1 Introduction

Robotic 3D Printers allow engineers and designers to rapidly prototype new mechanisms, products, or art. There are two related goals of this project. The first is to simulate the kinematics, and dynamics of a robotic 3D printer, and the filament particles it extrudes. The second is to simplify the simulation by removing the electric and drag forces acting on the particles, and optimize this simplified simulation using a Genetic Algorithm. The Genetic Algorithm was tasked with minimizing the difference between a desired path, and the path generated by the simulation. To do this, design strings of 4 paramters, 3 angular velocities, and the initial velocity of extrusion, were randomly generated, tested, ranked, and new strings were bred from the top performing design strings.

2 Background and Theory

The motion of the dispenser at the end of the robotic arm is described by:

$$\boldsymbol{r}^d = \boldsymbol{r}_0 + (x_d, y_d, z_d)$$

$$x_d = L_1 cos(\Theta_1) + L_2 cos(\Theta_2) + L_3 sin(\Theta_3)$$
$$y_d = L_1 sin(\Theta_1) + L_2 sin(\Theta_2)$$
$$z_d = L_3 cos(\Theta_3)$$
$$\boldsymbol{v}^d = (\dot{x}_d, \dot{y}_d, \dot{z}_d)$$
$$\dot{x}_d = -L_1\dot{\Theta}_1 sin(\Theta_1) - L_2\dot{\Theta}_2 sin(\Theta_2) + L_3\dot{\Theta}_3 cos(\Theta_3)$$
$$\dot{y}_d = L_1\dot{\Theta}_1 cos(\Theta_1) + L_2\dot{\Theta}_2 cos(\Theta_2)$$
$$\dot{z}_d = -L_3\dot{\Theta}_3 sin(\Theta_3)$$

When a droplet is extruded from the dispenser, it exits with an initial position and velocity:

$$\boldsymbol{r}_i^0 = \boldsymbol{r}^d,\ \boldsymbol{v}_i^0 = \boldsymbol{v}^d + \Delta\boldsymbol{v}^d$$

Where $\Delta\boldsymbol{v}^d$ is the extrusion velocity, one of the design parameters.

The dynamics of each droplet is described by Newton's second law:

$$m_i\boldsymbol{a}_i = \boldsymbol{\Psi}_i^{tot} = \boldsymbol{F}_i^{grav} + \boldsymbol{F}_i^{elec} + \boldsymbol{F}_i^{drag}$$

The force of gravity acting on each particle is:

$$\boldsymbol{F}_i^{grav} = (0, -m_i g, 0)$$

The force of the electric field is:

$$\boldsymbol{F}_i^{elec} = \sum_{p=1}^{N_c} \frac{q_p q_i}{4\pi\epsilon||\boldsymbol{r}_i - \boldsymbol{r}_p||^2}(\boldsymbol{r}_i - \boldsymbol{r}_p)$$

Where N_c is the number of point charges, q_p is their charge, and $\boldsymbol{r}_p$ are the positions of the point charges. The force of drag is:

$$\boldsymbol{F}_{d,i} = \frac{1}{2}\rho_a C_{D,i}||\boldsymbol{v}^f - \boldsymbol{v}_i||(\boldsymbol{v}^f - \boldsymbol{v}_i)A_i^D$$

The coefficient of drag depends on the Reynolds number of the particle:

$$Re = \frac{2R\rho_a||\boldsymbol{v}^f - \boldsymbol{v}_i||}{\mu_f}$$

$$C_{D,i} = \begin{cases} \frac{24}{Re} & 0 \leq Re \leq 1 \\ \frac{24}{Re^0.0646} & 1 \leq Re \leq 400 \\ 0.5 & 400 \leq Re \leq 3 * 10^5 \\ 0.000366 Re^0.4275 & 3 * 10^5 \leq Re \leq 2 * 10^6 \\ 0.18 & 2 * 10^6 \leq Re \end{cases}$$

Once the total force on a member is known, its position and velocity are updated using the Forward Euler method.

$$\boldsymbol{r}_i(t + \Delta t) = \boldsymbol{r}_i(t) + \boldsymbol{v}_i(t)\Delta t$$

$$\boldsymbol{v}_i(t + \Delta t) = \boldsymbol{v}_i(t) + \boldsymbol{\Psi}_i^{tot}(t)\frac{\Delta t}{m_i}$$

The filament used in this simulation is a mixture of two materials, therefore some of its material properties are a mixture of those of two disparate materials. Its effective density, and charge capacity are:

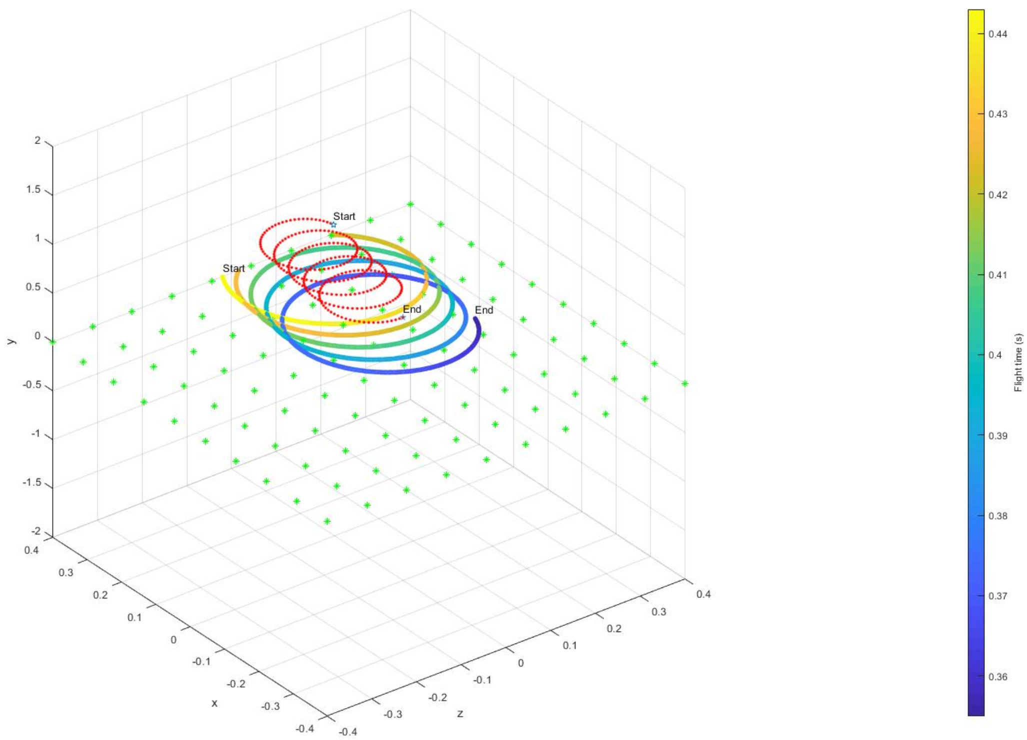
$$\rho^* = (1 - \nu_2)\rho_1 + \nu\rho_2$$

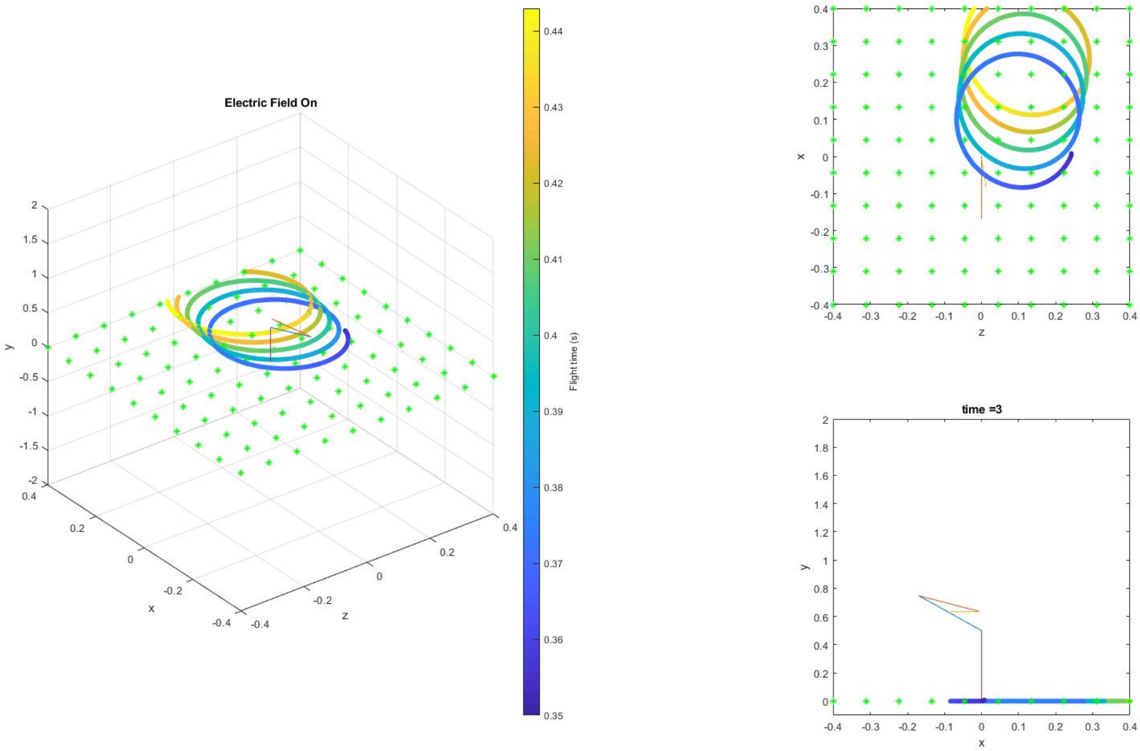
$$q^* = (1 - \nu_2)q_1 + \nu q_2$$

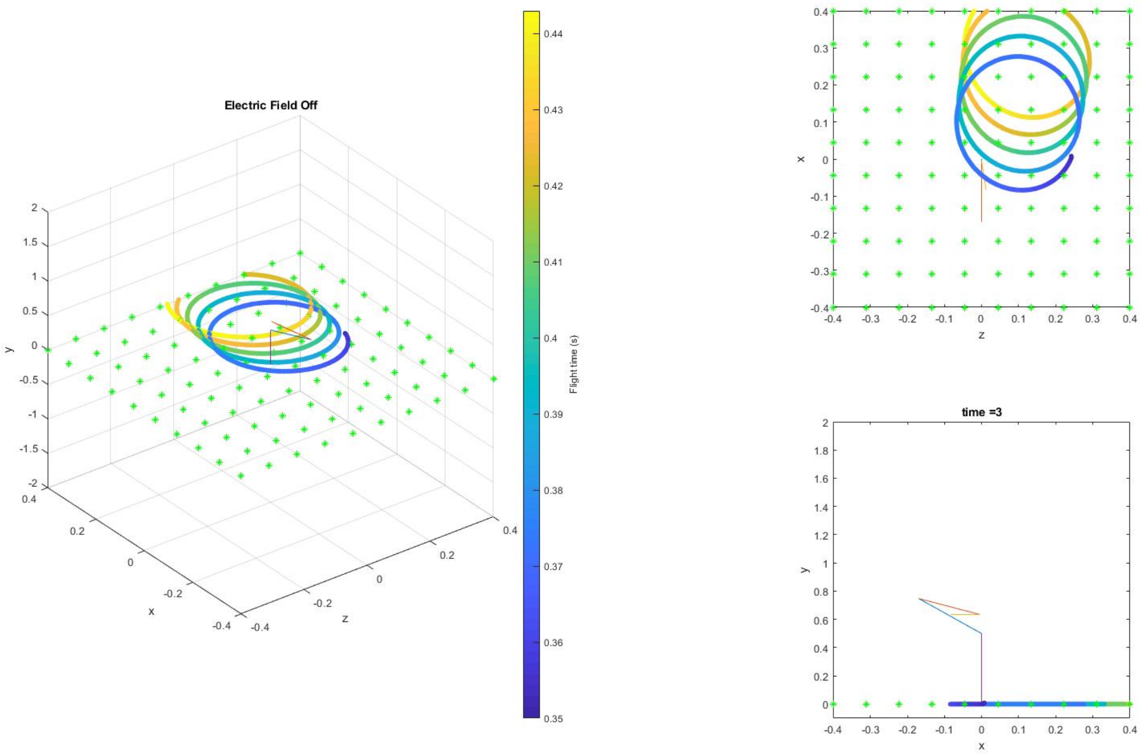
Where ρ_1 and ρ_2 are the densities, and q_1 and q_2 are the charge capacities of the materials in the mixture, and ν_2 is the volume fraction of the mixture.

3 Results and Discussion

3.1 Full Model With Electrical Forces



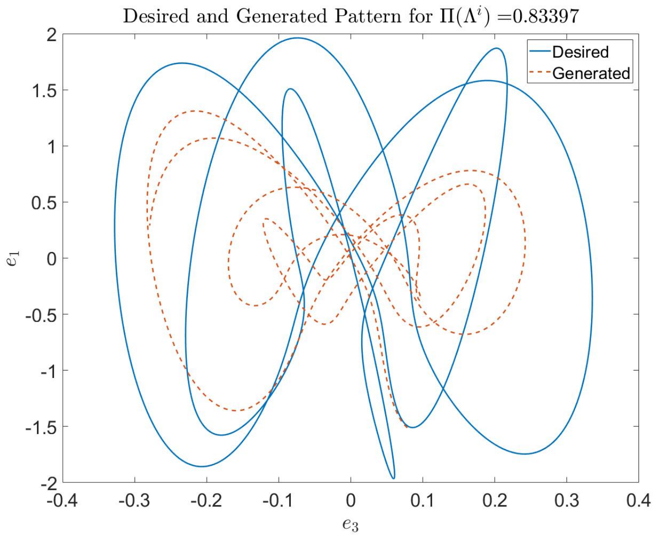




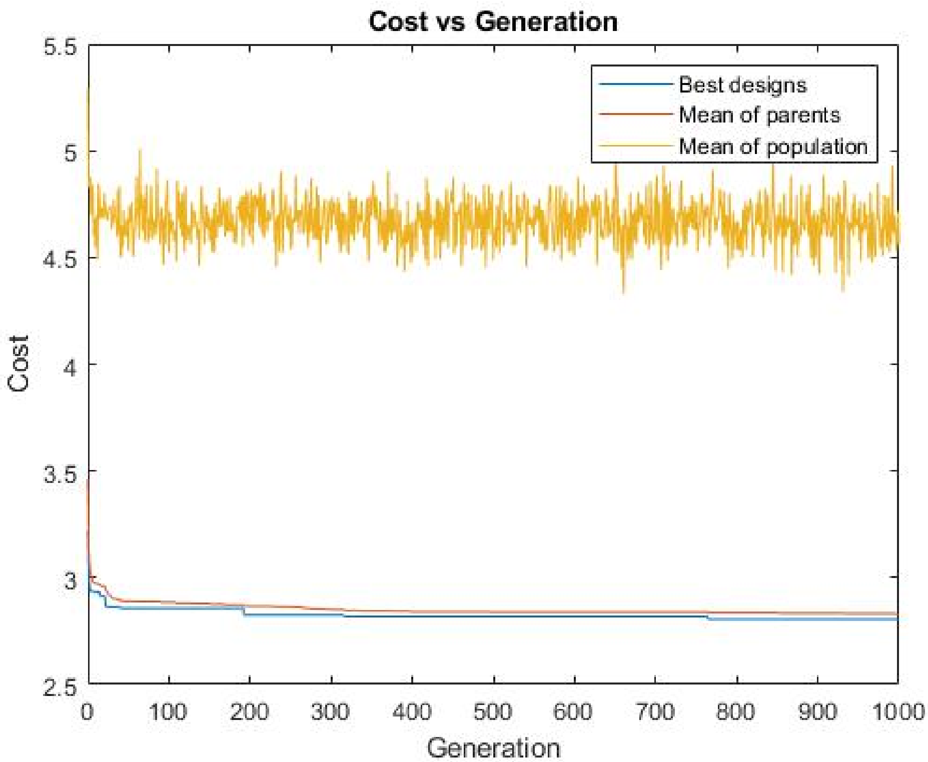
The effects of electrical force are very minimal on the simulation compared to gravity, and even drag. The very small charges, and high electric permittivity combine for a electrical force on the magnitude of 10^{-28}. The force of gravity is on the order of 10^{-4}, completely masking the electrical force. The major effect electrical force has on the simulation is its contribution to runtime. The operations that need to be performed on 3 dimensional arrays are costly, even without nested for loops. Dropping this term significantly speeds up the simulation, and has little affect on the outcome. In part 2, we will forgo electric forces, and drag in order to speed up our simulation, and run a Genetic Algorithm on it.

3.2 Simplified Free Fall Model With Genetic Algorithm

For this implementation of our Genetic Algorithm, the number of genetic strings per generation, S, was 100, the number of parents, P, was 10, and 1000 generations were tested. The design parameters that were optimized were the angular velocities of the linkages, and the extrusion velocity, Δv^d.



The genetic algorithm did not result in a generated outcome that matches the desired particle positions. The cost vs generation plot fell steeply in the beginning, then plateaus for hundreds of generations multiple times across the duration of the simulation. This could be the effects of inbreeding which would cause the simulation to stagnate around a false minimum between the best parent strings. The infrequent drops after plateauing could be due to randomly generated strings that performed better than the inbred strings.



$DESIGN$	Λ_1	Λ_2	Λ_3	Λ_4	Π
1	15.9990	15.1039	6.9735	-3.0081	0.8340
2	15.9980	15.1287	6.8965	-3.0396	0.8169
3	15.9933	15.2073	6.9673	-3.0033	0.7639
4	15.9942	15.1308	6.8558	-3.0325	0.8114

Table 1: Top 4 Design Parameter Strings. Π calculated using comparepattern.p.

4 Conclusion

In this project, a robotic 3D printer was simulated. The dynamics of the deposited particles were used to update their positions via forward Euler's method, and a string of design parameters controlled the rotational velocities of the linkages, and the extrusion velocity. It was determined that the effects of electricity on the particles is minimal, so in a second portion of the project, the forces of drag and electric field were discarded in favor of a fast run-time. In this second portion, this simplified simulation was fed into a Genetic Algorithm that randomly generated, and bred the design parameters across multiple generations to find a design string that resulted in an extruded pattern most similar to a reference pattern. The optimal design parameters after testing resulted in a path that was somewhat accurate to the desired pattern. Errors between the desired, and generated patterns could be due to inbreeding, or an error in the simulation.