scheme presented here borrows from the geometric optimization presented in [23] and expands on it.

4.3.1 Minimum Feature Size and Base Dimensions

For this motor optimization scheme we will assume some starting parameters from fabrication constraints. This assumes that because of electrostatic force density scaling relationships, if we have a desired minimum force density we can always achieve this below some minimum feature size. If our process has a width-to-height ratio α and our device layer has a thickness T we have a minimum feature size defined by

$$\lambda = \alpha T \tag{4.20}$$

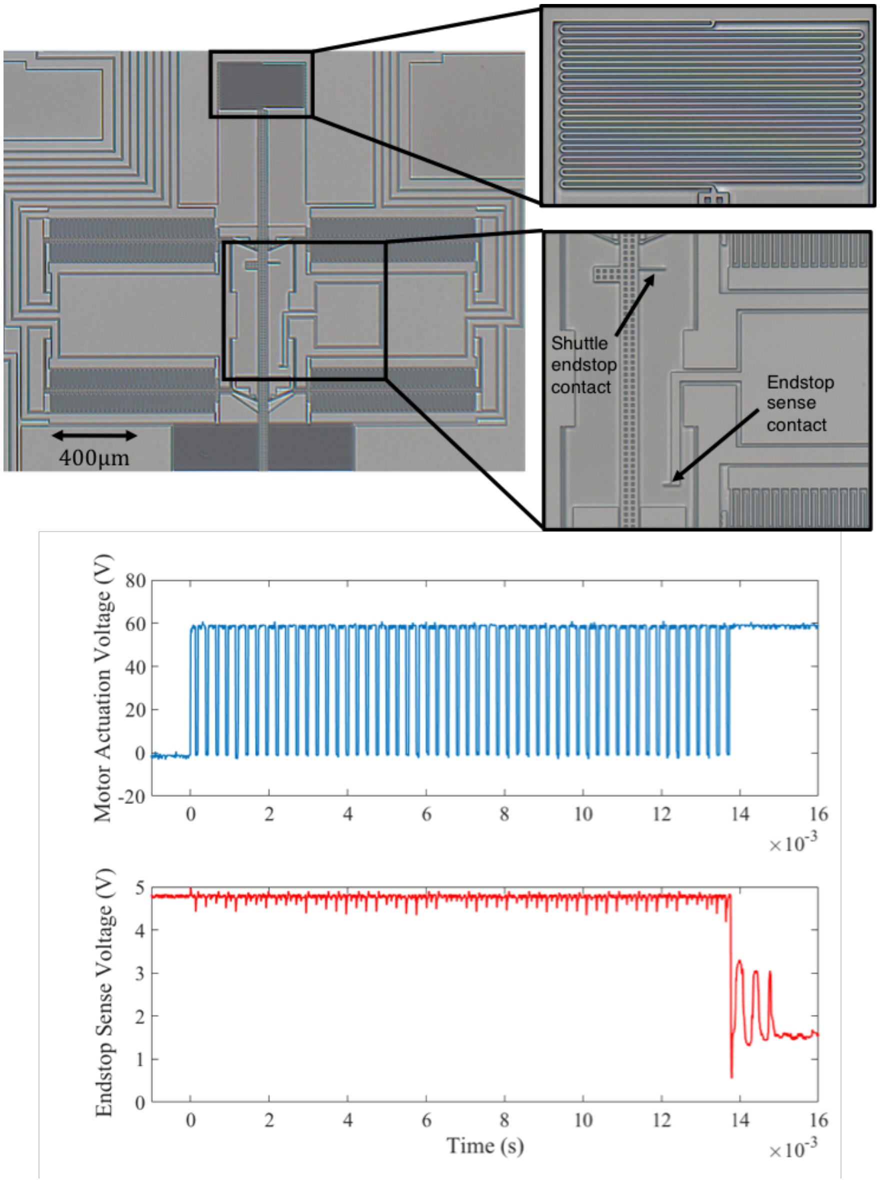


Figure 4.20: The endstop detection structures implemented around the inchworm motor (left) The acutation signal and sense signal from a motor operating with feedback (right)

It should be noted that because of process undercut all drawn features will be smaller by l_{uc} on all sides and gaps will be larger by l_{uc} on all sides once fabricated. Thus a beam of 2µm width will be 2µm$-2l_{uc}$ after fabrication. This scheme will not factor in the undercut. Here we assume the factors have been incorporated into the dimensions already and calibrated out after iteration of experimental designs.

We can define a set of base dimensions from the minimum feature size, shown in Table 4.3.

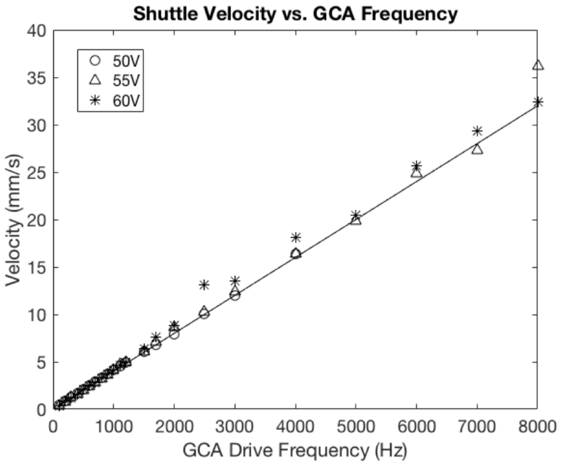


Figure 4.21: Measured inchworm velocity plotted against the expected result (solid line)

Table 4.3: Base Dimensions used in the Motor Optimization Scheme

Finger Width, w_f	Final Gap, x_f	Shuttle Stepsize, Δx	Anchor Size, L_A	Angled Arm Width, w_{arm}
2λ	λ	λ	25λ	2λ

These dimensions are labeled in Fig. 4.24. For the angled arm, we will assume a fixed angle ϕ to the shuttle. This variable is separately optimized in the scheme shown in [23] for a fixed set of constraints.

4.3.2 Gap Sizes

The shuttle-pawl distance, highlighted in Fig. 3.4, is given by

$$Y_I = \lambda + h_t + 2l_{od} - g_{fudge} \tag{4.21}$$

Since we are neglecting the undercut, l_{od} and g_{fudge} will be neglected assuming they have been calibrated after experimental design. If the tooth height, h_t, is set too the minimum feature size, then $Y_I = 2\lambda$.

We recall from Eq. 3.7 that $\Delta x = \frac{\Delta x_{shuttle}}{\tan\phi}$. The initial gap is given by

$$x_o = \Delta x + Y_I + x_f \tag{4.22}$$

The back gap, x_b, can be set to a constant times the front gap, $x_b = Bx_f$. Maximizing areal force, where the area is taken solely from active finger geometry, it can be analytically shown that $B \approx 2.4$. However this number is largely influenced by the amount of peripheral area and depends on the specific design and mechanical advantage of the array.

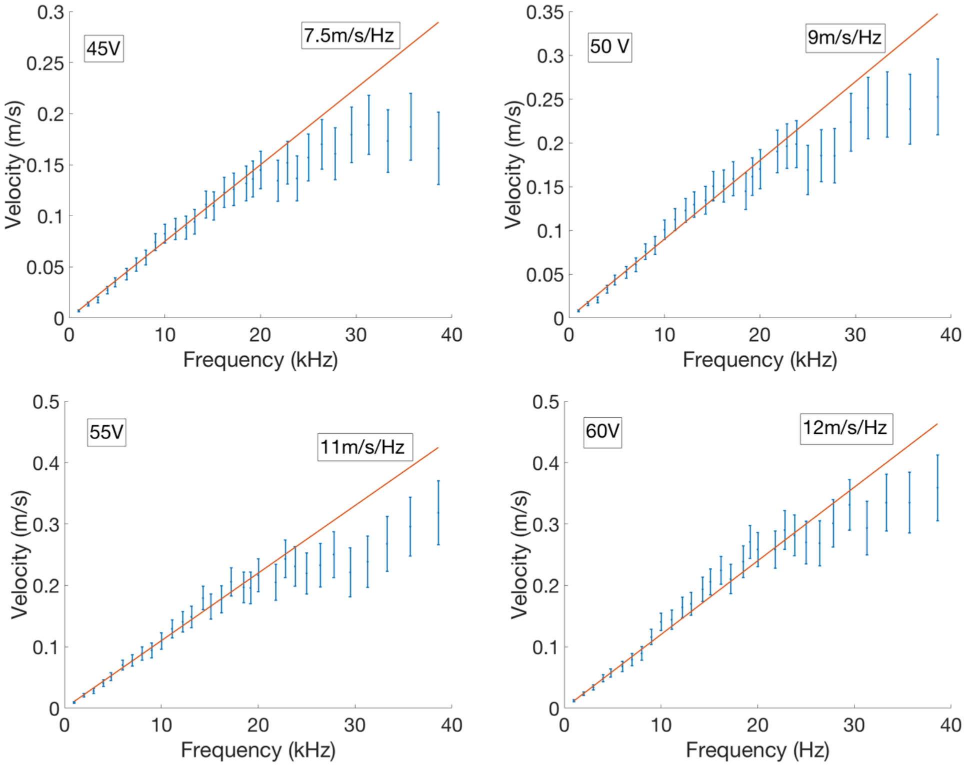


Figure 4.22: Velocity data for various voltages. The red line indicates the slope from the lower frequency data points

4.3.3 Finger Length

The finger length is chosen to prevent catastrophic pull-in at the maximum applied voltage. Osterberg derived an expression for pull-in of a cantilever under electrostatic force in [30]. This value was used to choose the length of the fingers in the current inchworm motors. However, from measurements and simulation shown in Section 4.1.4 it was seen that actual pull-in failure occurred at 70% of the analytical maximum voltage. Assuming a safety factor, C_l, to adjust the model based on various geometries, the maximum finger length will be given by

$$L_{max} = C_l \sqrt[4]{\frac{0.28 E w_f^3 x_f^3}{\epsilon_o V_{max}^2 \left(1 + 0.42 \frac{x_f}{T}\right)}} \tag{4.23}$$

This sets an upper bound on the finger length. We will define the support length, L_s, as $\frac{1}{10} L_{max}$ and we will say that $L_{max} = L_{ol} + L_s$. The force, the area of the array, and the capacitance will be proportional to the finger length so there will not be an optimum value for the overlap length for force density or efficiency. In terms of pure force output, we want L_{ol} to be as large as possible, so we can use L_{max} as our finger length in the optimization scheme.

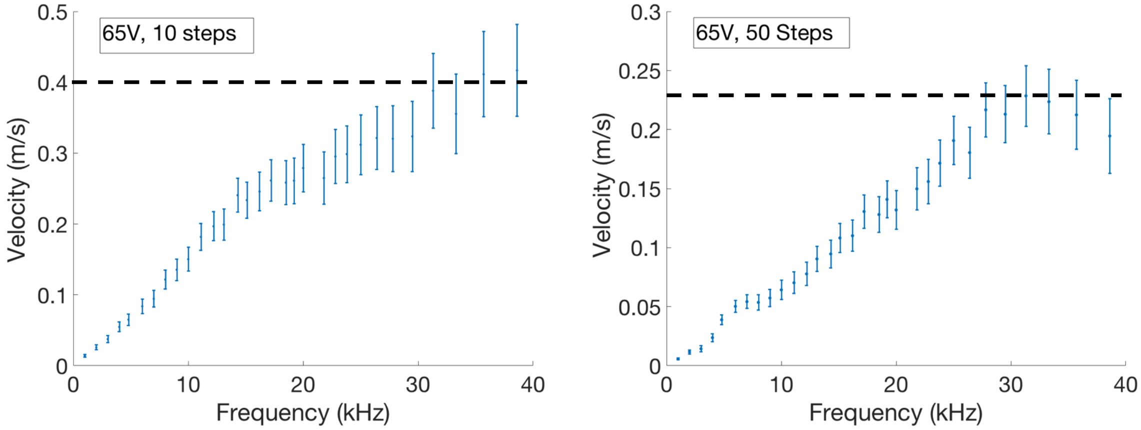


Figure 4.23: Velocity data for 65V actuation for a set number of steps of 10 (left) and 50 (right)

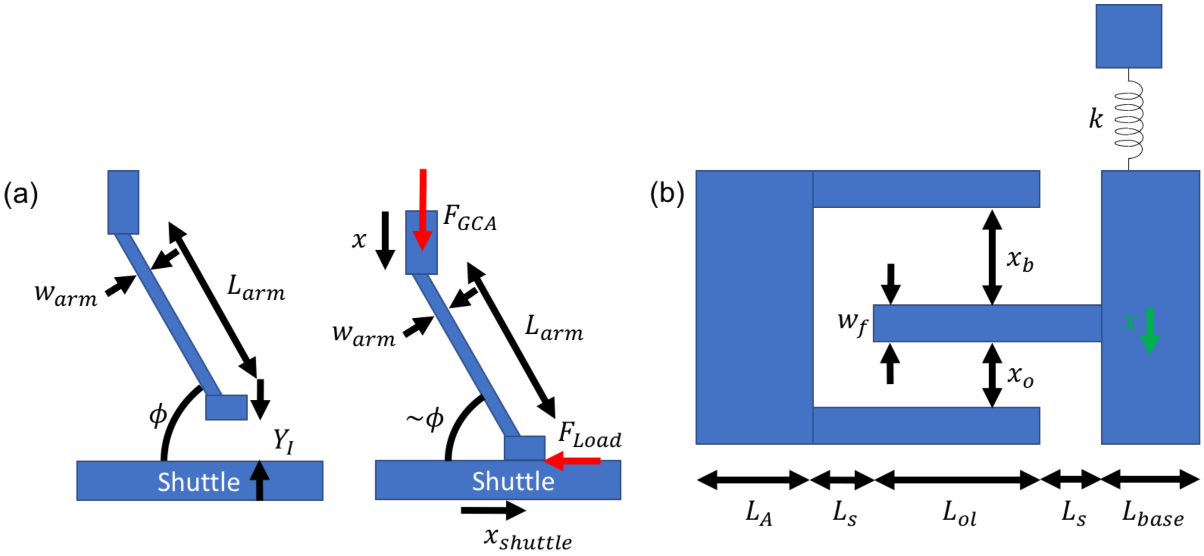


Figure 4.24: Dimensions for features in the optimization scheme of the angled arm (left) and the GCA fingers (right)

4.3.4 Capacitance and Force Output

We can find an expression for the capacitance of the array as The capacitance of the array will be

$$C_{tot}(x) = C_P + C_f(x) = C_P + \epsilon_o N L_{ol} T \left(\frac{1}{x_o - x} + \frac{1}{x_b + x} \right) \tag{4.24}$$

where C_P is the parasitic capacitance of the array, dependent on device geometry and external routing, and the second term is the capacitance between the fingers. We are ignoring fringing fields assuming that with the aspect ratio above 20:1 the open gap fringing fields will add approximately 10% to the capacitance, and proportionally smaller as the gap closes.

The electrostatic force from the GCA array as a function of x will be

$$F_{es}(x) = \frac{1}{2}\epsilon_o V^2 N L_{ol} T \left(\frac{1}{(x_o - x)^2} - \frac{1}{(x_b + x)^2} \right) \tag{4.25}$$

The GCA array will be held by some sort of suspension with spring constant k. Once the angled arm makes contact with the shuttle, the stiffness of the angled arm k_{arm}, found in Section 3.2.3, also comes into play.

We express the combination of these forces as a function of x. There are three regimes of interest for the force function given by

$$F_{GCA}(x) = \begin{cases} F_{es}(x) - kx & x < Y_I \\ F_{es}(Y_I) - kY_I & x = Y_I \\ F_{es}(x) - kx - k_{arm}(x - Y_I) & Y_I < x < x_o - x_f \end{cases} \tag{4.26}$$

The point at $x = Y_I$ is the engagement force. At any point after contact, the force on the shuttle is transmitted through the angled arm. This is given by

$$F_{load}(x) = \frac{F_{GCA}(Y_I \leq x)}{\tan \phi} \tag{4.27}$$

4.3.5 Array Area

With these dimensions, we can now define the dimensions and layout area, A_{array}, of the array

$$L_{array,1} = N(x_o + x_b + 2w_f) \tag{4.28}$$

$$L_{array,2} = L_A + L_{ol} + 2L_s + L_{base} \tag{4.29}$$

$$A_{arm} = L_{arm}^2 \cos\phi \sin\phi \tag{4.30}$$

$$A_{array} = L_{array,1} L_{array,2} + A_{arm} \tag{4.31}$$

Peripheral area including, routing, shuttle extension, and pads, will affect the total area. This is difficult to factor in beforehand and for this scheme we will focus on this as the active motor area. L_{base} and L_{arm} are found in following sections.

4.3.6 Base GCA Shuttle Length

It is important for the accuracy of the dynamics simulation to determine the mass of the movable finger array. According to our model, inertia has a significant impact at the size scales we are working in. The shuttle base dimension, L_{base}, can be determined from maximum load considerations. The shuttle base that all the movable fingers connect to will be under the full load of the GCA array when the gap is fully closed. It is desirable to keep this central base as small as possible for the sake of mass reduction in the dynamics equations. However, it must be able to withstand the full force of the GCA array at the closed gap at the maximum voltage

$$F_{max} = \frac{1}{2}\epsilon_o V_{max}^2 N L_{ol} T \left(\frac{1}{(x_f)^2} - \frac{1}{(x_b + x_o - x_f)^2} \right) \tag{4.32}$$

Finding an appropriate v_{max} is difficult before settling on the device geometry. Setting a fixed maximum voltage is not efficient since the smaller scale designs will produce the same force per unit area at lower voltages so setting too high a voltage constraint can produce needlessly large and robust geometries. As an iterative approach we can first define the same maximum voltage for every design, take the calculated V_{min} value of each design, and set a new array of V_{max} values that are proportional to the V_{min} of each design.

Many variations in the mechanical design of the suspension can be implemented. We consider a simple case here. There are two conditions that can set a limit on the width of the central base. The axial load cannot cause a strain above the strain limit of the material. For silicon we will use $\epsilon_{max} = 0.005$, a value safely under the typically reported 1-2% yield strain. The stress limit is given by

$$L_{base,1} = \frac{F_{max}}{T\sigma_{max}} \tag{4.33}$$

where σ_{max} is the maximum stress limit, $\epsilon_{max}E$.

Additionally, if the gap-stop is positioned at the end of the base making near the angled arm contact, the base must not buckle under the combined load from the fingers. In the case of buckling, the base dimension will be

$$L_{base,2} = \sqrt[3]{\frac{12L_{array,1}^2 F_{max}}{\pi^2 ET}} \tag{4.34}$$

This assumes a pinned boundary condition on a beam with a point load applied at one end. This is not the loading case for the actual GCA array, since the load is distributed across the fingers. However it has been shown that the case of a distributed load has a lower critical force than for a point load [31]. Therefore this sets a safe upper bound. The actual base length will be the maximum of these two values

$$L_{base} = max(L_{base,1}, L_{base,2}) \tag{4.35}$$

4.3.7 Angled Arm Length

In simulation from Section 3.2.6 the angled arm stiffness was bounded between the fixed-pinned and fixed-guided boundary conditions. We will assume fixed-guided boundary conditions as an upper bound case in the model. This is given by

$$k_{arm} = \frac{12EI_{arm}}{L_{arm}^3}\frac{x}{\cos\alpha} \tag{4.36}$$

where $I_{arm} = Tw_{arm}^3/12$. The length of the angled arm should be chosen to prevent buckling under the load from the GCA [23]. The maximum force applied to the angled arm will be when the GCA array fully closes. Assuming the deflection of the arm is small enough to not impact buckling, the ultimate buckling force is given by the Euler equation

$$F_{buckle} = \frac{\pi^2 EI_{arm}}{(KL)^2} \tag{4.37}$$

where K is an effective beam length depending on the boundary conditions. $K = 0.5$ for the fixed-fixed beam condition. This force should be greater than the component of the maximum GCA force axially through the arm

$$\frac{\pi^2 EI_{arm}}{(KL)^2} > F_{max}\sin\phi \tag{4.38}$$

where F_{max} is given by Eq. 4.26 at $x = x_o - x_F$. Solving for the length we get

$$L_{arm} = \sqrt{\frac{\pi^2 EI_{arm}}{nK^2 F_{max}\sin\phi}} \tag{4.39}$$

where we are using n as a safety factor to make sure that the buckling force is some factor above the load force.

4.3.8 Spring Constant, Minimum Force, and Pull-In Voltage

There are a variety of schemes to find a value for the spring constant k. k will have direct influence on the pull-in voltage of the GCA array and the minimum force output. We can define a minimum voltage, the pull-in voltage, in order to define the dynamic range of the actuator.

For our purposes, we will define a minimum output force density, Fd_{min}, on the shuttle. This is often more useful than defining a minimum operating voltage since motor designers focus on force output rather than drive voltage. From this Fd_{min} we must choose a pull-in voltage, V_{pi}, minimum applied voltage V_{min} and a corresponding maximum spring constant k_{max}. Fd_{min} is given by the force output at engagement multiplied by the mechanical advantage of the angled arm and divided by the array area

$$Fd_{min} = \frac{1}{A_{array}\tan\phi}\left(\frac{1}{2}\epsilon_o V_{min}^2 NL_{ol}T\left(\frac{1}{(x_o - Y_I)^2} - \frac{1}{(x_b + Y_I)^2}\right) - k_{max}Y_I\right) \tag{4.40}$$

We have another equation in the form of the pull-in voltage given by the iterative approach from Equation 4.17 where we begin with an arbitrary spring constant to solve the pull-in gap distance, x_{pi}. This will present two equations with three unknowns, V_{min}, V_{pi}, and k_{max}.

For a full system of equations, we can place a constraint between the minimum voltage and the pull-in voltage to ensure the motor has a desired dynamic range below the voltage V_{min}. A higher dynamic range of lower forces below F_{min} at lower voltages may be desirable to allow for lower force step sizes when desired (actuating a leg that has not touched the ground or has no load). We can define the range between V_{pi} and V_{min} by the ratio of the gap sizes where each voltage is found. This is given by

$$\frac{V_{pi}}{V_{min}} = \frac{x_o - Y_I}{x_o - x_{pi}} \tag{4.41}$$

This will ensure a margin between the pull-in voltage and the minimum voltage such that the arm can impart F_{min} on the shuttle after pull-in.

From these equations, we now have an upper bound on the spring constant, dependent on force output and pull-in of the GCA array. Setting the spring constant below k_{max} we can be assured that the GCA array will reach the pull-in voltage, V_{pi}, below V_{min} and output the desired Fd_{min} at or slightly below V_{min}.

4.3.9 Actuator Output

The useful force from the array occurs at $x = Y_I$. This is the point of minimum force on the shuttle arm. However the GCA array is capable of larger work output.

Fig. 4.25 shows the force output of a GCA array. The blue curve shows the raw force output of the GCA array alone. The red curve shows the GCA array force output with the angled arm spring at contact. The engagement force transmitted to the shuttle, $F_{load}(Y_I \leq x)$, is shown in yellow. The purple line shows the useful force output of the motor. This is the lowest point of force input onto the shuttle and the motor would be incapable of pushing against a load above this force. The motors useful work output is given by the area under the purple line.

4.3.10 Efficiency

The mechanical energy output of the array will be the engagement force times the shuttle displacement

$$U_{mech} = F_{load}(Y_I)\Delta x_{shut} \tag{4.42}$$

The input energy can be taken from the electrical energy to charge the capacitor. The output energy will be the theoretical of full recovery of the stored spring energy

$$U_{in} = C_{tot}(x = x_o - x_f)V^2 U_{out} = \frac{1}{2}k(x_o - x_f)^2 - \frac{1}{2}k_{arm}(x_o - x_f - Y_I)^2 \tag{4.43}$$

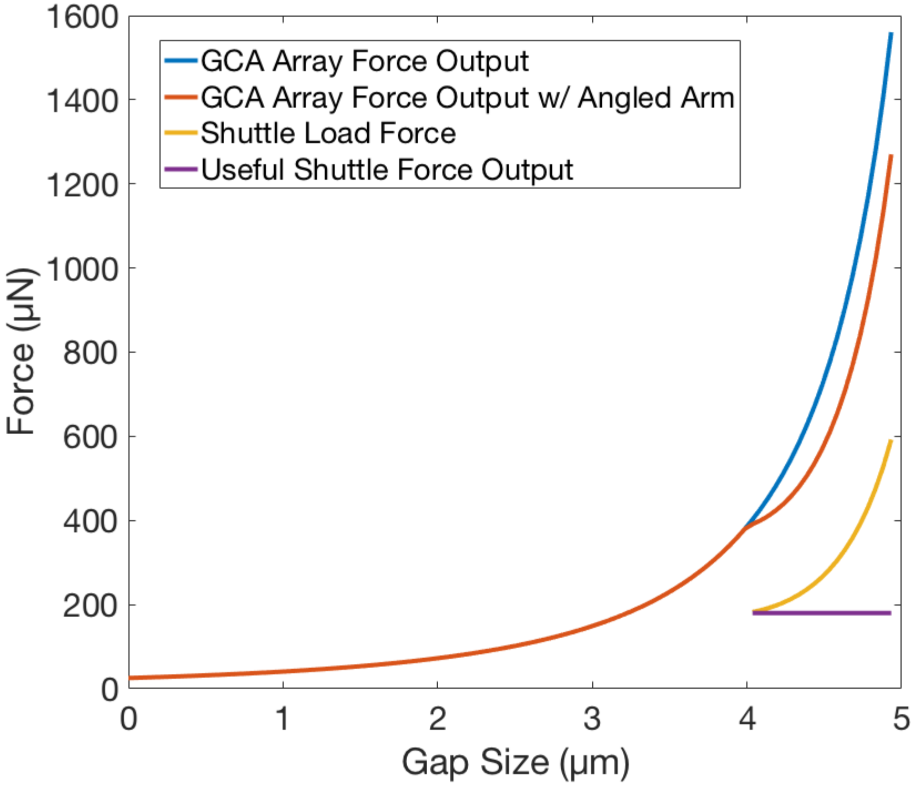


Figure 4.25: Force output of the GCA array/angled arm at various stages

The efficiency is thus

$$\eta = \frac{U_{mech}}{U_{in} - U_{out}} \tag{4.44}$$

4.3.11 Maximum Frequency

We are interested in the maximum achievable actuation frequency for our GCA arrays. For this model, we are using the dynamics model presented in 4.1. However we are modifying the stiffness term to include the stiffness of the angled arm upon contact with the shuttle. This is given by

$$k(x) = \begin{cases} kx & x < Y_I \\ kx + k_{arm}(x - Y_I) & Y_I \leq x < x_o - x_f \end{cases} \tag{4.45}$$

The dynamics model will output the pull-in time, t_{pi}, and the release time, t_r. A GCA array can be driven as fast as the time it takes for one GCA array to fully engage, plus the time it takes for the other GCA array to engage and hold on, plus the time for the first GCA to release. This will give a frequency of

$$f_{max} = \frac{1}{2t_{pi} + t_r} \tag{4.46}$$

4.3.12 Thickness and Aspect Ratio Optimization

We will use a desired minimum force density of 1mN/mm^2. The MATLAB script for this optimization is given in Appendix C.

Fig. 4.26 shows results from the optimization scheme. The thickness and aspect ratio are varied and the maximum drive frequency, maximum power output, efficiency, and minimum voltage for the desired force output are plotted.

We see a trend for higher GCA drive frequencies with smaller aspect ratios in Fig. 4.26(a). Fig. 4.26(b) shows mechanical power output density. The mechanical power density is taken as the mechanical energy output multiplied by the maximum frequency and divided by the mass of the array. This is given by

$$Pd_{mech} = \frac{U_{mech} f_{max}}{\frac{2}{3}(T + T_{sub})\rho_{si} A_{array}} \tag{4.47}$$

where T_{sub} is the substrate thickness holding the motor together and rho_{si} is the density of silicon. We are assuming that $T_{sub} = 2T$ for structural rigidity of the skeleton and that two thirds of the mass has been removed, abiding by fill rules.

We can see that mechanical power density follows the trend of the maximum frequency. The highest peak in mechanical power density of 570W/kg occurs at approximately 5μm thickness with 20:1 aspect ratio. This corresponds to a 250nm minimum feature size. 250nm lines and spaces are achievable in current Marvell Nanolab technology (ASML DUV Stepper 5500). Achieving this in actuality would require eliminating the approximate 0.4μm undercut in the current DRIE step.

The voltage to achieve the desired minimum force density of 1mN/mm^2 at 20:1 aspect ratio and 5μm thickness is approximately 11V. This is highly appealing for the sake of generating voltage with either solar cells or batteries.

With the current photolithography used on the robot, we can achieve 1μ gaps (assuming reduction of the undercut) in 20μm silicon with a 40V drive voltage. This still produces a respectable 430W/kg power output density.

Smaller thicknesses and higher aspect ratios see significant drops in power output. This is due to the scaling of the gap in the damping term. Since the rear and front gap of the array factor into play damping becomes a significant problem at reduced feature sizes. This can be alleviated by manipulating the size of the rear gap by the scaling term B since $x_b = Bx_o$. The simulation uses $B = 1.5$ in all results. Manipulating B in the optimization can be investigated further.

At larger aspect ratios there are gains in the efficiency term, but very small at less than 0.001. There is also no significant impact on the efficiency with respect to thickness.

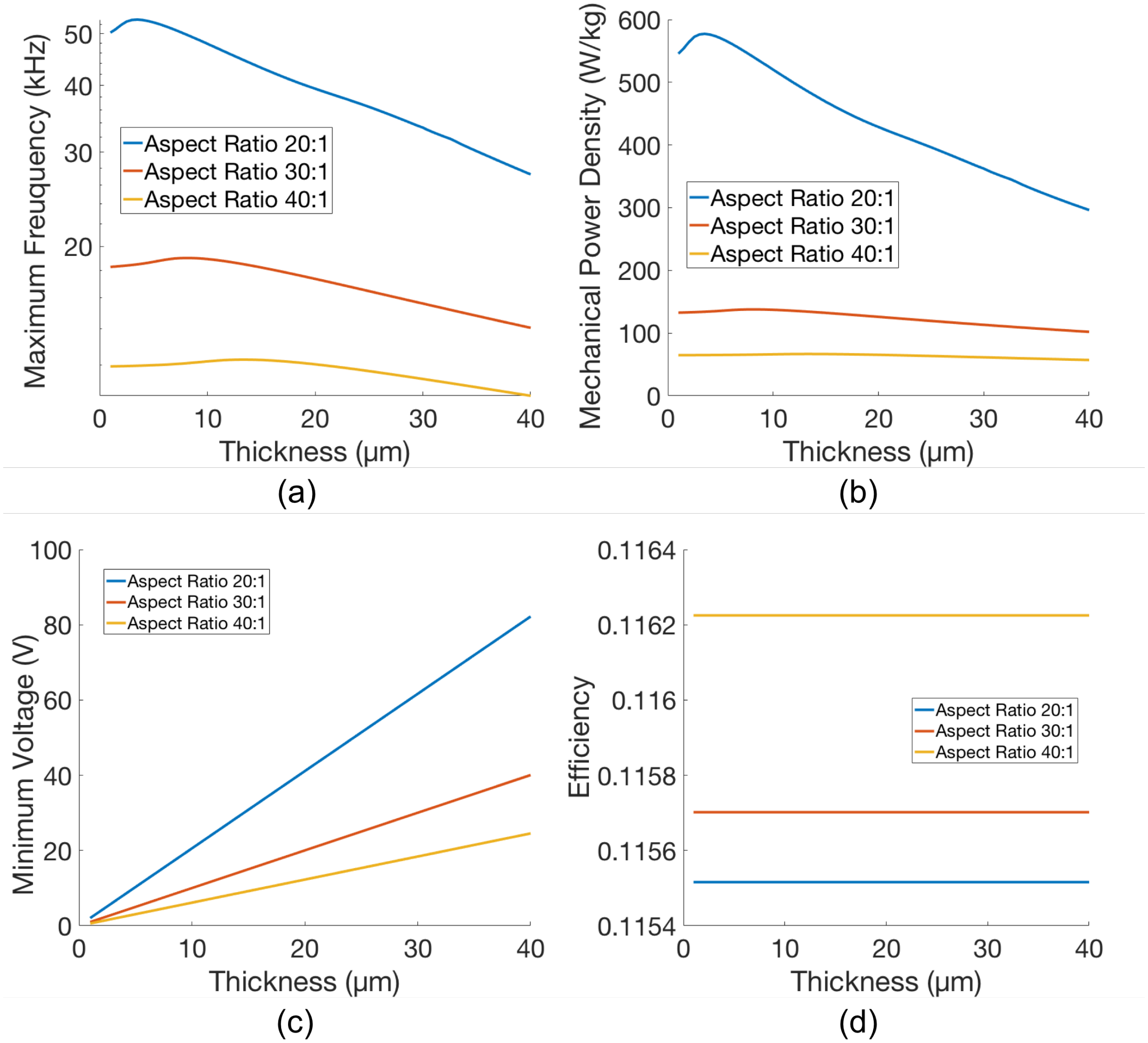


Figure 4.26: Results of the optimization scheme (a) The maximum operation frequency of the GCA arrays for various aspect ratios plotted against thickness (b) Mechanical power density (c) The minimum voltage for the desired force density output (d) The efficiency of the arrays

Chapter 5

The Single-Legged Robot

Silicon pin-joints and inchworm motors described in the previous chapter form the basis of the planar silicon robot leg. The robot leg was designed to achieve two degree of freedom actuation, actuate along the ground when the fabricated chip is flipped upright, and exert enough force to move itself forward. To verify operation of the leg unit cell, a single legged robot was developed.

5.1 Robot Design

The single-legged robot, shown in Fig. 5.1, measures 5mm long, by 6mm tall, by 0.5mm wide and weighs 18mg. Fabricated in a silicon-on-insulator (SOI) process, the robot is based on the previously mentioned electrostatic inchworm motors driving a 2 degree-of-freedom (DOF) planar silicon linkage that acts as the leg. The leg sweeps out an area of approximately 500µm x 500µm off the edge of the chip. To demonstrate walking, the chip is connected to power and control by long flexible copper wires which also act to support the robot upright. The robot exerts over 1.5x its weight in the vertical axis, enough to lift its body and push itself forward.

5.1.1 Linkage Design

The base leg design for our robot is a 5-bar linkage that is operated as a selective 4-bar linkage shown in Fig. 5.2. Two linear motors attach to the linkage as indicated. The linkage has 2 DOF and each DOF is actuated by a single motor at any given time while the other motor holds its attachment point and DOF anchored. In this way the linkage performs full circuit sweeps.

Additional links were included in the final design to isolate the inchworm motor shuttle from the effects of any lateral forces and moments that might be transmitted from the leg, shown in Fig. 5.3, as well as to orient the motors for compactness. The full leg is a 13-bar linkage with 15 rotary joints (pin-joints) and 2 prismatic joints (motor shuttles). At any

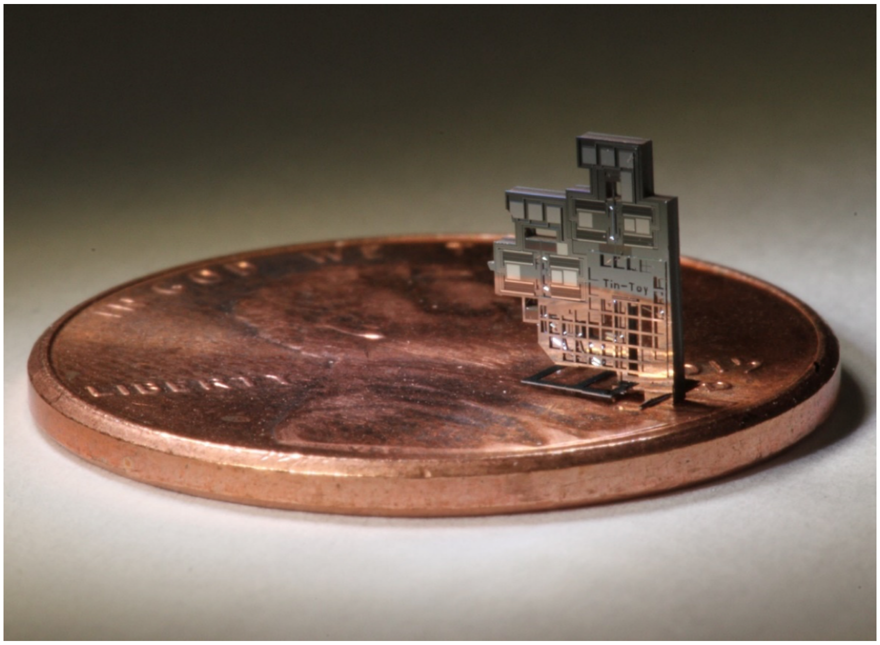


Figure 5.1: A photo of the robot standing upright on a US penny. The robot is held upright by microassembled support feet slotted into the foot and body of the robot

given time, when one motor is fixed the leg operates as an 8-bar linkage with 10 joints. The full linkage assembly can be seen in Fig. 5.4.

Fig. 5.5 shows a full fabricated robot with the motors and linkage highlighted. The actuation cycle of the robot is also shown with (1) the starting neutral leg position (2) the full vertical actuation sweep (3) the full horizontal actuation sweep and (4) the horizontal axis motor releases the leg and resets the vertical position. The horizontal axis motor releases at this point and the leg starts at position (1) once more.

Linkage Loss

Force output of the motors through the linkage was measured using spring-based vernier gauges fabricated alongside the structures. The vernier gauge is designed on a flexible support assembly such that it can be moved with probe tips so that the linkage is able to push on the gauge at various positions.

A diagram of the setup is shown in Fig. 5.6. The force output was measured at 50µm increments from the legs rest position. The linkage has a sweep of approximately 500µm in either direction and force measurements were taken to within 50µm of the max deflection. The leg pushes on the gauge until it stalls. This maximum displacement is read off the gauge and related to the output force by the spring constant of the gauge. All spring gauges are calibrated using comb drive resonators, as described in section 3.2.7.

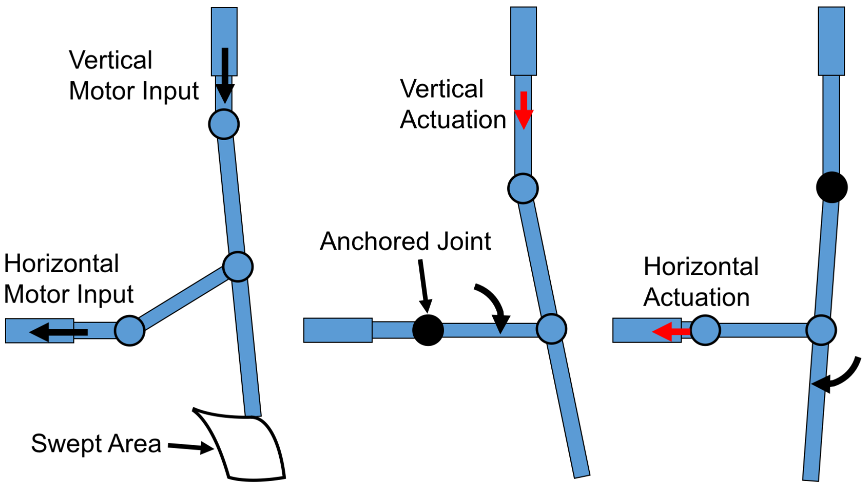


Figure 5.2: An illustration of a simplified leg's sweep and showing the actuation phases of each of the motors

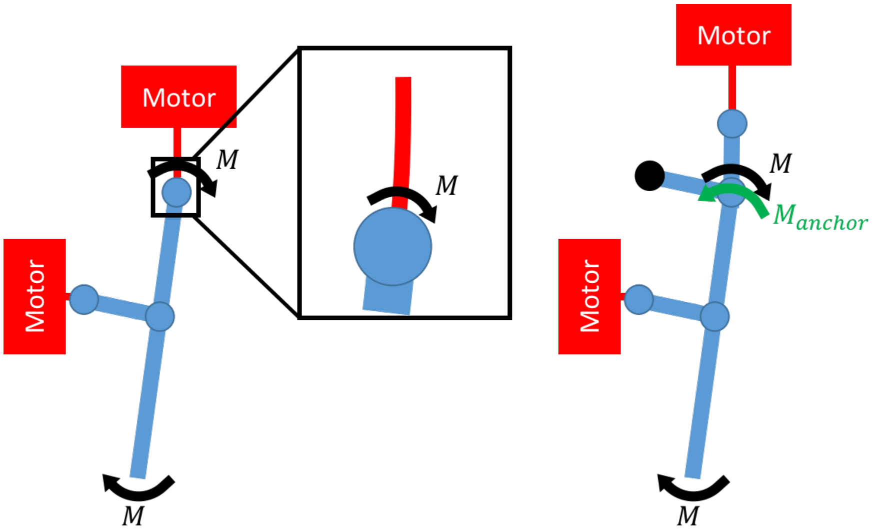


Figure 5.3: Diagram showing additional link included in the linkage to counteract external moments on the tip of the leg

Force output from the linkage in the vertical and horizontal axes is shown in Fig. 5.7 along with the modeled loss as the solid blue line. We see a drop off in force output as the leg extends in either axis. The loss is attributed to extension of the serpentine support springs, rotation of the pin-joint support springs, friction within the joints, and inherent mechanical disadvantage in the linkage.

The modeled loss is taken from a MATLAB Simulink simulation of the linkage. The simulation takes into account the extension of the serpentine springs attached to the inchworm

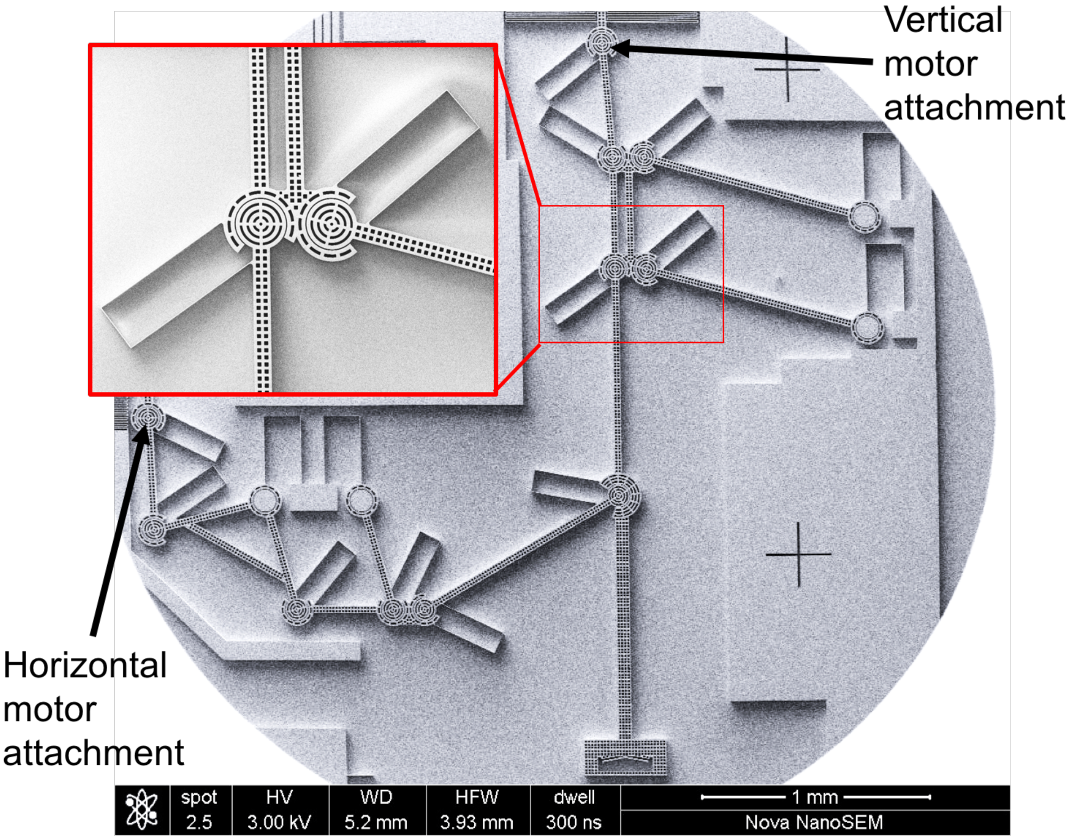


Figure 5.4: SEM micrograph of the linkage, highlighting the pin-joints used in the design and the motor attachment points

motor shuttle, approximately 0.3N/m, and the rotational stiffness of each joint, approximately 10^{-9}Nm. The rotational stiffness is taken from analysis and simulation given in Sec. 2.1.3. The combined effect of each stiffness is measured at each input of the simulated linkage during motion of the linkage and is subtracted from the raw motor force data. A majority of the modeled loss is the serpentine springs, indicated by the fact that the slope of the modeled line is close to the spring constant.

From the measured data of the vertical axis at the end of travel we see approximately 70% loss of force from the raw motor force when transmitted through the linkage. This is much larger than the 20% loss predicted from the model. This 40% additional loss is potentially from friction in the joints and friction between the linkage and the substrate. The stiffness of the joints could also be higher than the analytical model predicted. It is important to note that above 65V the leg is able to pick up the predicted mass of the robot.

From the measured data of the horizontal axis at the end of travel we see 75% loss. The model predicts about 63% loss. A majority of loss from the linkage's the mechanical advantage of half along the horizontal axis. There is better agreement with the model

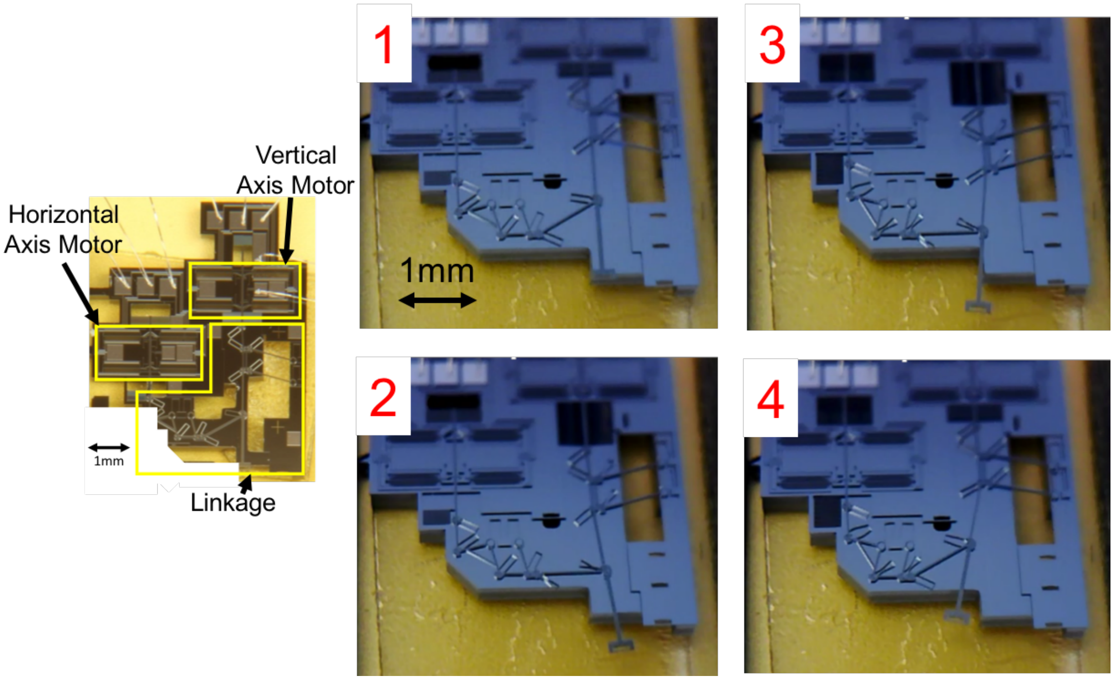


Figure 5.5: A diagram of the single-legged robot wirebonded to a chip package (left) The packaged robot sweeping it's leg through the actuation cycle. The end of each phase is shown in numerical order

compared with the vertical axis travel. Again, additional losses could come from friction of the linkage to the substrate and internal friction in the joints.

5.1.2 Microassembled Support

After fabrication, the robot leg is very fragile during handling and during actuation. Also, since it is single-legged, the robot is not statically stable. Several steps had to be taken to show the robot walking and support it upright.

Planar Stability

The silicon pin-joints are sensitive to out-of-plane forces above 100μN (See Section 2.1.3). Higher forces are enough to pop the pin-joints out of plane, disassembling them. In order to mitigate this issue, a protective screen is glued on top of the linkage using silver epoxy. This screen is bonded well enough to hold the linkage in plane. Layout for the protective screen as well as an assembled robot with a screen glued onto the linkage is shown in Fig. 5.8.

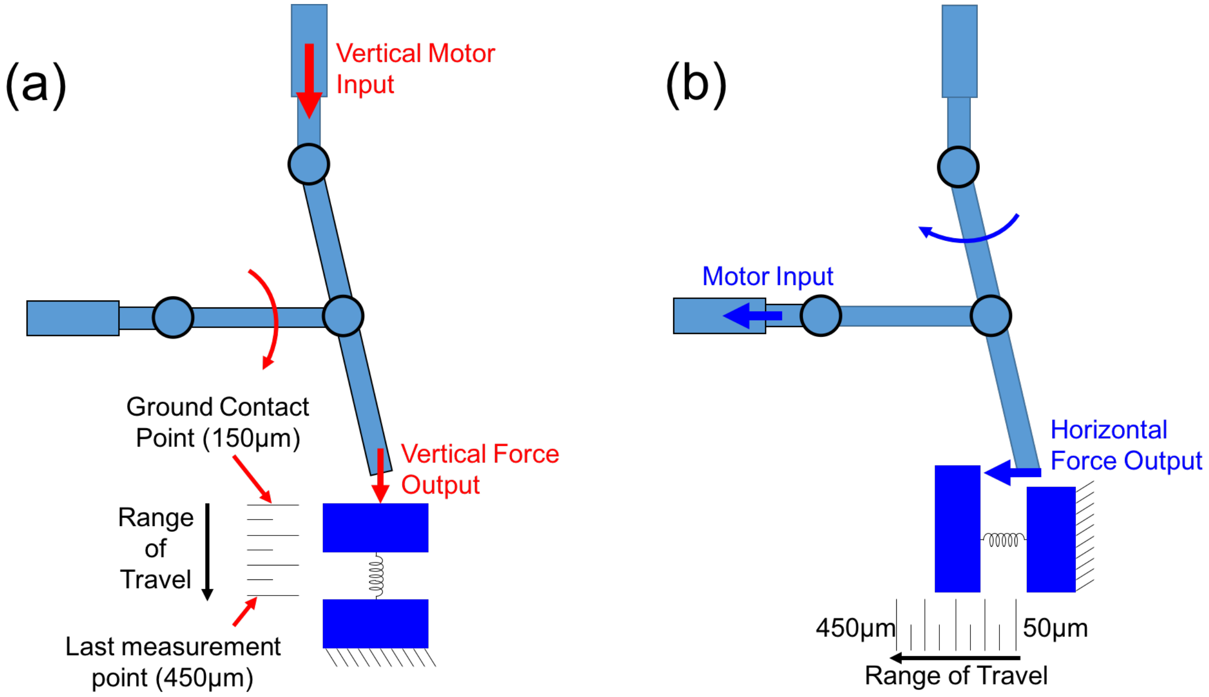


Figure 5.6: A photo of the robot standing upright on a US penny. The robot is held upright by microassembled support feet slotted into the foot and body of the robot

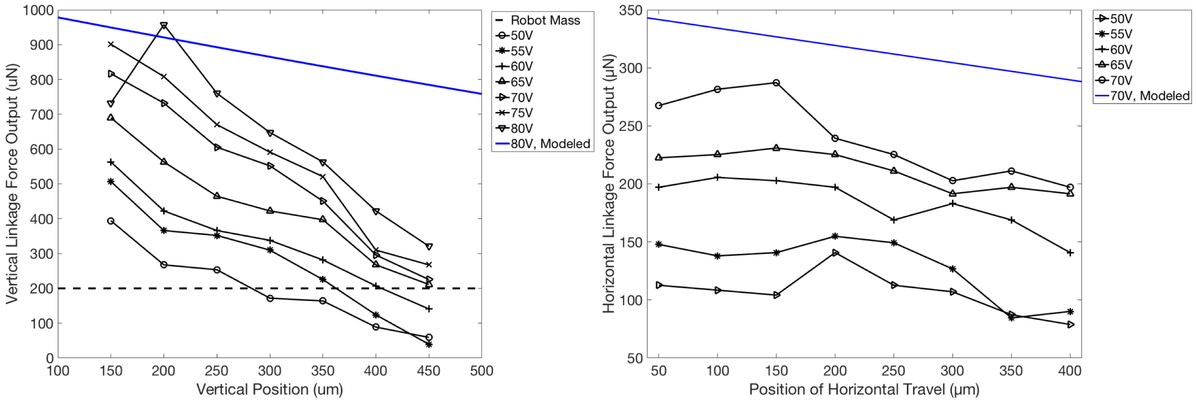


Figure 5.7: Force output of the linkage measured in the vertical axis. The dashed line indicates the mass of the robot (left) and in the horizontal axis (right) for varying voltages across the range of travel of the linkage. The solid blue line indicates the expected value at 80V (left) and 70V (right), factoring in the raw motor force output and mechanical advantage and linkage losses

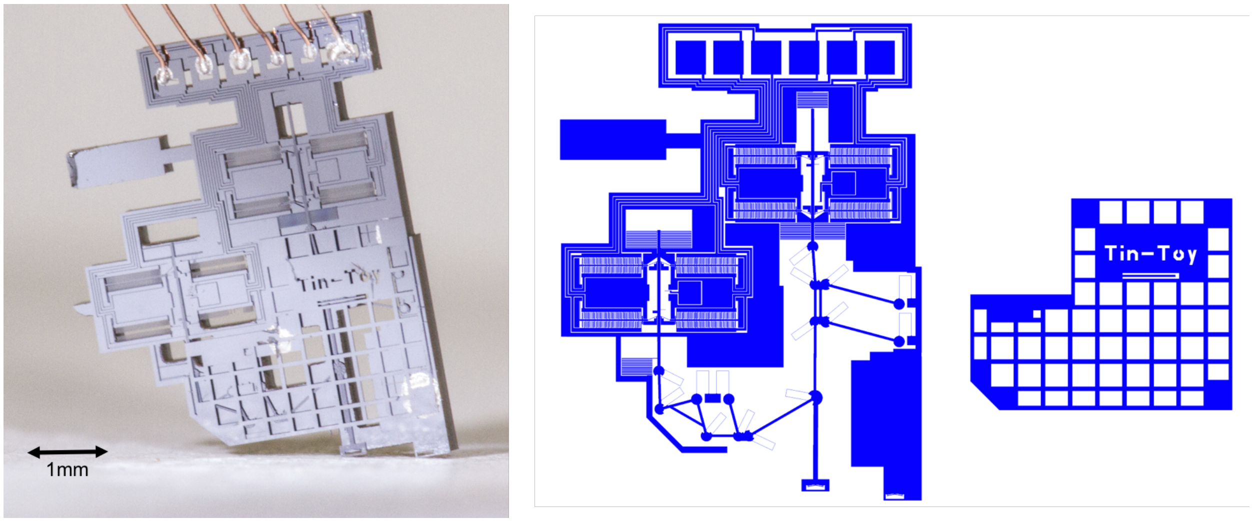


Figure 5.8: The robot with the protective screen glued over the linkage using silver epoxy (left) Layout of the linkage and the screen, showing how the two are oriented together (right)

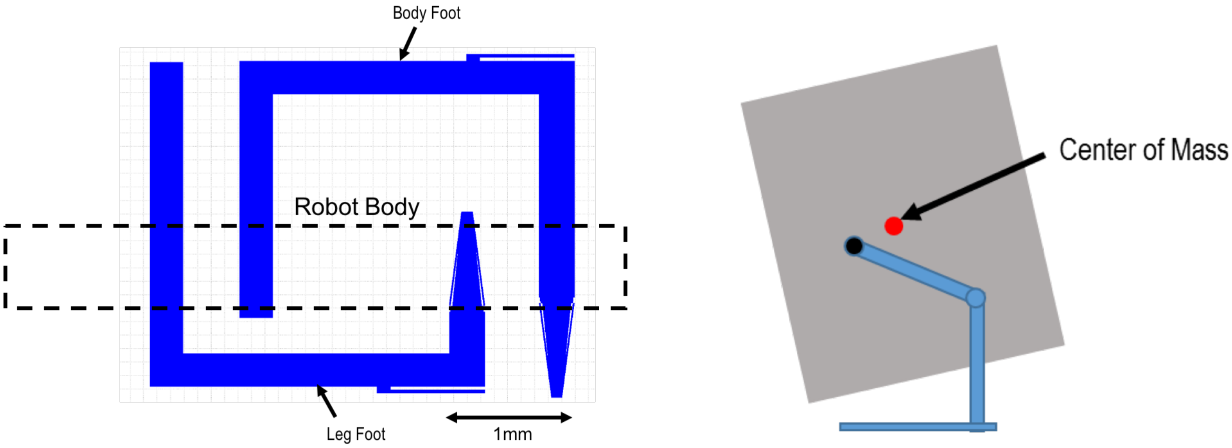


Figure 5.9:

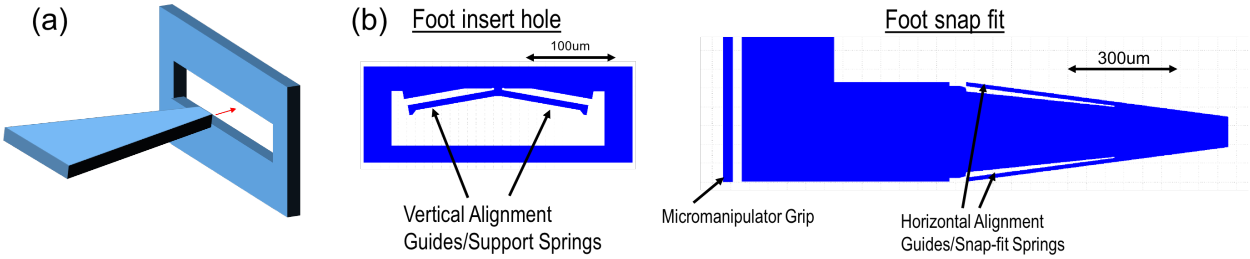


Figure 5.10: (a) A cartoon diagram of the insertion, with a tapered foot entering a hole (b) The insertion hole on the body of the robot with guiding cantilevers and the foot snap fit, meant to interface with the hole. A micromanipulator grip cantilever is used to hold on to the piece and manipulate it with a probe tip

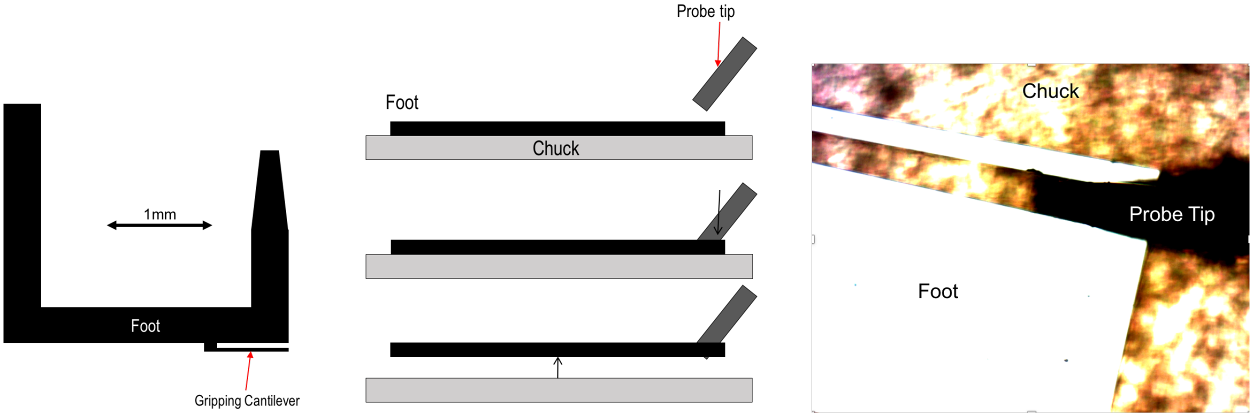


Figure 5.11: A diagram of the gripping cantilever, showing how the piece is lifted off of the chuck. The rightmost image shows a probe tip deflecting the gripping cantilever to lift the piece

Support Feet

To hold the robot upright during its walk cycle, microassembled support feet were designed. A diagram is shown in Fig 5.9. The assembled feet are made from device layer silicon. The size of the feet is meant to maintain the center of mass of the robot over the feet during each phase of its walk cycle. One foot is inserted into the end of the foot and another foot is inserted into the body of the robot. The assembly methodology, which involves snap fit springs and suspension was inspired by work shown in [32].

A detail of the assembly pieces is shown in Fig. 5.10. The insertion hole is fabricated on the foot and body of the robot. Vertical alignment guides are meant to guide the foot into place and deflect to hold the foot in place with the cantilever spring force. The springs are designed to exert 2mN of force at their full deflection after assembly. With a very low

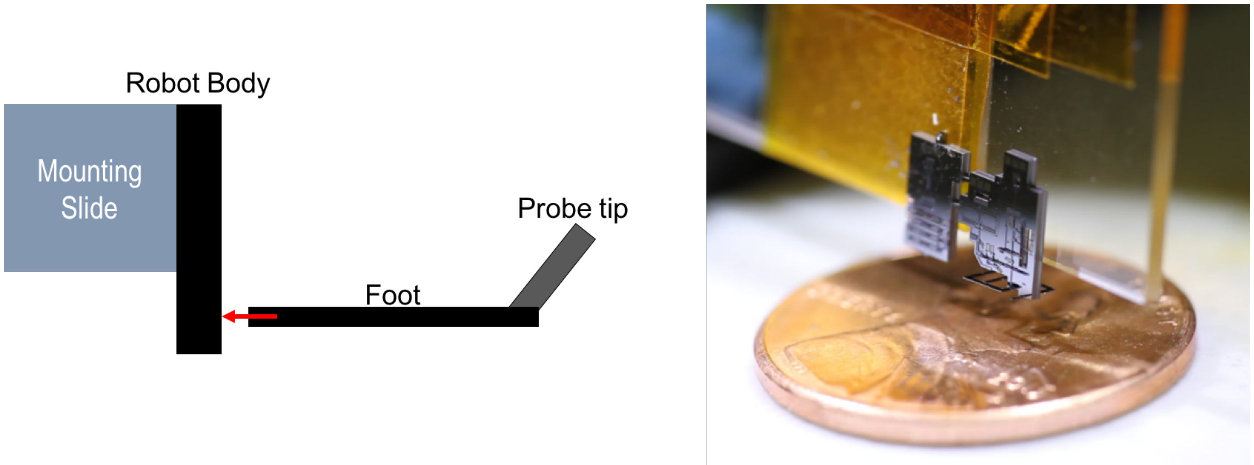


Figure 5.12: The robot is mounted onto a glass slide using kapton tape. This orients the hole for proper assembly. A robot with inserted feet is shown on the right. Once the piece is attached to the probe tip, the foot is aligned under the probe station microscope to insert it into the hole

friction coefficient of 0.1, this would hold the legs in place with over 200g's of acceleration on the foot.

The insertion end of the foot has horizontal alignment guides that deflect during insertion. Once the insertion is complete, these springs are released and lock the foot into place within the insertion hole. This end is also tapered to allow for easy alignment and insertion. The end of the foot begins at 80µm and ends at 300µm. The foot also has a cantilever grip that is meant to interface with a probe tip on a micromanipulator stage. This is how the foot is manipulated. Fig. 5.11 shows how the piece is lifted using a probe tip and manipulated. The cantilever grip also exerts 2mN of force at full deflection, holding onto the probe tip for over 200g's of acceleration on the foot.

Fig. 5.12 shows the mounting of the robot to orient the insertion hole during assembly. The foot is manipulated to align the parts using the probe station microscope. For the vertical axis, alignment is achieved using a 45° mirror.

Once the parts are aligned the foot is inserted into the hole. Fig. 5.13 shows the foot during insertion. The view is through a microscope, top down onto the robot, so the foot insertion hole is oriented 90° with respect to the image. The foot is inserted until the springs are fully deflected. Once the part is fully in place, the foot snaps into place. The probe tip is lifted and slips out of the gripping cantilever.

Fig. 5.14 shows the side view of the robot, seen through a 45° mirror. The deflection of the vertical guide springs can be seen, holding the piece together, along with the horizontal guide springs which have relaxed and locked the piece in place. The robot is seen standing upright on a US penny with it's support feet.

Although the feet could support the robot, attempting actuation with the feet proved

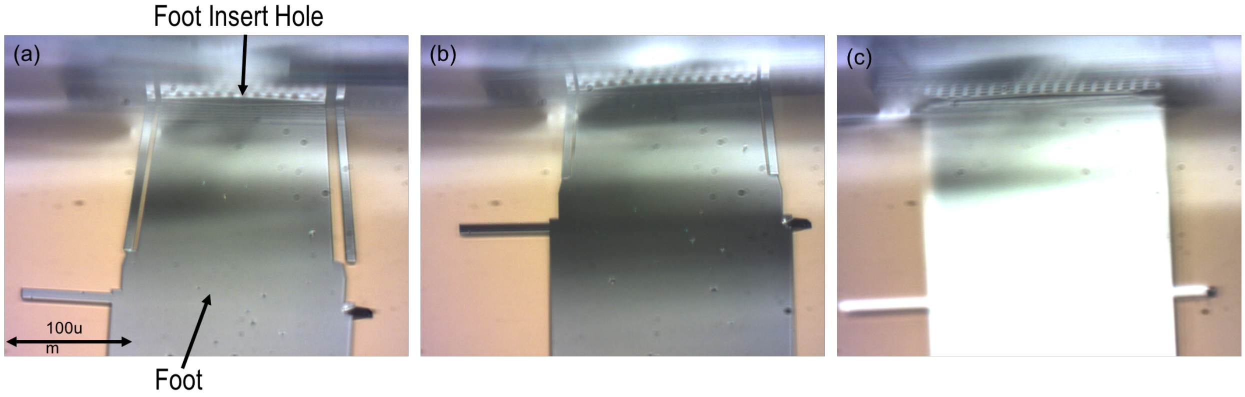


Figure 5.13: The process of insertion, seen from top down. The springs of the foot piece are seen deflecting during insertion

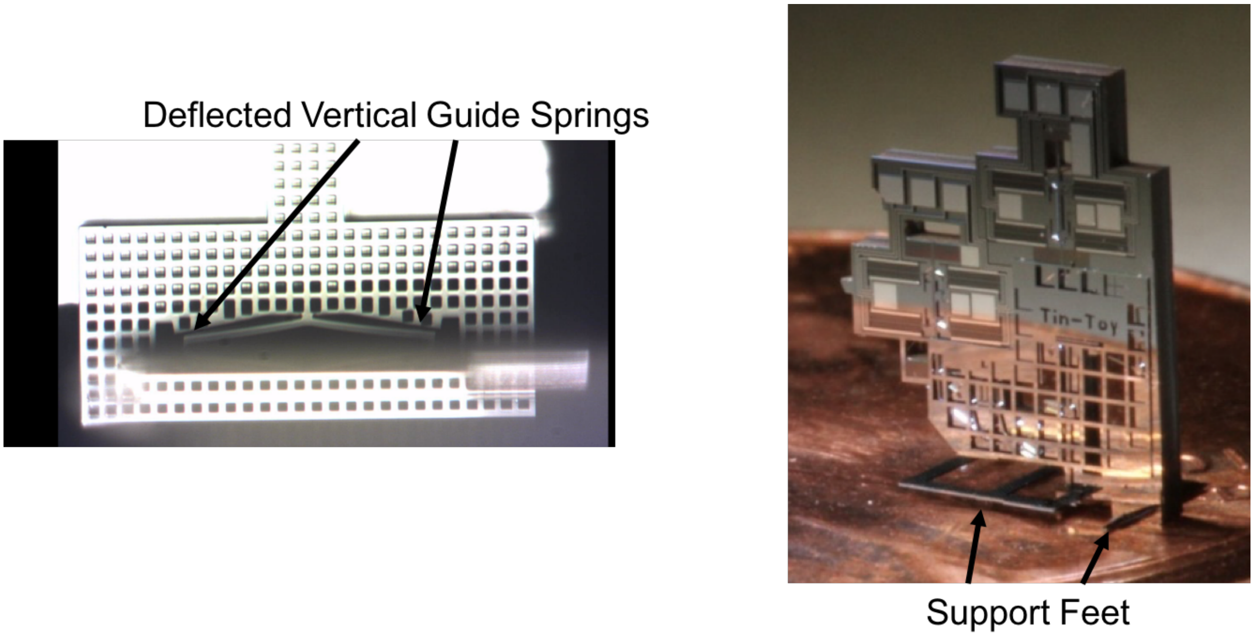


Figure 5.14: The hole with a foot inserted, showing the guide springs deflecting (left) The robot standing upright with its support feet (right)

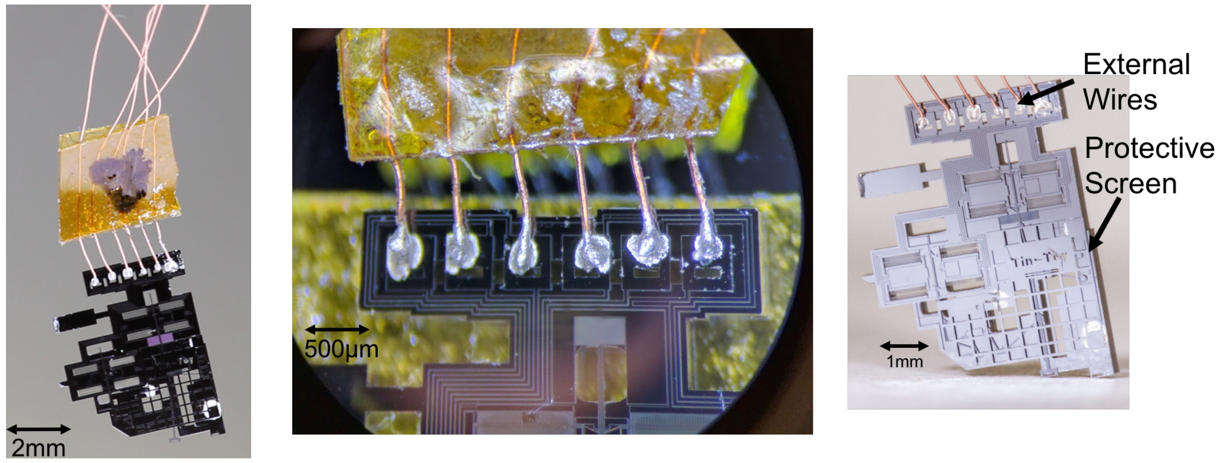


Figure 5.15: The robot suspended by its external wires (left) A closeup of the pad array and wires on each pad, attached with cured silver epoxy (center) The robot standing upright supported by the wire array (right)

difficult. In order to actuate the robot it was maintained upright by wires, described in the following section. The stiffness of the wires and light weight of the robot made accurate positioning without breaking the feet, and by extension the leg, difficult.

Wiring

The robot was tethered to external power and control using flexible insulated 60µm copper wires. Fig. 5.15 shows the wiring of the robot. An array of the wires is assembled by hand, spacing the wires appropriately with respect to the pads of the robot. A piece of paper with lines drawn at 600µm spacing are used to align the wires during assembly. The wires at the other end of the array are silver epoxied to header pins to interface into the breadboard circuit.

Cutting the wire exposes the copper end and conductivity is tested between the ends. The wire's exposed ends are dipped in silver epoxy and the array is placed on a micromanipulator stage using double-sided kapton tape. The array is then lowered onto the robot's pads using the micromanipulator. This is done on a hot plate under a dissection microscope. The silver epoxy is cured on the hot plate once the wires are in the proper position.

The robot is suspended by its wires from a vertical positioning stage. The stage is lowered onto a scale until the scale reads the robots mass of 18mg. The wire suspension is able to hold the robot upright without need for support during its walk cycle.

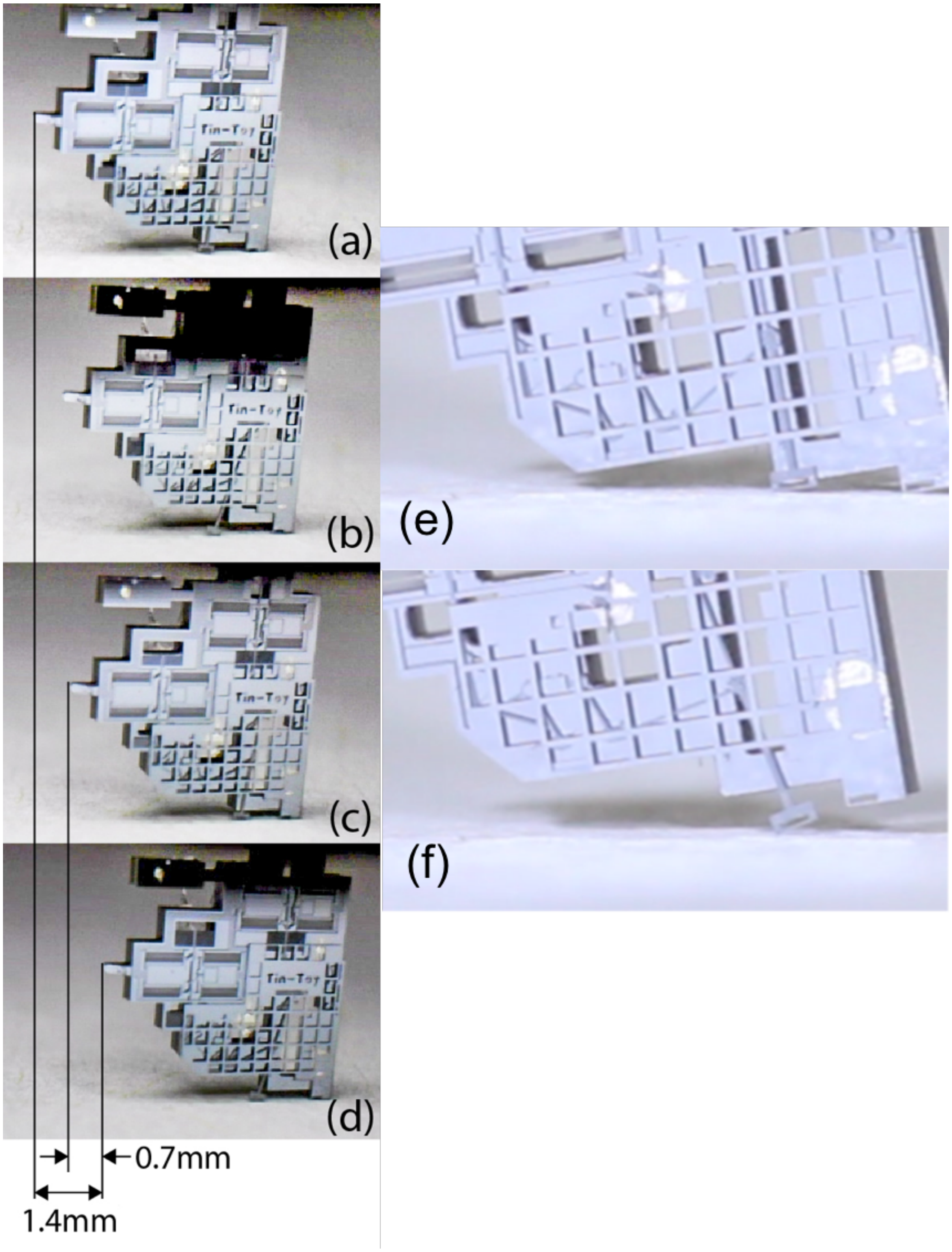


Figure 5.16: The robot walking, suspended by wires and shuffling forward on a piece of paper in (a)-(d). (e) and (f) show the robot extending its leg and lifting the robot's body

5.2 Walking

Fig. 5.16 shows the robot actuating on the ground while tethered. The robot was able to pick itself up with its foot actuating at 80V.

Fig. 5.16(a)-(d) show a sequence of the robot walking. The robot only had the vertical axis motor functioning. The horizontal axis was free to move as the vertical axis motor pushed the leg into the ground. Because of this, rather than lifting its body and taking a stride during its walk cycle, the leg would slip on the ground behind the robot. The robot was able to push itself forward in this manner, against its tethers and mass.

Actuating the motor GCA arrays at 500Hz, the walking speed was 0.7mm/s, about 0.15 body lengths per second. This frequency corresponds to an unloaded inchworm shuttle speed of about 2mm/s. At this speed, a full step would take about 0.5s giving a theoretical walking speed of 1mm/s. The lower measured walking speed could be attributed to the robot not being able to complete its full stride due to force loss. As shown in the previous chapter, a maximum inchworm motor velocity of 0.4m/s has been demonstrated, so the robot can potentially move much faster.

Fig.7 (e)-(f) shows the robot lifting its body. The robot is positioned such that the foot of the suspended robot is in contact with the ground. The leg extends and the robot is able to pick itself up. Occasionally the voltage would have to be raised to 90V for the robot to actuate. This was assumed to be caused by extra friction from the linkage on the protective screen. This robot weighed approximately 18mg.

5.2.1 Power Draw

Capacitance between the actuating fingers and between the drive fingers and the substrate was measured using an AD7746 evaluation board. When closed, the fingers measured 2.1pF. The capacitance to the substrate was about 10.3pF. At a given capacitance C, voltage V and frequency f, the power the motor consumes is given by

$$P = CV^2 f \tag{5.1}$$

Being driven at 500Hz, the total power draw for the robot is about 40μW. The solar cells presented in [13] were able to generate 100μW of power and weighed 2.3mg. Using these solar cells our robot would be able to carry this power source.

5.3 Next Steps

This robot served as the initial test case for a new design platform for silicon microrobots using low power electrostatic inchworm motors fabricated alongside silicon linkages, combined with lightweight energy sources and computation. The goal for this platform is two-fold; to develop robots with a tenable path towards autonomy as well as to create modular components fabricated in a reproducible process that can be widely disseminated. The next improvement to this robot involves adding legs to grant the robot better static stability and make it a better experimental platform with more interesting gaits.

Chapter 6

The Hexapod

Building off of the work on the single-legged robot, a multi-legged robot was designed as a logical next step for this microrobotics platform. The robot shown in this chapter is based on legs that use higher force electrostatic inchworm motors than previously demonstrated. These motors manipulate planar silicon linkages with a longer stride than was demonstrated with the single-legged robot. The robot has a total of 12 motors, 6 linkages, and 12 degrees-of-freedom.

6.1 Robot Description

Fig. 6.1 shows the robot standing upright. The robot is composed of three separate chips; two leg chips and a top hub chip. Using hand assembly, the robot is constructed from these discrete pieces. It is externally wired to a control and power source through pads on the hub chip, which route the actuator signals from the leg chips. The 3D MEMS assembly process successfully routes 24 separate signals to wires for external control. The robot weighs 200mg and measures 9.6mm wide, 7mm tall, and 13mm long and has walked forward under tethered operation.

6.1.1 Walk Cycle Design

A dual-tripod gait was chosen for the robot. This gait was chosen because it is has a large stability margin. At any given time, three points around the robots center of mass are always in contact with the ground.

The designed gait and cyclical leg motion of the legs are shown in Fig 6.2. The colors indicate the phase of each tripod. This gait is hardwired into the robot's routing. Three legs are actuated in tandem while the remaining set of three legs are actuated out of phase.

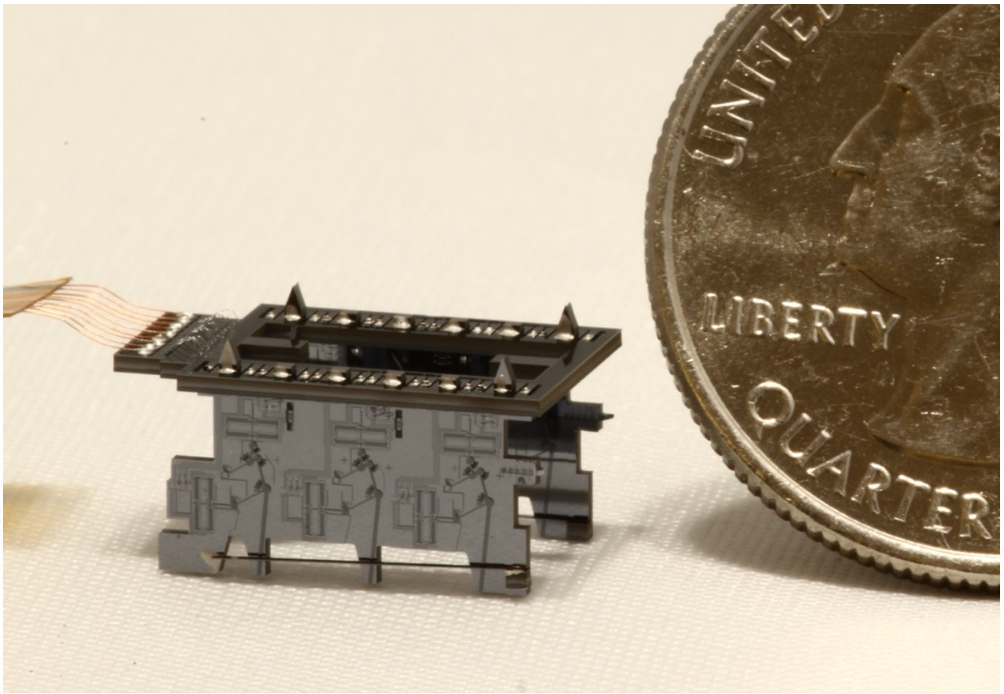


Figure 6.1: A photo of the robot next to a US quarter. The robot is wired to external control

6.2 Robot Chips

The robot is composed of three separate chips. These pieces are fabricated in the same process. This section gives an overview of the chips and the structures used to aid in their assembly.

6.2.1 Leg Chip

The leg chip is shown in Fig. 6.3. The leg chips have two main components, the legs and the assembly/routing pillars.

The assembly pillars are shown at the top of the leg chip. The motor signals are routed through wirebonds and silicon to terminate at the pillars. Each motor requires two high voltage signals and ground, meaning a single leg requires four signals and a ground. Each leg chip routes 12 signals and ground to the pillars. The ground for each motor is routed to a separate pillar, leaving 18 pillars in total.

Leg Design

The leg linkage design is similar to the design shown in [33]. Again, the design is a selective four-bar linkage. The stride length was modified to allow for a longer horizontal sweep when the leg is on the ground. The stride length is 1mm, twice what was demonstrated

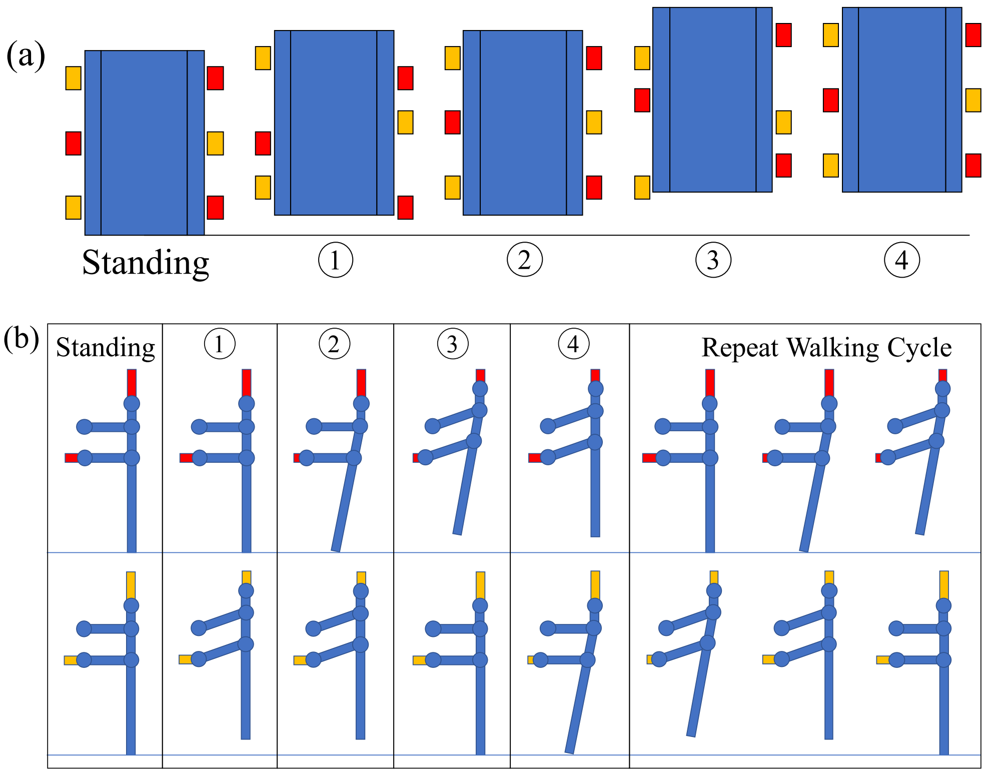


Figure 6.2: Cartoons detailing the intended walk cycle of the robot (a) The movement of all six legs around the robot body (b) The stride sequence of the legs. Orange vs. red indicates separate tripods

before. The linkage also uses newly designed rotary springs. These springs were designed to maintain a constant rotational stiffness through the angular travel of the joint.

Fig. 6.4 shows a diagram of the hexapod's leg. Two motors are attached to the leg linkage; one for vertical axis actuation and the other for horizontal axis actuation. The placement of the vertical axis motor, at $L_{v,input}$, was designed to be in-line with the end of the leg linkage for a 1:1 mechanical advantage. This is done in order to maximize the force output as the robot lifts itself. The horizontal axis motor was placed at $L_{h,input}$ with a 4:1 mechanical advantage, dropping the motors force output in favor of speed during the robots horizontal stride. Each leg has 2 degrees-of-freedom, granting the robot 12 degrees-of-freedom when fully assembled.

Rotary Joints

Fig. 6.5 shows a layout view of the rotary joint spring designed for the hexapod leg. The spring is made of meandering springs that attach to the linkage at a distance r from the center of the joint. The goal for this spring is to maintain a constant rotational stiffness, κ, as the linkage travels through its travel θ. The rotational stiffness, derived in Chapter 2, is

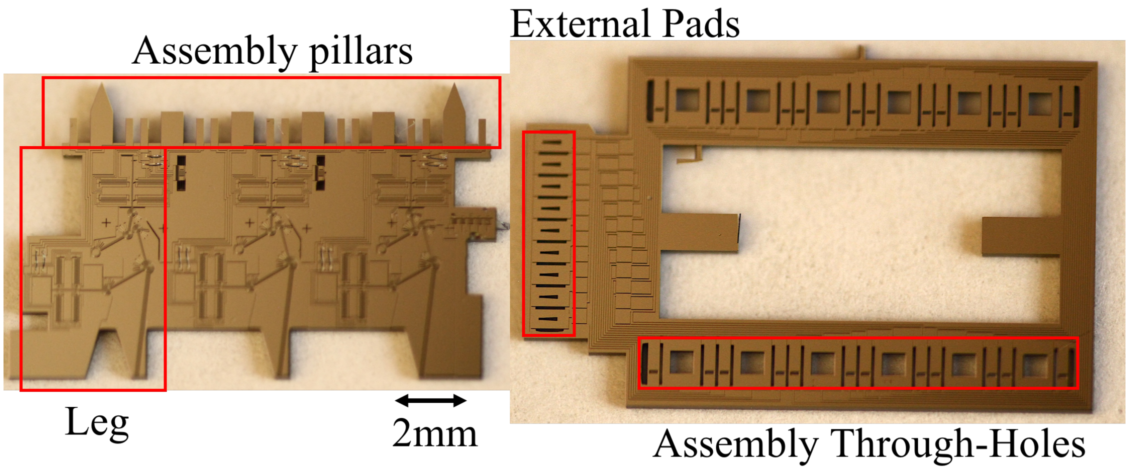


Figure 6.3: Photos of the chips that make up the robot. The leg chip with three separate legs, each with two motors (left) The hub chip used to hold the leg chips together and upright and provide access to external routing (right)

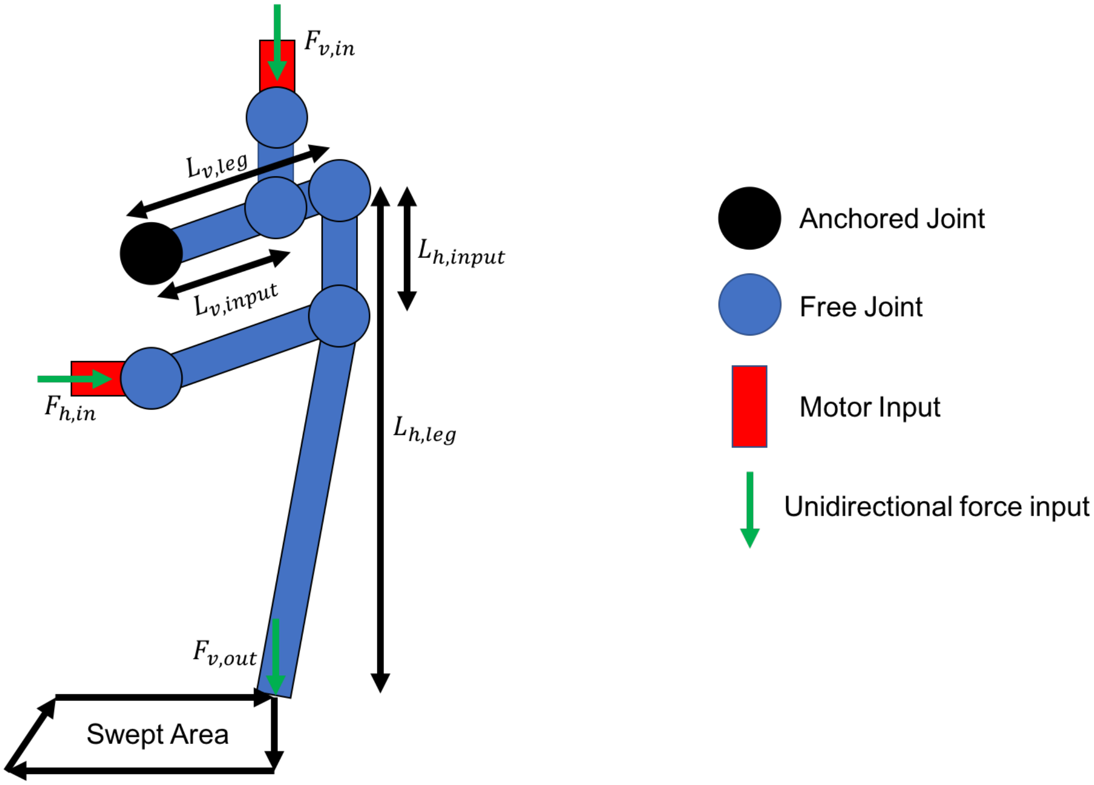


Figure 6.4: Diagram of the hexapod leg indicating mechanical advantage lengths and force inputs

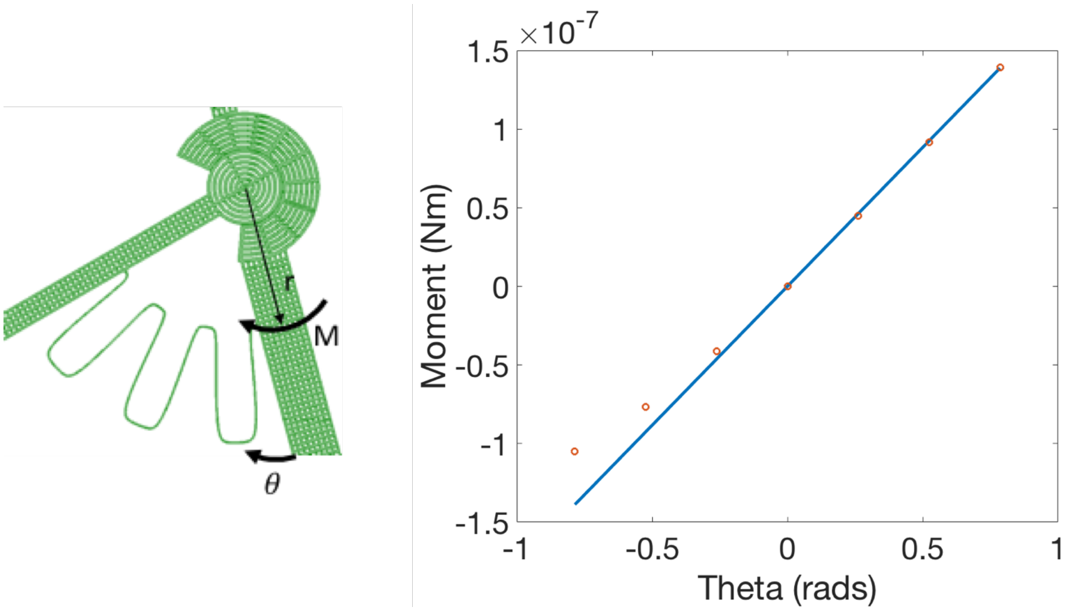


Figure 6.5: Diagram of the hexapod rotary joint spring on a leg joint (left) A plot comparing the analytical stiffness result to simulation in CoventorWare (right)

Table 6.1: Dimensions of the rotary joints used in the hexapod leg

T(µm)	w_s(µm)	N	L_s(µm)	r(µm)
40	3	6	150	140

given by

$$\kappa = k_s r^2 = \frac{ETw_s^3}{NL_s^3} r^2 \tag{6.1}$$

where we have expressed k_s by the linear serpentine spring constant for N beams in series. E is the Young's Modulus, T is the device thickness, w_s is the spring width, N is the number of beams along the length, L_s is the beam length, and r is the springs attachment radius to the joint. Rotary joint springs on the fabricated leg have the parameters in Table. 6.1. The analytical rotary stiffness was 1.6×10^{-7}Nm. This analytical expression is compared to simulation in Fig. 6.5. We see good agreement for positive extension, corresponding to spring stretching, but nonlinearity for negative extension, or spring compression. The actual fabricated springs have a stiffness of 5×10^{-8}Nm when using the analytical expression. This is due to the undercut during the etch reducing the width of the springs.

Linkage Loss

In order to measure the stiffness of the full linkage, a test structure was designed where the linkage was fabricated with spring-based force gauges attached to the point of vertical

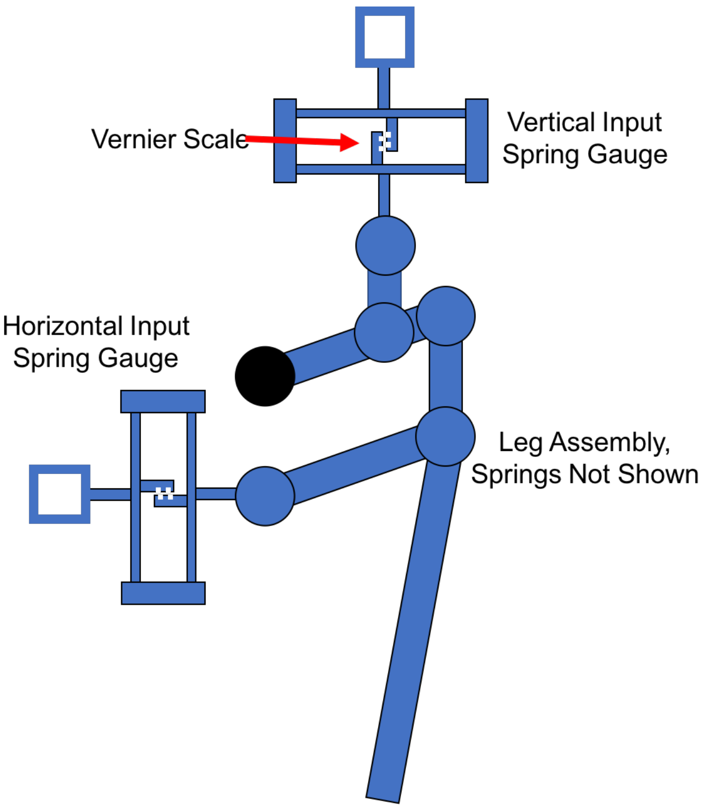


Figure 6.6: Diagram of the structure used to measure loss in the linkage suspension

input and horizontal input of the motor. A diagram of the structure is shown in Fig. 6.6. Spring gauges, similar to those used in Chapter 1 for the pull-out joint tests, are attached to the inputs of the linkage. These spring gauges were moved manually with a probe and the displacement of the spring was recorded and measured with displacement tracking software.

Fig. 6.7 shows the data from these test structures. The linkage was modeled in MATLAB Simulink. The geometry of the linkage was modeled and rotary stiffness was input as a parameter to each joint. The linear stiffness of the serpentine springs on the motors was also modeled on the input point in the simulation.

The simulation was modified to match the worst case slope of the data from the vertical travel. This involved changing the rotational stiffness to 3×10^{-8}Nm in the simulation. For this case 250nJ of energy are lost in the deformation of the linkage suspension for each full leg stride. For comparison, the full motor extension exerts 2.5μJ of energy at 100V.

Most points for the vertical travel data have a lower rotational stiffness than expected. The horizontal travel data suffered from the gauge sticking to sidewalls of the guides that the shuttle was sliding against. This lead to higher measured deformation of the gauge. It should be noted that a negative input force is expected for small values. This is because at the beginning of the horizontal travel phase the vertical travel has already displaced the

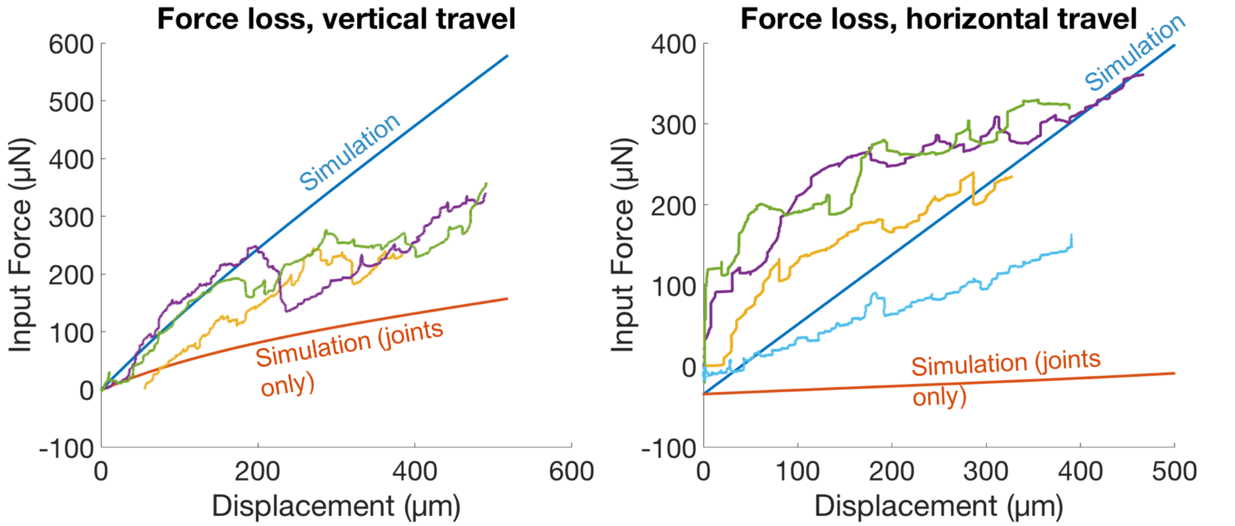


Figure 6.7: Experimental data on the loss and modeled loss of the linkage measured through suspensions around the linkage

linkage and deformed these springs, storing energy. This presents an interesting mechanism for storing energy during one part of a leg cycle, to be used in another.

High Force Inchworm Motors

The raw force output of the motor is shown in Fig. 6.8. In order to increase the force density of the motor to lift the mass of the robot, the motors were designed using a drawn 1µm feature size between the pawls and the shuttle of the motor. The force was measured using a spring gauge attached to the shuttle. The spring gauge was calibrated using an array of comb drive resonators. The total layout area of the motor measures 2.7mm^2. This leads to a higher force density of approximately 1.85mN/mm^2 while actuating at 100V. Normalizing to V^2, this is a 62% improvement over the previous maximum force density of 1.38mN/mm^2 at 110V [23].

6.2.2 Motor Lifetime

The motor shown in Chapter 3 had a lifetime over 28 hours, over 100 million steps for the GCA arrays. The high force motors used on the hexapod showed a much shorter lifetime.

Fig. 6.9 shows the motor displacement for approximately 50 cycles. The motor was programmed to extend the shuttle for the required number of steps to fully extend the shuttle 500 steps for 500µm. The drive voltage and frequency were 60V and 1kHz respectively. From the figure, the degradation in motor performance is obvious. The maximum extension drops significantly a short time into the motors operation. The motor velocity also drops significantly.

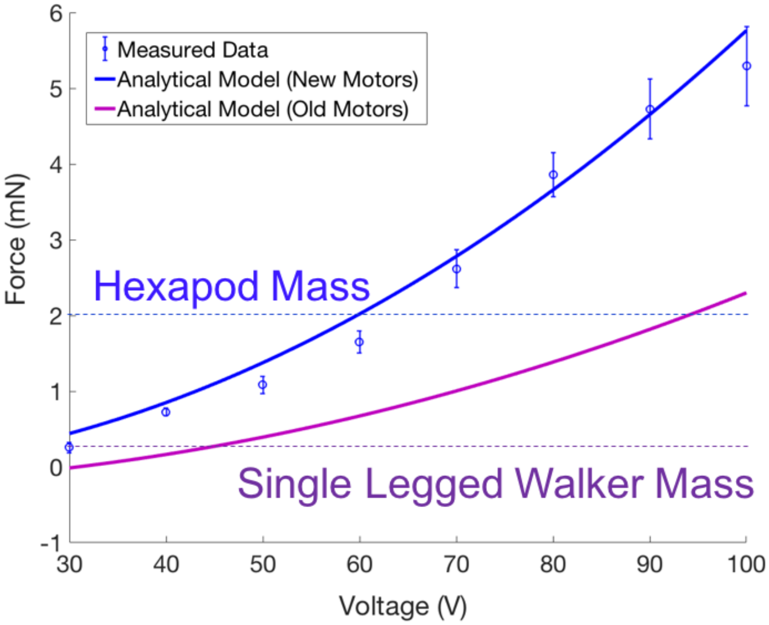


Figure 6.8: Plot of the force output data for the motor used on the hexapod (blue) compared with the motor used on the single-legged robot (magenta)

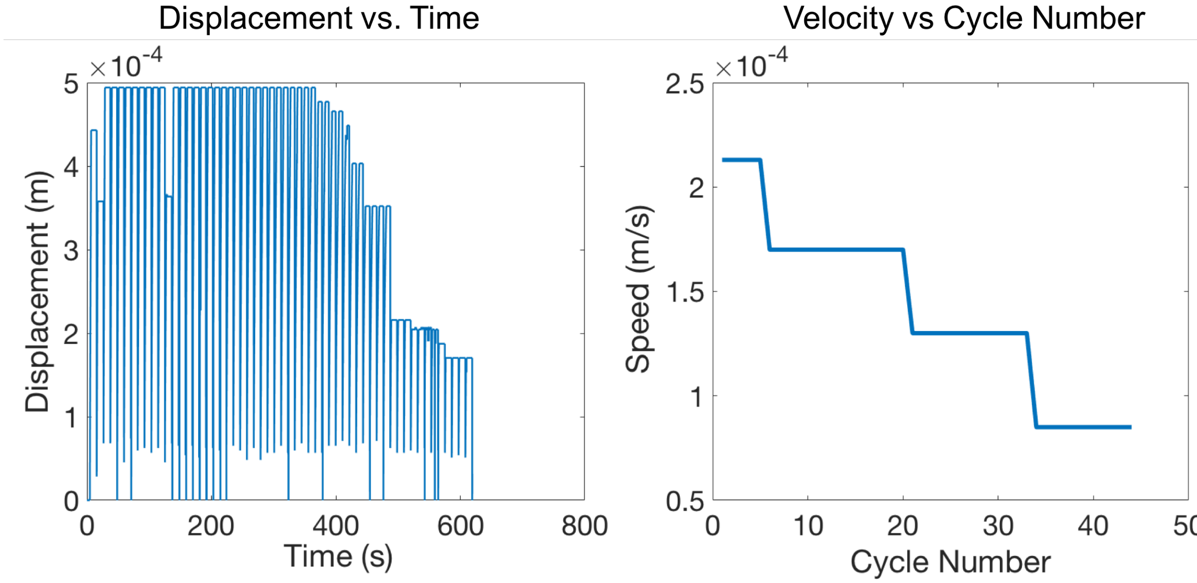


Figure 6.9: Plot showing the lifetime of the high force inchworm motors. Maximum displacement over time is shown (left) as well as velocity as a function of the cycle number (right)

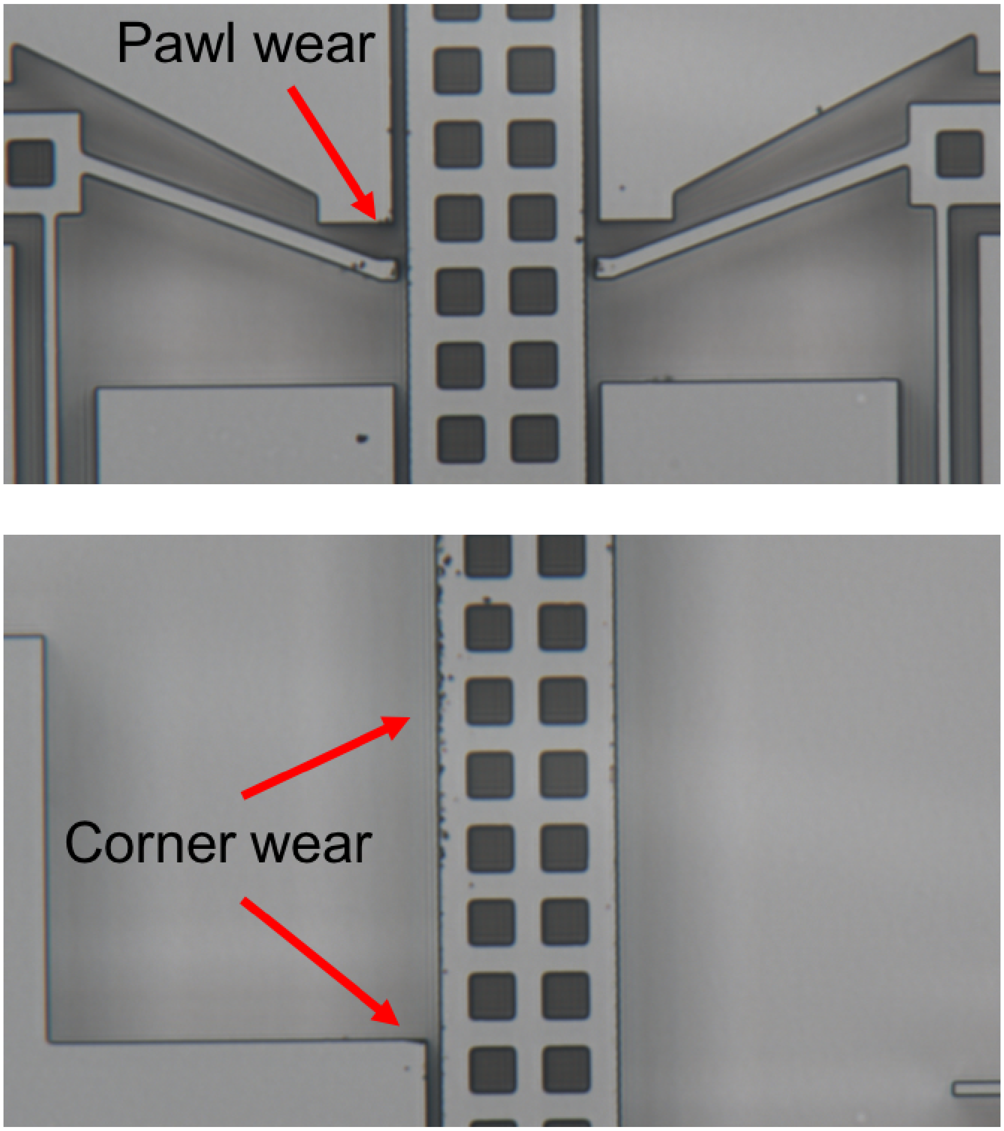


Figure 6.10: Figure indicating wear of the silicon on the pawls (left) and on the shuttle (right)

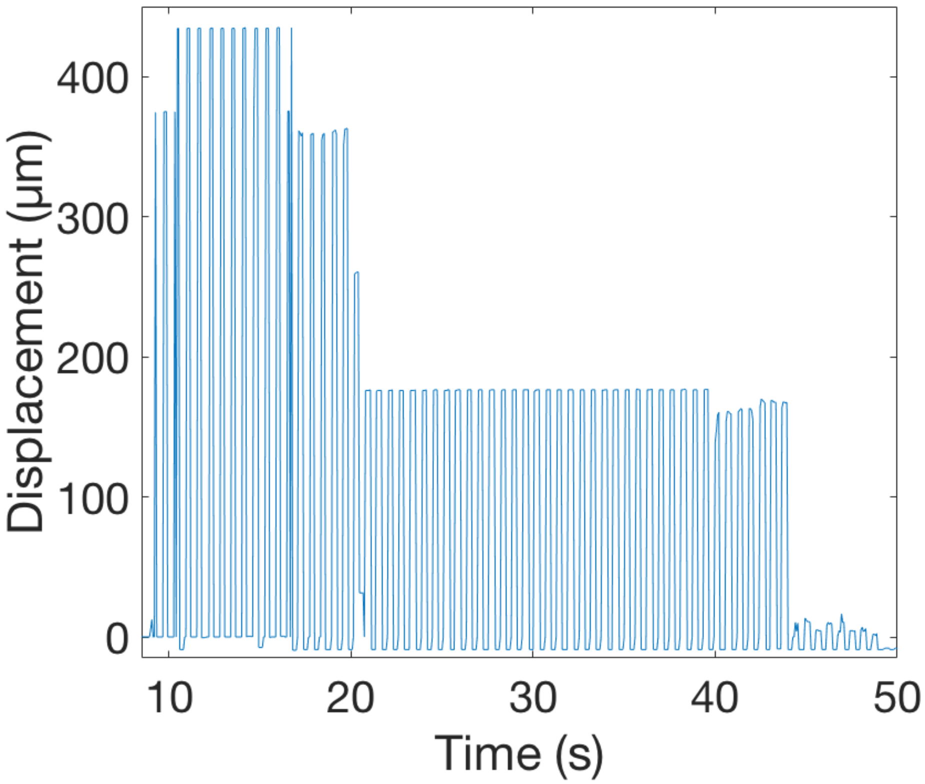


Figure 6.11: Plot showing the lifetime of the high force inchworm motors coated in a sputtered TiN layer

Fig. 6.10 shows the wear on the pawls and the shuttle from operation. The wear on the pawls is attributed to the high force impact. The impact force is on the order of 5mN for these motors. Wear on the shuttle from the surrounding silicon is also obvious.

Titanium nitride (TiN) was proposed as a protective coating for the motors due to the durability of the material. The motors were coated in a 20nm TiN sputtered layer. As seen in Fig. 6.11, there was no improvement over the previously uncoated motors. Operation significantly decreased after 20 cycles.

Wear was still obvious on the TiN coated motors, shown in Fig. 6.12. Degradation and chipping on the pawls and shuttle was still observed. It is likely that a sputtered TiN layer has imperfections that impede successful abrasion and impact resistance. An atomic layer deposition of TiN may be preferable for a more uniform coating.

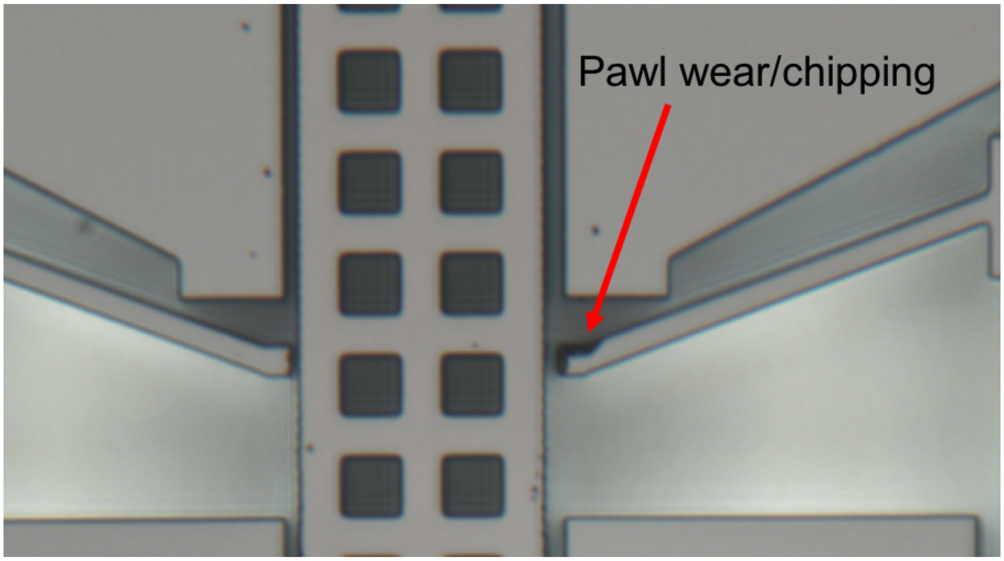


Figure 6.12: Wear is still present on motors sputtered with a 20nm TiN layer

6.2.3 Motor Tooth Measurements

In order to analyze wear on the shuttle and pawls, tooth arrays were analyzed using an Olympus LEXT OLS4000 3D Confocal Laser Microscope. Tooth arrays were fabricated held to the substrate by springs. The arrays were broken and flipped on their sides using a probe tip. The pieces would often stick onto the probe tip due to residue on the tip. This made them easy to manipulate. Fig. 6.13 shows the interface used to analyze the arrays. The array flipped on its side abutted against a sheet of device layer silicon on the same wafer is shown on the left. The array has teeth drawn at 0.25µm, 0.5µm, 0.75µm, and 1µm extrusion from the base of the array beam. True flat is measured against the device layer in the screen. Assuming the other side of the array is laying flat on the surface, the angle of the sidewall facing the camera should be halved.

Projecting the measured dimensions of the teeth, we can see what the engagement of the teeth would actually look like. This is shown in Fig. 6.14. We can see for the shallow 0.25µm used on the motor there is a very small amount of overlap between the pawl and the shuttle when the teeth are engaged. The engagement between the pawl and shuttle for 1.5µm extruded teeth is shown for comparison. This low engagement area could explain the drop in motor performance. As the small area is chipped away, the motor can begin to slip more and more.

Additionally undercut of the tooth arrays was measured, shown in Fig. 6.15. Undercut of the tooth arrays is smaller than in other locations on the mask. This is due to the small feature sizes of the gap between the teeth and surrounding silicon tending to have smaller undercuts throughout the etch. The gap between the arrays and the surrounding silicon was drawn at 1µm.

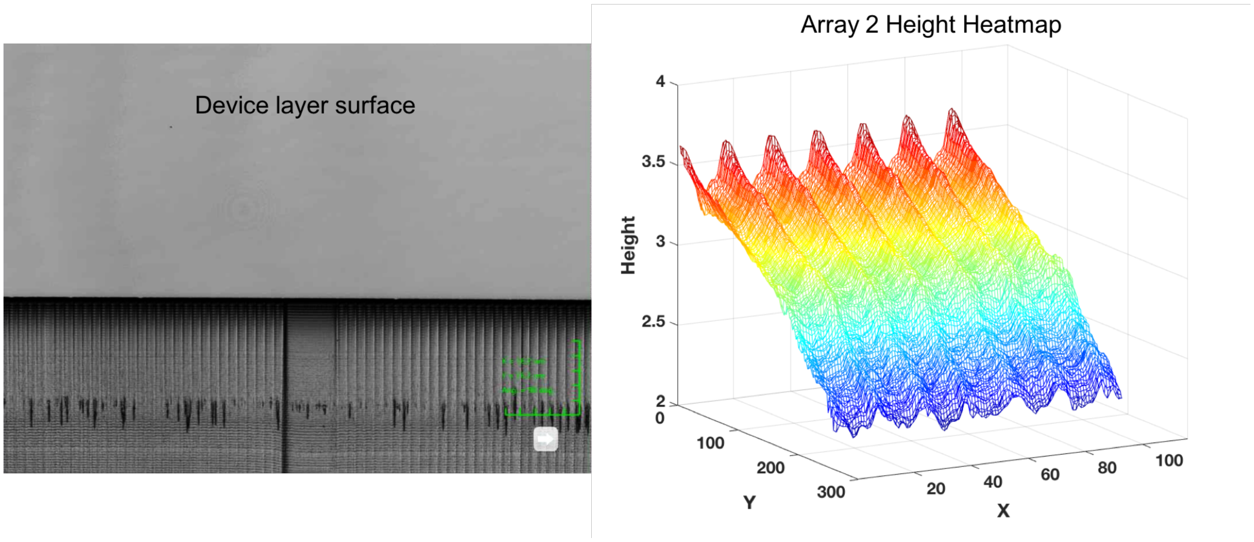


Figure 6.13: The confocal microscope's view of the array (left) and the heat map it produces (right)

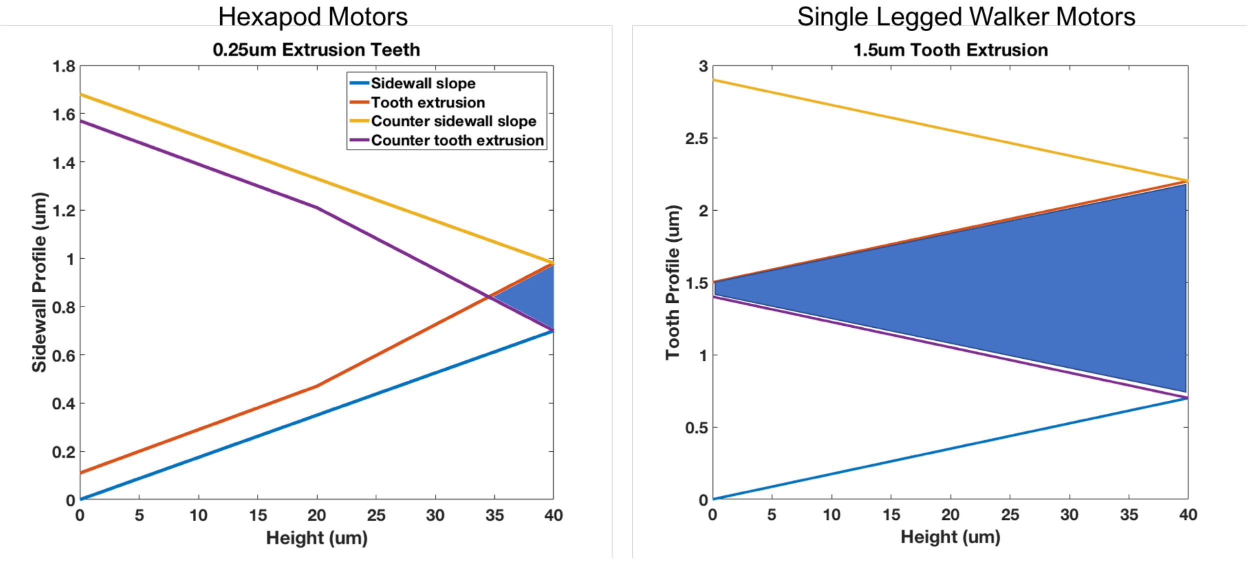


Figure 6.14: Projection of the engagement of the teeth for teeth with 0.25µm drawn extrusion (left) and 1.5µm drawn extrusion (right). The blue area signifies the area of overlap between the gear teeth on the pawl and on the adjacent shuttle

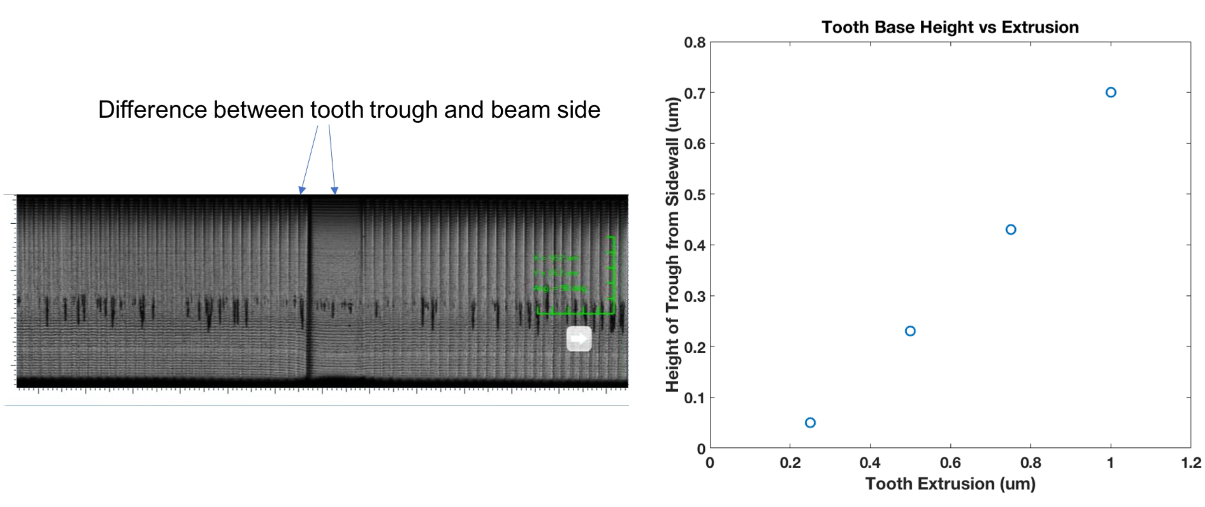


Figure 6.15: Plot of the height of the trough of the tooth arrays compared to the height of the base. The actual tooth undercut inside of the trough can be determined from this

6.2.4 Hub Chip

The hub chip is also show in Fig. 6.3. The hub chip is meant to fit on top of two adjacent leg chips using a pillar/through-hole assembly technique adapted from work shown in [34].

The purpose of the hub chip is two-fold. First, the hub chip is meant to hold the leg chips upright, such that they can actuate on the ground. Second, the hub chip routes electrical signals from the legs on the leg chip to pads that interface externally with long wire tethers. The robot routes the signals from the legs to demonstrate a dual tripod gait on the robot.

There are a total of 24 signals and 12 separate ground signals from the motors on the robot body. These signals are reduced to 8 signals and 4 ground signals by hardwiring the dual tripod gait in the silicon routing on the hub chip. An array of 12 wirebonds are necessary to route the signals to the exposed pads and tie the ground signals together onto one final pad. Thus only 9 wires are required to drive the robot.

6.2.5 Pillar Design and Tolerance

Fig. 6.16 details the design and dimensions of the pillars and the holes. There are three types of pillars used in this design; routing pillars for routing the high voltage signals made up of only device layer silicon, square block pillars made of device layer silicon and substrate silicon for mechanical rigidity and to protect the routing pillars, and triangular alignment pillars to align the holes and to gently guide the hub chip into place on the leg chips.

The alignment and block pillars share the same hole size. The hole on the hub chip is designed to have 10μm of tolerance on a given side around the pillars, factoring in the uncertainty in the thickness of the substrate. For the routing pillars, the tolerance is set to 15μm on a given side and 25μm in the dimension of the wafer thickness to account for

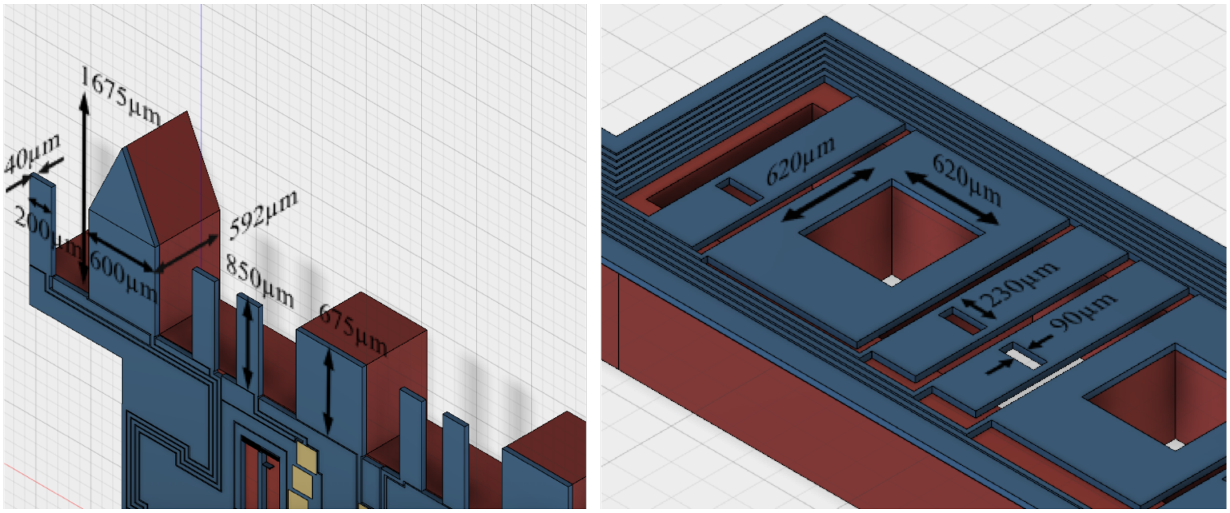


Figure 6.16: Detail on the leg chip pillars and the hub chip holes

uncertainty in the substrate thickness. With this larger tolerance, the alignment and block pillars protect the routing pillars as the hub chip falls into place.

6.3 Assembly

6.3.1 Preparing Assembly

Before assembling the pieces, signals on the leg chips had to be routed using wirebonds on bare silicon. 25µm Al-Si wire on a Westbond 747677E Wirebonder was used. The ultrasonic power was set above 325 (a unitless number internal to the tool) and the time was set to 30ms. Usually the power had to be increased depending on the cleanliness of the silicon surface. Bonds were easiest on bare silicon by using a vacuum chuck to hold the silicon piece down. To keep the piece from rotating, two vacuum holes were covered. Signals were also routed with wirebonds on the hub chip.

Additionally, a silicon bracket to maintain the legs in plane was assembled. The bracket is composed of two large substrate blocks and a thin device layer strip between them. The blocks are glued to the body of the robot using silver epoxy. This is meant to improve upon the protective screen on the single-legged robot's linkage. A small bracket is used to reduce friction that was present on the screen.

6.3.2 Multichip Assembly

The assembly process is shown in Fig. 6.17. In addition to the robot chips, additional structures for the purpose of assembly were also fabricated. This includes an assembly base, used to hold the leg chips upright during assembly and a bracket piece to place over the

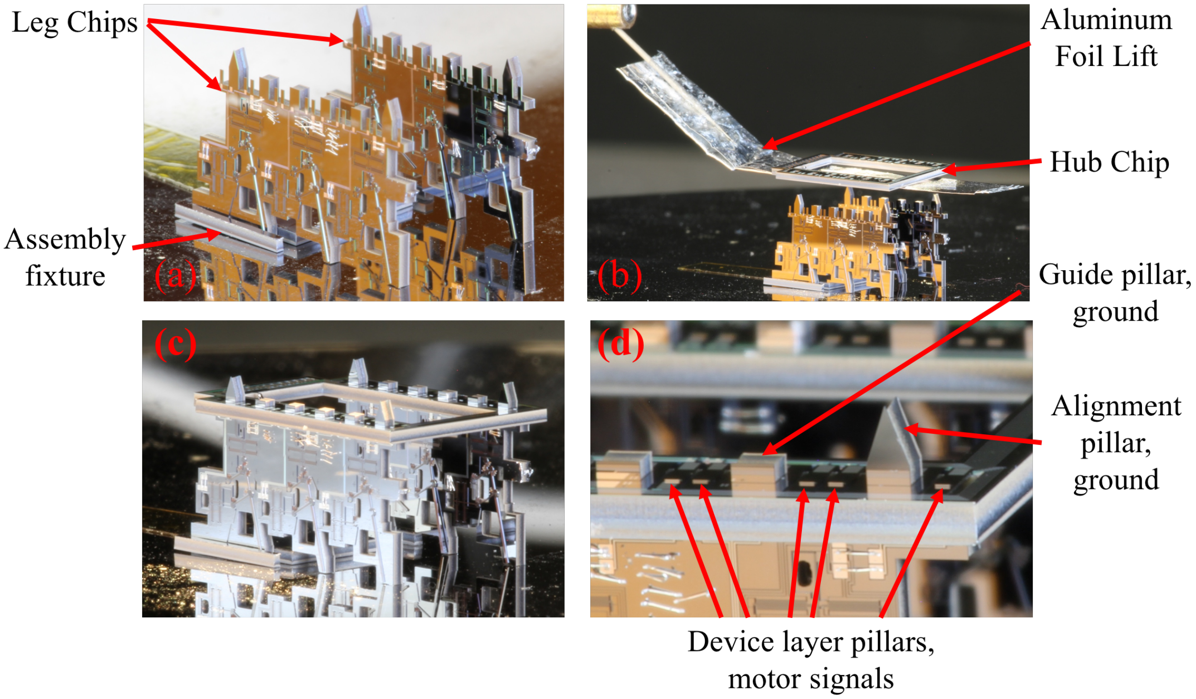


Figure 6.17: Photos detailing the robot's assembly

legs to keep ensure they do not come out of plane during actuation. The base is shown in Fig. 6.17(a) and the leg bracket is shown in Figure 9.

In Fig. 6.17(a) The leg chips are manually placed into the assembly base such that they stand upright. This structure is placed at a probe station where a microscope can be used for alignment of the hubchip for the next step.

In Fig. 6.17(b) a lift made from folded aluminum foil is glued onto a probe tip which is inserted into a micromanipulator for precision in alignment. The hub chip is manually placed on the lift and positioned above the leg chips. Alignment is performed through the microscope. The holes in the hub chip are positioned such that they line up with the pillars on the leg chips.

In Fig. 6.17(c), once the positioning is correct the hub chip is dropped with the micromanipulator to sit on the leg chips. Fig. 6.17(d) shows a zoom-in of the pillars through the holes on the hub chip, detailing the assembly, guide, and signal pillars.

The final assembled structure is brought to a hotplate and is permanently affixed using silver epoxy to join the pillars to the base of the holes. The epoxy also electrically routes the motor signals between legs that make up each of the tripods and to the hub chip pads. The silver epoxy joints on the pillars are shown in Fig. 6.18. The epoxy is dispensed from a Zephyrtronics ZT-5100 Air Mill handheld pneumatic liquid dispensing system. In order to dispense appropriately sized 200-300µm dots of silver epoxy, a pressure of 60psi and a

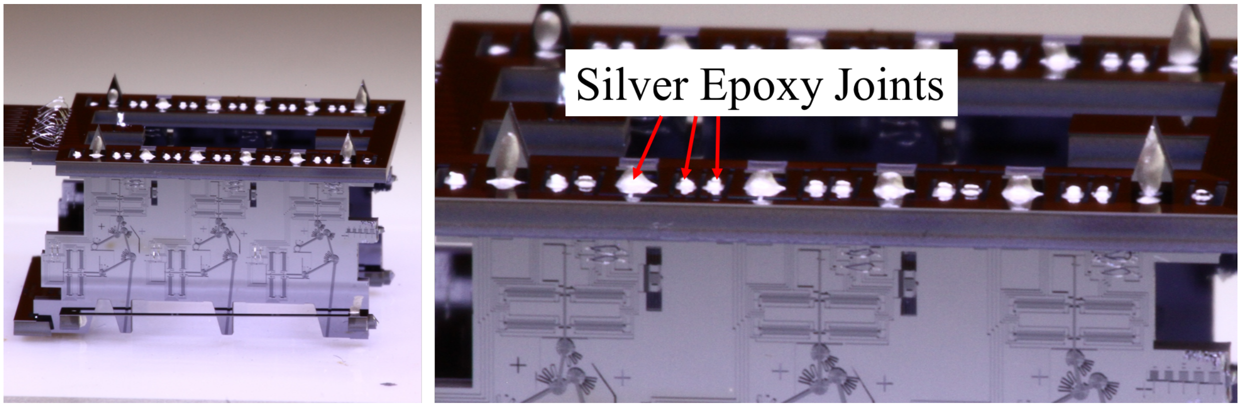


Figure 6.18: Photo showing the silver epoxy joints used to mechanically and electrically connect the pillars to the holes

dispense time of 0.01s were used.

6.3.3 Wiring

The hub chip has an array of pads extending off the rear of the chip. External wires are connected to these pads. Fig. 6.19 shows the process of wiring-up a robot. An array of 9 30cm long 60µm thick copper wires are aligned in an array to the pitch of the pads. The wire array is assembled on a piece of paper with lines printed at the spacing of the pads. Kapton tape is used to hold the array together at the correct spacing. The other end of the wire array is soldered onto a header pin that interfaces to a breadboard with the control circuit for the robot.

As with the single-legged robot, the wire array is trimmed to expose the ends. These ends are dipped in silver epoxy and the array is positioned with a micromanipulator onto the hub chip pads. Visual confirmation of wire contact to the substrate was observed by watching the silver epoxy on the wire adhere to the silicon surface. The whole assembly is assembled and cured on a hotplate at 150°C for half an hour.

6.4 Demonstrating Motion

6.4.1 Experimental Setup

Once the robot is assembled and wired it is placed on a flat surface to walk. Fig. 6.20 shows the experimental setup for the robot. The robot is positioned next to a ruler for scale and to track motion. The legs are acutated at 100V.

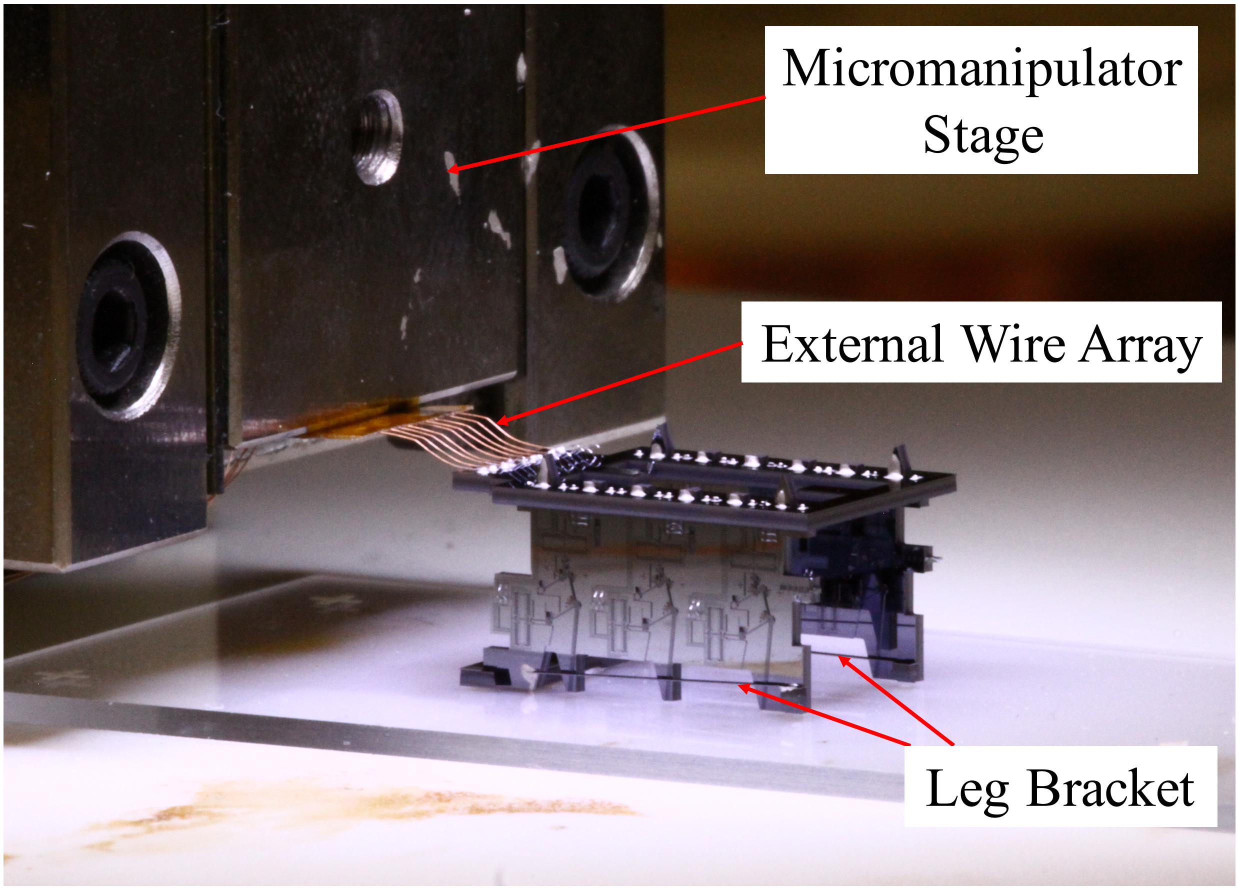


Figure 6.19: Figure showing the wiring process for the robot

6.4.2 Initial Attempts

During initial attempts, the robot demonstrated repeatable unintentional locomotion schemes. When the leg bracket was not put in place the legs of the robot would splay out of plane upon contact with the ground, and would move the robot to the side, like a crab. The legs would return to their initial position without breaking.

6.4.3 First Steps

Fig. 6.21 shows an assembled robot taking steps. The robot takes two full steps in the sequence shown. This robot had three functioning legs and was able to walk forward using two of the legs which were joined in the dual-tripod actuation scheme. The functioning legs are shown in the diagram in the same figure. The actuation voltage was 100V

The velocity of the horizontal stride of each step was measured. The first step was 7.5mm/s and the second step was 1mm/s. The motors were actuated at 100Hz, which translates to a horizontal speed of approximately 1mm/s. The first step was faster than

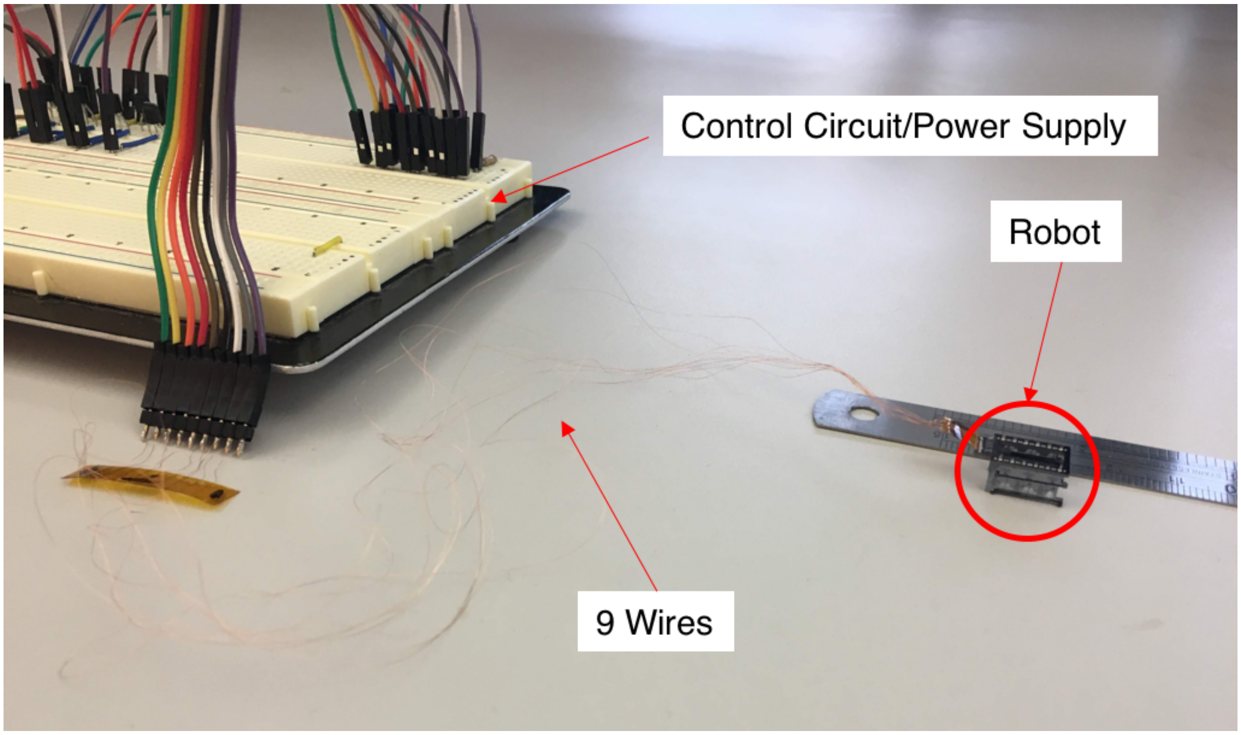


Figure 6.20: Setup used to show the robot taking steps

expected. This could be attributed to the weight of the body carrying the robot forward under the influence of gravity. The robot is designed to drop its mass as it takes its step.

The total capacitive load of a single actuator is estimated to be 26pF when the gaps are fully closed. Operating at 100Hz at 100V this leads to a power draw of 78µW for a single tripod. With top motor speeds of 34mm/s demonstrated in [28] and a 4:1 mechanical advantage, the robot could potentially demonstrate top speeds of over 13.6cm/s.

6.5 System Level Design

6.5.1 Improvements on Hub Chip

The hub chip is meant to support the robot's leg chips. It can also serve as a central base on which computational, power, and peripheral devices may be attached. The current state of the hub chip is free form and the silicon may be patterned to provide more mechanical contacts for MEMS ZIF sockets, flip-chip pads for bonding, or other connection methods.

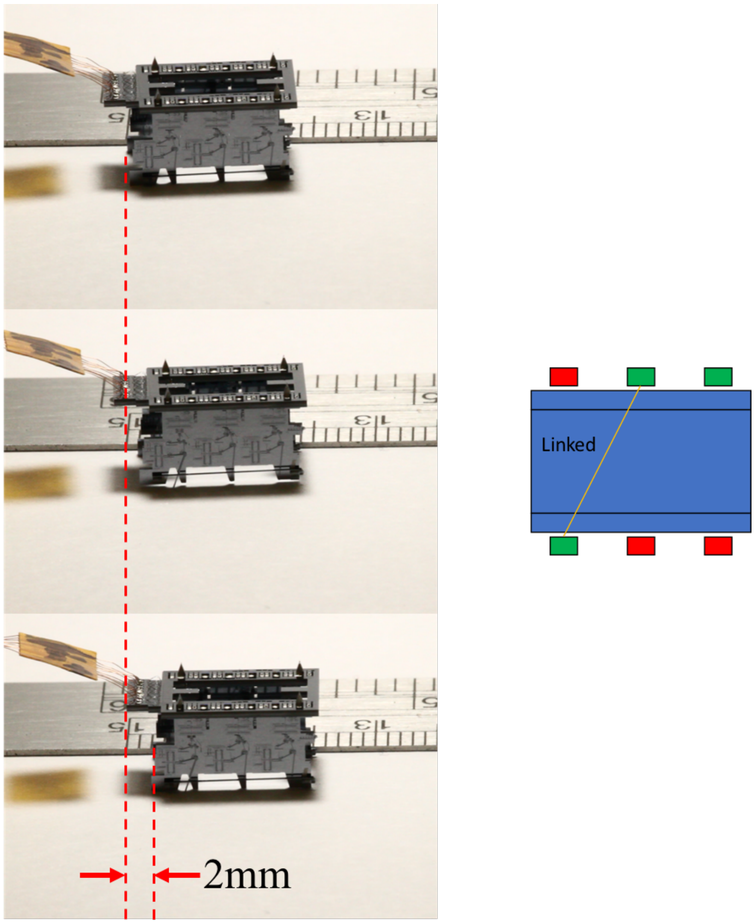


Figure 6.21: Sequence of the robot taking steps forward (left) Diagram showing the functioning legs of the robot (right)

6.5.2 Flexboard Backpack

In addition to the robot design made entirely of silicon, we have also done work on creating a robot controller and motor driver on a lightweight flexible circuit board substrate, hereby referred to as the robot backpack. The purpose of the robot backpack is to use the most lightweight commercial-off-the-shelf (COTS) and create an on-board controller that is within the robots payload.

The circuit board is shown in Fig. 6.22. The circuits mass is approximately 170mg. The backpack operates entirely on a 3.7V source. It uses an LT3482 boost DC/DC converter to generate a 90V signal from the 3.7V source. The high voltage signal is used as the high voltage rail for a network of high voltage transistors through pull-up resistors. These transistors are switched by a PIC16(L)F183XX microcontroller, which was chosen due to mass considerations. The board is meant to generate the high voltage signals for a dual tripod gait. With this board, the number of external connections drops from nine to two

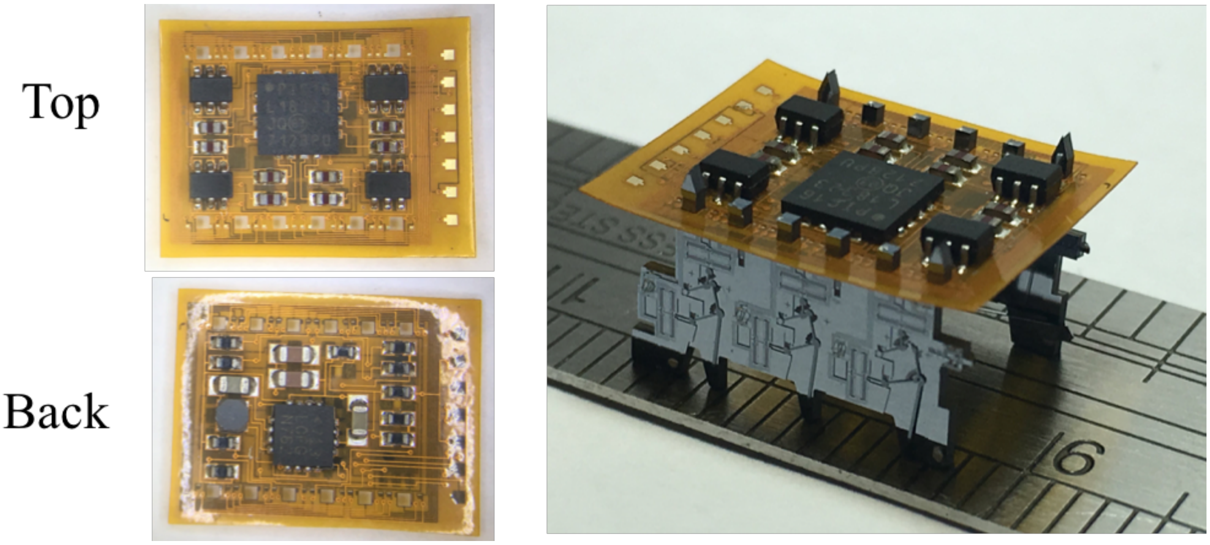


Figure 6.22: The flex backpack. Top and bottom of the board (left) The board with the leg chips inserted into the board. The inserted board only has the large ground signal pillars inserted. Issues with warping keep the device layer pillars from being safely inserted

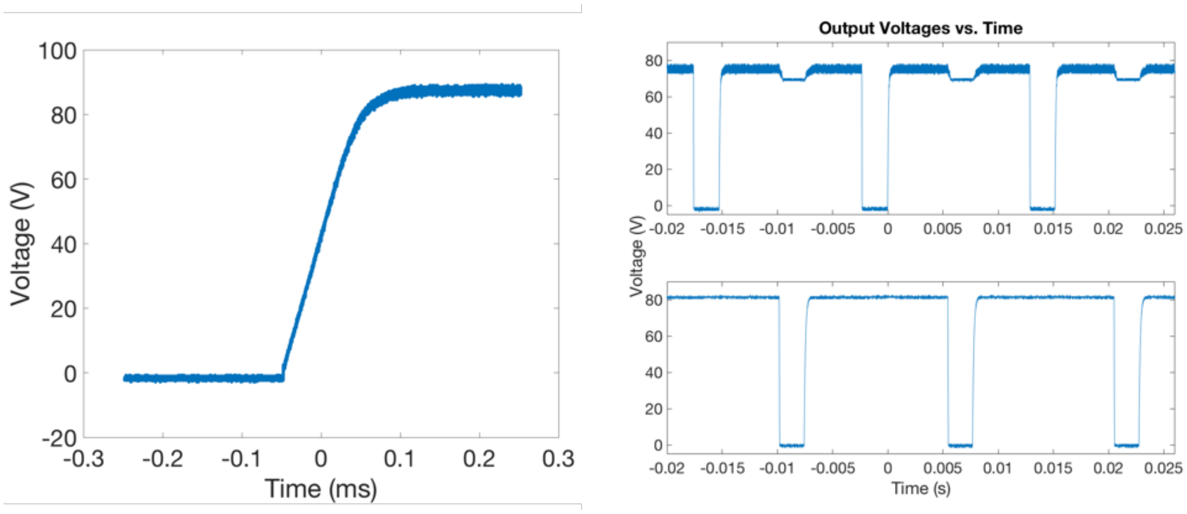


Figure 6.23: Output of the flexboard circuit. The board is able to generate 88V from a 3.7V source (left) The microcontroller uses resistively-loaded common source amplifiers to generate inchworm motor drive signals at 80V (right)

after the microcontroller has been programmed. The board has successfully generated an 88V signal that can drive the inchworm motors, as shown in Fig. 6.23.

The robot flex backpack has the same array of holes that are on the hub chip. The backpack is meant to interface to the leg chips in the exact same way that the hub chip interfaces to the leg chips. The flexible circuit board was patterned with a 2μm feature size in order to reduce the total size of the board. During assembly, the board breaks the device layer pillars because of warping in the board. More careful mechanical assembly is necessary to properly interface the pieces.

6.5.3 Future Work

We have demonstrated successful actuation of and steps taken by a six-legged silicon robot made of multiple chips, actuated through long tethers. The robot is able to move forward using only three of its six legs. The multichip assembly process provides reliable routing for the 3D MEMS structure, from the top hub chip to the side leg chips.

Current issues to address with the robot focus on yield and reliability of operation after assembly. Stiction frequently occurs during assembly and while the devices are in operation. This has been attributed to how sensitive the legs are to electrostatic forces and charging causing the legs to snap down and stick to the substrate. Preliminary experiments with self-assembled monolayers, such as FOTS, have shown promising results for preventing in operation stiction. Reduction of the substrate area can also be beneficial for stiction as well as reducing the robots total mass.

Chapter 7

Conclusion and Future Work

This chapter covers other topics such as other applications for the components discussed so far. We also discuss system level considerations for the robot.

7.1 Other Work

This work presented the individual components of the silicon pin-joint and the inchworm motors specifically with applications in the silicon robots in mind. However, these components can be useful in a variety of applications.

7.1.1 Inchworm Motor Chips

A standalone inchworm motor chip was designed, fabricated, and used in a variety of applications. The chip, pictured in Fig. 7.1, uses the same inchworm motor used on the hexapod. Signals are routed to pads toward the left end of the figure. The smaller pads are designed to interface with a zero-insertion-force (ZIF) socket. The larger pads are for probing with probe tips. There is a ring at the end of the shuttle. This end has also been designed as a U-shaped slot. The ring is 110μm in size. It was designed to interface with a 100μm rod with 10μm of tolerance.

Motors for Other Robots

The motor chip was initially designed to interface with the leg of another type of silicon robot, shown in Fig. 7.2. This work was presented in [35].

The inchworm motor ring interfaced with a bar on the leg, as shown in Fig. 7.3. A resistive switch network was used to drive the legs. Solar cells generating 60V at approximately 15μA were able to drive the leg through three quarters of its sweep cycle. The legs were not able to reset the leg during the passive retraction phase of the motor due to the serpentine springs on the motor shuttle being too weak to fully retract the leg.

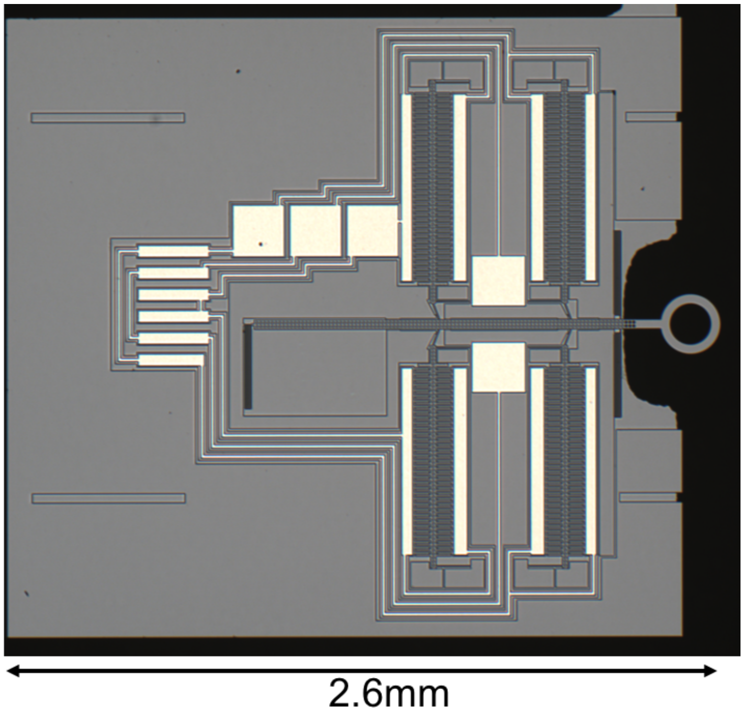


Figure 7.1: The standalone inchworm motor chip

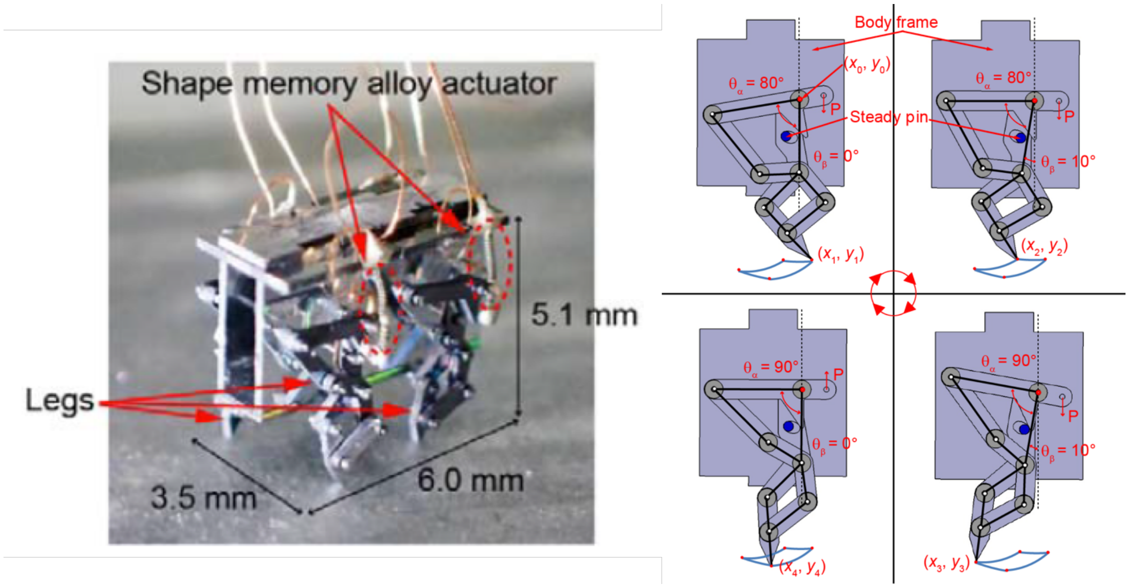


Figure 7.2: Silicon quadruped, designed with shape-memory alloy actuators (left) The sweep cycle of the leg used on the quadruped (right)

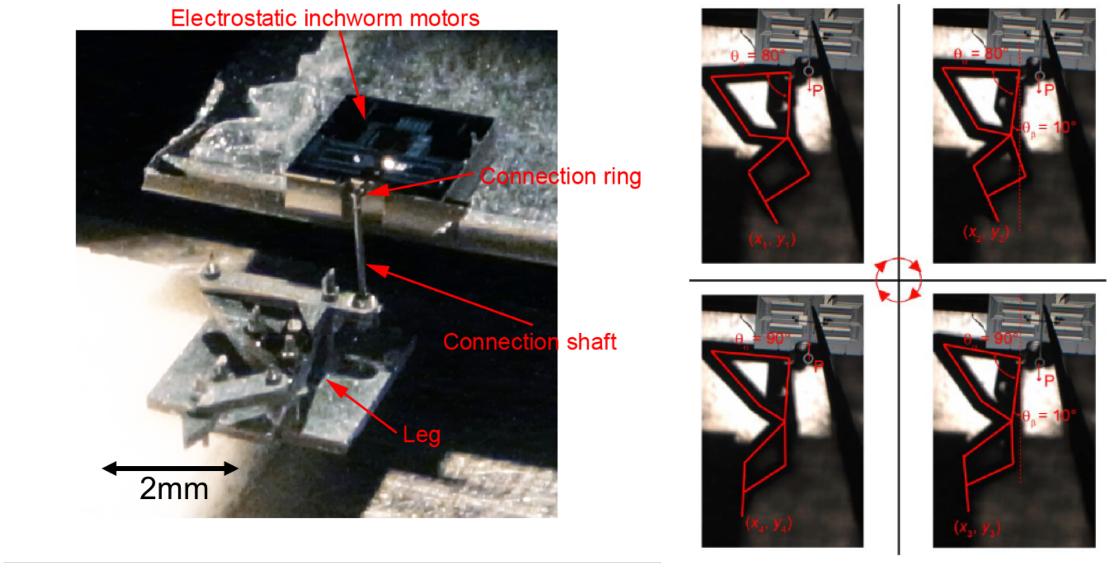


Figure 7.3: Experimental setup used to drive the quadruped leg with an electrostatic inchworm motor (left) The leg being driven by the inchworm motor (right)

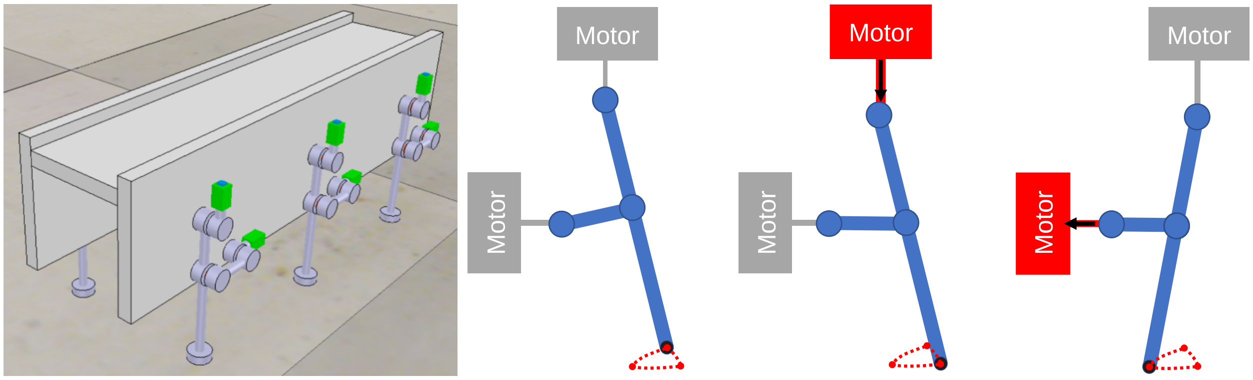


Figure 7.4: The simulated microrobot (left) Diagram of the legs sweep sequence, based on the hexapod leg (right)

7.1.2 Machine Learning for Microrobots

A machine learning model of the microrobot hexapod was developed in the robotics simulation package V-REP [36]. The model, shown in Fig. 7.4, is scaled up by a factor of 100 due to limitations in V-REP's simulation scales.

The algorithm uses central pattern generators (CPGs), which produce periodic outputs in vertebrates without any sensory input. In order to optimize the output of the simulation, Bayesian optimization is used to tune the parameters of the CPG. The target for the robot was to walk at a speed of 1cm/s using a variety of gaits.

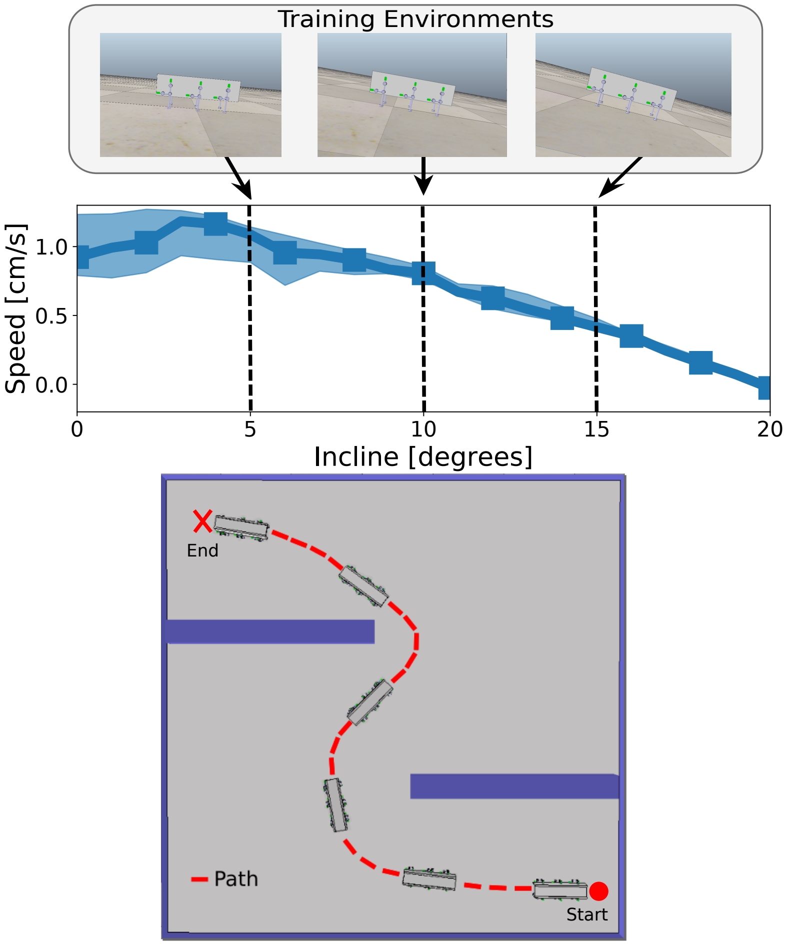


Figure 7.5: The simulated microrobot's performance on a variety of inclines (left) and through a small maze (right)

Using a variety of gaits (dual-tripod, ripple, wave, four-two) the simulated robot demonstrated learning to walk from nothing within 50 iterations. Fig. 7.5 shows results for the robot learning to walk in two environments. Using its learned dual-tripod gait, the robot showed moderate success on gradually increasing inclines. The robot simulation was also tested for path planning and was also able to successfully navigate a small maze. This also demonstrated that the planar legs of the robot allow it to make curved paths.

7.2 A 10cm/s Sub-mW Millimeter-Scale Autonomous Robot

The components presented in this work are capable of implementing on a robot that is capable of walking at many body lengths per second with low power draw. This section will develop a plan for this next generation robot.

7.2.1 Processing and Assembly Improvements

There are certain key issues with the current robot design that can be further improved by additions to the current process. These issues include the robot's mass, planar stability of the device layer components, the lack of a second routing layer, and robot assembly.

Mass Reduction and Backside Processing

The hexapod has a current mass of 200mg. Because of this mass, higher force motors with shorter lifetimes are necessary to lift the mass of the robot. 90% of this mass is from the 550μm thick substrate layer. This thickness is only useful during processing when handling the full wafer since thinner wafers are fragile and more prone to breaking. The exposure tool for the device layer, the GCA8500 Wafer Stepper (GCAWS6), also has a minimum thickness requirement for its focusing capabilities.

Multiple avenues of wafer thinning can be implemented. Vendors that supply SOI wafers are able to thin wafers although most do not thin below 400μm to 300μm due to handling difficulty. Commercial grinding options are also available. Thinning before handling has issues with handling during the initial steps of processing and using the tools for the device layer exposure. GCAWS6 has a 525μm thickness requirement.

The ASML DUV Stepper has a 200μm thickness requirement and it allows for smaller feature sizes than GCAWS6 (0.25μm resolution in a production environment). However, the tool requires deep UV photoresist rather than the i-line photoresist we traditionally use. Also the current undercut of the DRIE step of 0.45μm fundamentally prevents feature size improvements of structures unless the DRIE process is recharacterized. This photoresist would require process characterization for the DRIE and silicon selectivity. Current oxide etch selectivity of about 5:1 has been reported, comparable to selectivity of the i-line photoresist of about 3:1. Thinning the substrate further than these requirements is also possible if a specially machined chuck was made that could fit in the tools to meet the height requirements. Another wafer of the appropriate thickness temporarily bonded with a water drop could also be used.

The substrate can also be reduced during processing. Once the frontside process is complete, the wafer can be sent out for commercial grinding. A protective layer such as thick photoresist can be placed over the frontside for shipping and handling on the grinder. Application and removal of this protective layer would have to be characterized and incorporated in the process. Shipping wafers for grinding and waiting for there return would also add days to the process.

A faster alternative would be a backside KOH etch after frontside processing. The frontside layer can be protected with a special encasing that uses O-ring clamps to keep liquid from attacking the frontside. Reducing the substrate to 100μm would take 5.4 hours for a standard KOH etch rate (1.36μm/min at 80°C for 24% KOH concentration). This method is attractive due to its simplicity and quick turnaround time with respect to the other methods presented. Issues with this method include the presence of topography at the edge of the backside of the wafer due to the protection from the O-ring clamp at the edge. Photoresist spinning behavior due to this edge would have to be characterized.

Determining an ideal thickness. Assuming the substrate would serve solely as the robot's skeleton for structural purposes, it would have to handle loads typically associated with the robot's body. If the maximum forces the robot is expected to deal with come from the robot's own motors, than a typical load of 2mN should not cause a large deflection on the robot's skeleton. A cantilever beam that would be typical of the robot's body would be 1mm in length, 100µm wide (twice the device layer anchor width). Under a 2mN load with a 0.5µm maximum deflection, the beam would have to be 100µm thick.

If the backside is reduced by a factor of 1/5, the robot's mass would potentially be 60mg or less. A thinner substrate allows for a finer feature size allowing for finer mass reduction. For a 20:1 DRIE aspect ratio, 5µm feature size could be patterned. The contact lithography used for backside patterning allows for 2µm feature sizes, making gaps of this size in the backside conceivable. 5µm can be used for electrostatic actuation and sensing, allowing the use of the backside for active structures in addition to passive structures for support.

Planar Stability and Routing

Planar structures are kept in plane using assembled pieces of silicon after processing such as the screen used on the single legged walker and the brackets used on the leg of the hexapod. These protective structures can also be applied during processing using wafer bonding. The nanolab currently has a wafer bonding tool that can bond 6-inch wafers using metal, silicon, or silicon dioxide interfaces. This way, a second layer can be implemented that can serve to keep planar structures in-plane.

This second layer can also be used for routing, rather than the wire bonds that are currently applied after processing. Metal and silicon-dioxide pads can be defined on the frontside of the wafer. Opposing pads for interfaces would be defined on a second wafer for bonding. Once the wafers are bonded, processing can continue to reduce and pattern the backsides of the bonded wafers while the bonded device layers at the center are protected.

Mechanical Assembly

Another issue with the current assembly is reliance on silver epoxy for mechanical and electrical connectivity of the robot pieces. The silver epoxy is difficult to apply in repeatable fixed quantities even with pneumatic dispensing. The consistency of the silver epoxy also changes over time and achieving an optimal viscosity is difficult to target. When the epoxy is not viscous enough, it can short the routing around the robot through capillary action. If the epoxy is too viscous, it is difficult to apply and will not make good electrical connection when cured. Solder paste can be more reliable for repeatability and providing electrical conductivity but liquid flux in the solder can also run over the routing of the robot and ruin mechanical components.

There was also a concern with outgassing from the silver epoxy's resin. This could have been a reason for the tendency for legs stick down to the substrate after silver epoxy application and curing due to stiction caused by redeposition of outgassed chemicals. The self assembled monolayer FOTS was applied to try and prevent this post-assembly stiction. However the insulation from the FOTS coating made electrical contacts difficult to achieve. Ion milling was attempted to remove the FOTS from open areas where it was deemed unnecessary but stiction problems reoccurred after ion milling.

Another issue with the silver epoxy was making explicit contact between each substrate layer in the robot assembly. Electrostatic forces between the legs and the substrate were a suspected reason for leg stiction failure after assembly. Rather than relying on silver epoxy to connect the substrate layers and motor ground signals to a shared ground, this can be implemented in routing through mechanical assembly.

Rather than using silver epoxy or other paste-based methods for electro-mechanical connection, assembly can rely on purely mechanical spring-based support designs. The robot can use a zero-insertion force assembly method using probes on the hub chip to interface with pillars on the leg chips and hold everything in place with spring-based latches. Preliminary work in this area has shown success in using MEMS probes to interface with a CMOS chip.

Robot Process Description

Table 7.1 presents a process flow for further robot development, incorporating waferbonding and backside reduction. The full process involves 6 masks and 2 separate SOI wafers. The device wafer (labeled DEV) is used to define motors and legs and the bond wafer (labeled BOND) is used to define protective screens and routing to be bonded to the device wafer. Steps that can be performed in parallel are listed alongside each other. The process would involve 3 frontside alignment steps, and 1 backside alignment step.

The following lists assumptions to calculate the mass of a robot developed in this process.

- The robot is made from three separate chips, like the hexapod
- The device wafer backside is thinned to 100μm
- The robot is approximately 1cm long, 5mm wide, and 5mm tall
- 30% of the frontside/backside area is etched away abiding fill rules
- Frontside silicon on device wafer is 40μm thick
- Frontside silicon on bond wafer is 5μm thick

Under these assumptions, the robot would have a mass of approximately 35mg. Fill rules state that exposed area during an etch should be under 30%. This is to prevent undesirable etch effects such as grassing and pillaring.

Table 7.1: Process flow for next generation of robot production

Step (Device Wafer)	MASK	Step (Bond Wafer)	MASK	Notes
Photolithography (Liftoff)	METAL_DEV	Photolithography (Liftoff)	METAL_BOND	Metal routing/waferbond bondpad definition
Cr/Au Deposition 20nm/500nm		Cr/Au Deposition 20nm/500nm		Thickness typical of forming eutectic Si/Au bond [37]
Liftoff		Liftoff		
		Photolithography	SOI_RECESS_BOND	Define recessed areas of bond wafer silicon to reduce risk of mechanical parts on device wafer contacting bond wafer
		DRIE 1µm		
Photolithography	SOI_DEV	Photolithography	SOI_BOND	Define features in the device wafer (motors/linkages) and features in the bond wafer (grids for protective screens, routing lines)
DRIE t_{dev}		DRIE t_{bond}		t_{dev} and t_{bond} are the desired thickness of each of the SOI layers
Wafer bonding		Wafer bonding		Bond wafers together, all following steps apply to bonded wafer

Step (Bonded Wafer)	MASK	Notes
KOH Etch		Step for reduction of the substrates to ∼100µm. Bonded wafer is sealed with an outside ring of gold during metal deposition to prevent KOH from coming in and attacking SOI layer structures
Photolithography	TRENCH_DEV	Patterning features in the thinned device wafer substrate layer for mass reduction, skeleton definition, and device singulation
DRIE 100µm		
Vapor HF Release		Release structures from the oxide layer of the device wafer and the oxide layer of the bond wafer. Bond wafer substrate is completely removed during oxide etch

7.2.2 Robot Characteristics

In this section we repeat a similar series of calculations as in Chapter 1 for the assumed microrobot. Here we detail the characteristics of a robot with experimentally verified characteristics and some assumptions.

Characteristics and Assumptions

The following are experimentally verified characteristics of the robot motors in this work

- An inchworm motor generates F_m =1mN at 80V
- An inchworm motor has a velocity-frequency relationship of of 10µm/s/Hz
- An inchworm motor has a mechanical power density of 50W/kg
- A set of GCAs has finger capacitance of 0.33pF open, 2pF closed

The following are a list of assumptions about a proposed robot based on the hexapod design from this work

- An inchworm motor weigh 0.3mg
- The full robot weighs 35mg
- Robot legs have a mechanical advantage of 1:1 for the vertical stride
- Robot legs have a mechanical advantage of 5:1 for the horizontal stride
- Robot stride length is 3mm
- Each leg of the robot is independently controlled

The following are a list of assumptions about the power generation and circuitry for the proposed robot

- Charge recovery and layout make parasitic capacitance negligible
- Thin film battery energy density is Ed_b =0.36J/mg [15]
- Full sunlight solar power density is Pd_s =1000W/m^2
- Solar cells have η_{solar} = 10% efficiency
- Solar cells have an areal density of Ad_{solar} = 0.092kg/m^2 (40µm thick silicon)

Force Requirements

Weighing 35mg, the robot would require 350μN of force to lift itself. With a 1:1 mechanical advantage in the vertical axis and 1mN force output from a motor the robot is able exert a force of 2.5 times its weight with a single leg. This is desirable because it allows the use of a single leg for maneuverability and gait experimentation.

With three legs working in tandem as in a dual tripod gait, the robot can exert 8.5 times its weight. This allows 2.65mN of force for its payload. If it carried half of its payload capacity as a thin film battery, the robot would have over 50J of energy to use. If its entire surface was covered in solar cells the cells would provide 5mW of power and weigh 4.6mg.

The force limitation in the horizontal axis can come from a variety of factors. If we assume friction in the joints, the moment applied to the pin joint from gravity for a pin joint of radius 50μm and frictional coefficient of 0.3 would be (0.3)(35mg)(9.8m/s^2)(50μm)=5.25$\times 10^{-}$9Nm. The motor attached to the leg at 1mm from the joint can exert (1mN)(1mm)=1 $\times$ 10^{6}Nm, well in excess of the frictional moment.

There is also the concern of drag on the robot body. The drag equation is given by

$$F_D = \frac{1}{2}\rho v^2 A C_D \tag{7.1}$$

where ρ is the density of the fluid medium the object is moving through, v is the velocity of the fluid moving around the object, A is the cross-sectional area of the object moving through the fluid, and C_D is the unitless drag coefficient. We can solve for the maximum velocity as

$$v_{max} = \sqrt{\frac{2F_{D,max}}{\rho A C_D}} \tag{7.2}$$

Assuming a cuboid, $C_D = 1$. Assuming $F_D = 600$μN for three legs exerting 1mN each with 5:1 mechanical advantage, $\rho = 1.2$kg/m^3, $A = 1$cm^2, we get a max velocity of about 3m/s, well in excess of the target velocity of 10cm/s.

Power and Energy Requirements

To run at 10cm/s, the robot will have to actuate its GCAs at 2kHz. The electrical power the motors would draw for a dual tripod gait would be 154μW. The robot could run on solar cells with 4.8mW of excess power for control.

At 10μm for each step, the robot motor would need to take 60 steps to complete a stride for 5:1 mechanical advantage with a 1mm attachment point. This is a total of 1.5μJ of energy for each stride. With a battery pack providing 50J of energy the robot could take over 33 million steps totalling about 100km.

Appendix A

SUGAR Angled Arm Simulation

The SUGAR code for the angled arm simulations is presented here.

```
% MF_pinjoint.m
% Mechanical anchor
% model function defining a pin joint, anchored only in the z-axis
function [output] = MF_pinjoint(flag, R, param, q, t, nodes, varargin);

switch(flag)

case 'vars'

  output.ground =  {1 {'z'}};

case 'check'

  if (~isfield(param, 'l') | ...
      ~isfield(param, 'w') | ...
      ~isfield(param, 'h'))
    output = 'Missing length, width, or height parameters';
  else
    output = [];
  end

case 'pos'

  output = [0; 0; 0];

case 'abspos'
```

```
  if (isfield(param, ’x’) & isfield(param, ’y’) & isfield(param, ’z’))
    output = [param.x; param.y; param.z];
  else
    output = [];
  end

case ’display’

  q1 = zeros(12,1);
  displaybeam(q1, nodes(1).pos, R, param.l, param.w, param.h);

otherwise

  output = [];

end
```

The following is the angled arm netlist that used the pinjoint model function.

```
% angled arm netlist for simulation of statics of angled arm/shuttle
% interaction

% use berkeley SOI process file
uses SOI_berk.net

% parameters
% Fy - force on angled arms, oriented in the y-axis
% Fx - force on shuttle, oriented in the x-axis
% anchorS - default anchor size
% Wb - default beam width for suspensions
% Lb - default beam length for suspensions
% Ws - default shuttle width for suspensions
% Ls - default shuttle length for suspensions
% alpha - rotation of angled arms from the horizontal
% Larm - length of angled arm
% warm - width of angled arm
param Fy=0
param Fx=0
param anchorS=10u
param Wb=0.1u
param Lb=50u
param Ws=10u
param Ls=50u
param alpha=65*pi/180
param Larm=100u
param warm=3u


% define angled-arm subnet
subnet angledarm [nj pj] [ang=* rot=*]
[
    % define suspension
    beam3d    dev [a1 nj]  [l=Lb  w=Wb    oz=rot]
    beam3d    dev [nj a2]  [l=Lb  w=Wb    oz=rot]
    anchor    dev [a1]     [l=anchorS  w=anchorS   oz=-pi+rot]
    anchor    dev [a2]     [l=anchorS  w=anchorS   oz=rot]

    beam3d    dev [n2 a7]  [l=Lb  w=Wb oz=rot]
    beam3d    dev [n2 a8]  [l=Lb  w=Wb oz=-pi+rot]
    anchor    dev [a7]     [l=anchorS  w=anchorS   oz=rot]
    anchor    dev [a8]     [l=anchorS  w=anchorS   oz=-pi+rot]

    % define central shuttle angled arm is attached to
    beam3d    dev [nj n2]  [l=Ls  w=Ws   oz=pi/2+rot]

    % angled arm
    beam3d    dev [nj pj]  [l=Larm w=warm oz=ang]

    % force on angled arm from the y-direction towards the angled arm
    f3d     *     [nj]     [F=Fy oz=-pi/2+rot]

]

% call angled arm subnet with parameters
angledarm dev [n1 p1] [ang=-alpha rot=0]
angledarm dev [n2 p2] [ang=alpha rot=pi]

% pinjoint that interfaces the angled arm to the shuttle
pinjoint  dev [p1] [l=1n w=1n oz=-pi/2]
pinjoint  dev [p2] [l=1n w=1n oz=-pi/2]

% small beams solely meant to connect the pinjoint to the central shuttle
beam3d    dev [p1 nmid] [l=1n w=1n oz=-pi/2]
beam3d    dev [p2 nmid] [l=1n w=1n oz=pi/2]
```

```
% central shuttle 1
beam3d    dev [nmid n3] [l=Ls  w=Ws oz=-pi]

% central shuttle suspension 1
beam3d    dev [n3 a3]   [l=Lb w=Wb oz=pi/2]
beam3d    dev [n3 a4]   [l=Lb w=Wb oz=-pi/2]
anchor    dev [a3]      [l=anchorS w=anchorS oz=pi/2]
anchor    dev [a4]      [l=anchorS w=anchorS oz=-pi/2]

% central shuttle 2
beam3d    dev [nmid n4]  [l=Ls  w=Ws]

% central shuttle suspension 2
beam3d    dev [n4 a5]  [l=Lb w=Wb oz=pi/2]
beam3d    dev [n4 a6]  [l=Lb w=Wb oz=-pi/2]
anchor    dev [a5]     [l=anchorS w=anchorS oz=pi/2]
anchor    dev [a6]     [l=anchorS w=anchorS oz=-pi/2]

% force on the shuttle in the x-direction
f3d     *     [nmid]     [F=Fx oz=pi]
```

The following is the actual script that runs the angled arm netlist across a variety of variables to generate the plots in Section 3.2.5.

```
%force output over 2um displacement
clear;clc;

% define constants and variables
E=169e9;
alpha=65;
Larm=100e-6;
warm=3e-6;
t=40e-6;
Iarm=warm^3*t/12;
Kphi=E*Iarm/Larm;
Wb=0.1e-6;
Fy=(100:300:2000)*1e-6;
Fx=0;

% define parameter structure for netlist
param.Fx=Fx;
param.Fy=Fy(end);
param.Wb=Wb;
param.Larm=Larm;
param.warm=warm;
param.alpha=alpha*pi/180;

% take first step with netlist
net = cho_load('angled_arm.net',param);
dq1 = cho_dc(net);
figure(1); cho_display(net);
figure(2); cho_display(net,dq1);
dy1 = dqval(net,dq1,'n2','y');
dy2 = dqval(net,dq1,'n1','y');
dx1 = dqval(net,dq1,'p1','x');
dx2 = dqval(net,dq1,'nmid','x');
dstop=0e-6;

for i=1:length(Fy)
    Fx=0;
    param.Fx=Fx;
    param.Fy=Fy(i);
    while dx2>dstop
        net = cho_load('angled_arm.net',param);
        dq  = cho_dc(net);
        dy1 = dqval(net,dq,'n1','y');
        dy2 = dqval(net,dq,'n2','y');
        dx1 = dqval(net,dq,'p1','x');
        dx2 = dqval(net,dq,'nmid','x');
        Fx=Fx+10e-6;
        param.Fx=Fx;
    end
    Fout(i)=Fx;
    dx1=1;
    dx2=1;
    dy1=-1;
    dy2=1;
end

Foutcalc=2*(Fy/tand(alpha));

figure(3);plot(Fout*1e6,Fy*1e6,Foutcalc*1e6,Fy*1e6)
xlabel('F_x (N)','FontSize',30)
```

```
ylabel('F_y (N)','FontSize',30)
set(gca,'fontsize',30)

Fy=1000e-6;
Fx=0;
param.Fy=Fy;
param.Fx=Fx;
net = cho_load('angled_arm.net',param);
dq = cho_dc(net);
dx  = dqval(net,dq,'p1','x');
i=1;
while dx>5e-6
    param.Fx=param.Fx+1e-6;
    net = cho_load('angled_arm.net',param);
    dq  = cho_dc(net);
    dx = dqval(net,dq,'p1','x');
    dy = dqval(net,dq,'n2','y');
end

dx1(1)=dx;
dy1(1)=dy;

while dx1>0
    dx1(i) = dqval(net,dq,'p1','x');
    Fxout(i)=param.Fx;
    param.Fx=param.Fx+1e-6;
    net = cho_load('angled_arm.net',param);
    dq  = cho_dc(net);
    dy1(i) = dqval(net,dq,'n2','y');
    i=i+1;
end

Fxcalc=2*(Fy/tand(alpha)-3*Kphi*dx1/(Larm^2*sind(alpha)^2));
dxcalc=dy1*tand(alpha);

figure(4);plot(dx1*1e6,dy1*1e6,dxcalc*1e6,dy1*1e6)
xlabel('\Deltax (N)','FontSize',30)
ylabel('\Deltay (N)','FontSize',30)
set(gca,'fontsize',30)
figure(5);plot(dx1*1e6,Fxout*1e6,dx1*1e6,Fxcalc*1e6)
xlabel('\Deltax (N)','FontSize',30)
ylabel('F_x (N)','FontSize',30)
```

```
set(gca,'fontsize',30)
```

Appendix B

MATLAB Dynamics Model

The following is the GCA dynamics master script for the MATLAB model.

```
% clear workspace and console
clear;clc;

% plotting parameters
% FS - fontsize
% LW - line width
% MS - marker size
% MW - marker width
FS=25;
LW=3;
MS=10;
MW=2;

% define global variables to pass between functions
global V e0 mu m S1 S2 T k x0 xf beta N xb Ts h0 wf er bcon kcon mcon...
    fcon kconb A rho_si Ls alpha E

%%
% forcing equation variable constants
% multiplicative factors to manipulate the results of the forcing function
% kconb adjusts spring constant during release phase. this constant
% provides a better fit to pull-out voltage data
bcon=1;
kcon=1;
kconb=1.2;
mcon=1;
fcon=1;
```

```
%%
% Undercut value around structure and area calculation of the central
% shuttle. measured from layout
% xuc_m, undercut around the central shuttle for the sake of mass reduction
% xuc_spr, undercut of springs, for spring width reduction
% xuc_fing, undercut around the GCA fingers
% AshutArr, calculating the layout area of the central shuttle using the
% undercut xuc_m
xuc_m=0.4e-6;
xuc_spr=0.4e-6;
xuc_fing=0.4e-6;
AshutArr=(-4.249e-6*xuc_m+1.52475e-8);

%%
% define physical constants
% e0, vacuum permittivity
% er, relative permittivity
% E, young's modulus of silicon
% mu, viscosity of air
% rho_si, density of silicon
% rho_air, density of air
% rho_h2o, density of water
% alpha, constant for finger resonant frequency
e0=8.85e-12;
er=1;
E=169e9;
mu=1.8e-5;
rho_si=2300;
rho_air=1;
rho_h2o=1000;
alpha=1.875;


%%
% constant GCA geometry dimensions
% L0 - finger overlap length
% N - number of fingers
% T - device thickness
% wf - finger width
% x0 - initial forward gap
% xb - initial rear gap
% h0 - substrate gap
% mfp - mean free path of air
% Ks - knudsen number
% Ts - effective thickness for damping due to substrate proximity
% Ls - base support finger length
% xf - final gap, defined by gapstop
% Lspr - length of spring
L0=76.5e-6;
N=70;
T=40e-6;
wf=5e-6-2*xuc_fing;
x0=4.83e-6+2*xuc_fing;
xb=7.75e-6+2*xuc_fing;
h0=2e-6;
mfp=68e-9;
Ks=mfp/h0;
Ts=T+0.81*(1+0.94*Ks)*h0;
Ls=10e-6;
xf=1e-6;
LSpr=240.85e-6;



%%
% measurement arrays, for data
LArrMeas=[0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]*L0;
wSprMeas=(2:0.5:5.5)*1e-6-2*xuc_spr;
kArrMeas=2*E*T*wSprMeas.^3/LSpr^3;



%%
% simulation arrays
% LArrSim1 - array of overlap lengths, less granular, for voltage theory
% comparison
% LArrSim2 - array of overlap lengths, granular, for length theory
% comparison
% wSprSim1 - spring width array
% kArrSim1 - spring constant array, less granular for voltage theory
% comparison
% wSprSim2 - spring width array, granular
% kArrSim2 - spring constant array, granular for spring theory comparison
```

```
LArrSim1=[0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1]*L0;
LArrSim2=linspace(0.2,1,100)*L0;
wSprSim1=[2 2.5 3 3.5 4 4.5 5 5.5]*1e-6-2*xuc_spr;
kArrSim1=2*E*T*wSprSim1.^3/LSpr^3;
wSprSim2=(2:0.1:5)*1e-6-2*xuc_spr;
kArrSim2=2*E*T*wSprSim2.^3/LSpr^3;

% betaArrSim - array of beta correction factors according to overlap
%              length/thickness aspect ratio
for i=1:length(LArrSim1)
    if(LArrSim1(i)/T<=1)
        betaArrSim(i)=-0.6*LArrSim1(i)/T+1;
    else
        betaArrSim(i)=-0.6*T/LArrSim1(i)+1;
    end
end

%%
% set up voltage simulation arrays
% calculate pull-in voltages
x=linspace(0,1,10000)*x0;
for i=1:length(LArrSim1)
    % Constant for solving Vpi and xpi
    const1=(fcon*1/2*N*e0*LArrSim1(i)*T)*(1./(x0-x).^2-1./(xb+x).^2);

    % Solve pull-in parameters
    Vtemp=sqrt(kArrSim1(3)*x./const1);

    % Pull-in voltage will be the maximum of the solved equation, dF/dx=0
    % xpi is a fraction w.r.t g0
    [M,I]=max(Vtemp);
    xpi(i)=x(I)/x0;
    VpiArrL(i)=max(Vtemp);
end
x=linspace(0,1,10000)*x0;
for i=1:length(kArrSim1)
    % Constant for solving Vpi and xpi
    const1=(fcon*1/2*N*e0*LArrSim1(end)*T)*(1./(x0-x).^2-1./(xb+x).^2);

    % Solve pull-in parameters
    Vtemp=sqrt(kArrSim1(i)*x./const1);
    % Pull-in voltage will be the maximum of the solved equation, dF/dx=0
    % xpi is a fraction w.r.t g0
    [M,I]=max(Vtemp);
    xpi(i)=x(I)/x0;
    VpiArrk(i)=max(Vtemp);
end

% calculate release voltages
po_const1=(1/2*N*e0*LArrMeas*T)*(1/(xf)^2-1/(xb+x0-xf)^2);
VpoArrL=sqrt(kconb*kArrSim1(3)*(x0-xf)./po_const1);

po_const2=(1/2*N*e0*LArrMeas(end)*T)*(1/(xf)^2-1/(xb+x0-xf)^2);
VpoArrk=sqrt(kconb*kArrSim1*(x0-xf)/po_const2);


% load and plot measured data
data_20180208

%% define analysis arrays
VArrL1(1,:)=linspace(V1_Arr1(1),V1_Arr1(end));
VArrL1(2,:)=linspace(V2_Arr1(1),V2_Arr1(end));
VArrL1(3,:)=linspace(V3_Arr1(1),V3_Arr1(end));
VArrL1(4,:)=linspace(V4_Arr1(1),V4_Arr1(end));
VArrL1(5,:)=linspace(V5_Arr1(1),V5_Arr1(end));
VArrL1(6,:)=linspace(V6_Arr1(1),V6_Arr1(end));
VArrL1(7,:)=linspace(V7_Arr1(1),V7_Arr1(end));
VArrL1(8,:)=linspace(V8_Arr1(1),V8_Arr1(end));
VArrL1(9,:)=linspace(V9_Arr1(1),V9_Arr1(end));

VArrL1_r(1,:)=linspace(V1_Arr1_r(1),V1_Arr1_r(end));
VArrL1_r(2,:)=linspace(V2_Arr1_r(1),V2_Arr1_r(end));
VArrL1_r(3,:)=linspace(V3_Arr1_r(1),V3_Arr1_r(end));
VArrL1_r(4,:)=linspace(V4_Arr1_r(1),V4_Arr1_r(end));
VArrL1_r(5,:)=linspace(V5_Arr1_r(1),V5_Arr1_r(end));
VArrL1_r(6,:)=linspace(V6_Arr1_r(1),V6_Arr1_r(end));
VArrL1_r(7,:)=linspace(V7_Arr1_r(1),V7_Arr1_r(end));
VArrL1_r(8,:)=linspace(V8_Arr1_r(1),V8_Arr1_r(end));
VArrL1_r(9,:)=linspace(V9_Arr1_r(1),V9_Arr1_r(end));

VArrk1(1,:)=linspace(V1_Arr2(1),V1_Arr2(end));
VArrk1(2,:)=linspace(V2_Arr2(1),V2_Arr2(end));
VArrk1(3,:)=linspace(V3_Arr2(1),V3_Arr2(end));
```

```
VArrk1(4,:)=linspace(V4_Arr2(1),V4_Arr2(end));
VArrk1(5,:)=linspace(V5_Arr2(1),V5_Arr2(end));
VArrk1(6,:)=linspace(V6_Arr2(1),V6_Arr2(end));
VArrk1(7,:)=linspace(V7_Arr2(1),V7_Arr2(end));
VArrk1(8,:)=linspace(V8_Arr2(1),V8_Arr2(end));

VArrk1_r(1,:)=linspace(V1_Arr2_r(1),V1_Arr2_r(end));
VArrk1_r(2,:)=linspace(V2_Arr2_r(1),V2_Arr2_r(end));
VArrk1_r(3,:)=linspace(V3_Arr2_r(1),V3_Arr2_r(end));
VArrk1_r(4,:)=linspace(V4_Arr2_r(1),V4_Arr2_r(end));
VArrk1_r(5,:)=linspace(V5_Arr2_r(1),V5_Arr2_r(end));
VArrk1_r(6,:)=linspace(V6_Arr2_r(1),V6_Arr2_r(end));
VArrk1_r(7,:)=linspace(V7_Arr2_r(1),V7_Arr2_r(end));
VArrk1_r(8,:)=linspace(V8_Arr2_r(1),V8_Arr2_r(end));

VArrL2=[45 50 55 60];
VArrk2=[45 50 55 60];

%%
% pull-in data measurements and plots
Vpi_meas_Arr1=mean(Vpi_Arr1);
Vpi_meas_Arr2=Vpi_Arr2;

figure(1)
subplot(2,2,1)
hold on
errorbar(LArrMeas*1e6,Vpi_meas_Arr1,VErr1,'.','MarkerSize',MS,...
    'LineWidth',MW)
stack=plot(LArrMeas*1e6,VpiArrL,'LineWidth',LW);
uistack(stack,'bottom')
axis tight

subplot(2,2,2)
hold on
errorbar(kArrMeas,Vpi_Arr2,VErr2,'.','MarkerSize',MS,'LineWidth',MW)
plot(kArrMeas,VpiArrk,'LineWidth',LW)
stack=uistack(stack,'bottom');
axis tight

set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% release data measurements and plots
```

```
Vpo_meas_Arr1=Vpo_Arr1;
Vpo_meas_Arr2=Vpo_Arr2;

subplot(2,2,3)
hold on
errorbar(LArrMeas*1e6,Vpo_Arr1,VErr1,'.','MarkerSize',MS,'LineWidth',MW)
stack=plot(LArrMeas*1e6,VpoArrL,'LineWidth',LW);
uistack(stack,'bottom')
axis tight

subplot(2,2,4)
hold on
errorbar(kArrMeas,Vpo_Arr2,VErr2,'.','MarkerSize',MS,'LineWidth',MW)
stack=plot(kArrMeas,VpoArrk,'LineWidth',LW);
uistack(stack,'bottom')
axis tight

set(findall(gcf,'-property','FontSize'),'FontSize',FS)

%%
% area and mass arrays
AmSim = AshutArr+N*(Ls+LArrSim1)*wf;
mArrSim = mcon*rho_si*AmSim*T;
meff=mArrSim+N*rho_air*LArrSim1.^2*T^2./(2*(LArrSim1+T));


%%
% define initial conditions and time range for numerical solution
xinit=[0,0];
tspan=0:1e-6:1e-2;

% options for ODE solver
refine=4;
options1 = odeset('Events',@eventsPI,'OutputSel',1,'Refine',refine);
tpiSimArrL1=zeros(length(LArrSim1),length(VArrL1(1,:)));
tpiSimArrk1=zeros(length(kArrSim1),length(VArrk1(1,:)));
tpiSimArrL2=zeros(length(LArrSim2),length(VArrL2(1,:)));
tpiSimArrk2=zeros(length(kArrSim2),length(VArrk2(1,:)));


%% find pull-in data wrt voltage for varying overlap lengths
for i=1:length(LArrSim1)
```

```
    for j=1:length(VArrL1(i,:))
        S1=max(LArrSim1(i),T);
        S2=min(LArrSim1(i),T);
        beta=betaArrSim(i);
        m=meff(i);
        V=VArrL1(i,j);
        k=kArrSim1(3);
        A=AmSim(i);
        [t1,x1,tpiSimArrL1(i,j),ye,ie]=ode45(@pull_in,tspan,xinit...
            ,options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end


%%
% recalculate beta array with LArrSim2
for i=1:length(LArrSim2)
    if(LArrSim2(i)/T<=1)
        betaArrSim(i)=-0.6*LArrSim2(i)/T+1;
    else
        betaArrSim(i)=-0.6*T/LArrSim2(i)+1;
    end
end
% recalculate area and mass arrays with LArrSim2
AmSim = AshutArr+N*(Ls+LArrSim2)*wf;
mArrSim = mcon*rho_si*AmSim*T;
meff=mArrSim+N*rho_air*LArrSim2.^2*T^2./(2*(LArrSim2+T));

%% find pull-in data wrt overlap length for varying voltages
for i=1:length(LArrSim2)
    for j=1:length(VArrL2)
        S1=max(LArrSim2(i),T);
        S2=min(LArrSim2(i),T);
        beta=betaArrSim(i);
        m=meff(i);
        V=VArrL2(j);
        k=kArrSim1(3);
        A=AmSim(i);
        [t1,x1,tpiSimArrL2(i,j),ye,ie]=ode45(@pull_in,tspan,xinit,...
            options1);
```

```
%         plot(t1,x1(:,1));
%         hold on
    end
end


%% find pull-in data wrt voltage for varying spring constant
for i=1:length(kArrSim1)
    for j=1:length(VArrk1(i,:))
        S1=max(LArrSim1(end),T);
        S2=min(LArrSim1(end),T);
        beta=betaArrSim(end);
        m=meff(end);
        V=VArrk1(i,j)+2;
        k=kArrSim1(i);
        A=AmSim(end);
        [t1,x1,tpiSimArrk1(i,j),ye,ie]=ode45(@pull_in,tspan,xinit,...
            options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end


%% find pull-in data wrt spring constant for varying voltage
for i=1:length(kArrSim2)
    for j=1:length(VArrk2)
        S1=max(LArrSim1(end),T);
        S2=min(LArrSim1(end),T);
        beta=betaArrSim(end);
        m=meff(end);
        V=VArrk2(j)+2;
        k=kArrSim2(i);
        A=AmSim(end);
        [t1,x1,tpiSimArrk2(i,j),ye,ie]=ode45(@pull_in,tspan,xinit,...
            options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end
```

```
%% release simulation ODE setup
% define initial conditions and time range for numerical solution
tspan=0:1e-6:1e-3;
% options for ODE solver
refine=4;
options1 = odeset('Events',@eventsPO,'OutputSel',1,'Refine',refine);
tpoSimArrL1=zeros(length(LArrSim1),length(VArrL1(1,:)));
tpoSimArrk1=zeros(length(kArrSim1),length(VArrk1(1,:)));
tpoSimArrL2=zeros(length(LArrSim2),length(VArrL2(1,:)));
tpoSimArrk2=zeros(length(kArrSim2),length(VArrk2(1,:)));

%% recalculate beta array and area and mass arrays for LArrSim1
for i=1:length(LArrSim1)
    if(LArrSim1(i)/T<=1)
        betaArrSim(i)=-0.6*LArrSim1(i)/T+1;
    else
        betaArrSim(i)=-0.6*T/LArrSim1(i)+1;
    end
end
AmSim = AshutArr+N*(Ls+LArrSim1)*wf;
mArrSim = mcon*rho_si*AmSim*T;
meff=mArrSim+N*rho_air*LArrSim1.^2*T^2./(2*(LArrSim1+T));


%% find release time wrt voltage for varying overlap length
for i=1:length(LArrSim1)
    for j=1:length(VArrL1_r(i,:))
        S1=max(LArrSim1(i),T);
        S2=min(LArrSim1(i),T);
        beta=betaArrSim(i);
        m=meff(i);
        k=kArrSim1(3);
        V=VArrL1_r(i,j);
        A=AmSim(i);
        vinit=calc_vinit(LArrSim1(i),m);
        xinit=[x0-xf,vinit];
        [t1,x1,tpoSimArrL1(i,j),ye,ie]=ode45(@release,tspan,xinit,...
            options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end
```

```
%% recalculate beta and area and mass arrays for LArrSim2
for i=1:length(LArrSim2)
    if(LArrSim2(i)/T<=1)
        betaArrSim(i)=-0.6*LArrSim2(i)/T+1;
    else
        betaArrSim(i)=-0.6*T/LArrSim2(i)+1;
    end
end
AmSim = AshutArr+N*(Ls+LArrSim2)*wf;
mArrSim = mcon*rho_si*AmSim*T;
meff=mArrSim+N*rho_air*LArrSim2.^2*T^2./(2*(LArrSim2+T));
% define LArr for vinit
LArr=LArrSim2;

%% find release time wrt varying overlap length for varying voltage
for i=1:length(LArrSim2)
    for j=1:length(VArrL2)
        S1=max(LArrSim2(i),T);
        S2=min(LArrSim2(i),T);
        beta=betaArrSim(i);
        m=meff(i);
        k=kArrSim1(3);
        V=VArrL2(j);
        A=AmSim(i);
        vinit=calc_vinit(LArrSim2(i),m);
        xinit=[x0-xf,vinit];
        [t1,x1,tpoSimArrL2(i,j),ye,ie]=ode45(@release,tspan,xinit,...
            options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end

%% find release time wrt varying voltage for varying spring constant
for i=1:length(kArrSim1)
    for j=1:length(VArrk1_r(i,:))
        S1=max(LArrSim2(end),T);
        S2=min(LArrSim2(end),T);
        beta=betaArrSim(end);
        m=meff(end);
        k=kArrSim1(i);
        V=VArrk1_r(i,j);
        A=AmSim(end);
        vinit=calc_vinit(LArrSim2(end),m);
        xinit=[x0-xf,vinit];
        [t1,x1,tpoSimArrk1(i,j),ye,ie]=ode45(@release,tspan,xinit,...
            options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end

%% find release time wrt varying spring constant for varying voltage
for i=1:length(kArrSim2)
    for j=1:length(VArrk2)
        S1=max(LArrSim2(end),T);
        S2=min(LArrSim2(end),T);
        beta=betaArrSim(end);
        m=meff(end);
        k=kArrSim2(i);
        V=VArrk2(j);
        A=AmSim(end);
        vinit=calc_vinit(LArrSim2(end),m);
        xinit=[x0-xf,vinit];
        [t1,x1,tpoSimArrk2(i,j),ye,ie]=ode45(@release,tspan,xinit,...
            options1);
%         plot(t1,x1(:,1));
%         hold on
    end
end


%% Plotting
color='k';

% pull-in plots of arrays with varying overlap length
figure(2)
for i=1:length(VArrL1(:,1))
   subplot(3,3,i)
   hold on
   stack=plot(VArrL1(i,:),tpiSimArrL1(i,:)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
```

```
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% release plot of arrays with varying overlap length
figure(3)
for i=1:length(VArrL1_r(:,1))
   subplot(3,3,i)
   hold on
   stack=plot(VArrL1_r(i,:),tpoSimArrL1(i,:)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% pull-in plots of arrays with varying spring constant
figure(4)
for i=1:length(VArrk1(:,1))
   subplot(4,2,i)
   hold on
   stack=plot(VArrk1(i,:),tpiSimArrk1(i,:)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% release plots of arrays with varying spring constants
figure(5)
for i=1:length(VArrk1_r(:,1))
   subplot(4,2,i)
   hold on
   stack=plot(VArrk1_r(i,:),tpoSimArrk1(i,:)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% pull-in plots against overlap lengths
figure(6)
for i=1:length(VArrL2)
   subplot(2,2,i)
   hold on
   stack=plot(LArrSim2*1e6,tpiSimArrL2(:,i)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% release plots against overlap length
figure(7)
for i=1:length(VArrL2)
   subplot(2,2,i)
   hold on
   stack=plot(LArrSim2*1e6,tpoSimArrL2(:,i)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% pull-in plots against spring constant
figure(8)
for i=1:length(VArrk2)
   subplot(2,2,i)
   hold on
   stack=plot(kArrSim2,tpiSimArrk2(:,i)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)

% release plots against spring constant
figure(9)
for i=1:length(VArrk2)
   subplot(2,2,i)
   hold on
   stack=plot(kArrSim2,tpoSimArrk2(:,i)*1e6,'LineWidth',LW);
   uistack(stack,'bottom');
   set(gca,'XScale','log','YScale','log')
   grid on
   axis tight
end
set(findall(gcf,'-property','FontSize'),'FontSize',FS)
```

```
%% Functions

function dxdt = pull_in(t,x)
    global V e0 mu m S1 S2 k x0 beta N xb Ts h0 wf er bcon kcon mcon fcon A
    % define differential equation for pull-in analysis

    % electrostatic force
    Fes = fcon*(1/(x0-x(1))^2-1/(xb+x(1))^2)*1/2*N*e0*er*V^2*(S1)*(S2);

    % damping
    b2=mu*A/h0;
    Fd  = bcon*((1/(x0-x(1))^3+1/(xb+x(1))^3)*N*beta*mu*S1*S2^3+b2)*x(2);

    % spring
    Fk  = kcon*k*x(1);

    % setup ODE to solve
    dxdt = [x(2); (Fes - Fd - Fk)/(m*mcon)];
end

function dxdt = release(t,x)
    global mu m S1 S2 k x0 beta N xb Ts h0 wf bcon kconb mcon A
    % define differential equation for release analysis

    % damping
    b2=mu*A/h0;
    Fd  = bcon*((1/(x0-x(1))^3+1/(xb+x(1))^3)*N*beta*mu*S1*S2^3+b2)*x(2);

    % spring
    Fk = k*x(1);

    % setup ODE to solve
    dxdt = [x(2); (-Fd - Fk)/(m*mcon)];
end

function [value,isterminal,direction] = eventsPI(t,x)
    % define end state for pull-in analysis (gap equals the final gap)
    global x0 xf
    value = x0-x(1)-xf;
    isterminal = 1;
    direction = -1;
```

```
end

function [value,isterminal,direction] = eventsPO(t,x)
    % define end state for pull-in analysis (gap equals the initial gap)
    value = x(1);
    isterminal = 1;
    direction = -1;
end

% calculate initial velocity
function vinit = calc_vinit(LArr,mArr)
    global V e0 T xf N wf fcon rho_si E Ls alpha

    % find max velocity of finger upon release, from strain energy in
    % the finger's deflection
    If=wf^3*T/12;
    mf0=T*wf*rho_si;
    omega=alpha^2*sqrt(E*If/(mf0*(LArr+Ls)^4));
    kf=6.9*E*If/(LArr+Ls)^3;
    mfeff=kf/omega.^2;
    Fxf=fcon*1/2*e0*V^2*LArr*T*(1/(xf)^2);
    xfing=Fxf/kf;
    varr=0;%-omega*xfing/2;

    % find max velocity of the GCA shuttle upon release, from strain
    % energy stored in the shuttle's deflection
    wshut=20e-6;
    Lshut=860e-6;
    Ashut=T*wshut;
    kshut=E*Ashut/Lshut;
    xshut=N*Fxf/kshut;
    mshut=T*Ashut*rho_si;
    omegashut=sqrt(kshut/mshut);
    vshut=0;%-omegashut*xshut;

    % find initial velocity of entire GCA assembly using momentum
    % transfer
    vinit=(N*mfeff*varr+mshut*vshut)/mArr;
end
```

Appendix C

Thickness Optimization

The following is the code used to generate the thickness optimization plots.

```
clear;clc;
global V e0 mu m S1 S2 xo beta N xb rho_si wf xf Arot h0 T Ls E k YI...
    karm L1tot Lbase

% physical constants
E=169e9;
e0=8.85e-12;
mu=1.8e-5;
rho_si=2300;
emax=0.0005;
sigmaMax=emax*E;
eox=3.9;

% alpha - aspect ratio Tarr - dev layer thickness array luc - process
% undercut Cl - safety factor on max length VmaxArr - maximum voltage
% array. found through iteration. run script with
%   single Vmax of 100, then input an array following the trend of VminArr
%   with some desired proportionality (VmaxArr(i) will always be
%   1.5xVminArr(i)). Using a line proportional to T here for simplicity.
% Fmin - minimum output force B - back gap multiplicative factor N - number
% of gaps n - angled arm buckling safety factor K - angled arm buckling
% effective arm length phi - angled arm angle to shuttle h0 - substrate gap
alpha=1/10;
Tarr=(1:1:40)*1e-6;
luc=0.4e-6;
Cl=0.8;
VmaxArr=1e6*Tarr;
```

```
Fmin=1000;
B=1.5;
N=50;
n1=10;
n2=1;
K1=0.5;
K2=1;
phi=65;
for i=1:length(Tarr)
    T=Tarr(i);
    Vmax=VmaxArr(i);

    % lambda - minimum feature size
    lambda=alpha*T;

    % derived dimensions h0 - gap to substrate, assume some proportionality
    % with thickness wf - finger width xf - gapstop size dxshut - shuttle
    % stepsize Ls - finger support length LA - anchor dimension warm -
    % angled arm width Iarm - angled arm moment of inertia dx - GCA step
    % translated from shuttle step size YI - shuttle-pawl gap xo - initial
    % gap size xb - initial back gap size xpi - pull-in position, found
    % from value for B. a new B value needs a new
    %   xpi. Run pull_in_voltage.m to find a pull-in voltage for a new B
    % Lmax - maximum finger length Lol - overlap finger length
    h0=T/10;
    wf=2*lambda;
    xf=lambda/2;
    dxshut=lambda;
    LA=25*lambda;
    warm=4*lambda;
    Iarm=warm^3*T/12;
    dx=dxshut/tand(phi);
    YI=2*lambda;
    xo=dx+YI+xf;
    xb=B*xo;
    xpi=0.28*xo;
    Lmax=Cl*(0.28*E*wf^3*xf^3/(e0*Vmax^2*(1+0.42*xf/T)))^(1/4);
    Ls=Lmax/10;
    Lol=Lmax-Ls;


    % Fxf - electrostatic force at gapstop
```

```
Fxf=1/2*N*e0*Vmax^2*T*Lol*(1/(xf)^2-1/(xb+xo-xf)^2);

% Larm - length of angled arm karm - stiffness of angled arm along GCA
% axis
Larm=sqrt(pi^2*E*Iarm/(n1*K1^2*Fxf*sind(phi)));
karm=12*E*Iarm/(Larm^3*cosd(phi));

% array layout area calculation L1tot - length of the array along
% finger array Lbase - base GCA shuttle width meeting maximum strain
% requirement or buckling load requirement, whichever comes first L2tot
% - length of the array along finger length
L1tot=N*(2*wf+xo+xb);
Lbase1=Fxf/(T*sigmaMax);
Lbase2=(12*n2*K2^2*L1tot^2*Fxf/(pi^2*T*E))^(1/3);
Lbase=max(Lbase1,Lbase2);
L2tot=(LA+2*Ls+Lbase+Lol);
A=(L1tot)*(L2tot)+Larm^2*cosd(phi)*sind(phi);

% Find spring constant based on minimum output force and pull-in
% voltage
syms Vpisym Vminsym ksym
[Vpi, Vmin, kmax] = solve(...
    Fmin    == 1/tand(phi)*(1/2*N*e0*Vminsym^2*Lol*T*(1/(xo-YI)^2-...
    1/(xb+YI)^2)-ksym*YI)/A,...
    Vpisym  == sqrt(2*ksym*xpi/(e0*N*Lol*T)*(1/(xo-xpi)^2-...
    1/(xb+xpi)^2)^(-1)),...
    Vminsym == (xo-xpi)/(xo-YI)*Vpisym,...
    Vpisym,Vminsym,ksym);

% Vpi - pull-in voltage Vmin - voltage for minimum output force kmax -
% max spring constant V - voltage value to pass onto dynamics equations
Vpi  = double(vpa(Vpi));
VminArr(i) = double(vpa(Vmin));
kmax = double(vpa(kmax));
V=VminArr(i);

% Fxo - force at engagement
Fxo=1/2*N*e0*VminArr(i)^2*T*Lol*(1/(xo-YI)^2-1/(xb+YI)^2);

% dynamics factors damping factors S1 - max between Lol and T S2 - min
% between Lol and T beta - damping constant dependent on aspect ratio
S1=max(Lol,T);
```

```
    S2=min(Lol,T);
    if(Lol/T<=1)
        beta=-0.6*Lol/T+1;
    else
        beta=-0.6*T/Lol+1;
    end

    % mass factors Arot - layout area of GCA array rotor for couette flow
    % damping m - mass of GCA array rotor
    Arot=N*Lmax*wf+L1tot*Lbase;
    Ap=N*Lmax*wf+LA*L1tot;
    m=Arot*T*rho_si;

    % spring factors
    k=kmax;
%     kArr=linspace(0,k); for j=1:length(kArr)
%         uMech=(Fxo-kArr(i)*YI)*dx; uElec=e0*VminArr(i)^2*N*Lol*T/xf;
%         eta(j)=uMech/uElec;
%     end figure(1) plot(kArr,eta) count=1; x=linspace(0,xo-xf); for
%     j=1:length(x)
%         if x(j)<=YI
%             Farray1(j)=1/2*N*e0*VminArr(i)^2*Lol*...
%             T*(1/(xo-x(j))^2-1/(xb+x(j))^2)-k*x(j);
%             Farray2(j)=1/2*N*e0*VminArr(i)^2*Lol*...
%             T*(1/(xo-x(j))^2-1/(xb+x(j))^2)-k*x(j);
%         else
%             xen(count)=x(j);
%             Farray1(j)=1/2*N*e0*VminArr(i)^2*Lol*...
%             T*(1/(xo-x(j))^2-1/(xb+x(j))^2)-k*x(j);
%             Farray2(j)=1/2*N*e0*VminArr(i)^2*Lol*...
%             T*(1/(xo-x(j))^2-1/(xb+x(j))^2)-k*x(j)-karm*(x(j)-YI);
%             Fx(count)=1/tand(phi)*(1/2*N*e0*...VminArr(i)^2*Lol*T*
%             (1/(xo-x(j))^2-1/(xb+x(j))^2)-k*x(j)-karm*(x(j)-YI));
%             count=count+1;
%         end
%     end figure(2) Fxolin=1/tand(phi)*(Fxo-k*YI)*ones(1,length(xen));
%     plot(x*1e6,Farray1*1e6,x*1e6,Farray2*1e6,xen*1e6,Fx*1e6,xen*1e6,Fxolin*1e6)
%end


    % define initial conditions and time range for numerical solution
    refine=4;
    xinit1=[0,0];
    tspan=0:1e-9:1e-3;
    options1 = odeset('Events',@eventsPI,'OutputSel',1,'Refine',refine);
    options2 = odeset('Events',@eventsPO,'OutputSel',1,'Refine',refine);
    [t1,x1,tpiArr,yepi,iepi]=ode45(@pull_in,tspan,xinit1,options1);

    vinit=calc_vinit(Lmax,m);
    xinit2=[xo-xf,vinit];
    [t2,x2,tpoArr,yepo,iepo]=ode45(@release,tspan,xinit2,options2);

    fmax(i) = 1/(2*tpiArr+tpoArr);
    FshutYI(i) = 1/(tand(phi))*(Fxo-k*YI);
    Fd(i)=FshutYI(i)/(A*T*rho_si);

    uMech=FshutYI(i)*dxshut;
    Pdmax(i)=fmax(i)*uMech./((3*T*2/3)*rho_si*A);
    Cfing=e0*N*Lol*T/xf;
    uIn=Cfing*VminArr(i)^2;
    uOut=1/2*k*(xo-xf)^2+1/2*karm*dx^2;
    eta(i)=uMech/(uIn-uOut);
    %     Fxolin=FshutYI*ones(1,length(x)); plot(x,Fx,x,Fshut,x,Fxolin)
end
figure(1)
hold on
plot(Tarr*1e6,fmax*1e-3)
figure(2)
hold on
plot(Tarr*1e6,Pdmax)
figure(3)
hold on
plot(Tarr*1e6,eta)
figure(4)
hold on
plot(Tarr*1e6,VminArr)

function dxdt = pull_in(t,x)
global V e0 mu m S1 S2 k xo beta N xb h0 Arot YI karm
% define differential equation for pull-in analysis

% electrostatic force
Fes = (1/(xo-x(1))^2-1/(xb+x(1))^2)*1/2*N*e0*V^2*(S1)*(S2);
```

```
% damping
b2=mu*Arot/h0;
Fd  = ((1/(xo-x(1))^3+1/(xb+x(1))^3)*N*beta*mu*S1*S2^3+b2)*x(2);

if x(1)<YI
    % spring
    Fk  = k*x(1);
else
    Fk  = k*x(1)+karm*(x(1)-YI);
end

% setup ODE to solve
dxdt = [x(2); (Fes - Fd - Fk)/(m)];
end

function dxdt = release(t,x)
global mu m S1 S2 k xo beta N xb h0 Arot karm YI
% define differential equation for release analysis

% damping
b2 = mu*Arot/h0;
Fd  = ((1/(xo-x(1))^3+1/(xb+x(1))^3)*N*beta*mu*S1*S2^3+b2)*x(2);

% spring
if x(1)<YI
    % spring
    Fk  = k*x(1);
else
    Fk  = k*x(1)+karm*(x(1)-YI);
end
% setup ODE to solve
dxdt = [x(2); (-Fd - Fk)/(m)];
end

function [value,isterminal,direction] = eventsPI(t,x)
% define end state for pull-in analysis (gap equals the final gap)
global xo xf
value = xo-x(1)-xf;
isterminal = 1;
direction = -1;
end
```

```
function [value,isterminal,direction] = eventsPO(t,x)
% define end state for pull-in analysis (gap equals the initial gap)
% end defined as within 100th of the gap size
global xo xf
value = x(1)-(xo-xf)/100;
isterminal = 1;
direction = -1;
end

% calculate initial velocity
function vinit = calc_vinit(LArr,mArr)
global V e0 T xf N wf rho_si E Ls L1tot Lbase
alpha1=1.875;

% find max velocity of finger upon release, from strain energy in the
% finger's deflection
If=wf^3*T/12;
mf0=T*wf*rho_si;
omega=alpha1^2*sqrt(E*If/(mf0*(LArr+Ls)^4));
kf=6.9*E*If/(LArr+Ls)^3;
mfeff=kf/omega.^2;
Fxf=1/2*e0*V^2*LArr*T*(1/(xf)^2);
xfing=Fxf/kf;
varr=-omega*xfing;

% find max velocity of the GCA shuttle upon release, from strain energy
% stored in the shuttle's deflection
Lshut=L1tot;
wshut=Lbase;
Ashut=T*wshut;
kshut=E*Ashut/Lshut;
xshut=N*Fxf/kshut;
mshut=T*wshut*Lshut*rho_si;
omegashut=sqrt(kshut/mshut);
vshut=-omegashut*xshut;

% find initial velocity of entire GCA assembly using momentum transfer
    vinit=(N*mfeff*varr+mshut*vshut)/mArr;
end
```

Bibliography

[1] Andrew T. Baisch, Onur Ozcan, Benjamin Goldberg, Daniel Ithier, and Robert J. Wood. “High speed locomotion for a quadrupedal microrobot”. In: *The International Journal of Robotics Research* 33.8 (2014), pp. 1063–1082. URL: http://journals.sagepub.com/doi/abs/10.1177/0278364914521473 (visited on 02/22/2017).

[2] Remo Brühwiler, Benjamin Goldberg, Neel Doshi, Onur Ozcan, Noah Jafferis, Michael Karpelson, and Robert J. Wood. “Feedback control of a legged microrobot with on-board sensing”. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 5727–5733. URL: http://ieeexplore.ieee.org/abstract/document/7354190/ (visited on 02/22/2017).

[3] Michael Rubenstein, Christian Ahler, and Radhika Nagpal. “Kilobot: A low cost scalable robot system for collective behaviors”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3293–3298. URL: http://ieeexplore.ieee.org/abstract/document/6224638/ (visited on 02/22/2017).

[4] Ron Pelrine, Annjoe Wong-Foy, Brian McCoy, Dennis Holeman, Rich Mahoney, Greg Myers, Jim Herson, and Tom Low. “Diamagnetically levitated robots: An approach to massively parallel robotic systems with unusual motion properties”. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 739–744. URL: http://ieeexplore.ieee.org/abstract/document/6225089/ (visited on 03/21/2017).

[5] Ron Pelrine, Annjoe Wong-Foy, Allen Hsu, and Brian McCoy. “Self-assembly of milli-scale robotic manipulators: A path to highly adaptive, robust automation systems”. In: *Manipulation, Automation and Robotics at Small Scales (MARSS), International Conference on*. IEEE, 2016, pp. 1–6. URL: http://ieeexplore.ieee.org/abstract/document/7561728/ (visited on 03/21/2017).

[6] Bruce R. Donald, Christopher G. Levey, Craig D. McGray, Igor Paprotny, and Daniela Rus. “An untethered, electrostatic, globally controllable MEMS micro-robot”. In: *Journal of microelectromechanical systems* 15.1 (2006), pp. 1–15. URL: http://ieeexplore.ieee.org/abstract/document/1588903/ (visited on 02/04/2017).

[7] Dana Vogtmann, Ryan St Pierre, and Sarah Bergbreiter. "A 25 MG magnetically actuated microrobot walking at >5 body lengths/sec". In: *Micro Electro Mechanical Systems (MEMS), 2017 IEEE 30th International Conference on.* Las Vegas, Nevada, USA: IEEE, 2017, pp. 179–182. URL: `http://ieeexplore.ieee.org/abstract/document/7863370/` (visited on 03/21/2017).

[8] Ryan St. Pierre and Sarah Bergbreiter. "Gait Exploration of Sub-2 g Robots Using Magnetic Actuation". In: *IEEE Robotics and Automation Letters* 2.1 (Jan. 2017), pp. 34–40. ISSN: 2377-3766, 2377-3774. DOI: `10.1109/LRA.2016.2523603`. URL: `http://ieeexplore.ieee.org/document/7395294/` (visited on 02/28/2018).

[9] Thorbjörn Ebefors, Johan U. Mattsson, Edvard Kälvesten, and Göran Stemme. "A walking silicon micro-robot". In: *Proc. Transducers 99.* 1999, pp. 1202–1205. URL: `https://www.researchgate.net/profile/Goeran_Stemme/publication/228602606_A_walking_silicon_micro-robot/links/5609163008ae576ce63dd036.pdf` (visited on 02/04/2017).

[10] Ken Saito, Kazuto Okazaki, Tatsuya Ogiwara, Minami Takato, Katsutoshi Saeki, Yoshifumi Sekine, and Fumio Uchikoba. "Locomotion Control of MEMS Microrobot Using Pulse-Type Hardware Neural Networks". en. In: *Electrical Engineering in Japan* 186.3 (Feb. 2014), pp. 43–50. ISSN: 04247760. DOI: `10.1002/eej.22473`. URL: `http://doi.wiley.com/10.1002/eej.22473` (visited on 02/21/2017).

[11] Ken Saito, Kei Iwata, Yuki Ishihara, Kazuki Sugita, Minami Takato, and Fumio Uchikoba. "Miniaturized Rotary Actuators Using Shape Memory Alloy for Insect-Type MEMS Microrobot". en. In: *Micromachines* 7.4 (Mar. 2016), p. 58. ISSN: 2072-666X. DOI: `10.3390/mi7040058`. URL: `http://www.mdpi.com/2072-666X/7/4/58` (visited on 02/21/2017).

[12] Mingjing Qi, Yangsheng Zhu, Zhiwei Liu, Xiaoyong Zhang, Xiaojun Yan, and Liwei Lin. "A fast-moving electrostatic crawling insect". In: *Micro Electro Mechanical Systems (MEMS), 2017 IEEE 30th International Conference on.* Las Vegas, Nevada, USA: IEEE, 2017, pp. 761–764. URL: `http://ieeexplore.ieee.org/abstract/document/7863519/` (visited on 03/21/2017).

[13] Seth Hollar, Anita Flynn, Colby Bellew, and K. S. J. Pister. "Solar powered 10 mg silicon robot". In: *Micro Electro Mechanical Systems, 2003. MEMS-03 Kyoto. IEEE The Sixteenth Annual International Conference on.* IEEE, 2003, pp. 706–711. URL: `http://ieeexplore.ieee.org/abstract/document/1189847/` (visited on 02/16/2017).

[14] M. Takato, Y. Naito, K. Maezumi, Y. Ishihara, Y. Okane, H. Oku, M. Tatani, K. Saito, and F. Uchikoba. "Development of mountable pulse-type hardware neuron model integrated circuit on MEMS microrobot". In: *Electronics Packaging and iMAPS All Asia Conference (ICEP-IACC), 2015 International Conference on.* IEEE, 2015, pp. 413–417.

[15] Aminy E. Ostfeld, Abhinav M. Gaikwad, Yasser Khan, and Ana C. Arias. "High-performance flexible energy storage and harvesting system for wearable electronics". In: *Scientific Reports* 6 (May 2016), p. 26122. ISSN: 2045-2322. DOI: 10.1038/srep26122. URL: http://www.nature.com/articles/srep26122 (visited on 02/26/2017).

[16] Raymond J. Roark, Warren C. Young, and Richard G. Budynas. *Roark's formulas for stress and strain*. 7th ed. New York: McGraw-Hill, 2002, pp. 267–380. ISBN: 978-0-07-072542-3.

[17] Matthew A. Hopcroft, William D. Nix, and Thomas W. Kenny. "What is the Young's Modulus of Silicon?" en. In: *Journal of Microelectromechanical Systems* 19.2 (Apr. 2010), pp. 229–238. ISSN: 1057-7157, 1941-0158. DOI: 10.1109/JMEMS.2009.2039697. URL: http://ieeexplore.ieee.org/document/5430873/ (visited on 06/18/2018).

[18] Jeremy E. Frank, Gary H. Koopmann, Weicheng Chen, and George A. Lesieutre. "Design and performance of a high-force piezoelectric inchworm motor". en. In: ed. by Norman M. Wereley. June 1999, pp. 717–723. DOI: 10.1117/12.350747. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=982164 (visited on 06/18/2018).

[19] John H Comtois and Victor M Bright. "Applications for surface-micromachined polysilicon thermal actuators and arrays". In: *Sensors and Actuators A: Physical* 58 (1997), pp. 19–25.

[20] Murat M. Okyar, Xiqing Sun, and William N. Carr. "Thermally excited inchworm actuators and stepwise micromotors: analysis and fabrication". en. In: ed. by Kevin H. Chau and Patrick J. French. Sept. 1997, p. 372. DOI: 10.1117/12.284538. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.284538 (visited on 06/18/2018).

[21] Richard Yeh, Ezekiel J J Kruglick, and Knstofer S J Pister. "Surface-Micromachined Components for Articulated Microrolbots". en. In: *JOURNAL OF MICROELECTROMECHANICAL SYSTEMS* 5.1 (1996), p. 8.

[22] R. Yeh, S. Hollar, and K.S.J. Pister. "Single mask, large force, and large displacement electrostatic linear inchworm motors". en. In: *Journal of Microelectromechanical Systems* 11.4 (Aug. 2002), pp. 330–336. ISSN: 1057-7157. DOI: 10.1109/JMEMS.2002.800937. URL: http://ieeexplore.ieee.org/document/1022844/ (visited on 02/23/2017).

[23] I Penskiy and S Bergbreiter. "Optimized electrostatic inchworm motors using a flexible driving arm". In: *Journal of Micromechanics and Microengineering* 23.1 (Jan. 2013), p. 015018. ISSN: 0960-1317, 1361-6439. DOI: 10.1088/0960-1317/23/1/015018. URL: http://stacks.iop.org/0960-1317/23/i=1/a=015018?key=crossref.4f6da5044bbf32981d8ee15d86b83c99 (visited on 02/19/2017).

[24] P G Steeneken, Th G S M Rijks, J T M van Beek, M J E Ulenaers, J De Coster, and R Puers. "Dynamics and squeeze film gas damping of a capacitive RF MEMS switch". In: *Journal of Micromechanics and Microengineering* 15.1 (Jan. 2005), pp. 176–184. ISSN: 0960-1317, 1361-6439. DOI: `10.1088/0960-1317/15/1/025`. (Visited on 02/16/2017).

[25] Minhang Bao and Heng Yang. "Squeeze film air damping in MEMS". en. In: *Sensors and Actuators A: Physical* 136.1 (May 2007), pp. 3–27. ISSN: 09244247. DOI: `10.1016/j.sna.2007.01.008`. URL: `http://linkinghub.elsevier.com/retrieve/pii/S0924424707000118` (visited on 02/03/2017).

[26] Minhang Bao and Heng Yang. "Squeeze film air damping in MEMS". In: *Sensors and Actuators A: Physical* 136 (May 2007), pp. 3–27.

[27] Mo Li, Vashwar T. Rouf, and David A. Horsley. "Substrate effect in squeeze film damping of lateral oscillating microstructures". In: *Micro Electro Mechanical Systems (MEMS), 2013 IEEE 26th International Conference on*. Taipei, Taiwan: IEEE, 2013, pp. 393–396.

[28] Daniel S. Contreras and Kristofer SJ Pister. "Dynamics of electrostatic inchworm motors for silicon microrobots". In: *Manipulation, Automation and Robotics at Small Scales (MARSS), 2017 International Conference on*. Montreal, Canada: IEEE, 2017.

[29] Raymond J. Roark, Warren C. Young, and Richard G. Budynas. *Roark's formulas for stress and strain*. 7th ed. New York: McGraw-Hill, 2002, p. 765. ISBN: 978-0-07-072542-3.

[30] Peter M Osterberg and Stephen D Senturia. "M-TEST: A Test Chip for MEMS Material Property Measurement Using Electrostatically Actuated Test Structures". en. In: *JOURNAL OF MICROELECTROMECHANICAL SYSTEMS* 6.2 (1997), p. 12.

[31] H.H. Vaziri and J. Xie. "Buckling of columns under variably distributed axial loads". en. In: *Computers and Structures* 45.3 (Oct. 1992), pp. 505–509. ISSN: 00457949. DOI: `10.1016/0045-7949(92)90435-3`. URL: `http://linkinghub.elsevier.com/retrieve/pii/0045794992904353` (visited on 08/05/2018).

[32] Matthew E. Last, V. Subramaniam, and Kristofer SJ Pister. "A Microassembled Large-Deflection Tip/Tilt Micromirror from a Single-Mask DRIE Process". In: *Solid-State Sensors, Actuators and Microsystems Workshop, Hilton Head Island, South Carolina*. 2006. (Visited on 11/28/2016).

[33] Daniel S. Contreras, Daniel S. Drew, and Kristofer S. J. Pister. "First steps of a millimeter-scale walking silicon robot". In: *Solid-State Sensors, Actuators and Microsystems, Transducers, 2017 19th International Conference on*. IEEE, 2017, pp. 910–913. URL: `http://ieeexplore.ieee.org/abstract/document/7994197/` (visited on 08/16/2017).

[34] Gaopeng Xue, Masaya Toda, and Takahito Ono. "Comb-Drive XYZ-microstage With Large Displacements Based on Chip-Level Microassembly". In: *Journal of Microelectromechanical Systems* 25.6 (Dec. 2016), pp. 989–998. ISSN: 1057-7157, 1941-0158. DOI: 10.1109/JMEMS.2016.2607233. URL: http://ieeexplore.ieee.org/document/7576632/ (visited on 02/23/2017).

[35] Ken Saito, Daniel Contreras, Yudai Takeshiro, Yuki Okamoto, Yuya Nakata, Taisuke Tanaka, Satoshi Kawamura, Minami Kaneko, Fumio Uchikoba, Yoshio Mita, and Kristofer S. J. Pister. "Study on silicon device of microrobot system for heterogeneous integration". en. In: IEEE, Apr. 2018, pp. 33–37. ISBN: 978-4-9902188-5-0. DOI: 10.23919/ICEP.2018.8374664. URL: https://ieeexplore.ieee.org/document/8374664/ (visited on 08/03/2018).

[36] *Robot simulator V-REP*. http://www.coppeliarobotics.com/. 2018.

[37] Liwei Lin, Yu-Ting Cheng, and Khalil Najafi. "Formation of Silicon-Gold Eutectic Bond Using Localized Heating Method". In: *Japanese Journal of Applied Physics* 37.Part 2, No. 11B (Nov. 1998), pp. L1412–L1414. ISSN: 00214922. DOI: 10.1143/JJAP.37.L1412. URL: http://stacks.iop.org/1347-4065/37/L1412 (visited on 07/10/2018).