In Vivo Silicon Lance Array Transfection of Plant Cells

Taylor Andrew Brown
Brigham Young University

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Engineering Commons

BYU ScholarsArchive Citation
Brown, Taylor Andrew, "In Vivo Silicon Lance Array Transfection of Plant Cells" (2020). Theses and Dissertations. 8398.
https://scholarsarchive.byu.edu/etd/8398

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.
In Vivo Silicon Lance Array Transfection of Plant Cells

Taylor Andrew Brown

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Brian Jensen, Chair
Sandra Hope
Mark Colton

Department of Mechanical Engineering
Brigham Young University

Copyright © 2020 Taylor Andrew Brown
All Rights Reserved
Arrays of silicon lances were made using photolithographic and STS DRIE Bosch techniques. Arrays consist of a 10 mm square grid pattern of lances measuring 100 \( \mu \)m tall and having a 3 \( \mu \)m diameter, each lance being spaced 30 \( \mu \)m apart. The tips of lances are pointed, allowing easier penetration through plant cell walls. A nanoinjector device was also made to accept the silicon lance arrays and perform nanoinjections. A nanoinjection consisted of 2 silicon lance arrays, with lances oriented towards each other, being moved into and out of a plant cotyledon placed between them. Prior to the nanoinjection, polar molecules in solution can be attracted to the lances through a process utilizing the nanoinjector device’s ability to control the electrical current between the 2 lance arrays. During the nanoinjection the displacement between the lances, the force exerted on the plant cotyledon and the electrical current between the lance arrays are controlled. Once the lances are inserted into the cells, the electrical current between the lance arrays is reversed, repelling the molecular load from the lance array. Propidium iodide (PI) and Cotton Leaf Crumple Virus (CLCrV) were used as molecular loads in nanoinjections. The nanoinjector also records and outputs data from the nanoinjection for analysis. Nanoinjections were performed on Arabidopsis and Cotton cotyledons. Changes in the force applied during a nanoinjection and varying the number of repeated nanoinjections on the same cotyledon were observed. Too much force or too many repeated injections causes physical damage to the cotyledon. An optimal force and number of repeated injections can be performed without causing physical damage to the cotyledon. Successful transfection of PI and CLCrV was not observed in a relatively small number of performed nanoinjection procedures on either Arabidopsis or Cotton cotyledons. Possible interacting variables and recommendations for further work are discussed.

Keywords: nanoinjection, transfection, CRISPR, silicon, injection, lance, array
# TABLE OF CONTENTS

**LIST OF TABLES** ................................................................. v

**LIST OF FIGURES** ............................................................. vi

**Chapter 1 Introduction** .................................................... 1
  1.1 Prior Nanoinjection Work .............................................. 2
  1.2 Motivation ............................................................... 3
  1.3 Scope ........................................................................ 4

**Chapter 2 Silicon Lance Arrays for Plant Cells** ...................... 5
  2.1 Introduction .............................................................. 5
    2.1.1 Animal Cell Lance Array Geometry ........................... 6
    2.1.2 Changes needed for plant cells ............................... 6
  2.2 Requirements ........................................................... 7
    2.2.1 Material Properties ............................................. 7
    2.2.2 Geometry ............................................................ 7
    2.2.3 Reusability ......................................................... 9
  2.3 Fabrication ............................................................... 11
    2.3.1 Introduction ....................................................... 11
    2.3.2 Photolithography ................................................ 11
    2.3.3 STS Etcher ........................................................ 12
  2.4 Results ..................................................................... 15
    2.4.1 Geometry ............................................................ 16
    2.4.2 Testing ............................................................... 17
    2.4.3 Manufacturing Challenges ...................................... 19
  2.5 Conclusion ............................................................... 19

**Chapter 3 Implementation of Injection System** ....................... 20
  3.1 Intro ....................................................................... 20
  3.2 Design Requirements for Injection Device ......................... 20
    3.2.1 Mounting Silicon Lance Arrays ............................... 20
    3.2.2 Current and Voltage Control ................................. 22
    3.2.3 Space for Plant Specimen ..................................... 23
    3.2.4 Repeatable Force Application ............................... 24
    3.2.5 Other Design Advantages .................................... 25
    3.2.6 Design Disadvantages ........................................ 26
  3.3 Fabrication ............................................................... 26
  3.4 Testing .................................................................. 26
    3.4.1 Electrical ........................................................... 26
    3.4.2 Force ................................................................. 28
    3.4.3 Linear Movement ................................................. 28
  3.5 Conclusion ............................................................... 29
Chapter 4  Implementation Results .................................................. 30
  4.1  Introduction ................................................................. 30
  4.2  Force Experiment ......................................................... 30
    4.2.1  Multiple Injections ................................................. 31
  4.3  Injection Results ........................................................ 33
  4.4  Conclusion ............................................................... 34

Chapter 5  Conclusion ............................................................. 36
  5.1  Conclusion ................................................................. 36
  5.2  Further Work ............................................................. 36
    5.2.1  Possible Interacting Variables ................................. 37
    5.2.2  Recommendations ................................................ 38

REFERENCES ................................................................. 39

Appendix A  Supplemental Material ............................................. 42
  A.1  Injection Device BOM .................................................. 42
  A.2  Microcontroller Code .................................................. 42
  A.3  Graphing Code - Python ............................................... 60
LIST OF TABLES

3.1 Nanoinjector Movement .................................................. 28
4.1 Significant Nanoinjection Experiments .............................. 34
2.1 A computer generated comparison of the two geometries of lances and spacing side by side. Animal cell lances are on the left and plant cell lances are on the right. 6

2.2 A light microscope image of the surface of a silicon lance array after a nanoinjection. The chip has not been cleaned and surface contaminants are still visible. Missing lances in the lance array are also evident. See arrows. 10

2.3 A light microscope top view of a silicon wafer that sustained damage at the edge of the lance array. This type of damage happened while manipulating the lance array into place for a nanoinjection procedure. 11

2.4 A light microscope side view of a damaged lance array. The lances in the plane of focus are gone, having snapped when too much force was applied laterally at the tip of the lances. 12

2.5 Silicon lance etching progression. Step One: Photoresist is patterned onto the silicon wafer. Step Two: Isotropic etch step undercuts photo-resist. Step Three: Vertical anisotropic etch removes the bulk of the silicon, exposing the lance shaft. Step Four: Final isotropic etch forms the final geometry of the lance. 13

2.6 wafer 2-1-17-2018 post bake 14

2.7 A light microscope top view of the silicon wafer before the final dicing step. Some lances have been intentionally knocked over with a razor blade so that their final height can be measured. 15

2.8 A light microscope side view image of the silicon lance array. Due to excessive light reflection during imaging, black ink was applied to the edge of the lance array. The application of black ink left behind dark contaminant on the first row of lances. 16

2.9 A modified tweezer injection device used for plant nanoinjections. 17

2.10 SEM image of the surface of an Arabidopsis leaf used for nanoinjections. Lances were intentionally sheared off of the silicon wafer mid injection and several lances can still be seen partially inserted into the surface of the leaf. 18

3.1 Top is a picture of the nanoinjector device. Bottom is isometric, front and right views of the nanoinjector device exported from a CAD program. 21

3.2 Schematic and circuit drawing of the nanoinjector control board. 23

3.3 Nanoinjection procedure overview. 1) DNA in solution is pipetted onto lower lance array. 2) The lance arrays are charged to attract DNA to either lower or upper lance array. 3) The lance arrays are separated but still charged. 4) The plant specimen is positioned and the lance arrays are inserted. The lance arrays undergo a reversed charge pulsing cycle to repel the DNA. 5) The lance arrays are separated and the plant specimen is removed. 27

4.1 Experimental nanoinjection results for varying the force applied to the plant specimen. Significant damage occurs to the plant specimen above a 20 N force, or a force of 0.18 mN per lance. 31

4.2 Experimental nanoinjection results for varying the number of sequential injections applied to a plant specimen. For varying numbers of sequential injections 1, 5 and 10 no additional damage was seen on the plant specimens. 32
4.3 Experimental nanoinjection results for varying the number of sequential injections applied to a plant specimen. No damage was seen on the plant specimen injected 10 sequential times but additional damage on the surface of the plant specimen was observed at the injection site for sequential injections of 15 and 30 times at 7 dpi.

4.4 Experiment Number 17 results. The CLCrV was not manifested in the treatment plant specimen. The control plant specimens remained unchanged and unharmed by the nanoinjection procedure.
CHAPTER 1. INTRODUCTION

The basic unit of life is the cell [1]. Introducing foreign molecules into the cell is difficult in large part because of the small size of the cell [2]. Cells are microscopic but contain complex processes that affect every aspect of the organism [3]. The information to carry out these processes is protected by the cell membrane, as well as the cell wall for plant cells [2]. These semi-permeable structures allow certain molecules or objects to pass into the cell and bar others [4] [5]. Delivering specific synthesized molecular payloads, such as gene editing CRISPR proteins, into the cell can alter the behavior of the cell and its DNA [6]. The DNA inside every cell nucleus contains the instructions for what it will become, how it will act and how it will respond to various stimuli throughout its life [7]. Each cell also has an effect on surrounding cells [8]. Successfully altering the DNA can cause the cell to grow or behave differently and cause the altered genome to be passed on to its descendants [9]. The altered genome of a single cell or group of cells within a plant can then be passed on to become a significant part of the plant [9]. If the altered part of the plant produces seeds, those seeds will also contain the altered genome. Manipulating the DNA of plants allows specific traits to be emphasized, resulting in more desirable plants [10].

Techniques for delivering molecular loads into a cell fall into three separate categories. Each has advantages and disadvantages; the ability to transfect many cells is usually sacrificed in order to not kill a large percentage of cells [11] [12].

• Chemical - Chemical vectors have a low transfection rate. Increasing the concentration can result in higher cell transfection rates but also higher rates of cell death [13].

• Viral - Viral vectors have high transfection rates but can be expensive and dangerous due to immune responses [14].

• Physical - Physical methods can be less dangerous and expensive but can also be limited in the ways cells can be isolated and immobilized [15].
1.1 Prior Nanoinjection Work

Animal cells have been successfully injected with polar molecular payloads at BYU using an electro-mechanical injection method called nanoinjection [16]. The animal cells selected for initial testing of the nanoinjection process were immortal HeLa cells and were injected in vitro. The cells grew a single layer thick on the surface of the petri dish and were then used to test the process of nanoinjection.

The nanoinjection process utilizes an array of pointed solid lances fabricated on the surface of a silicon wafer to pierce and deliver molecular loads inside target cells. Unlike needles, the lances do not have a hollow core where the injection load is stored. Instead, molecules are delivered inside the cell by adhering to the lance itself as well as diffusion through the gaps in the cell membrane left by the lances. This process is enhanced by electrically manipulating the lance arrays to produce electroporation as well as to attract and repel the molecular payload at key points in the nanoinjection procedure.

The single layer of HeLa cells provided a flat surface advantageous to the nanoinjection process because the array of nanoinjection lances are made on the surface of a flat silicon wafer, allowing the two surfaces to mate completely. The lance array was designed in a rectangular pattern where lances were spaced 10 \( \mu \text{m} \) apart, which corresponded to the spacing of the HeLa cells. Although the HeLa cells are irregularly shaped and spaced, each cell had a high probability of being injected at least once by one of the lances in the lance array.

This work began by creating an array of microscopic silicon lances using a photolithographic process to produce the requisite mask and then an etching process to produce the long shaft of the lances and pointed tip. The lances are fragile and can be damaged easily but they can be re-used for multiple nanoinjection procedures.

After the lance array was made it was mounted to the injection device. The injection device was responsible for moving the lances into and out of the HeLa cell specimen while controlling the force applied and the voltage potential between the lance arrays. First, a positive voltage is applied to the array of lances which attracts the negatively charged molecules in solution. The lances are then moved downward into the HeLa cell culture where they pierce the cell membranes. The voltage is then reversed, causing the polar molecules to be repelled by the lances into the interior of the cell. The lances are then removed.
Nanoinjection of propidium iodide (PI) and green fluorescent protein plasmid were successfully tested as injection payloads. The factors such as the maximum force applied, the material used for the lances and the injection speed were tested. It was shown that the method has resulted in a high level of efficiency and cell viability compared to other transfection methods [11].

1.2 Motivation

More advanced nanoinjection has the potential to increase transfection efficiency, increase the types of cells that can be transfected and reduce the cost of transfection. Easy, fast and cheap transfection would be a boon to academic and commercial development of transgenic organisms. For example, genome editing has potential for increasing crop yields and for a subsequent economic impact [17]. It was proposed to apply nanoinjection technology to in vivo plant cell specimens, including Arabidopsis and Cotton.

Transfection efficiency is particularly high with nanoinjection because of the ability to introduce a molecular payload to many cells simultaneously without causing the cell irreparable physical damage resulting in cell death. Other physical injection techniques, such as microinjection, result in greater trauma and lower survival rates than nanoinjection. [18].

Nanoinjection also has the potential to be applied successfully to in vivo plant specimens. This allows the analysis of post-transfection results to be observed much more quickly in some scenarios than a pronuclear transfection. Nanoinjection gene targeting, utilizing a tool such as CRISPR-Cas9, would allow faster expression of applied genes and development of gene therapies [19].

Nanoinjection has the potential to be cheaper than other transfection processes. Nanoinjection only consumes electricity, without requiring any expensive materials. The most expensive component of nanoinjection is the lance arrays. The material is silicon, which is widely available and not cost prohibitive, but the process to create the lances from the silicon requires specialized equipment. The individual cost of each lance array could be brought down significantly when producing larger amounts of arrays. The nanoinjection lances are also reusable through a cleaning process that takes place in between nanoinjections. Being able to repeat nanoinjections with the same equipment reduces the costs of transfection.
1.3 Scope

This work was focused on the mechanical implementation of the nanoinjection system for plant cells, comprising the silicon lance arrays and the injector device. Nanoinjections were performed in order to test the nanoinjection system but more tests are needed to confirm the viability or non-viability of nanoinjection as a transfection method for plants. Only 3 cotton plants were attempted to be transfected with CLCrV using the final nanoinjection parameters, this sample size is not adequate to provide a conclusive recommendation on whether nanoinjection is a viable transfection method for plants. A full testing of at least 100 injection samples, as demonstrated by McGarry, Roisin and Ayre [20], should be done to make a conclusive recommendation.

There are significant differences nanoinjecting in vivo plant cells when compared to in vitro animal cells. This work is focused on overcoming those differences in order to create a viable cell transfection method. Some of these differences include the non-uniform surface of plants and the additional fortification of the cell wall. In order to overcome the differences between transfection of plant and animal cells a modified method was developed.

This new nanoinjection method required a change in lance array geometry as well as a change in the nanoinjector device. These changes are discussed in the following chapters and then the implementation results are shown.
CHAPTER 2. SILICON LANCE ARRAYS FOR PLANT CELLS

2.1 Introduction

Nanoinjection requires microscopic lances to deliver macro molecules through the cell wall and to deposit them into the interior of the cell. These microscopic lances are produced using a subtractive manufacturing technique from a flat silicon wafer. The manufacturing process creates thousands of lances in separate rectangular arrays, spaced approximately a cell’s width apart [12]. During nanoinjection these arrays are linearly advanced in and out of the injection sample causing thousands of injections simultaneously where individual lances pierce individual cells [21] [22].

Nanoinjecting plant cells is significantly different than nanoinjecting animal cells and requires changes to the geometry of the silicon lance arrays. The geometrical differences can be seen in Figure 2.1. The changes to geometry also require additional changes to the existing manufacturing process. The changes are designed to maximize transfection efficiency and are necessitated by the differences between injection samples as well as cell types [23]. Previous work done using animal cells was done with in vitro HeLa cells while this work is done using cotton and arabidopsis cotyledons in vivo.

The difference in geometry of the sample requires a change to the lance array geometry. The animal cell samples were HeLa cells grown in a single layer on a flat bottomed petri dish. The top of the cells was the injection surface and was flat and uniform [12]. Each animal cell sample was geometrically very similar. In contrast, each plant sample was a unique cotyledon (the first leaves developed during germination), which has its own unique curved surface, orientation, shape, and size [24]. Cotyledons are also many cell layers thick instead of just one. Plant cells also have a cell wall in addition to a cell membrane [25].

These differences necessitate a modified nanoinjection approach. The main differences are plant cell lances are longer and thicker than animal cell lances in order to penetrate through multiple cell layers. Plant cell lances are also spaced further apart because the individual plant cells
Figure 2.1: A computer generated comparison of the two geometries of lances and spacing side by side. Animal cell lances are on the left and plant cell lances are on the right.

are larger than the previously-used HeLa cells. The array is designed to potentially inject each cell one or more times.

2.1.1 Animal Cell Lance Array Geometry

In previous work, the nominal geometry of each silicon lance was 1-1.5 μm in diameter, 10 μm tall and spaced in a rectangular grid pattern 10μm apart on a 2 cm square chip, as shown in Figure 2.1. Therefore each individual chip contained about 4 million lances. The geometry of the lance array follows from the size and spacing of the sample cells being targeted. The HeLa cells being targeted were about 10 μm in diameter, irregularly shaped and spaced together.

2.1.2 Changes needed for plant cells

The changes made to the silicon lance arrays for plant cell nanoinjection are geometrical. Plant cell lances are longer, thicker and spaced farther apart from each other due to the size and spacing of plant leaf cells. Arabidopsis cells are near 50 μm in diameter and irregularly spaced, with some cells being larger and some smaller [26]. Instead of attempting to inject only the first layer of cells of the sample the longer lances are able to inject several layers of cells with a given
molecular payload. This is advantageous not only for injecting more cells but because the cells located in the middle of the leaf have a higher rate of cell division [27].

Plant cell arrays need to have smaller chip sizes because the surface area of the injection sample is smaller. Some cotyledons may have an injectable area of less than 3 square centimeters, and the lance array must fit completely within the area for an accurate injection. If the lance array hangs over the edge of the leaf then those lances are not able to inject a cell and the force exerted per lance increases an unpredictable amount.

2.2 Requirements

2.2.1 Material Properties

The electrical conductivity of the lance array is important. During nanoinjection the voltage and current between the two arrays are controlled in order to maximize the transfection efficiency. If the material used is not conductive then no current will be able to flow and the attraction and repulsion of polar molecules will be unattainable. Pure silicon is a poor conductor of electricity but doped silicon wafers are much better conductors of electricity [28]. The wafers used in this work have a resistivity of 1-5 ohm-cm. The lances are attached to a flat electrical probe with a conductive carbon tape.

It is important that the lances do not buckle, bend or snap during the nanoinjection procedure. If the lance breaks then it can no longer complete the electrical circuit. It also cannot be removed from the sample or fully inserted into the sample. The ideal material would be very rigid and also very strong. Silicon is similar to glass and can fracture dramatically but is very rigid. In order to minimize fracture the lances are only intended to be subjected to compression and not exposed to a lateral force.

2.2.2 Geometry

The nominal lance height for nanoinjection of plant cells in this work is 100 µm. The nominal lance diameter is 3 µm. The size of each chip is 11 mm square with a 10 mm square lance array in the center. This leaves the edges of each chip empty of lances. The empty edges
make it easier to manipulate lance arrays without damaging the lances, such as when installing and uninstalling the lance arrays on the injector device.

The lances are spaced 30 \(\mu\)m apart in a rectangular array because the target plant cells are near 30 \(\mu\)m in their longest dimension. This way each plant cell will likely be injected at least once. The tip of each lance is pointed in order to pierce the cell wall without excessive damage. The lances have vertical walls in order to be inserted into the sample their entire length. Figure 2.1 shows a computer model of an array for injecting plant cells next to the array designed for HeLa cell injection.

An individual lance can be modeled as a vertical column fixed at one end and free at the opposite end. Euler’s critical load can be calculated using the nominal lance geometry and silicon’s material properties.

\[
F = \frac{\pi EI}{(KL)^2}
\]

where
F = vertical load on column
E = modulus of elasticity
R = radius of the column
I = smallest area moment of inertia of the cross section of the column
L = unsupported length of the column
K = column effective length factor (one end fixed, one end pinned = 0.707)

For

\[
E = 195 \text{ GPa}
\]
\[
R = 1.5e^{-6} \text{ m}
\]
\[
I = \pi R^4/4 = 3.98e10^{-24} \text{ m}^4
\]
\[
L = 1e^{-4} \text{ m}
\]
\[
K = 0.707
\]
The critical buckling load for a single lance is,

\[ F = 0.41 \text{ mN} \]

When each lance experiences the critical buckling load the total force applied to the lance array is,

\[ F_{\text{array}} = F_{cr} \times NL \]

where

\( F_{cr} \) = critical buckling load

NL = number of lances per lance array

For

\[ F_{cr} = 0.41 \text{ mN} \]

\[ NL = 110,889 \text{ lances} \]

The total force applied to the lance array is

\[ F_{\text{array}} = 45.464 \text{ N} \]

2.2.3 Reusability

Lance arrays that have not been damaged after being used for nanoinjection can be reused. They should be cleaned before reuse to remove organic material and the injection molecular payload. Organic residue left over from nanoinjection can be seen on a silicon lance array in Figure 2.2. In order to do this the lance array is rinsed using organic solvents acetone and isopropynol followed by an oxygen plasma etching. During cleaning lance arrays are handled by hand or with tweezers. They are stored in a container designed to constrain them by their free edges in order not to damage the individual lances. If the lances are too close to the edge of the silicon chip they can be damaged when moving them by holding the edges. Edge damage can be seen on the chip in Figure 2.3. Individual lances are too small to see with the naked eye so to assess damage
Figure 2.2: A light microscope image of the surface of a silicon lance array after a nanoinjection. The chip has not been cleaned and surface contaminants are still visible. Missing lances in the lance array are also evident. See arrows.

A microscope is required. Most damage can be assessed with a light microscope; missing lances usually appear in regions near the edges of the lance array. Figure 2.4 shows a side view taken with a light microscope of an array that was damaged. As shown in Figure 2.4 damaged lances usually fracture where the lance connects to the rest of the wafer. The fracture event usually results in the lance being launched off the wafer so all that remains is a pock mark where a lance used to be. If a region of the lance array has missing lances then that portion of the chip will not result in successful injections. A nanoinjection procedure does not usually result in damage to the lance array. Even though direct contact occurs between the lances and the sample the injector device moves linearly along one axis which results in a largely uniaxial compressive force.
Figure 2.3: A light microscope top view of a silicon wafer that sustained damage at the edge of the lance array. This type of damage happened while manipulating the lance array into place for a nanoinjection procedure.

2.3 Fabrication

2.3.1 Introduction

Lance arrays are made from a single piece of silicon in wafer form. Silicon is etched away from the surface of the wafer until an array of lances remain. The process takes place using a modified photolithography procedure followed by isotropic and anisotropic etching steps as seen in Figure 2.5. The wafer is then diced into individual arrays used for nanoinjection.

2.3.2 Photolithography

First, a chromium plated mask was made for manufacturing the plant cell lance arrays. The mask consisted of chromium circles 5 \( \mu \text{m} \) in diameter spaced 30 \( \mu \text{m} \) apart. After a positive
photoresist photolithographic procedure a pattern of short photoresist cylinders was created on the silicon surface as seen in Figure 2.6.

The first step in this process is to clean and then dehydrate a wafer. Photoresist is then spun on followed by a soft bake. Then the photoresist is exposed through the mask and then developed. After a longer than typical hard bake the photoresist layer height is measured and then it is ready to be etched. The longer hard bake results in a photoresist layer that is more durable during the etching steps. The etching procedure is corrosive to the photoresist layer so the photoresist layer must meet a certain height in order to not be totally eaten away.

2.3.3 STS Etcher

Then, an isotropic etch undercuts the photoresist and starts the pointed tip of the lance. A long anisotropic etch removes the bulk of the material by etching vertically downward. After measuring the lance height the photoresist is removed followed by final isotropic etch. The final isotropic etch finishes the tip of the lance and shrinks the diameter.

It is important to verify the etching depths to confirm the lance geometry using a profilometer during the manufacturing process. The profilometer measures the profile of the surface of the
Figure 2.5: Silicon lance etching progression. Step One: Photoresist is patterned onto the silicon wafer. Step Two: Isotropic etch step undercuts photo-resist. Step Three: Vertical anisotropic etch removes the bulk of the silicon, exposing the lance shaft. Step Four: Final isotropic etch forms the final geometry of the lance.

After etching the 100 mm wafer is diced into 41 lance arrays. The wafer is diced in order to free the individual lance arrays from each other so they can be used one at a time for nanoinjection.
Figure 2.6: A light microscope top view of the silicon wafer after the final hard bake and before the wafer has been etched. There are 5 \( \mu \text{m} \) circles of photoresist patterned in a 30 \( \mu \text{m} \) spaced rectangular array.

After dicing the wafer into individual arrays they are measured in order to be ready to be used for nanoinjections.

In addition to the profilometer the geometry of the lance arrays are measured by taking pictures of the lance arrays and then using a computer measuring tool. The computer measuring tool is calibrated by using the known size of a feature in the image, usually the spacing of the individual arrays, which is 30 \( \mu \text{m} \). The geometry that can be measured this way using a light microscope is the length, or height, of the individual lances and their diameter. A light microscope can also be used to identify damage that has occurred to the lance array or the location of missing lances. In order to view the lance height using a light microscope some sacrificial lances are intentionally broken which causes them to lay horizontally on the wafer surface. Without breaking the lances it is difficult to view the lances on some light microscopes because of lack of a convenient way to hold the lance array on its side. This is especially difficult before the wafer has been diced.
Figure 2.7: A light microscope top view of the silicon wafer before the final dicing step. Some lances have been intentionally knocked over with a razor blade so that their final height can be measured.

because of the large size of the silicon wafer and small viewing volume of many laboratory light microscopes.

The tip of the lance is more difficult to measure because of differences in tip shape as well as being difficult to bring into focus. The tip of the lance can be seen in the light microscope images, but not as clearly as in the scanning electron microscope images. The lance arrays can be seen clearly using an SEM.

2.4 Results

A process to generate silicon lance arrays suitable for injecting plant cells was generated, tested and verified. Silicon lance arrays were fabricated and then measured to verify they conformed with the required plant cell lance array geometry.
2.4.1 Geometry

Varying lance lengths were made in the process of creating a manufacturing process for the generation of lance arrays. The maximum length of the lances generated reached up to 120 µm. The manufacturing process is suitable to produce lances even longer as well as any shorter length. The process to manufacture lances allows for a wide range of lance height but not a wide range of lance diameter. The final length of the lance was measured by breaking a sacrificial lance so that it layed perpendicular to the imaging plane of a light microscope, as shown in Figure 2.7. The length of the lance could then be measured using an imaging software by comparing the lance length with the known spacing of the lances in the lance array. A second method of verifying the geometry is by viewing the silicon lance array by positioning the lance array on its side relative to the microscope, as seen in Figure 2.8. This is a difficult way to image the silicon lances if the
Figure 2.9: A modified tweezer injection device used for plant nanoinjections.

light microscope being used is set up to image flat glass slides because the silicon lance array is too large to fit under the lens of the microscope.

The diameter of the lances generated is nominally 3 \( \mu \text{m} \). The diameter of the lance is coupled with the shape of the tip of the lance as well as the geometry of the mask. The lance diameter can be changed by etching shorter or longer. The longer the etch the smaller the diameter of the lance. The tip reaches an optimum shape when it is most pointed. Etching longer or shorter blunts the pointed shape of the tip of the lance. In order to change the diameter of the lance without sacrificing the shape of the tip a mask must be made with different sized circles.

The spacing and geometry of the lances is controlled by the mask. Lances are spaced in a rectangular array 30 \( \mu \text{m} \) apart. In order to change the spacing width a new mask must be created.

2.4.2 Testing

Initial testing of the silicon lance arrays was done on vivo young Arabidopsis leaves. Arabidopsis was chosen as an advantageous plant for initial nanoinjection testing because of previous work where its genome was successfully targeted [29]. After evidence of successful initial testing of the silicon lance arrays for plant cells was observed the primary plant used in this work was changed from Arabidopsis to Cotton. This was done because of the requirements for funding and the ability of cotton to greater highlight the potential for nanoinjection to improve crop yields, nutritional quality and other advantages [30] [31]. The testing began with two lance arrays that were inserted on either side of the Arabidopsis leaf in a clamping fashion using a modified tweezer injection device shown in Figure 2.9. After insertion the lance arrays were slid away from each
Figure 2.10: SEM image of the surface of an Arabidopsis leaf used for nanoinjections. Lances were intentionally sheared off of the silicon wafer mid injection and several lances can still be seen partially inserted into the surface of the leaf.

other in an effort to shear the lances from the silicon base. The lance arrays were then removed from the leaf. The leaf was prepared for imaging in the SEM.

Imaging in the SEM, see Figure 2.10, showed evidence that at least some lances had been inserted into cells of the Arabidopsis leaf. This was shown by the location and position of the lances on the leaf surface. Many lances are seen laying horizontally on the surface of the leaf, while others are held erect by the tip of the lance being encompassed by the leaf surface.
2.4.3 Manufacturing Challenges

Manufacturing on the microscopic scale brings its own challenges [32]. Steps cannot be visually confirmed and so more expensive and time consuming checks must be performed.

The photoresist itself is subject to the etching processes and is etched away as well as the silicon but at a different rate. The measurements taken by the profilometer are in relation to the photoresist so when the photoresist is etched the true etch depth of the silicon is skewed. This makes it difficult to measure the actual silicon etch depth with the profilometer.

One of the most delicate steps is the final anisotropic etching phase where the final lance is produced, including sharpening the point of the lance. This final step doesn’t result in a very large change in the overall length of the lance and so its effect can be hard to measure using just a light microscope. In addition there is randomness to what the tips of the lances end up looking like. Their shape can only be confirmed with an electron microscope which can be time consuming and relatively expensive to operate.

2.5 Conclusion

A process for manufacturing silicon lance arrays for plant cells was created and silicon lance arrays suitable for use in nanoinjection of plant cells were produced.

These results provide evidence that nanoinjection of plant cells may be a viable method of transfection.
CHAPTER 3. IMPLEMENTATION OF INJECTION SYSTEM

The purpose of this chapter is to explain the nanoinjection system, its design requirements, fabrication and testing.

3.1 Intro

This injection system implementation is similar to the injection system used in injecting animal cells in vitro [11]. It is capable of mounting silicon lance arrays and then moving them in a controlled linear fashion, as well as sending an electric waveform through the specimen during a nanoinjection procedure. While similar in function these systems have no parts in common. The injection system made for plant nanoinjection went through several design iterations and includes several improvements over the animal cell injection system and is optimized for accepting plant specimens. The nanoinjector system implementation is shown in Figure 3.1.

3.2 Design Requirements for Injection Device

The injection device must fulfill four main requirements: be capable of mounting silicon chip lance arrays designed for plant injections, have control over the voltage potential between the lance arrays as well as the current passing through the plant specimen, contain a space for accepting the plant specimen, and move the lance arrays in order to repeatably apply a force upon the plant specimen.

3.2.1 Mounting Silicon Lance Arrays

The injection system will routinely change lance arrays when lance arrays become damaged or unsuitable for injections again due to possible contamination from the material being used to inject. Therefore, the mounting design needs to be able to release and acquire new lance arrays
routinely. The mounts also must move together and apart freely until the arrays contact each other so the mount should not interfere with the injection process itself. The injection device should be able to move freely together and apart with the lance arrays installed.

The injection device must be capable of accepting silicon lance arrays for use in plant nanoinjection. The arrays need to be mounted in parallel planes because the injection device moves one array only along a linear path. If the array is not parallel with the other array then one of the silicon array edges will contact the plant specimen first, before the lances, and cause damage. The
edge of each silicon lance array chip has sharp edges so imperfections in parallelism can easily cause damage.

In addition to being parallel the lance arrays also need to line up when they come together. This enables pressure on the plant specimen to be calculated from the known area of the plant cell lance array. From the force per silicon lance array chip the force per lance can also be calculated from the known number of lances on each silicon lance array chip. If the lance arrays do not line up then the area of plant specimen that the force is transferred through will be smaller. Therefore, the pressure on the plant specimen surface being injected increases past the safe value, possibly damaging the plant. When fatal damage occurs to the plant specimen the plant cells die, make the injection unsuccessful.

The mount pattern designed into the injection device must also take into account that the silicon lance arrays must complete an electrical circuit. The electrical circuit will take its path through one lance array, into the plant specimen and then into the second lance array. This can be accomplished by using an electrically conductive carbon based tape to adhere the lance array to the rest of the injection device. The current injection device takes this approach to adhering the lance arrays to its mounts. Wires are connected from the control box of the injection device to thin metal probes, which are adhered to the lance arrays with carbon tape. The metal probes are attached to the mounting parts of the injection device.

### 3.2.2 Current and Voltage Control

The nanoinjection process is enhanced by the ability to control the current and voltage through the plant specimen. This causes electroporation and increases the efficiency of the injection procedure [12] [33]. The previous work injecting animal cells used a technique called voltage control. This injection system uses a technique called current control. It is advantageous to control the current at different points in the injection process because it is indicative of how many ions are moving in solution and also is a greater factor for causing damage to the cell than voltage.

The current is controlled by the injection device control box. The injection device control box is shown in Figure 3.2. The control box uses two current control integrated circuits (IC’s) that have been configured for certain currents. The current control IC’s need to be pre-set to their desired current, then the control box can switch between allowing no current to flow between
the lance arrays and the two pre-set current values. The current control IC’s are fed by a boost converter, which acts as a voltage source set at 25 volts. They work by changing the voltage between the two lance arrays so that the pre-set constant current will flow. The current control IC’s hit the 25 volt rail when two much resistance exists between the lance arrays to force enough current through the circuit. Therefore, when the voltage between the lance arrays saturates it indicates that the current flow between the lance arrays is below the pre-set value. The control box can switch between current sources rapidly and the current source IC’s can also respond rapidly. This allows the control box to send pulses of current through the plant specimen that enhance the nanoinjection process. Plants are not as damaged by pulses of current as by prolonged exposure to current. Therefore, the current source IC’s must be able to respond rapidly because the resistance changes from one plant specimen to another and also as a function of how close the lance arrays are in relation to each other. The closer that the plant lance arrays are together the less resistance exists between them and the lower the voltage needs to be to drive the desired current.

3.2.3 Space for Plant Specimen

The injection system must include a space for the plant specimen to be inserted and then injected. This space should accommodate a wide variety of plant specimens and also not require or cause any damage to the plant specimen to make it available for injection.
The plant specimens used are cotyledons: they are typically delicate and low to the ground. It is imperative to be able to manipulate the injection device into position relative to the plant specimen while not needing to move plant specimen. If the plant specimen needs to be bent or stretched into position in order to be injected then the chances of tearing the plant specimen into pieces is far greater. Since the cotyledons are low to the ground the lower lance array needs to have a slim profile in order to be maneuvered beneath it. There is usually a large amount of space available above the cotyledon because it is growing facing upwards with nothing in between its surface and the sun. Therefore, the top lance array must be able to be moved at least a few cotyledons width away from the bottom lance array but there is no maximum distance that they must be separated.

The maneuverability of the injection device is also enhanced by the ability to raise and lower the injection device by means of an adjustable stand. The stand itself does not obstruct the injection device from getting close to the plant sample. The stand allows the injection device to adjust its height relative to the plant sample. This is useful because each cotyledon grows at a different height and being able to bring the injection device to the level of the plant sample reduces the possibility of damaging the plant sample prematurely.

3.2.4 Repeatable Force Application

In order to compare the nanoinjection results to one another the force applied per lance should be controlled. The ability to move the lances through a desired deflection was also considered as a way to standardize the nanoinjection process. This method was not used because of the variability in plant samples. A set distance between the lances might only be optimal for a certain thickness of plant specimen. A distance too close would cause significant damage to the plant specimen from excessive pressure on each cell and cell tissues while a distance too far away may not even cause the lances to pierce the surface of the plant specimen. Therefore, while the variability in cotyledons can’t be fully accounted for from specimen to specimen a force was found that would not cause damage to the plant specimen but also fully insert the lance array.

The force transferred through the plant cell arrays is measured by an aluminum cantilever beam attached with a strain gauge, a Wheatstone bridge and an Analog to Digital Converter (ADC) IC. The force measurement is calibrated using a set of known weights. Then the force is read
during the nanoinjection process by a micro controller in the injection device control box. The nanoinjection control box can adjust the force exerted on the plant specimen by the lance arrays by moving the lance arrays farther apart or closer together. If the lance arrays have not contacted the plant specimen then no force is transferred. If the lance arrays are in contact with the plant specimen surface then the closer the lance arrays are together the greater the pressure on the plant specimen, the farther apart the lance arrays are from each other the lower the pressure on the plant specimen.

3.2.5 Other Design Advantages

This nanoinjection system has advantages when compared to the animal cell injection system and previous iterations of this plant cell injection system. Notable advantages include:

- Made largely from standard components
- Records and saves injection data
  - Voltage reading from each lance array
  - Voltage reading from boost converter
  - Force reading from load cell
  - Time stamp
- Injection parameters can be easily changed
  - Force through specimen
  - Pulse frequency and duty cycle
  - Injection speed
  - Injection phase durations
  - Data acquisition frequency
  - Linear step size
- Injects top and bottom of specimen
3.2.6 Design Disadvantages

- Lance arrays are mounted using electrically conductive carbon tape. Removing the lance array from the tape can cause the lance array to crack, making the lance array not reusable.

- The nanoinjector device code does not fully utilize the capability of the nanoinjector in terms of maximum or minimum speed, acceleration or minimum step size.

3.3 Fabrication

The nanoinjection system was designed using a computer aided design program. After the design was completed there are four 3D printed parts that must be made in order to complete the nanoinjector. These parts include the nanoinjector stand, the bottom probe mount, the top probe mount and the load cell mount. No machining is required to assemble the rest of the nanoinjector. The remaining parts can be sourced off the shelf and assembled by hand. No special tools are required besides metric hex wrenches. After the nanoinjector is assembled it can be tested to verify it is functioning correctly.

3.4 Testing

The injection system’s ability to produce the required electrical, force and movement requirements was measured. The electrical and force tests were also double verified using the microcontroller inside the control box. The test for linear movement cannot be completely verified by the internal microcontroller because the movement is not a completely closed loop system. However, the force generated at the load cell changes when linear movements are made when the probes are in contact with each other.

3.4.1 Electrical

The injection system is required to manipulate the current and voltage through the probes. The ability of the injection system to generate voltage at the probe ends was measured through the use of a general purpose volt meter. In addition to this measurement the microcontroller in the
control box can also sample the voltage generated at the probes and output the value to a computer. Both of these methods confirmed that the injection system could generate the required voltage.

The injection system is also required to generate its current pulses at a rate of at least 1000 Hz. This was measured by connecting a lab oscilloscope to the probes to see the pulses generated. The internal microcontroller was also used to sample at several times the pulse frequency to measure the voltage pulses generated at the probes. Both of these methods confirmed that the pulse and pulse timing could be generated fast enough by the injection system.

The injection system also is required to generate several different currents at different phases of operation. This was measured using a lab handheld current meter. The internal microcontroller also shows a corresponding change in voltage when there is a change in resistance between the probes. This indicates that the current is staying the same while the voltage and resistance in the circuit are changing in order to satisfy the Ohm’s Law equation.
Table 3.1: Nanoinjector Movement

<table>
<thead>
<tr>
<th>Step Size</th>
<th>Linear Travel [μm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full (1)</td>
<td>25</td>
</tr>
<tr>
<td>Half (1/2)</td>
<td>12.5</td>
</tr>
<tr>
<td>Quarter (1/4)</td>
<td>6.25</td>
</tr>
<tr>
<td>Eighth (1/8)</td>
<td>3.13</td>
</tr>
<tr>
<td>Sixteenth (1/16)</td>
<td>1.56</td>
</tr>
</tbody>
</table>

3.4.2 Force

The force is measured with an off the shelf load cell and is calibrated per the manufacturers instructions. This involves matching the output with a set of standard laboratory masses by changing a scale factor used by the ADC’s internal microcontroller. In order to further reduce error, the load cell is tared by the control box microcontroller before it is used for nanoinjection.

3.4.3 Linear Movement

The movement of the injection probe is governed by the step inputs to the stepper motor that powers the linear rail. The bottom probe is held stationary while the top probe is attached to the linear slide. The stepper motor step size can range from a full step to 1/16 step. The speed is governed by the frequency at which steps are commanded.

The step size is translated to linear movement with this equation:

\[ \Delta X = \frac{p}{(f \times s)} \]

where:
- \( X \) = linear travel
- \( p \) = ball screw pitch
- \( f \) = full steps/rev
- \( s \) = step size

The range of step sizes to linear travel are calculated and shown in Table 3.1.
The movement of the positive and negative probes of the nanoinjector device are shown in Figure 3.3. A nanoinjection procedure overview is as follows; 1) DNA in solution is pipetted onto lower lance array. 2) The lance arrays are moved together to form a fluid bridge between them, and then charged to attract DNA to either lower or upper lance array. 3) The lance arrays are separated but still charged. 4) The plant specimen is positioned and the lance arrays are inserted. The lance arrays undergo a reversed charge pulsing cycle to repel the DNA. 5) The lance arrays are separated and the plant specimen is removed.

3.5 Conclusion

A nanoinjection system was created. It was fabricated and tested successfully and adheres to the required design constraints.
4.1 Introduction

A set of experiments was performed to optimize the effect of certain variables when performing a nanoinjection. An experiment was created and performed to determine a range of allowable forces to be used during nanoinjections. An experiment was created and performed to determine a range of allowable number of times an injection could be performed sequentially on the same plant specimen. The nanoinjector device and the silicon lance arrays were used to perform injections on cotton leaves.

4.2 Force Experiment

The nanoinjector device is capable of producing a force on the plant specimen that causes fatal damage to plant cells. It was desired to know the maximum amount of force that could be generated between the silicon lance arrays that would not result in a fatal amount of pressure for the plant cells. Different forces, varying from 1 N to 200 N, were applied to plant specimens during a nanoinjection procedure. Results for the plant specimens applied forces of 20 N, 100 N and 200 N can be seen in Figure 4.1 where photographs are shown for 0 and 7 days post injection (dpi).

These results show that fatal damage to plant cells occurred in a force applied greater than 20 N. The 20 N force did not damage the plant specimen. For plant specimens with a force applied less than 20 N no damage was observed. A force of 20 N per silicon lance array translates to an average force of 0.18 mN per lance.

As can be seen in Figure 4.1, the voltage between the probes remains constant after a sufficient amount of force has been achieved. The nanoinjection device is applying a constant current through the plant specimen while the probes are in motion and the force between them is increasing. This indicates that the resistance between the probes is also remaining constant due
to Ohm’s Law. If the resistance remains constant while the force increases then the silicon lances have stopped getting closer together. The conclusion is either they have reached a full insertion or they are not capable of piercing the plant specimen and the tips of the lances remain pressing on the surface of the plant specimen.

4.2.1 Multiple Injections

The nanoinjector device is capable of following a procedure involving multiple sequential injections on the same plant specimen. This procedure was designed to maximize transfection while minimizing damage to the plant specimen. It was desired to know the maximum amount of sequential injections on a single plant specimen which could occur before damage to the plant specimen.
Figure 4.2: Experimental nanoinjection results for varying the number of sequential injections applied to a plant specimen. For varying numbers of sequential injections 1, 5 and 10 no additional damage was seen on the plant specimens.

specimen occurred. In order to test this variable a different quantity of injections was chosen for six different plant specimens. The results can be seen in Figure 4.2 and in Figure 4.3.

During the procedure and after each time the lance arrays were extricated from the plant specimen, the plant specimen was moved slightly in order to ensure that the silicon lances did not insert directly into the same location on the plant specimen on the next injection.

The results for the plant specimen injected 15 times and the plant specimen injected 30 times both sustained damage. The plant specimens injected fewer times did not sustain damage from the nanoinjection. This is evidence that injecting 10 or fewer times will not sustain damage to the plant specimen.
Figure 4.3: Experimental nanoinjection results for varying the number of sequential injections applied to a plant specimen. No damage was seen on the plant specimen injected 10 sequential times but additional damage on the surface of the plant specimen was observed at the injection site for sequential injections of 15 and 30 times at 7 dpi.

4.3 Injection Results

Table 4.1 is a summary of the most important plant specimen nanoinjection experiments. As experiments were performed the nanoinjection procedure was changed in an effort to maximize transfection.

The nanoinjection device was used to perform a nanoinjection on a cotton cotyledon. The results are shown in Figure 4.4. The cotton cotyledon was injected with Cotton Leaf Crumple Virus, abbreviated CLCrV. If this viral DNA was inserted into the interior of the cells of the plant specimen then it would produce a crumpled effect and a bleaching effect on the plant specimen’s surface. The results for the plant specimen treated with CLCrV and other control plant specimens included in the nanoinjection procedure can be seen in Figure 4.4.
Table 4.1: Significant nanoinjections

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Device</th>
<th>Substance</th>
<th>Plant Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HeLa Cell/ V. Control</td>
<td>PI</td>
<td>Arabidopsis</td>
<td>Positive result due to cell wall permeability to PI.</td>
</tr>
<tr>
<td>2</td>
<td>Manual/ V. Control</td>
<td>None</td>
<td>Arabidopsis</td>
<td>Shear applied during injection can break lances.</td>
</tr>
<tr>
<td>3</td>
<td>Manual/ V. Control</td>
<td>CLCrV</td>
<td>Cotton</td>
<td>Changed to cotton plants.</td>
</tr>
<tr>
<td>6</td>
<td>Manual/ I. Control</td>
<td>CLCrV</td>
<td>Cotton</td>
<td>Change electronics to current controlled.</td>
</tr>
<tr>
<td>9</td>
<td>Plant Cell/ I. Control</td>
<td>CLCrV</td>
<td>Cotton</td>
<td>Change to automated plant injector device from manual.</td>
</tr>
<tr>
<td>12</td>
<td>Plant Cell/ I. Control</td>
<td>CLCrV</td>
<td>Cotton</td>
<td>Adjust force using numbers from force experiments.</td>
</tr>
<tr>
<td>14</td>
<td>Plant Cell/ I. Control</td>
<td>CLCrV</td>
<td>Cotton</td>
<td>Adjust attraction time on nanoinjection.</td>
</tr>
<tr>
<td>17</td>
<td>Plant Cell/ I. Control</td>
<td>CLCrV</td>
<td>Cotton</td>
<td>Inject three times on each specimen.</td>
</tr>
</tbody>
</table>

4.4 Conclusion

The plant specimens nanoinjected do not show any additional harm or signs of damage following the nanoinjection procedure when injected 10 or less sequential times at an injection force of .18 mN per lance.

In order to validate successful transfection through nanoinjection in plant specimens a nanoinjected gene should be expressed by the plant specimen. While the plant specimens are not harmed by the procedure there is not evidence for successful transfection of CLCrV. While the symptoms of CLCrV were not seen, leaf bleaching and crumpling, a different injection gene may result in a successful nanoinjection.
Figure 4.4: Experiment Number 17 results. The CLCrV was not manifested in the treatment plant specimen. The control plant specimens remained unchanged and unharmed by the nanoinjection procedure.
CHAPTER 5. CONCLUSION

5.1 Conclusion

Silicon lances for nanoinjection were made using photolithographic and DRIE Bosch processes. The silicon lances were separated into arrays for use in nanoinjecting cotyledons of both Arabidopsis and Cotton plants. The lance array geometry was measured to verify they matched the design requirements. The lance arrays showed evidence of successful cell penetration on Arabidopsis cotyledons. A nanoinjector device was made that used the silicon lance arrays in a nanoinjection protocol. This nanoinjector was capable of repeated nanoinjections as well as varying the applied force, timing, speed, voltage and electrical current parameters for nanoinjection. The nanoinjector was used with the silicon lance arrays to conduct nanoinjections. Evidence of successful penetration of cell walls was observed with light and SEM microscopes. The effects of force and repeated injections on Cotton cotyledons was observed. The parameters for the nanoinjection procedure were obtained such that plant specimens do not sustain visible damage. However, the relatively small number of nanoinjection procedures that were performed on cotton plants did not show evidence of a successful injection by manifesting symptoms of CLCrV virus. More work is needed to understand in greater detail the conditions that would create a successful nanoinjection procedure.

5.2 Further Work

The scope of this work was focused on the mechanical implementation of the nanoinjection system for plant cells, comprising the silicon lance arrays and the injector device. Nanoinjections were performed in order to test the nanoinjection system but more tests are needed to confirm the viability or non-viability of nanoinjection as a transfection method for plants. Only 3 cotton plants were attempted to be transfected with CLCrV using the final nanoinjection parameters, this sample size is not adequate to provide a conclusive recommendation on whether nanoinjection is a viable
transfection method for plants. Further work should focus on the variables and recommendations discussed next.

### 5.2.1 Possible Interacting Variables

There are several possible reasons that successful transfection was not observed. Successful nanoinjection transfection for plants relies on correctly identifying and controlling the affecting conditions during a nanoinjection. It’s not currently known which conditions allow for a successful nanoinjection. Further work should be done to identify these conditions or variables and their allowable parameters. Some potential variables are discussed below:

- **Larger Sample Size** - A nanoinjection may not have a 100 percent success rate, therefore additional nanoinjections may need to be performed on multiple plants or on the same plant multiple times to be sure of a successful transfection.

- **Plant type** - Each species of plant may react to nanoinjection more or less favorably.

- **Temperature**
  
  Plant growing temperature - The temperature of the growing conditions. Plants may be more or less favorable to transfection in some temperatures.

  Nanoinjection temperature - The temperature conditions of the nanoinjection procedure itself may affect transfection efficiency.

- **Soil Nutrients** - Abundance or deficits in specific soil macro and micro nutrients.

- **Water** - Over and under watering may create more or less favorable conditions for nanoinjection.

- **Light**

  Intensity - The intensity of light available to the plant specimens may affect the nanoinjection efficiency.

  Duration - Shortening or lengthening the day of available light. - Altering the spectrum of wavelengths for growing conditions.
• Humidity - High or low levels of humidity may create more or less favorable conditions for nanoinjection.

• Virus - Some virus’ may be easier to transfect with using nanoinjection than others.

• Injection location - Varying location of nanoinjection may lead to different levels of transfection efficiency due to the local vascular system and tissue [30].

• Seed preparation - Variations due to seed preparation before germination may affect nanoinjection transfection efficiency [34].

• Plant age - Nanoinjections at varying stages of plants lifespan may affect transfection efficiency.

• Silicon Lance Array - Varying the dimensions (i.e. height, diameter, spacing) may affect the transfection efficiency.

5.2.2 Recommendations

Further work done on plant nanoinjection should focus on the recommendations discussed next.

The large number of interacting variables is a barrier to understanding plant nanoinjection in greater detail. Iterations on the nanoinjection procedure could be more rapidly improved if results could be observed more rapidly. One way to accomplish this is through the use of an injection substance that manifested itself faster. For example, causing the plant cells to produce green or red fluorescent protein would be evident in 2-3 days rather than up to a month for CLCrV expression.

Further work should also focus on doing larger numbers of experiments to fully validate whether the nanoinjection procedure was successful, or to what extent it was successful. Starting with ideal plant, such as Arabidopsis, to achieve the first successful plant nanoinjections is recommended. An active control for the virus or injection media should also be added, i.e. a check to make sure the virus or media is still viable.
REFERENCES


APPENDIX A. SUPPLEMENTAL MATERIAL

A.1 Injection Device BOM

<table>
<thead>
<tr>
<th>Description</th>
<th>Model/ Manufacturer</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Actuator</td>
<td>FUYU</td>
<td>1</td>
</tr>
<tr>
<td>Stepper Motor Driver</td>
<td>A4988</td>
<td>1</td>
</tr>
<tr>
<td>Microcontroller</td>
<td>STM32F103C8T6</td>
<td>1</td>
</tr>
<tr>
<td>Current Source</td>
<td>LM334</td>
<td>2</td>
</tr>
<tr>
<td>Screen</td>
<td>SSD1306 LCD</td>
<td>1</td>
</tr>
<tr>
<td>Load Cell</td>
<td>HX711</td>
<td>1</td>
</tr>
<tr>
<td>Voltage Booster</td>
<td>SX1308</td>
<td>1</td>
</tr>
<tr>
<td>Relay</td>
<td>TLP222AF-ND</td>
<td>6</td>
</tr>
<tr>
<td>Power Supply</td>
<td>Dr.Meter</td>
<td>1</td>
</tr>
<tr>
<td>Tactile Switch</td>
<td>Adafruit</td>
<td>3</td>
</tr>
<tr>
<td>Various Resistors</td>
<td>Adafruit</td>
<td>14</td>
</tr>
<tr>
<td>Various Capacitors</td>
<td>Adafruit</td>
<td>2</td>
</tr>
<tr>
<td>Various Wires</td>
<td>Adafruit</td>
<td>-</td>
</tr>
<tr>
<td>3D Printed Parts</td>
<td>Taylor Brown</td>
<td>1</td>
</tr>
</tbody>
</table>

A.2 Microcontroller Code

```c
#include <SPI.h>  // SSD1306
#include <Wire.h> // SSD1306
#include <Adafruit_SSD1306.h>  // SSD1306
#include "HX711.h"  // Load Cell
```
# define BOARD_LED_PIN PC13
#define OLED_RESET 4 // SSD1306
#define UP HIGH
#define DOWN LOW

// Use Timer 1,2
HardwareTimer timer_one(1); // State Driving timer
HardwareTimer timer_two(2); // Motor & Pulse timer

Adafruit_SSD1306 display(OLED_RESET); // Set-up for LCD (SSD1306)

// Setup timer periods
int timer_one_period = 2;
int timer_two_period = 1;

// Pin assignments
int button_1 = PB12; // Button one pin number
int button_2 = PB13; // Button two pin number
int button_3 = PB14; // Button three pin number
int step_pin = PA8; // Step pin on motor driver
int direction_pin = PB15; // Direction pin on motor driver
int msl1 = PA9;
int ms2 = PA10;
int ms3 = PA11;
int pos_probe_node = PA2; // Analog node for positive probe
int neg_probe_node = PA1; // Analog node for negative probe
int boost_node = PA0; // Analog node for boost converter
int relay_1 = PA15;
// Digital pin activating relay 1, Neg Probe --> GND
int relay_2 = PB3;
// Digital pin activating relay 2,
// Neg Probe —> CS1 (Current Source 1)
int relay_3 = PB4;
// Digital pin activating relay 3,
// Neg Probe —> CS2 (Current Source 2)
int relay_4 = PB5;
// Digital pin activating relay 4, Pos Probe —> GND
int relay_5 = PB8;
// Digital pin activating relay 5, Pos Probe —> CS1
int relay_6 = PB9;
// Digital pin activating relay 6, Pos Probe —> CS2

// Load Cell Parameters
int load_cell_clk = PB10;
int load_cell_dt = PB11;
int pre_contact_load = 25;
int max_load = 2000; // Newtons (need to calibrate for newtons)
long current_load = 0;

// Setup Load Cell
HX711 scale(load_cell_dt, load_cell_clk);

// Program Parameters/ Variables

int led = 0; // Led status (0=LOW, 1=HIGH)
volatile int state = 0; // Program state number
long debouncing_time = 200;
// Debounce time in milliseconds for push buttons
volatile unsigned long last_millis = 0;
int state6 = 0;

long start_time = 0;
int stabilize_time = 3; // sec
int attraction_time = 30; // sec
int repulsion_time = 10; // sec
volatile int pulse_length = 1000; // Pulse length in us
int current_insertion = 0; // Initialize
int number_of_insertions = 4;
// Number of insertions (attraction counts as an insertion)

// Stepper Motor Parameters
volatile int current_retraction_distance = 0;
int retraction_distance = 500;
// Steps to retract after attraction & insertion
volatile int stepper_period = 10000; // ms
volatile int stepper_period2 = 1000;

void setup() {

setup_SSD1306();
setup_pins();

attachInterrupt(button_1, button_1_isr, RISING);

configure_timer_one();
configure_timer_two();

setup_serial();
initialize_load_cell();

// Enter state 1 after setup()
state = 1;

}

void loop() {

    // Main program State Machine
    switch (state) {

        case 1:
            update_screen();
            while (state == 1);
            break;

        case 2:
            update_screen();
            set_timer2_period(stepper_period);

            // Move probe in accordance with buttons 2 & 3
            while (state == 2){

                if (digitalRead(button_2) == HIGH){
                    digitalWrite(direction_pin, HIGH);
                    timer_two.resume();

        46
else if (digitalRead(button_3)==HIGH) {
    digitalWrite(direction_pin, LOW);
    timer_two.resume();
}
else{
    timer_two.pause();
}
}

break;

case 3:
    update_screen();
    set_timer1_period(stabilize_time);

    // Initialize time stamp only at the beginning
    if (current_insertion==0){
        start_time = millis();
    }
    // Collect data during "stabilize_time"
    while (state==3){
        collect_data();
    }
    break;

case 4:
    update_screen();
    set_timer1_period(attraction_time);
while (state == 4) {
    collect_data();
}
break;

case 5:
    update_screen();
    set_timer2_period(stepper_period);
    stepper_resolution(1);

while (state == 5) {
    collect_data();
    if (current_retraction_distance > retraction_distance) {
        // update screen
        display.setTextSize(1);
        display.setTextColor(WHITE);
        display.setCursor(0, 0);
        display.print("Nano Injector V4 S:");
        display.println(state);
        display.println();
        display.setTextSize(2);
        display.print("Retraction done");
        display.display();
        display.clearDisplay();
        // turn off timer 2
        timer_two.pause();
        // reset current retraction distance
        current_retraction_distance = 0;
        // Increment current_insertion_var,
        }
// attraction counts as an insertion
    current_insertion ++;
}
if (current_insertion == number_of_insertions) {
    state = 3;
    digitalWrite(relay_6, LOW);  
    digitalWrite(relay_4, HIGH);  
    digitalWrite(relay_3, LOW);  
    digitalWrite(relay_1, HIGH);
}
}
break;

case 6:
    update_screen();
    set_timer2_period(stepper_period);
    while (state == 6) {
        collect_data();
        if (abs(current_load) >= pre_contact_load && state6 == 0) {
            state6 = 1;
            stepper_resolution(8);
            set_timer2_period(stepper_period2);
        }
        else if (abs(current_load) >= max_load) {
            state = 7;
            state6 = 0;
            // Connect (+) probe to GND
            digitalWrite(relay_6, LOW);  
            digitalWrite(relay_4, HIGH);  

49
```c
  digitalWrite(relay_1, LOW);
}

}
break;

case 7:
  update_screen();
  set_timer1_period(repulsion_time);
  set_timer2_period(pulse_length);

  while(state==7){
    collect_data();
  }
break;

default:
  // Something went wrong
  break;
} // End of switch(state)

} // End of loop()

void configure_timer_one() {
  // Pause the timer while we're configuring it
  timer_one.pause();

  // Set up period
  timer_one.setPeriod(timer_one_period * 1000000);
```
// in microseconds (maximum 59 seconds)

// Set up an interrupt on channel 1
timer_one.setChannel1Mode(TIMER_OUTPUTCOMPARE);
timer_one.setCompare(TIMER_CH1, 1);
// Interrupt 1 count after each update
timer_one.attachCompare1Interrupt(timer_one_isr);

// Refresh the timer's count, prescale and overflow
timer_one.refresh();

// Start the timer counting
//timer_one.resume();
}

void configure_timer_two() {
    // Pause the timer while we're configuring it
timer_two.pause();

    // Set up period
timer_two.setPeriod(timer_two_period * 1000000);
    // in microseconds (maximum 59 seconds)

    // Set up an interrupt on channel 1
    timer_two.setChannel1Mode(TIMER_OUTPUTCOMPARE);
timer_two.setCompare(TIMER_CH1, 1);
    // Interrupt 1 count after each update
timer_two.attachCompare1Interrupt(timer_two_isr);

    // Refresh the timer's count, prescale and overflow
timer_two.refresh();

// Start the timer counting
timer_two.resume();
}

void timer_one_isr() {
    led = ~led;
    digitalWrite(BOARD_LED_PIN, led);

    if (state==3 && current_insertion!=number_of_insertions){
        state=4;
        // Connect (+) probe to CS1
        digitalWrite(relay_4, LOW);
        digitalWrite(relay_5, HIGH);
    }
    else if (state==3 && current_insertion==number_of_insertions){
        state=1;
        current_insertion=0;
    }
    else if (state==4){
        state=5;
        // Connect (+) probe to CS2
        digitalWrite(relay_5, LOW);
        digitalWrite(relay_6, HIGH);
        timer_one.pause();
    }
    else if (state==7){
        state=5;
        digitalWrite(relay_2, LOW);
    }
digitalWrite(relay_3, HIGH);
timer_one.pause();
timer_two.pause();
}

void timer_two_isr() {
    led = !led;
    digitalWrite(BOARD_LED_PIN, led);

    if (state == 2) {
        digitalWrite(step_pin, led);
    }
    if (state == 5) {
        digitalWrite(direction_pin, DOWN);
        digitalWrite(step_pin, led);
        current_retraction_distance++;
        // Increment each time
        // (only steps once per 2x thru interrupt)
    }
    if (state == 6) {
        digitalWrite(direction_pin, UP);
        digitalWrite(step_pin, led);
    }
    if (state == 7) {
        // Alternate (-) probe with CS1 & CS2
        digitalWrite(relay_2, led);
        digitalWrite(relay_3, !led);
    }
}
void button_1_isr () {
    if (millis() - last_millis > debouncing_time){
        // if not a false press do this
        led = ~led;
        digitalWrite(BOARD_LED_PIN, led);

        if (state==1){
            state = 2;
        }
        else if (state==2){
            state = 3;
            // Turn off timer 2
            timer_two.pause();
        }
        else if (state==5){
            state = 6;
            digitalWrite(relay_3, LOW);
            digitalWrite(relay_1, HIGH);
            digitalWrite(relay_4, LOW);
            digitalWrite(relay_6, HIGH);
        }
    }

    last_millis = millis();
}

void setup_serial () {

}
Serial.begin(9600);
while (!Serial);  // Wait for serial monitor
Serial.println("Serial Communication Established!");
}

void setup_SSD1306() {

#if (SSD1306_LCDHEIGHT != 64)
#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif

// Setup SSD1306 Display
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
// initialize with the I2C addr 0x3C (for the 128x64)
display.clearDisplay();
display.display();

update_screen();
}

void set_timer1_period(int timer1_period) {
  // Change timer_one
  // Pause the timer while we’re configuring it
timer_one.pause();
  // Set up period
timer_one.setPeriod(timer1_period * 1000000);
  // (*1000000 = us) (maximum 59 seconds)
  // Refresh the timer’s count, prescale and overflow
timer_one.setCompare(TIMER_CH1, 1);
  // Interrupt 1 count after each update
void set_timer2_period(int timer2_period) {
    // Change timer_one
    // Pause the timer while we’re configuring it
    timer_two.pause();
    // Set up period
    timer_two.setPeriod(timer2_period);
    // (us) (maximum 59 seconds)
    // Refresh the timer’s count, prescale and overflow
    timer_two.setCompare(TIMER.CH1, 1);
    // Interrupt 1 count after each update
    timer_two.refresh();
    timer_two.setCount(5);
    timer_two.resume();
}

void update_screen(){

    char* state_header[] = {"connect serial monitor",
                            "Press the button to start",
                            "Align for attraction",
                            "Collecting data",
                            "Attracting",
                            "Retracting",
                            "Inserting",
                            "Repelling"};


// Prompt for state
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(0,0);
display.println("Nano Injector V4 S:");
display.println(state);
display.println();
display.setTextSize(2);
display.println(state_header[state]);
display.display();
display.clearDisplay();
}

void setup_pins(){
  // Set up the LED to blink
  pinMode(BOARD_LED_PIN, OUTPUT);
  pinMode(button_1, INPUT_PULLDOWN);
  pinMode(button_2, INPUT_PULLDOWN);
  pinMode(button_3, INPUT_PULLDOWN);
  pinMode(step_pin, OUTPUT);
  pinMode(direction_pin, OUTPUT);
  pinMode(ms1, OUTPUT);
  pinMode(ms2, OUTPUT);
  pinMode(ms3, OUTPUT);
  pinMode(pos_probe_node, INPUT_ANALOG);
  pinMode(neg_probe_node, INPUT_ANALOG);
  pinMode(boost_node, INPUT_ANALOG);
  pinMode(relay_1, OUTPUT);
  pinMode(relay_2, OUTPUT);
}
pinMode(relay_3, OUTPUT);
pinMode(relay_4, OUTPUT);
pinMode(relay_5, OUTPUT);
pinMode(relay_6, OUTPUT);

// Initial output pin states
digitalWrite(BOARD_LED_PIN, HIGH);
digitalWrite(step_pin, LOW);
digitalWrite(direction_pin, LOW);
digitalWrite(ms1, LOW);
digitalWrite(ms2, LOW);
digitalWrite(ms3, LOW);
digitalWrite(relay_1, HIGH);
digitalWrite(relay_2, LOW);
digitalWrite(relay_3, LOW);
digitalWrite(relay_4, HIGH);
digitalWrite(relay_5, LOW);
digitalWrite(relay_6, LOW);

} 

void collect_data(){
    if (state==3 || state==4 || state==5 || state==7){
        Serial.print(millis()-start_time);
        Serial.print("t");
        Serial.print(float(analogRead(pos_probe_node))/112.5);
        Serial.print("t");
        Serial.print(float(analogRead(neg_probe_node))/112.5);
        // *11*3.3/4085
    }
Serial.print("\t");
Serial.println(float(analogRead(boost_node))/112.5);
}
else if(state ==6){
current_load = abs(scale.get_units());
Serial.print(millis()-start_time);
Serial.println("\t");
Serial.print(analogRead(float(pos_probe_node))/112.5);
Serial.println("\t");
Serial.print(float(analogRead(neg_probe_node))/112.5);
Serial.println("\t");
Serial.print(float(analogRead(boost_node))/112.5);
Serial.println;
Serial.println(current_load);
}
}

void initialize_load_cell(){

scale.power_up();
scale.set_scale(113.f);
// this value is obtained by calibrating
// the scale with known weights;
// [grams] 1kg load cell = 892, 20kg = 113

if( scale.get_units(10) != 0){
    Serial.println("Load Cell Initialized!");
}
else{

59
Serial.println("Load Cell Malfunction!");

scale.tare(); // reset the scale to 0


void stepper_resolution(int step_size)
{
if (step_size == 1)
{
digitalWrite(ms1, LOW);
digitalWrite(ms2, LOW);
digitalWrite(ms3, LOW);
}
if (step_size == 8)
{
digitalWrite(ms1, HIGH);
digitalWrite(ms2, HIGH);
digitalWrite(ms3, LOW);
}
if (step_size == 16)
{
digitalWrite(ms1, HIGH);
digitalWrite(ms2, HIGH);
digitalWrite(ms3, HIGH);
}

A.3 Graphing Code - Python

import matplotlib.pyplot as plt
import numpy as np
import csv
import sys

# Data for plotting

filename = 'filename goes here.txt'

reader = csv.reader(open(filename, 'r'), delimiter='\t')

# reader = csv.reader(x.replace('0', ' ') for x in filename)

f = list(reader)

# initialize variables

t = []
boost = []
pos_probe = []
neg_probe = []
force = []
xticks = []
yticks = []
voltage_difference = []
error = 0

#volt = 5*5.5/1024
volt = 1

print(filename, 'is', len(f), 'lines long')
for x in range(4, len(f), 1):
    # for x in range(4, 28000, 1):

    # error variable starts as zero every time through the loop
    error = 0

    # add value to "t" variable
    try:
        if float(f[x][0]) < 1000000:
            t.append(int(f[x][0])/1000)
        else:
            t.append(0)
    except ValueError:
        print('Line', x, 'Time: Value Error')
        t.append(0)
        error = 1

    except IndexError:
        print('Line', x, 'Time: Index Error')
        t.append(0)
        error = 1

    # add value to "pos_probe" variable
    try:
        if float(f[x][1])*volt < 30:
            pos_probe.append(float(f[x][1])*volt)
        else:
            pos_probe.append(0)
    except ValueError:
        print('Line', x, 'Pos_Probe: Value Error')
pos_probe.append(0)
error = 1

except IndexError:
    print('Line', x, 'Pos_Probe: Index Error')
    pos_probe.append(0)
    error = 1

# add value to "neg_probe" variable
try:
    if float(f[x][2]) * volt < 30:
        neg_probe.append(float(f[x][2]) * volt)
    else:
        neg_probe.append(0)
except ValueError:
    print('Line', x, 'Neg_Probe: Value Error')
neg_probe.append(0)
error = 1

except IndexError:
    print('Line', x, 'Neg_Probe: Index Error')
neg_probe.append(0)
error = 1

# add value to "boost" variable
try:
    if float(f[x][3]) * volt < 30:
        boost.append(float(f[x][3]) * volt)
    else:
        boost.append(0)
except ValueError:
    print('Line', x, 'Boost: Value Error')
    boost.append(0)
    error = 1

except IndexError:
    print('Line', x, 'Boost: Index Error')
    boost.append(0)
    error = 1

# add value to "force" variable
try:
    if float(f[x][4]) < 20000:
        force.append(abs(float(f[x][4]) * .0098))
    else:
        force.append(0)
except ValueError:
    force.append(0)

except IndexError:
    force.append(0)

# add value to "voltage_difference" variable
try:
    voltage_difference.append(pos_probe[len(pos_probe)-1] - neg_probe[len(pos_probe)-1])
except ValueError:
```python
print('Line', x, 'voltage_difference: Value Error')
voltage_difference.append(0)

except IndexError:
    print('Line', x, 'voltage_difference: Index Error')
voltage_difference.append(0)

# remove last value if error accrued
if error == 1:
    t.pop()
pos_probe.pop()
neg_probe.pop()
boost.pop()
force.pop()
voltage_difference.pop()

print(len(t))
print(len(pos_probe))
print(len(neg_probe))
print(len(boost))
print(len(force))
print(len(voltage_difference))

# Setup x, y ticks on graph
# time at end in seconds (integer value)
end_t = int(t[len(t) - 1])
```
for x in range(0, end_t, 30):
    xticks.append(x)

for x in range(-30, 30, 10):
    yticks.append(x)

# Note that using plt.subplots below is equivalent to using
# fig = plt.figure and then ax = fig.add_subplot(111)

fig, ax1 = plt.subplots()

#ax1.plot(t, boost, color = '#FFFF00', label='Boost V.')
ax1.plot(t, voltage_difference, color = '#FF0000', label='Voltage')
#ax1.plot(t, pos_probe, color = '#FF0000', label='Pos. Probe V.')
#ax1.plot(t, neg_probe, color = '#000000', label='Neg. Probe V.')
ax1.set_xlabel('Time (s)')
ax1.set_ylabel('Voltage (V)')
ax1.set_title('Voltage & Force v. Time — ' + filename)
ax1.set_xticks(xticks)
ax1.set_yticks(yticks)

ax2 = ax1.twinx()
# instantiate a second axes that shares the same x-axis
ax2.set_ylabel('Force (N)')
ax2.plot(t, force, '−−', color = '#008000', label='Force')
# ask matplotlib for the plotted objects and their labels
lines, labels = ax1.get_legend_handles_labels()
lines2, labels2 = ax2.get_legend_handles_labels()
ax2.legend(lines + lines2, labels + labels2, loc=3)

fig.savefig(filename + ".png")
plt.show()

exit()