ENPM667 - Control of Robotic Systems

Project 1

Tetherless Manipulation of Suture Needles using Electromagnets

Authors:
Sparsh Jaiswal (UID 117433968)
Divyansh Agrawal (UID 117730692)

November 2021

Contents

1 Introduction

1.1 What is a Surgical Suture?

Surgical suture is a clinical gadget used to hold body tissues together after an injury or medical procedure. Application generally involves using a needle with an attached length of thread. The primary goal of suturing is to bring together the wound margins and eliminate dead space between wound walls and edges so that underlying tissues are held together. Healing can only occur properly if the two severed areas of tissue align and remain intact. If dead space is not eliminated blood may pool in the wound leading to hematoma and consequently wound strength will be compromised, increasing the risk of infection.

1.2 Aim and Motivation

We aim to develop an autonomous magnetic needle suturing controller for controlling magnetic suture needles in a 2-dimensional (2-D) viscous environment. Present suturing methods can cause multiple problems like increased healing time, greater chances of infection, blood clot formation, and swelling near the wound. With the help of our model, the suturing task will become a lot easier and more accessible. Although, suturing takes place in a 3-dimensional environment, this project is currently in development phase and hence a 2-D workspace is considered for the proof of concept study. Our work has been divided into three major parts:

1. a mathematical model representing the force and torque due to the electromagnets
2. a kinematically constrained model to make the needle move only in 2-D
3. a velocity controller which takes current as input and then makes the needle move with a specific linear and angular velocity

1.3 Problem Formulation

In this paper, the authors are controlling a Neodymium (NdFeB) magnetic needle of length 18mm and radius 0.8mm with the help of 4 electromagnets. The residual flux density, or the remanence of the needle is 1.33 T which is a measure of the value of the flux density remaining when the external field returns from the high value of saturation magnetization to 0. The needle is suspended in a viscous fluid of dynamic viscosity 0.43 Pa-s inside a circular Petri dish of radius 85 mm. The needle can move only in 2-dimensions that is lateral motion of the needle is not allowed. The needle is controlled with the help of magnetic field generated by 4 electromagnets whose centers are a distance of 80 mm from the center of the dish and whose centerlines intersect at the center of the dish. The electromagnets are composed of approximately 12 tightly wound layers of 54 turns of AWG 16 polyimide-coated copper wire ($N_{em} = 12{\times}54$). The inner diameter of the EM is 85 mm, their outer diameter is 98 mm (average diameter 2 $\rho_{em} = 91.5$ mm), and their length is 60 mm.

2 Literature

2.1 Tissue Structure

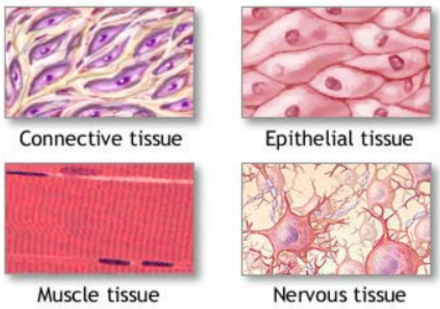

Figure 1: Types of tissues in human body

- Basically, a cell consists of three parts: the cell membrane, the nucleus, and between the two, the cytoplasm.
- The cell nucleus contains genetic material and regulates activities of the cell. It determines how the cell will function, as well as the basic structure of that cell.
- All of the functions for cell expansion, growth and replication are carried out in the cytoplasm of a cell.
- Tissue is a group of cells that have similar structure and that function together as a unit. Primary types of body tissues include epithelial, connective, muscular, and nervous tissues as shown in figure 1.

2.2 Present Needle Design and Suturing Methods

Most surgical suture needles are made of stainless steel. A needle needs to be sharp so it penetrates tissues easily, strong so it resists bending, and it should have some flexibility so resists breaking. Needle designs may vary according to:

- method of suture attachment
- shape and amount of curvature
- size of the needle
- design of needle point

Different types of needles used are shown below in Figure 2. Various suturing methods are available depending on the type of needle and type of wound. A few of them have been described below [1].

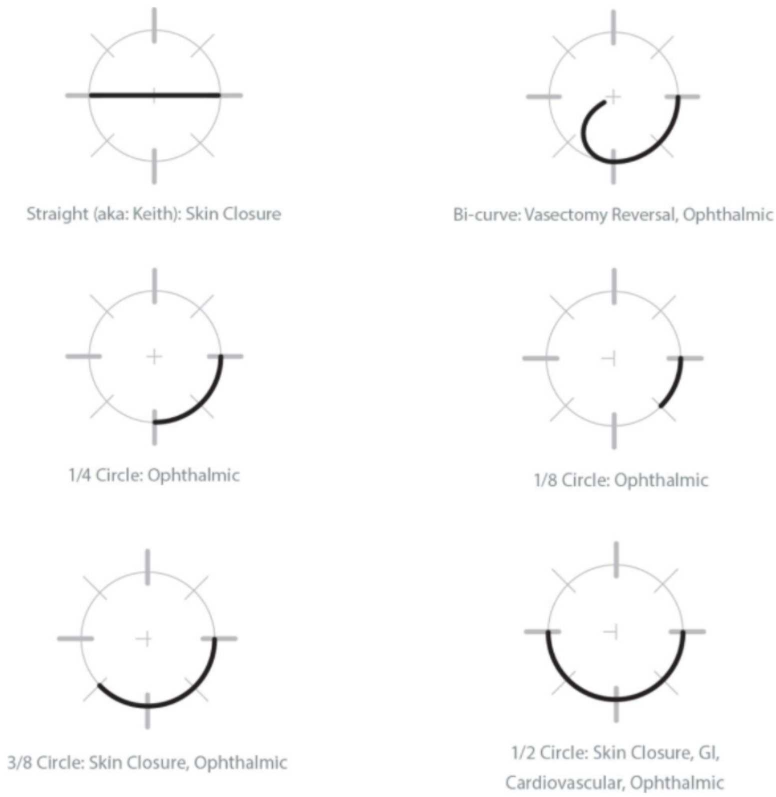

Figure 2: Types of suture needles

- Continuous sutures: This technique involves a series of stitches that use a single strand of suture material. This type of suture can be placed rapidly and is also strong, since tension is distributed evenly throughout the continuous suture strand.

- Interrupted sutures: This suture technique uses several strands of suture material to close the wound. After a stitch is made, the material is cut and tied off. This technique leads to a securely closed wound. If one of the stitches breaks, the remainder of the stitches will still hold the wound together.

- Deep sutures: This type of suture is placed under the layers of tissue below (deep) to the skin. They may either be continuous or interrupted. This stitch is often used to close fascial layers.

- Buried sutures: This type of suture is applied so that the suture knot is found inside (that is, under or within the area that is to be closed off). This type of suture is typically not removed and is useful when large sutures are used deeper in the body.

- Purse-string sutures: This is a type of continuous suture that is placed around an area and tightened much like the drawstring on a bag. For example, this type of suture would be used in your intestines in order to secure an intestinal stapling device.

- Subcutaneous sutures: These sutures are placed in your dermis, the layer of tissue that lies below the upper layer of your skin. Short stitches are placed in a line that is parallel to your wound. The stitches are then anchored at either end of the wound.

2.3 Basics of PID Controller

A PID algorithm consists of three basic coefficients; proportional, integral and derivative which are varied to get optimal response. The basic idea behind a PID controller is to read a sensor, then compute the desired actuator output by calculating proportional, integral, and derivative responses and summing those three components to compute the output.

2.3.1 Proportional Controller

The simplest controller is the proportional controller. With this term proportional, the feedback control signal u(t) is computed in proportion to the feedback error e(t) with the formula, $u(t) = K_c e(t)$ where K_c is the proportional gain and the feedback error is the difference between the reference signal r(t) and the output signal $y(t)(e(t) = r(t)y(t))$. The block diagram for the closed-loop feedback control configuration is shown in Figure 3 where R(s), E(s), U(s), and Y(s) are the Laplace transforms of the reference signal, feedback error, control signal, and output signal, respectively. G(s) represents the Laplace transfer function of the plant [2]. Because of its simplicity, the proportional controller is often used in the cases when little information about the system is available and the required control performance in steady-state operation is not demanding. As the controller only involves one parameter to be determined, it is possible to choose K_c without detailed information about the plant [3].

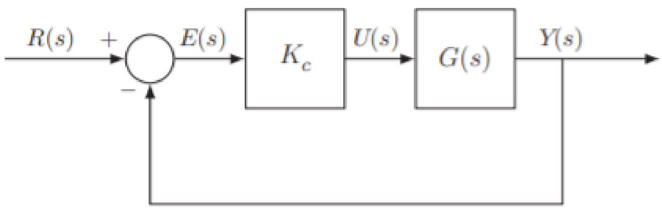

Figure 3: Block diagram of a basic PID control algorithm

2.3.2 PD Controller

In many applications, a proportional controller Kc is not sufficient to achieve a particular control objective such as stabilization or producing adequate damping for the closed-loop system. A PD controller is described as:

$$u(t) = k_p e(t) + k_d \dot{e}(t) \tag{1}$$

The derivative component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable. Increasing the derivative time (T_d) parameter will cause the control system to react more strongly to changes in the error term and will increase the speed of the overall control system response. Most practical control systems use very small derivative time (T_d), because the derivative response is highly sensitive to noise in the process variable signal. If the sensor feedback signal is noisy or if the control loop rate is too slow, the derivative response can make the control system unstable

2.3.3 PI Controller

A proportional plus integral (PI) controller is the most widely used controller among PID controllers. With the integral action, the steady-state error that had existed with the proportional controller alone is completely eliminated. The output of the controller u(t) is the sum of two terms, one from the proportional function and the other from the integral action, having the form,

$$u(t) = k_p e(t) + k_i \int e(t)dt \tag{2}$$

where $e(t) = r(t)y(t)$ is the error signal between the reference signal $r(t)$ and the output $y(t)$, K_c is the proportional gain, and τ_I is the integral time constant. The parameter τ_I is always positive, and its value is inversely proportional to the effect of the integral action taken by the PI controller. A smaller τ_I will result in a stronger effect of the integral action.

2.3.4 Gain Tuning

The process of setting the optimal gains for P, I and D to get an ideal response from a control system is called tuning. There are different methods of tuning of which the most used ones are "Trial and Error" method and the Ziegler Nichols method.

The gains of a PID controller can be obtained by trial and error method. In this method, the I and D terms are set to zero first and the proportional gain is increased until the output of the loop oscillates. As one increases the proportional gain, the system becomes faster, but care must be taken not make the system unstable. Once P has been set to obtain a desired fast response, the integral term is increased to stop the oscillations. The integral term reduces the steady state error, but increases overshoot. Some amount of overshoot is always necessary for a fast system so that it could respond to changes immediately. The integral term is tweaked to achieve a minimal steady state error.

Once the P and I have been set to get the desired fast control system with minimal steady state error, the derivative term is increased until the loop is acceptably quick to its set point. Increasing derivative term decreases overshoot and yields higher gain with stability but would cause the system to be highly sensitive to noise.

2.4 NdFeB Needle Design

Commercially available suture needles, Ethicon ST-4 needle, are commonly made of austenitic SAE 316 stainless steel which is magnetically soft and has a naturally low magnetic permeability and a low remnant magnetization. For manuplation of such needles significantly large field gradients would be required to achieve the magnetic force. Thus for the purpose of magnetic manipulation, magnetically hard materials that have stable remanent magnetizations such as Neodymium Iron Boron (NeFeB) are prefered. As the force is also dependant the the volume of the object to which the magnetic field is applied, the needle formed for magnetic manipulation is a NeFeB cylindrical needle of length 18mm and diameter 1.6mm with one end sharpened using sand paper(400 and 2000 grit) on a high speed rotating tool.

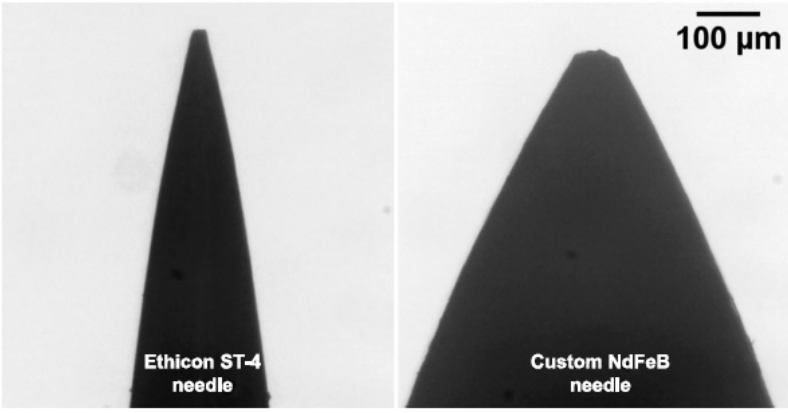

Figure 4: NdFeB Needle

3 Needle Controller Design

In this section we will design a blueprint to control the magnetic needle autonomously using electromagnets. We begin by stating our model assumptions followed by mathematical calculations.

3.1 Model Assumptions

Following assumptions were made in the work:

1. We are using a dipole model here, i.e. each electromagnet is a dipole and the needle is also a dipole. Distance between the 2 dipoles (needle and an electromagnetic coil) is much greater than the distance between the charges of the dipole.

2. We are considering Laminar fluid Flow. The fluid particles follow smooth paths in layers, with each layer moving smoothly past the adjacent layers with little or no mixing.

3. The movement of needle is restricted to 2 dimensions, x and y directions and there is no lateral movement of the needle.

4. Force required for the initial penetration of the needle into the skin/tissue is not taken into consideration. It is assumed that the needle is already present inside the body part.

3.2 Coordinate System

The domain of operation, the interior of the dish, will be denoted by $D \in R^2$ as a circular area of radius r_{dom} centered at the origin. Consider the general setting of 4 EM whose centers are located at $r_k \in R^d, k \in$ 1,...,4, and without loss of generality assume that they are uniformly spaced a distance $r_{em} = ||r_k||$ from the origin. As stated above, we assume these EM are aligned such that their axes of symmetry intersect at the origin, i.e., that their axes of symmetry are $\hat{r}_k = r_k/r_{em}$.

We wish to control the centroid of a needle $r \in D$ and its heading (in the north pole direction) given by θ by manipulating external magnetic field acting on the needle through the controlled currents of the m EM. For convenience denote $x = [r^T, \theta^T]$ as the state of the needle.

3.3 Magnetic Moment and Field

The magnitude of the Magnetic moment of the needle is $M = NIA$ where $N =$ Number of turns in the coil , $I=$ current in the coil , $A=$ area of cross section of the coil with direction perpendicular to the current loop in the right-hand-rule direction.

$$M(x) = \frac{\pi \rho_{ndl}^2 \ell_{ndl} B_{ndl}^r}{\mu_0} h(\theta) \tag{3}$$

where $\mu_0 = 4\pi \times 10^{-7}$ H/m is the permeability of vacuum and $h(\theta) = [cos(\theta), sin(\theta), 0]^T$ is the heading vector (3×1 vector) of the needle (considering the z component too as MATLAB requires 3 dimensions for taking cross product). Here, ρ_{ndl} is the radius of the needle, ℓ_{ndl} is the length of the needle and B^r_{ndl} is the residual flux density (remanence). The magnetic moment of the electromagnet is

$$M_k(I_k) = \pi \rho_{em}^2 N_{em} \hat{r_k} I_k \tag{4}$$

where ρ_{em} is the radius of the EM, N_{em} is the number of turns in the EM coil. Vector from each electromagnet to the needle's centroid is given by

$$d_k = r - r_k, \delta_k = ||d_k||, \hat{d_k} = \frac{d_k}{\delta_k} \tag{5}$$

$r_{k1} = [0.08, 0, 0]^T$; $r_{k2} = [0, 0.08, 0]^T$; $r_{k3} = [-0.08, 0, 0]^T$; $r_{k4} = [0, -0.08, 0]^T$ as r_k represents the vector from center of dish to the center of electromagnet as shown in figure 5.

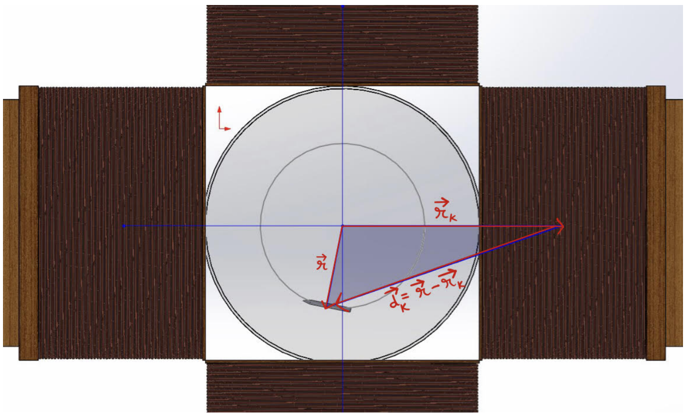

Figure 5: Vectors used in our work

The magnetic field generated by a dipole at a distance r is given by

$$B(r) = \frac{\mu_0}{4\pi} \frac{[3(\vec{m} \cdot \hat{r})\hat{r} - \vec{m}]}{|r|^3} \tag{6}$$

Here, r $= d_k$, $\vec{m}$ is given by (2), so substituting these values in (4) gives

$$B_k(x, I_k) = -\frac{\mu_0 \rho_{em}^2 N_{em}}{4\delta_k^3}(\hat{r_k} - 3\hat{d_k}\hat{d_k^T}\hat{r_k})I_k \tag{7}$$

3.4 Force and Torque

Calculating the torque (τ) on the needle due to all the electromagnets. We know that torque τ is defined as the cross product of magnetic moment, $\vec{m}$ and the magnetic field, B.

$$\tau = M(x) \times B_k(x, I_k) \tag{8}$$

So, due to all the 4 coils, torque will be

$$\tau = \sum_{k=1}^{4} M(x) \times B_k(x, I_k) \tag{9}$$

$$= \frac{\pi \rho_{ndl}^2 \ell_{ndl} B_{ndl}^r}{\mu_0} h(\theta) \times -\frac{\mu_0 \rho_{em}^2 N_{em}}{4\delta_k^3} (\hat{r_k} - 3\hat{d}_k \hat{d}_k^T \hat{r_k}) I_k$$

$$= \sum_{k=1}^{4} \frac{C}{\delta_k^3} (h(\theta)^T S \hat{r_k} - 3h(\theta)^T S \hat{d}_k \hat{d}_k^T \hat{r_k}) I_k$$

$$C = \frac{\pi \rho_{ndl}^2 \rho_{em}^2 l_{ndl} B_{ndl}^r N_{em}}{4}$$

$$S = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Here, we are using a skew-symmetric matrix to perform the cross product calculation. If we have 2 vectors, $\vec{a}$ and $\vec{b}$, both of them are 3×1 vectors, then

$$\vec{a} \times \vec{b} = [b]^T * a = \begin{bmatrix} 0 & b_3 & -b_2 \\ -b_3 & 0 & b_1 \\ b_2 & -b_1 & 0 \end{bmatrix} \begin{bmatrix} a1 \\ a2 \\ a3 \end{bmatrix} \tag{10}$$

The force exerted by the EM is given by the gradient of the magnetic potential [4] [5]

$$F = -\sum_{k=1}^{4} \nabla (M(x)^T B_k(x, I_k)) \tag{11}$$

$$= \sum_{k=1}^{4} \frac{\hat{d}_k \hat{r}_k^T h + h r_k^{\hat{T}} \hat{d}_k + \hat{r}_k \hat{d}_k^T h - 5 r_k^{\hat{T}} \hat{d}_k \hat{d}_k^T h}{\delta^4} I_k$$

Here, we are omitting the argument to $h(\theta)$ for representational purposes.

3.5 Needle Dynamics

We are assuming the first order dynamics for the needle due to laminar fluid flow from low Reynolds number and negligible inertia terms, so

$$\dot{r} = \frac{1}{\sigma_r} F, \dot{\theta} = \frac{1}{\sigma_\theta} \tau \tag{12}$$

where $\sigma_r = 0.15141$ and $\sigma_\theta = 7.002 * 10^{-7}$ are the fluid drag constants in the translational and rotational motion respectively that have been calculated in the MATLAB code. We are assuming that there is no translational motion in the z direction, that is needle can only translate along its axis. So the cross product of heading vector and the velocity vector should be 0 as they both are pointing in the same direction.

$$\dot{r} \times h(\theta) = 0 \tag{13}$$

$$\dot{r}^T S h(\theta) = 0$$

The linear velocity of the needle becomes

$$v = h(\theta)^T \dot{r} \tag{14}$$

$$= [cos(\theta), sin(\theta), 0]^T * \frac{\sum_{k=1}^{4} \frac{\hat{d}_k \hat{r}_k^T h + h \hat{r_k^T} \hat{d}_k + \hat{r_k} \hat{d_k^T} h - 5 \hat{r_k^T} \hat{d}_k \hat{d_k^T} h}{\delta^4} I_k}{\sigma_r}$$

$$= \frac{3C}{\sigma_r} \sum_{k=1}^{4} \frac{2h^T \hat{d}_k \hat{r}_k^T h + \hat{r_k^T} \hat{d}_k - 5 \hat{r_k^T} \hat{d}_k \hat{d_k^T} h}{\delta^4} I_k$$

Similarly, $\omega = \tau / \sigma_\theta$ where ω is the angular velocity. Let,

$$g_k(x) = \begin{bmatrix} \frac{3C}{\sigma_r} \frac{2h^T \hat{d}_k \hat{r}_k^T h + \hat{r_k^T} \hat{d}_k - 5 \hat{r_k^T} \hat{d}_k \hat{d_k^T} h}{\delta^4} \\ \frac{C}{\sigma_\theta} \frac{(h(\theta)^T S \hat{r_k} - 3h(\theta)^T S \hat{d}_k \hat{d_k^T} \hat{r_k})}{\delta_k^3} \end{bmatrix} \tag{15}$$

such that if $g(x) = [g_1(x), ..., g_4(x)]$ and $I = [I_1, ..., I_4]^T$ are stacked versions of these vector fields and EM currents, then we get

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = g(x) I = \sum_{k=1}^{4} g(x)_k I_k \tag{16}$$

The kinematically constrained dynamics of the needle becomes

$$\dot{x} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} g(x) I \tag{17}$$

3.6 Feedback Linearizing Control

We want the needle to have a certain desired velocity (v^d, ω^d). Assume a mapping,

$$I \triangleq g(x)^T y \tag{18}$$

for some $y \in R^2$. From (14), the actual linear and desired velocities may be expressed in terms of this mapping, i.e.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = g(x) I = g(x) g(x)^T y \tag{19}$$

As long as at least two of the fields generated by the EM are linearly independent at every $r \in D$, the matrix $g(x)g(x)^T$ will be full-rank. Thus, solving for the desired velocities gives

$$y = (g(x) g(x)^T)^{-1} \begin{bmatrix} v^d \\ \omega^d \end{bmatrix} \tag{20}$$

or in terms of our control current

$$I = g(x)^T (g(x) g(x)^T)^{-1} \begin{bmatrix} v^d \\ \omega^d \end{bmatrix} \tag{21}$$

This is equivalent to finding the Moore-Penrose inverse of $g(x)$ (as $g(x)$is not a square matrix), which can be shown to be the optimal solution to the problem.

$$min \sum_{k=1}^{4} I_k^2 \tag{22}$$

$$s.t.\ g(x)I = \begin{bmatrix} v^d \\ \omega^d \end{bmatrix}$$

3.7 Closed-Loop Control Schemes

We are performing tracking of a reference motion pattern encoded through time-varying positions $r_d(t)$ by setting

$$z_d = k_r(r_d - r) + \dot{r}_d \tag{23}$$

where $k_r > 0$ is a tuning parameter, and mapping into linear and angular velocities via the transformation

$$\begin{bmatrix} v^d \\ \omega^d \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\frac{1}{\lambda}\sin\theta & \frac{1}{\lambda}\cos\theta \end{bmatrix} z_d \tag{24}$$

where $\lambda > 0$ is a tuning parameter to control the aggressiveness in which the controller favors turning to track the reference (e.g., $\lambda = ndl/2$ allows us to achieve the desired reference z_d at the tip of the needle).

4 Simulation & Code

We have performed the simulation using COMSOL MultiPhysics and Solidworks. Programming was done in MATLAB.

4.1 COMSOL Simulation & Solidworks Model

This section shows the magnetic field due to single turn of an electromagnetic coil. An attempt was made to simulate the entire system modeled in solidworks as well. The attempt was un-successful because COMSOL was being operated through a virtual environment. With the limited resources of the virtual environment we successfully modeled and verified the magnetic flux densities and the 2D surface magnetic field vectors.

Finite Element Method (FEM) was used to model the single magnetic coil in COMSOL5.6. A 2D axis symmetrical model (cross-section) of the coil, having 12x54 tightly wound wires of cross section area $1.31mm^2$ and an air core, bounded by a magnetic insulation is shown in Fig 6. It also shows the direction of the magnetic field lines and the flux densities around the visible corss-section.

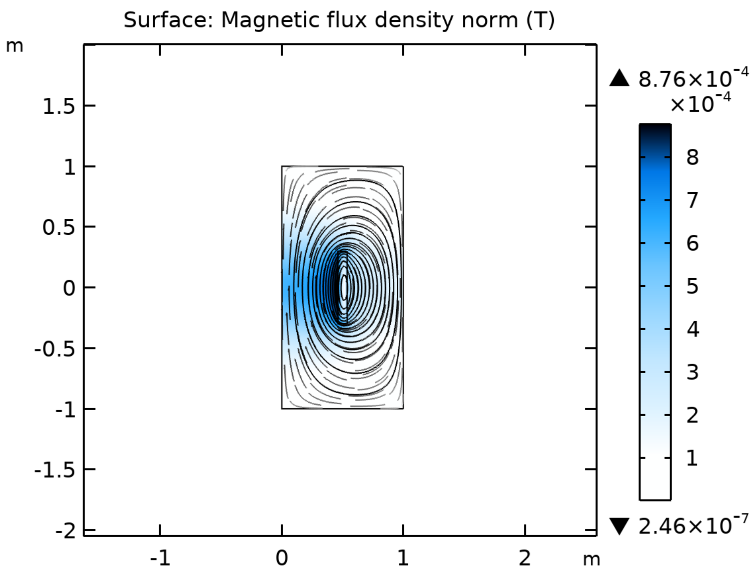

Figure 6: Magnetic Flux Density Norm

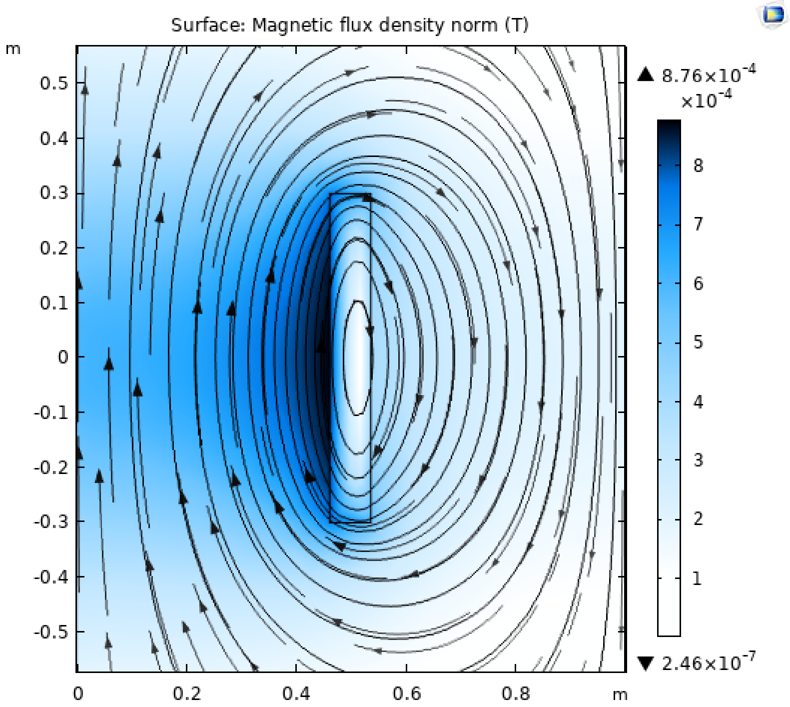

Figure 7: Close-up View of the Magnetic Fields

It is observed that the flux densities vary from $24mT$ to $88mT$. These flux densities were observed when 1A of current was passed through the coil. Fig 8 shows the 2D revolved model of the coil revolved to 255^o. The coil can be seen inside the cylindrical magnetic boundary (cyan highlight).

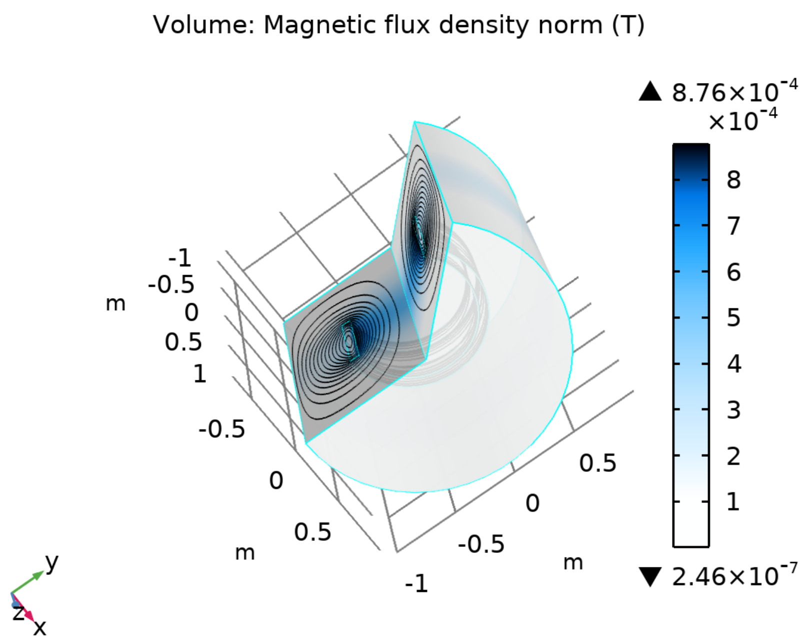

Figure 8: Cross-Sectional Area of Coil displaying Flux Density

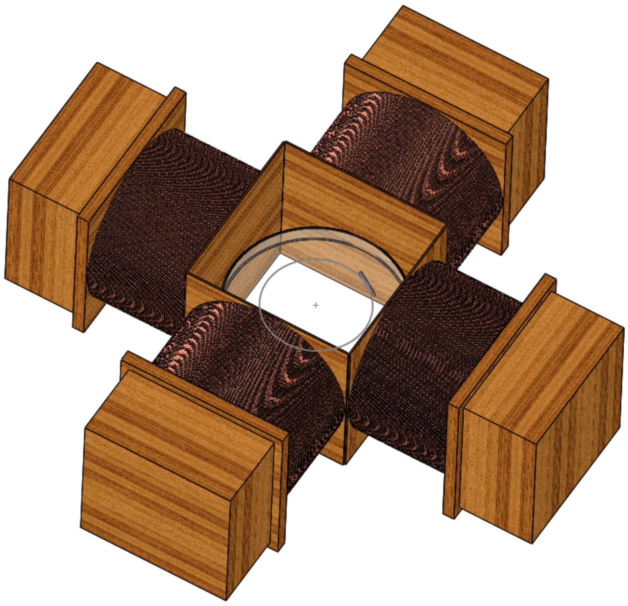

Figure 9: System Model - made using Solidworks

4.2 MATLAB Algorithm

Algorithm 1: MATLAB Algorithm

Result: Final result will be the velocity of the needle, linear as well as angular
Data: $\vec{r}$: Vector from center of Petri dish to the center of needle
Data: $\vec{r}$: Vector from center of Petri dish to the center of EM
Data: σ_r and σ_θ which are the damping coefficients
Data: $h(\theta)$: heading vector
Data: Magnetic Moment of Needle
initialization;
calculating magnetic field, force and torque;
using these to calculate angular and linear velocities;
using a specific time interval, calculate the actual position of the needle;
if *error* $\neq$ *0* **then**
    compare actual position and desired position;
    tuning the error using PD controller;
else
    follow the exact path as error equals 0;

5 Results

In the original MagnetoSuture paper [6], the authors have descibed a manual way to control the needle in a viscous environment.

In [7], the authors have used an autonomous method on physical setup to demonstrate magnetic needle suturing using electromagnets. They have used Hough Line Transform with appropriate pre-processing and post-processing for the needle pose estimation. This way they got the feedback of the needle position and then they calculated error using a PD controller.

In our work, we have used inverse kinematics to obtain the current input required to calculate the velocity of the needle. From this velocity, we are using forward kinematics to obtain the position of the needle for its next destination.

Since our work is limited to simulation of the needle movement in the viscous environment, any image processing work done in the paper has not been re-created due to unavailability of the physical setup. For getting feedback of the needle position, the actual trajectory of the needle (found using the mathematical calculations above) is compared with a desired trajectory which is defined explicitly by us as a circle.

5.1 Problems Faced

- Insufficient literature on Dipole Model of a Coil
- Unavailability of realtime localization data for simulation
- Restricted accessibility to COMSOL MultiPhysics software

6 Conclusion

This paper has successfully demonstrated magnetic manipulation on mesoscale. This proof of concept was novel to mesoscale but has already been implemented on the macroscale, like XPlanar, which is an industrial non-contact motion system used in clean rooms. Not only that, new studies have begun on this stepping stone to magnetically control mesoscale objects in 3 dimensions as well with greater degrees of freedom. Future works will also aim to increase the coil separation inorder to incorporate larger volume between the coils to perform tests on anesthetized rats and eventually human limbs. Feedback mechanism cannot be visual when it is applied in the real world thus work is being done on non visual feedback for localization data of the needle.

References

[1] Sonal Choudhary, Keyvan Nouri, Ploypailin Jungcharoensukying, and Mohamed Elsaie. *Suturing Techniques*, pages 25–29. 09 2012. ISBN 9781444330670. doi: 10.1002/9781118412633.ch4.

[2] Kamran Iqbal. Introduction to control systems, 02 2021.

[3] Liuping Wang. *Basics of PID Control*, pages 1–30. 2020. doi: 10.1002/9781119469414.ch1.

[4] Andrew J. Petruska and Bradley J. Nelson. Minimum bounds on the number of electromagnets required for remote magnetic manipulation. *IEEE Transactions on Robotics*, 31(3):714–722, 2015. doi: 10.1109/TRO.2015.2424051.

[5] Kar W. Torrance Yung, Peter B. Landecker, and Daniel D. Villani. An analytic solution for the force between two magnetic dipoles. *Physical Separation in Science and Engineering*, 9:39–52, 1998.

[6] Lamar O. Mair, Xiaolong Liu, Bhargav Dandamudi, Kamakshi Jain, Sagar Chowdhury, Jeremy Weed, Yancy Diaz-Mercado, Irving N. Weinberg, and Axel Krieger. Magnetosuture: Tetherless manipulation of suture needles. *IEEE Transactions on Medical Robotics and Bionics*, 2(2):206–215, 2020. doi: 10.1109/TMRB.2020.2988462.

[7] Matthew Fan, Xiaolong Liu, Kamakshi Jain, Dani Lerner, Lamar Mair, Irving Weinberg, Yancy Diaz-Mercado, and Axel Krieger. Towards autonomous control of magnetic suture needles. 04 2020.

MATLAB CODE:

```
syms theta I_1 I_2 I_3 I_4 y v_d omega_d x y xd yd kr rdd vf lam vdc wdc;


% Inverse Kinematics
x1=0.01;        % Feedback from vision sensor
y1=0.01;
theta11 = pi/4;

vd = 0.00001;   % Input
wd = 0;

rk_1 = [0.08; 0];   % Vectors to the centers of the coils
rk_2 = [0; 0.08];
rk_3 = [-0.08; 0];
rk_4 = [0; -0.08];

rc_1 = [1; 0];  % Unit vectors to the centre of the coils
rc_2 = [0; 1];
rc_3 = [-1; 0];
rc_4 = [0; -1];

rc_1t = transpose(rc_1); % Done to match the dimension to perform matrix/vector
multiplications
rc_2t = transpose(rc_2);
rc_3t = transpose(rc_3);
rc_4t = transpose(rc_4);

sigma_r = 2 * pi * 0.43 * 18 * 10^(-3) * log(18/0.8);                       %
Constants
sigma_t = 2 *pi * 0.43 * 0.8 * 10^(-3) * (18 * 10^(-3))^2;
c = pi * 0.8*10^(-3) * 0.8*10^(-3) * 18 * 10^(-3) * 1.33 * 0.0915 * 12 * 54;

r = [x1; y1];   % Position vector from the centre of the petri dish to the centre of
the needle
h = [cos(theta11); sin(theta11)];   % Heading Vector
ht = transpose(h);

dk_1 = r - rk_1;    % Vector from the centre of the coils to the centre of the needle
dk_2 = r - rk_2;
dk_3 = r - rk_3;
dk_4 = r - rk_4;

mod_1 = norm(dk_1);
mod_2 = norm(dk_2);
mod_3 = norm(dk_3);
mod_4 = norm(dk_4);

dc_1 = dk_1/mod_1;  % Unit vector from the centre of the coils to the centre of the
needle
dc_2 = dk_2/mod_2;
dc_3 = dk_3/mod_3;
dc_4 = dk_4/mod_4;
```

```
dc_1t = transpose(dc_1);
dc_2t = transpose(dc_2);
dc_3t = transpose(dc_3);
dc_4t = transpose(dc_4);

s=[0,-1;1,0];  % 2x2 Skew symmetric matrix for calculating cross product

vv1 = 3*c*(2*ht*dc_1*rc_1t*h + rc_1t*dc_1 -
5*rc_1t*dc_1*dc_1t*h)/(sigma_r*mod_1^4);    % Parametric velocity component due to
magnetic feilds each coil
vv2 = 3*c*(2*ht*dc_2*rc_2t*h + rc_2t*dc_2 - 5*rc_2t*dc_2*dc_2t*h)/(sigma_r*mod_2^4);
vv3 = 3*c*(2*ht*dc_3*rc_3t*h + rc_3t*dc_3 - 5*rc_3t*dc_3*dc_3t*h)/(sigma_r*mod_3^4);
vv4 = 3*c*(2*ht*dc_4*rc_4t*h + rc_4t*dc_4 - 5*rc_4t*dc_4*dc_4t*h)/(sigma_r*mod_4^4);

w1 = c*(ht*s*rc_1 - 3*ht*s*dc_1*dc_1t*rc_1)/(sigma_t*mod_1^3);  % Parametric angular
velocity component due to magnetic feilds of each coil
w2 = c*(ht*s*rc_2 - 3*ht*s*dc_2*dc_2t*rc_2)/(sigma_t*mod_2^3);
w3 = c*(ht*s*rc_3 - 3*ht*s*dc_3*dc_3t*rc_3)/(sigma_t*mod_3^3);
w4 = c*(ht*s*rc_4 - 3*ht*s*dc_4*dc_4t*rc_4)/(sigma_t*mod_4^3);

gx_r = [vv1 ,vv2 , vv3 ,vv4; w1 ,w2, w3, w4 ];

% disp(size(gx));

gi = pinv(gx_r); % MoorePenrose Inverse for non square matrices

% disp(size(gi));

m = [vd ; wd];  % Desired velocities

I = gi*m;   % Control currents for each coil

% disp(size(I));

disp(I);


% Forward Kinematics

r_k_1 = [0.08; 0; 0];
r_k_2 = [0; 0.08; 0];
r_k_3 = [-0.08; 0; 0];
r_k_4 = [0; -0.08; 0];

sigma_r = 2 * pi * 0.43 * 18 * 10^(-3) * log(18/0.8);
sigma_theta = 2 *pi * 0.43 * 0.8 * 10^(-3) * (18 * 10^(-3))^2;

I_1 = I(1,1);
I_2 = I(2,1);
I_3 = I(3,1);
I_4 = I(4,1);

r = [x; y; 0];
rd = [xd;yd];
```

```
heading = [cos(theta); sin(theta); 0]; % 3 cross 1

heading_T = transpose(heading);  % 1 cross 3
d_k_1 = r - r_k_1;
d_k_2 = r - r_k_2;
d_k_3 = r - r_k_3;
d_k_4 = r - r_k_4;

mod1 = norm(d_k_1);
mod2 = norm(d_k_2);
mod3 = norm(d_k_3);
mod4 = norm(d_k_4);

M_needle = pi * 0.8*10^(-3) * 0.8*10^(-3) * 18 * 10^(-3) * 1.33 / (4 * pi * 10^(-
7))  * heading;

B_constant = 4 * pi * 10^(-7) * 0.0915 * 12 * 54;

I = [I_1; I_2; I_3; I_4];

B1 = (-B_constant / (4 * mod1^3)) * ((r_k_1/norm(r_k_1)) - 3 * (d_k_1 / norm (d_k_1))
* transpose(d_k_1 / norm (d_k_1)) * (r_k_1/norm(r_k_1)))*I_1; % Magnetic feild due to
each coil
B2 = (-B_constant / (4 * mod2^3)) * ((r_k_2/norm(r_k_2)) - 3 * (d_k_2 / norm (d_k_2))
* transpose(d_k_2 / norm (d_k_2)) * (r_k_2/norm(r_k_2)))*I_2;
B3 = (-B_constant / (4 * mod3^3)) * ((r_k_3/norm(r_k_3)) - 3 * (d_k_3 / norm (d_k_3))
* transpose(d_k_3 / norm (d_k_3)) * (r_k_3/norm(r_k_3)))*I_3;
B4 = (-B_constant / (4 * mod4^3)) * ((r_k_4/norm(r_k_4)) - 3 * (d_k_4 / norm (d_k_4))
* transpose(d_k_4 / norm (d_k_4)) * (r_k_4/norm(r_k_4)))*I_4;

F1 = -gradient(dot(transpose(M_needle), B1)); % Forces due to each coil on the needle
F2 = -gradient(dot(transpose(M_needle), B2));
F3 = -gradient(dot(transpose(M_needle), B3));
F4 = -gradient(dot(transpose(M_needle), B4));

tau_1 = cross (M_needle, B1); % Torque due to each coil on the needle
tau_2 = cross (M_needle, B2);
tau_3 = cross (M_needle, B3);
tau_4 = cross (M_needle, B4);

r_dot_1 = F1 / sigma_r; % Magnitude of velocities
r_dot_2 = F2 / sigma_r;
r_dot_3 = F3 / sigma_r;
r_dot_4 = F4 / sigma_r;

v1 = transpose(heading) * r_dot_1;  % Components of velocity due to each coil
v2 = transpose(heading) * r_dot_2;
v3 = transpose(heading) * r_dot_3;
v4 = transpose(heading) * r_dot_4;

omega1 = tau_1 / sigma_theta;   % Angular velocity due to each coil
omega2 = tau_2 / sigma_theta;
omega3 = tau_3 / sigma_theta;
omega4 = tau_4 / sigma_theta;
```

```
g1 = [vpa(norm(v1)) ; vpa(norm(omega1))];
g2 = [vpa(norm(v2)) ; vpa(norm(omega2))];
g3 = [vpa(norm(v3)) ; vpa(norm(omega3))];
g4 = [vpa(norm(v4)) ; vpa(norm(omega4))];
gg = g1 + g2 + g3 + g4;                     % Resultant velocities

theta = pi/4;
x = 0.05;
y = 0.05;
r=[x;y];
vi = [subs(gg);0];
disp(vi);

zd = kr*(rd - r) + rdd;        % PD control equation
t = [cos(theta),sin(theta); -sin(theta)/lam , cos(theta)/lam ] * zd;
vdc = t(1,1);
wdc = t(2,1);
```