Conceptual Design of Nail Painting Robot

Brandon Bier
Department of Mechanical Engineering
Columbia University
New York. NY, 10027, USA
bb2856@columbia.edu

Rahul Jinturkar
Department of Mechanical Engineering
Columbia University
New York. NY, 10027, USA
rj2512@columbia.edu

Jianghao Ou
Department of Mechanical Engineering
Columbia University
New York. NY, 10027, USA
jo2526@columbia.edu

Avni Shah
Department of Mechanical Engineering
Columbia University
New York. NY, 10027, USA
as5437@columbia.edu

***Abstract*— Robotic manipulators are increasingly automating human tasks. This is due to emerging technology that can improve the speed, efficiency, and quality of jobs and services. The manicure industry in the United States grew to a record 8.53 billion dollars in 2016 with more than 17,000 nail salons [1]. Manicures make up 22% of the revenue in salons. Clearly, there is a lucrative opportunity for a precise automated robot to assist with nail painting. For manicurists, painting clients' nails requires skills to apply the polish evenly and steadily. To manicure one's own nails, one must additionally possess sufficient ambidextrous precision to coat the nails of the non-dominant hand. In order to improve the performance, artistry and convenience of nail painting, we propose 'Nailbot', a five degree of freedom manipulator to effectively and efficiently automate nail painting.**

I. Introduction

The field of cosmetics is a multibillion dollar industry that could benefit greatly from the precise and deliberate automation of a robot. The objective of Nailbot is to paint nails with a precise and uniform coat of polish. The scope of this project was to design and simulate the robot.

For the polish application tool, we considered three approaches to coat the nails: using a brush end effector, using a roller end effector, and using and nozzle end effector. We chose to proceed with a nozzle end effector as it will be the most efficient way to ensure an even polish application. Additionally, a nozzle avoids contact friction between the end effector and nail and scopes out the consideration of compliance or elasticity of the tool. Furthermore, it is more hygienic than a brush or roller as it does not directly come in contact with the user's nail.

One consideration was how to contain the polish within the nail boundaries. To avoid spillage onto the skin surrounding the nails, we recommended the user to employ the use of stick-on nail guards, which are currently available on the market. These guards will serve a second purpose, as they would enable the robot's sensors to easily detect the nail boundaries.

For the purposes of this project, we acquired the nail boundaries from experimental data [2]. We used these obtained geometric nail properties, to develop inverse kinematics equations and perform motion planning for the robot. This allowed us to position and orient the end effector in the desired starting location for each nail and to then traverse the nail trajectories. Finally, we successfully modeled the Nailbot in SolidWorks and simulated its automation using MATLAB.

II. Solution Approach

A. Manipulator Design

We chose to design a 5 degree of freedom robot with three prismatic joints followed by two revolute joints. The prismatic joints are used to position the end effector in terms of Cartesian coordinates (x, y, and z). The first revolute joint aligns the end effector with respect to the axis parallel to the length of the finger. This joint is especially useful while painting the thumbnail, as the user's thumb will be pointing in a different direction than the rest of the fingers. The second revolute joint is used to orient the end effector perpendicular to the nail contour curvature, allowing for a consistent and professional polish application.

Utilizing a standard Denavit Hartenberg (DH) convention, we assign joint variables as shown in Figure 1. The corresponding DH-table follows in Table 1. The three prismatic joints result in three variables, d_1*, d_2*, and d_3* which correspond to the base frame z, y, and x axes, respectively. The revolute joints are shown separately for illustrative purposes, but in reality they are combined at one location ($d_4 = 0$). These rotating joints produce two additional variables, θ_4*and θ_5*. The length of the end effector, is defined as a_5.

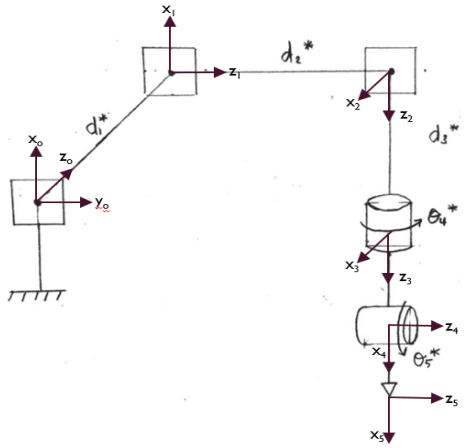

Figure 1: Manipulator Design

TABLE 1. DENAVIT HARTENBERG

Link	a	α	d	θ
1	0	π/2	d_1*	0
2	0	-π/2	d_2*	π/2
3	0	0	d_3*	0
4	0	π/2	0	θ_4*
5	a_5	0	0	θ_5*

B. *Aquiring Inputs for Motion Planning*

To map out the motion planning, we need to acquire inputs to set boundaries and paths for our end effector. As the scope of this project was to simulate the robot motion, we used measurement data from a technical paper as our inputs. The collected data comes from a study of three-hundred participants of varying ages and reports the average nail width, length, and radii of curvature for the nails [2].

Using these geometric properties, we developed equations to model a parabolic path along the width of the nail for the end effector to traverse. The starting point, or initial boundary condition, for each finger was then used to identify the position and orientation of the end-effector. Thereafter we inputted the positions and orientations of the end-effector in the inverse kinematics equations to identify the positions and orientations of the joints.

While we used average nail geometric properties as inputs, in practice (a production Nailbot), sensors would locate the positions of the fingers along with their respective nail geometries. Additionally, the sensors would account for the θ_4 values and θ_5 boundaries, i.e. finger orientation and nail boundaries.

C. *Developing Inverse Kinematics Equations*

We used inverse kinematics to calculate the necessary joint variables given a desired position in terms of the end effector frame (x_e, y_e, and z_e) and orientation. For simplicity of calculation, we assumed that the index, middle, ring, and little fingers are parallel to the base frame z_0 axis, and therefore the θ_4 values for those fingers are zero. We assumed the thumb to be oriented 45° away, and therefore θ_4 is set to pi/4. We were able to calculate the required value of θ_5 based on the nail geometry. Thus, moving forward, the θ_4 and θ_5 values are treated as knowns. To calculate d_1*, d_2*, and d_3* with known x_e, y_e, z_e, θ_4, and θ_5 values, we used equations (1), (2), and (3):

$$d_1 = z_e - a_5 sin\theta_5 \quad (1)$$

$$d_3 = 24 - (x_e + a_5 cos\theta_5 + 2) \quad (2)$$

$$d_2 = y_e - a_5 sin\theta_5 \, cos\theta_4 + 2 \quad (3)$$

D. *End Effector Motion Planning*

In order to determine the path travelled by the end effector, we needed to first define a trajectory for the transverse nail curvature. We assumed that the nail maintains a constant cross section along the length of the nail, and therefore the same parabolic equation defining x_e as a function of ye can be used for any value of z_e as seen in equations (4) and (5):

$$\theta = \text{asin}\left(\frac{w}{2R}\right) \quad (4)$$

$$x_e = \sqrt{R^2 - y_e} - Rcos\theta \quad (5)$$

Where θ, w, and R are defined as the following dimensions when looking at the nail cross section (Figure 2):

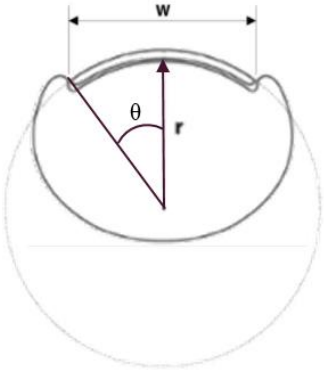

Figure 2: Nail Cross Section

$$s = 2R\theta \quad (6)$$

We discretized the arc length (6) into N points and then calculated the θ_5, ye, and x_e values for each point on the arc length. The path of the end effector was defined to follow the transverse trajectory of the nail, taking one second to move from point to point and one second to dispense the paint at each location.

Once reaching the end of a trajectory, the end effector incrementally moves along the length of the nail (z_o direction) to eventually cover the entire nail area. To operate efficiently, the end effector moves in a "snake" pattern: once it moves along the nail parabola in the positive y_o direction, it will increment along the length of the nail to the next zo value, and then transverse the nail in the negative y_o direction.

To complete the motion planning, we defined a matrix containing end effector position and orientation as a function of time. The columns of the matrix are the following: time, x_e, y_e, z_e, θ_4, and θ_5. Then, we were able to use this motion planning matrix as inputs for the inverse kinematics to determine the joint variables as a function of time. Lastly, we were able to use these joint variables to populate the DH table for each point in time to develop a robot simulation.

III. Results

A. *Joint Angles vs Time*

Using the motion planning and inverse kinematics codes, we created five plots showing each joint variable versus time for each finger. The plots of joint variables for each finger have a similar pattern, excluding that of θ_4. The d_1* increments along the length of the nail with time until it is equal to length of the nail. Both d_2* and d_3* exhibit sinusoidal behavior versus time as they increase and decrease with each nail parabola. Meanwhile, θ_5* will change due to the changing d_2* and d_3* values. Figures 3-8 show all the joint variable vs time plots for the thumb and the θ_4* vs time plot for index finger (to show how it differs).

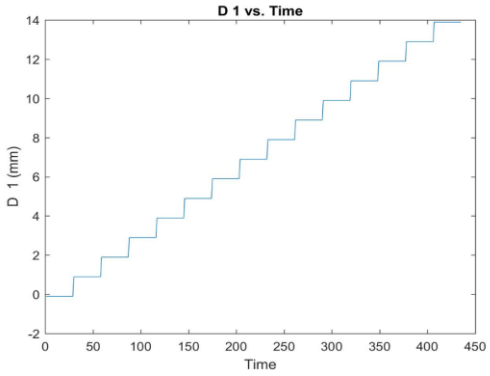

Figure 3: Thumb d_1* vs Time

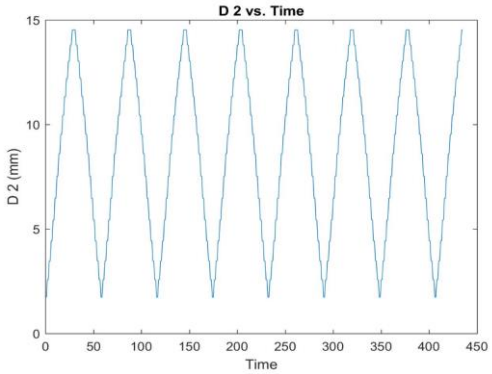

Figure 4: Thumb d_2* vs Time

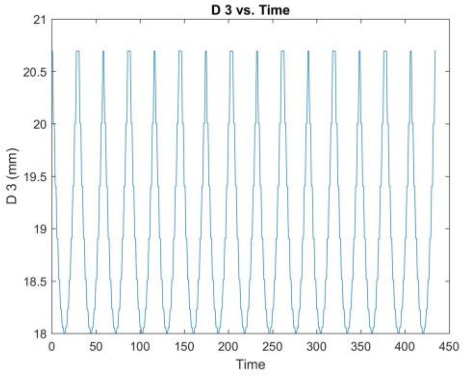

Figure 5: Thumb d_3* vs Time

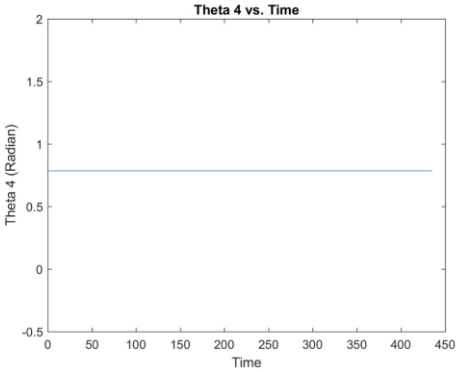

Figure 6: Thumb θ_4* vs Time

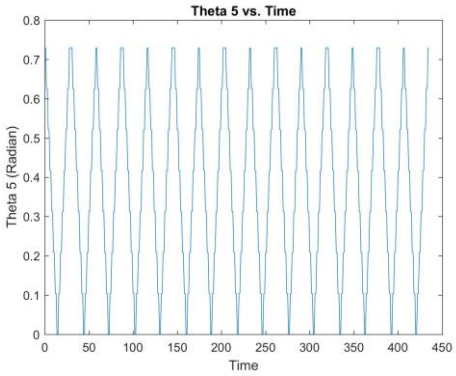

Figure 7: Thumb θ_5* vs Time

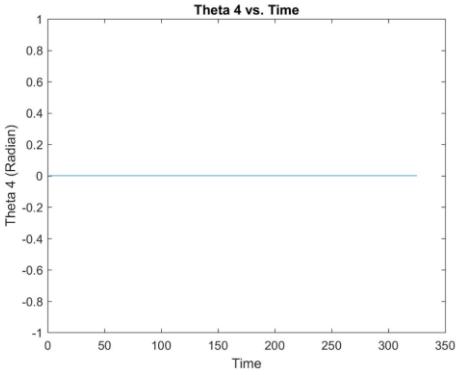

Figure 8: Index θ_4* vs Time

B. Simulation Results

Using the inverse kinematics equations, the values of the DH table can be determined for each moment in time. Then, the DH table values are inputted into the forward kinematic equations and the drawRobot code is used to simulate the robot. Figure 9 shows a snapshot of the simulation. The red dot represents the end effector. In the presented orientation, the end effector moves from left to right. After it finishes one arc length of nail, it then moves backwards in the z-axis direction. This process will continue until it the complete nail area is covered.

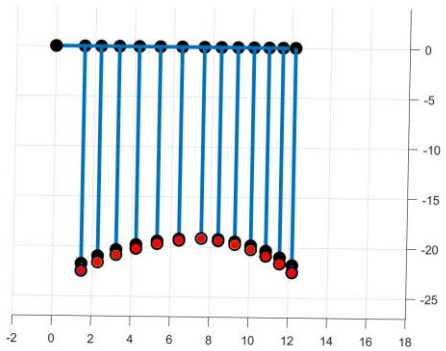

Figure 9: Simulation Snapshot

IV. Conclusion and Future Applications

Nailbot presents a novel approach to automated nail painting, accounting for the different geometries, positions and orientations of the fingernails. The five degrees of freedom allow Nailbot to position itself in the x, y and z directions, account for the finger angles (specifically the thumb, which naturally lies on an angle), and orient the end effector orthogonal to the nail's surface. These freedoms allow the manipulator to cover the contour of the nail with a clean and consistent coat of paint. We calculated a parabolic trajectory based on each nail's geometry, so that the manipulator could traverse this path while pausing at discretized units to spray the paint. Whereupon completion of the curve, the manipulator moved along the length of the finger, and traversed the parabolic curve in reverse. This motion planning continued until covering the entire length of the nail. We successfully simulated the complete transverse parabolic trajectory for each discretized point along the length of the nail. This shows that for provided geometries of any fingernail, our manipulator can automate the nail painting process.

Allowing one second to spray at discretized points, and one second for the manipulator to position along the length of the nail after a completion of a parabolic path, plus two seconds to adjust to the thumb finger, the total time to coat five fingers is fifteen minutes. Although the speed wasn't the focus of this paper, the time is consistent and competitive with a human performing the manicure. For future extensions, Nailbot can be fitted with different end effector attachments for a complete manicure and pedicure. Such attachments would provide nail cleaning, filing, and even possibly a hand massager. Extended functions include tattoo and henna applications.

References

[1] OPI Products, "2012-2013 Industry Statistics." Nails Magazine, files.nailsmag.com/Market-Research/NAILSbb12-13stats.pdf.

[2] J. W. Jung, K. S. Kim, J. H. Shin, Y. J. Kwon, J. H. Hwang, and S. Y. Lee, "Fingernail Configuration," Archives of Plastic Surgery, vol. 42, no. 6, Nov. 2015, pp. 753–760.

[3] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*. New York: Wiley, 1989.