RBE 501 - Final Report

Patient-specific Neurosurgical Robot Optimization

Team GreyMatters
Michael Brauckman Daniel Miller

***Abstract*—Robot-assisted neurosurgery performed within the confines of an MRI scanner can provide numerous benefits for both the patient and the surgeon. The surgeon can view progress in real time, ensuring a high degree of accuracy and control, decreasing procedure duration, and reducing the frequency of errors. In this paper, we calculate the forward and inverse kinematics for the 7 DOF RCM manipulator, and to customize the arm's linkages to best fit a given patient and operation. These optimized linkage configurations will provide some mechanical collision avoidance, while guaranteeing full access to the surgical site.**

I. Introduction

Neurosurgery has traditionally been limited by a lack of vision inside the cranial cavity during the surgical procedure. Typically the surgeon uses an image taken previously with an MRI machine to plan and perform the surgery. This method gives no visual feed-back to the surgeon. By performing the surgery with a robotic device inside of an MRI machine, real-time visual feedback can be provided to the surgeon increasing the ability to adapt to situations arising during an operation, reducing the duration of the procedure, and reducing the incidence of surgical errors. A device capable of performing such surgeries is currently being developed at WPI. The device consists of a 4-DOF remote center of motion mechanism (RCM) capable of achieving any approach angle around a fixed point, and a 3-DOF planner actuator which moves the RCM effectively adjusting the location of the center of motion. The target workspace inside the cranium, as well as the shape of the cranium itself, and the allowable entry location on the surface of the cranium can vary greatly between different patients or types of surgery. Additionally the space inside the bore of an MRI machine is limited and the device must work around the patients head. Consequently it may not be possible to design a single device, for general use with all patients and surgeries, which remains within the confines of the MRI machine.

This project seeks to generate a set of parameters for the links of the mechanism which will allow a surgery to be performed without unwanted collisions between the mechanism and the MRI machine or the patient's cranium. When given the size of the MRI machine, a model of the patient's head, the desired workspace inside the brain, and the size and location of the allowable entry region to the cranium, the program must find a set of link parameters that allow the workspace to be reached through the entry region without collisions.

In order to accomplish this task we propose a set of kinematic models describing the position of each link, combined with collision detection algorithm allowing configurations to be tested for suitability, and an iterative design cycle. The design cycle consists of proposing candidate designs, evaluating the suitability of a candidates, and altering those designs until a configuration is found which satisfies all the constraints. the following section describes the work done so far on this proposal and indicates tasks which will be developed through future work.

II. Background

A. Genetic Optimization and Robotics

Traditional optimization methodologies such as hill climbing algorithms or numerical optimization methods may fail in large and complex domains. Genetic algorithms excel in very large and potentially non-linear search spaces. These algorithms are based on a heuristic approach and mechanisms of evolutionary biology and genetics. A population of candidate solutions is generated in a random fashion. Each individual in the population is then evaluated heuristically to determine how well it approximates a solution to the optimization problem. A new generation is created by probabilistically selecting individuals based on their fitness evaluation and generating new individuals by combining attributes of those selected a process called crossover. Further adaptations are created be making random by alterations to some attributes this is call mutation. The new population is evaluated again and the process continues until it converges on a solution with a high fitness function. It has been suggested "... that optimization at large may benefit greatly from the adaptive optimization exhibited by natural systems when attempting to solve complex optimization problems and that the determinism of classical optimization models may sometimes be an obstacle in nonlinear systems". [3]

B. Collision

Collision detection between models in a virtual environment is a critical feature in a number of fields. Video games, physics simulations, and especially robotics make heavy use of collision detection. As early as 1995, computer scientists were devising ways to efficiently handle complex physical interactions in cluttered environments [2]. However, as Cohen, Lin, Manocha, and Ponamgi demonstrate, the collision detection run time is proportional to the number of objects (polytopes) and the level of detail on each object.

The graphics industry has changed dramatically since 1995, and the models used by scientists and engineers have only become increasingly complex. A number of different methods have been proposed which allow for efficient detection of collisions in these complex environments [1]. However, currently, the most popular method is to approximate the volume of a complex, concave model with a number of geometric primitives. Collision detection using shape primitives has proven to be an extremely salable solution to the complex collision problem. Numerous methods for high-accuracy collision detection exist, and have been tested thoroughly [4]. As such, due to time constraints, detailed collision detection is considered to be outside the scope of this project.

III. Methods

The goal of this project is to find a set of link parameters that allows the robot to reach the entirety of a given workspace inside of a patient's brain by extending the needle through a designated entry point on the surface of the scull. It is critically important for patient safety that the robot does not collide with either the patient or the bore of the MRI machine. In order to satisfy this highly non-linear set of constraints, a Genetic Optimization algorithm is used to search the very large space of possible configurations. The fitness of each candidate configuration will be based on the existence of solution to the inverse kinematic problem, which both pass through the desired entry location, and are non-colliding. For a design to be accepted it must have such a solution for each point in the discretized workspace. The following sections detail the process of: developing the kinematic solutions for the surgical robot, modeling the patient, robot, and MRI machine, performing collision detection, and performing the genetic optimization.

A. Kinematic Model

The first step in evaluating a particular linkage configuration is calculating the kinematics of the system. Joint angles needed to reach a particular goal point in the workspace must be calculated, and the link models must be moved and rotated to align with that configuration. Kinematic models allow this calculation to be performed. Forward kinematics are used to find the location and orientation of the coordinate frames for each link and can be used to place the link models appropriately. Inverse kinematics are used to determine the joint angles needed to place the tip of the end effector at a given point.

1) Forward Kinematics: The forward kinematics can be derived by the Denavit-Hartenberg (DH) convention. Using the DH convention a coordinate frame is defined for each link using four parameters. The mechanism's 3D CAD models were used to visually aid in placing the coordinate frames and identify the corresponding DH parameters shown in Table 1. Note that links 5 and 6 use the same joint angle variable this represents the mechanical link in the RCM while allowing a reference frame on both links. Once the DH parameters were determined, a MATLAB function was written to calculate the transformation matrix of each joint, and plot the position of the arm. In addition to joint parameters the Length of each link can be passed to the function which is needed to test alternative link length and shapes.

Link	d	θ	a	α
1	Z	0	0	$-\pi/2$
2	Y	$\pi/2$	0	$\pi/2$
3	X	π	0	$\pi/2$
4	0	θ_1	0	$\pi/2$
5	0	θ_2	L1	0
6	0	$\pi/2-\theta_2$	L2	0
7	0	$\pi/2+\theta_2$	L3	$-\pi/2$

TABLE I
DH Parameter Table

2) Inverse Kinematics: Solving the inverse kinematic problem allows us to find a set of required joint angles, given a target location. The RCM mechanism used has 7 degrees of freedom and which means that it is a redundant mechanism and infinitely many solutions could exist to the inverse kinematic problem. Many iterative methods exist for finding a single inverse kinematic solution. However, in an effort to properly measure the quality of a given linkage configuration, a geometric IK solution was generated.

A geometric IK approach allows searching along the null space of a given IK solution. If a specific point of entry to the skull is chosen then a line from the desired point through this entry point can be drawn this line defines the orientation that the arm must be in to reach the goal location through the chosen entry location. In other word the needle must be position along the line from the goal point through the entry point. There are still an infinite number of solutions along this line corresponding to movement of the XYZ table some distance along the line and extension of the needle in the by the same distance. However this allows us to search along the line by changing a single parameter the extension of the needle. If no solution is found and the allowable entry area is larger than one point a different point could be chosen and the process repeated.

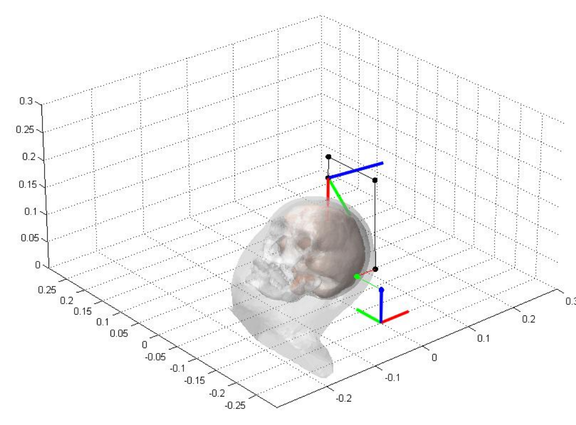


Fig. 1. MATLAB Stick and Head Model

B. Parametric Models

The chosen collision detection algorithm requires a set of points distributed over the surface of the RCM to be checked for collision. a Model of the RCM was needed which could provide these points and which was adaptable to changes of link length and radius during the optimization routine. Parameterized models of the links were constructed in MATLAB. These links can be arranged to represent the full RCM using the transformations for each link generated by the DH parameters. The Models are constructed as 4xN arrays where each column contains 4 points which are the vertices of a surface. Keystone shapes are generated to approximate the curvature of the links to a user defined precision. Figure 2 shows one link generated in this way, and Figure 3 shows a fully model of the RCM assembled from these links. The vertices of each surface defined are used for collision detection and the patch command is used to visually represent the arm.

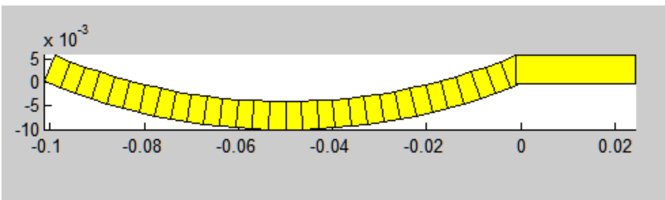


Fig. 2. MATLAB model of a single link

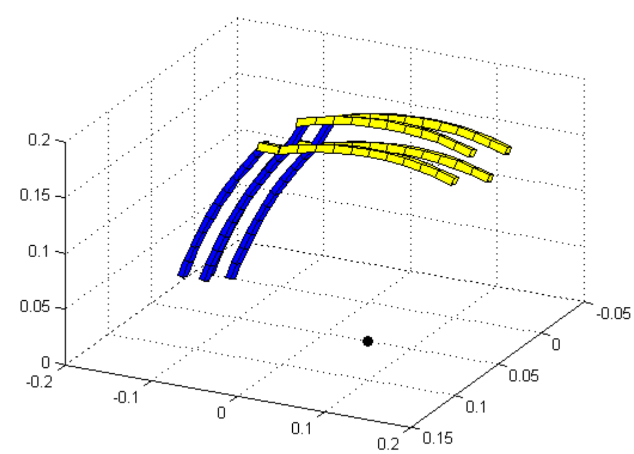


Fig. 3. MATLAB model of full RCM

C. Collision Detection

As expressed above, transformations for individual links can be calculated from the DH parameters. These transformations will be used to determine the placement of each link for use in collision detection. During the optimization process, the physical parameters of these models (such as link length and curvature) must be altered in order to test different configurations.

After a brief review, the team discovered that detection of collisions between arbitrary manifolds (i.e. not necessarily convex) is quite difficult. In an effort to simplify the collision detection portion of this project, the team explored the feasibility of many other collision detection methods, including discretizing the workspace as a voxel grid, and approximating the various models as primitive shapes. The team elected to use shape primitives for collision detection, as they are significantly faster than many mesh-based approaches. As detailed previously, these geometric primitives are a suitable alternative to mesh-based collision detection.

In this paper, the various models are approximated with shape primitives. The patient's head is comprised of a set of spheres, whereas the MRI machine itself is simply modeled as a cylinder. As detailed in section III-B, the linkages within the RCM mechanism are constructed geometrically, as a set of points. Although these geometric primitives to not accurately reflect the finer details of the models within the simulation, they serve as a very good first (and fast) approximation. The set of spheres used to define the shape of the patient's head is shown in Figure 4.

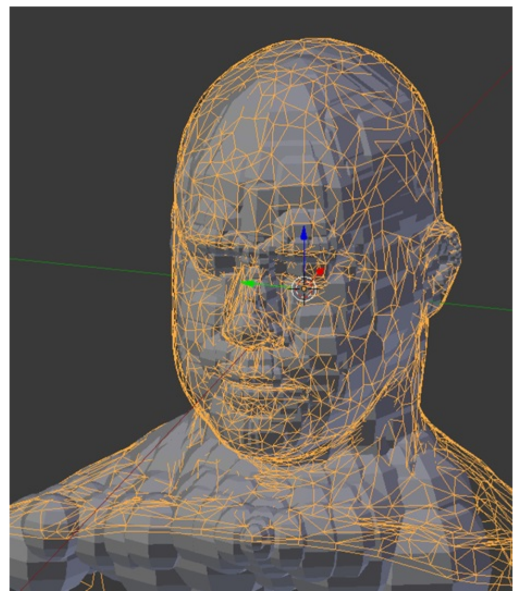


Fig. 4. Head modeled as a set of spheres

Using these shape primitives, collisions are easily detected.

D. Genetic Optimization

MATLAB's Optimization toolbox was used to implement the genetic algorithms. This tool boxes has built in methods for selection, crossover, mutation, and generation. The toolbox provides a number of options for each of these genetic operators allowing for customization to the problem at hand. After some testing with various parameters the generation mechanism was set to only allow feasible solutions and constraints were added to limit the range of values for each link and to constrain the radius of a link to be less than half of that links length. The first constraint prohibits individuals with links that are too long or less than zero which would not be practical, and the second prevents links where the bend in the link would be more than half a circle which will no longer aid in collision avoidance and would break our link models causing inaccurate collision detection. The optimization routine generates values for a defined set of parameters and evaluates them by calling a user defined fitness function which must take those parameters as arguments. The fitness function should return smaller values

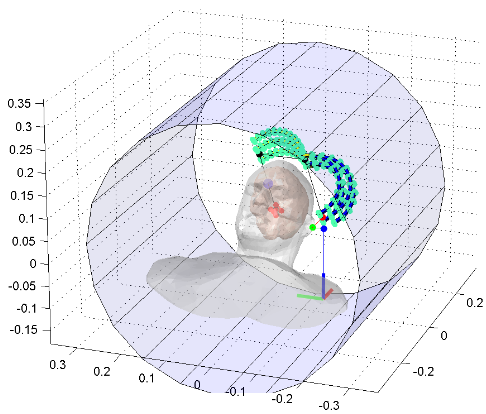


Fig. 5. Simulation #21: Candidate linkage after 35 generations with a population of 25

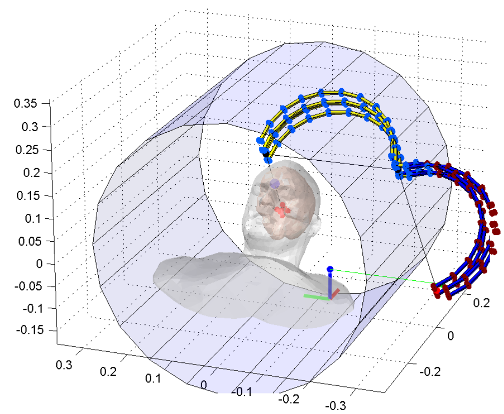


Fig. 7. Simulation #3: Invalid candidate linkage

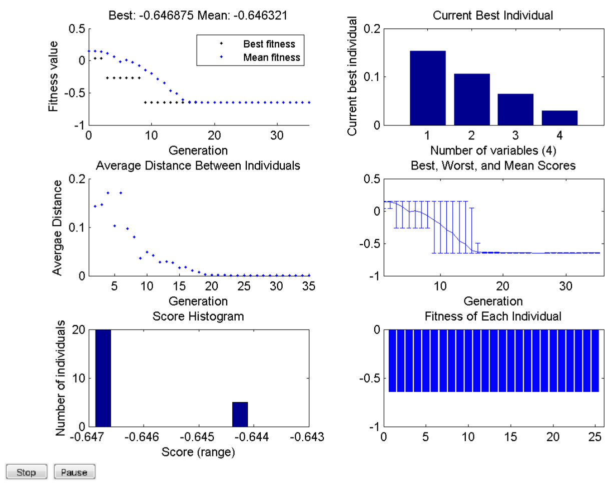


Fig. 6. Statistics of simulation #21

for more fit functions. Currently our fitness function takes into account the percentage of the workspace with at least one non colliding inverse kinematic solution, and the totally percentage of inverse kinematic solutions which do not collide. The fitness function is calculated as a weighted sum resulting in a number between 0 and 1 where 1 would indicate the all inverse kinematic solutions were collision free and 0 that all inverse kinematic solutions collided. Since our primary acceptance criteria for a configuration is that all points in the workspace have at least one non colliding solution that percentage is weighted most heavily. Finally the function is inverted to meet the requirement of the optimization tools that the lower values are better in the fitness function.

IV. Results

The optimization routine was tested on a simplified example with only 7 points in the workspace and a single allowable entry location. 35 tests were run allowing populations of 10 configurations to evolve through 35 generations. 33 of these test converged onto an acceptable design within 35 generations, while 2 failed to produce a design which did not collide with the MRI machine. In the cases where no working design was found it was observed that no design was ever generated with a fitness function less than 0, that is to say that no design was ever generated that had any non-colliding inverse kinematic solutions. A higher population size would greatly reduce the probability of this occurring; however, it also suggested that more granularity in the fitness function may be needed. As long as one design was found with a non zero fitness function the algorithm quickly converged on a solution hence a refined fitness function which gave some evaluation of a design even if it collided in all configuration would direct the search and aid in convergence.

Figures 5 and 6 show the resulting linkage and simulation statistics of a simulation conducted with 35 generations of 25 individuals each. This simulation was repeated 35 times, producing a variety of unique linkages, some of which are shown in the appendix. Figure 7 shows the resulting linkage of a simulation which did not yield a viable candidate linkage. Archives of each simulation are available with this document.

V. Conclusions

In this document, it has been shown that genetic optimization is well suited to patient-specific optimization of a neurosurgical robotic manipulator. Using an approximate collision detection, numerous permutations of a parametric linkage model are compared. These various linkage candidates were compared against one another in a genetic optimization. These optimizations were run multiple times, with random starting configurations. The populations in these simulations reliably converged to viable linkage candidates with relatively few generations.

VI. Future Works

Further experimentation is needed to find an improved fitness function. There are many desirable parameters we wished to include in the system's optimization, but were not included due to time constraints. For example, weighting the average or minimum distances to the nearest obstacle would greatly improve the resulting candidate linkages. To avoid populations which have no viable candidates, a parameter which measures a model's penetration into an obstacle. By reducing fitness proportionally to this penetration measure, the genetic optimization will begin to favor links which collide less. This will prevent populations from getting ”stuck” if it has no viable candidates.

The linkage models used in MATLAB were simple approximations of the real RCM mechanism. Further work is required to accurately reflect the geometry of the arm, including imposing mechanical limits on the various DOF. Ideally, the parameters generated by the program would be piped directly into CAD software, updating the models accordingly. These models could then be 3D printed

VII. List of Deliverables

- Final Report (This Document)
- Software archive (code.zip)
- Optimization results archive (results.avi)
  - Trial 1: 35 generations, 25 population (good fitness function)
  - Trial 2: 35 generations, 25 population (poorly formed fitness function)

References

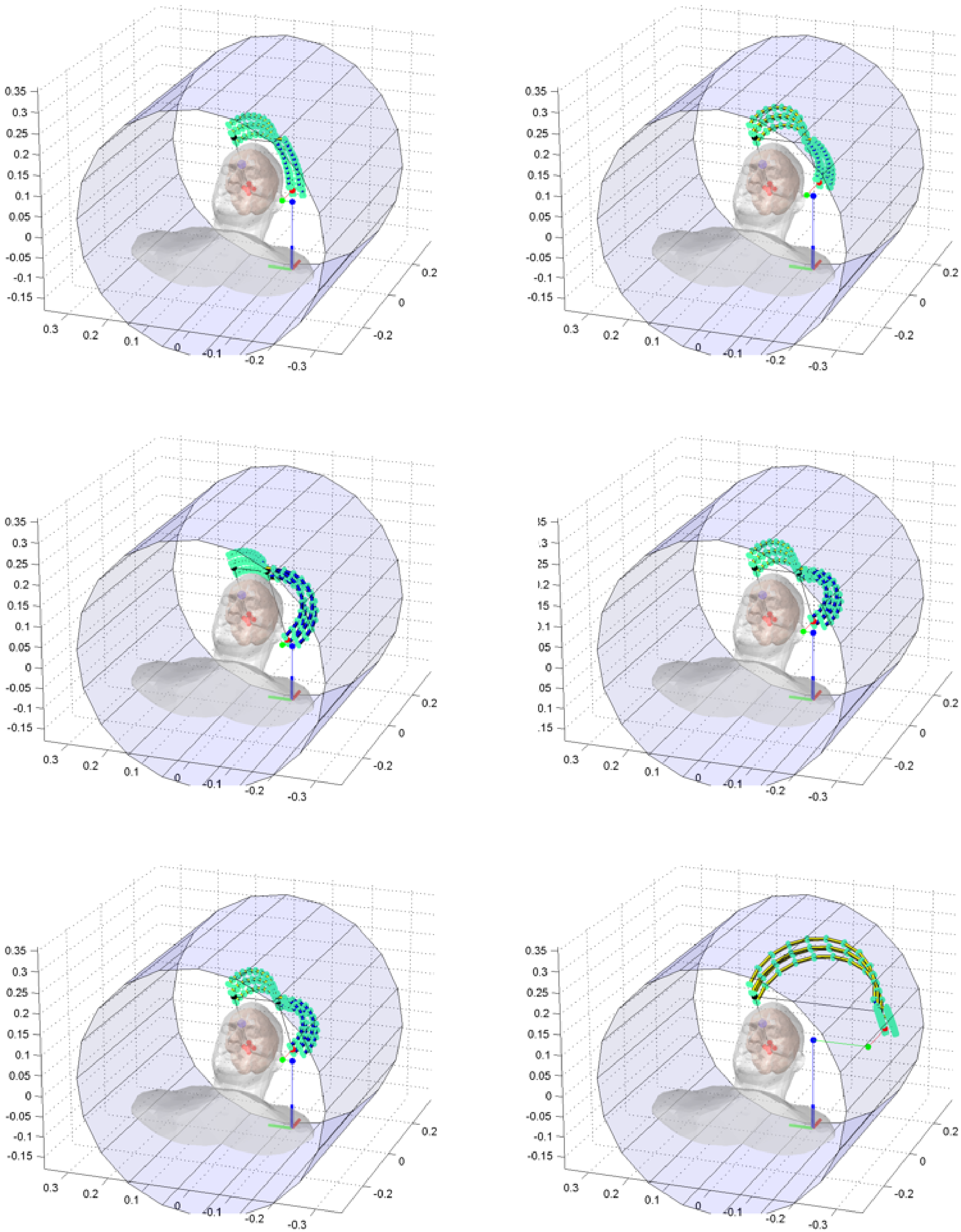
[1] Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.

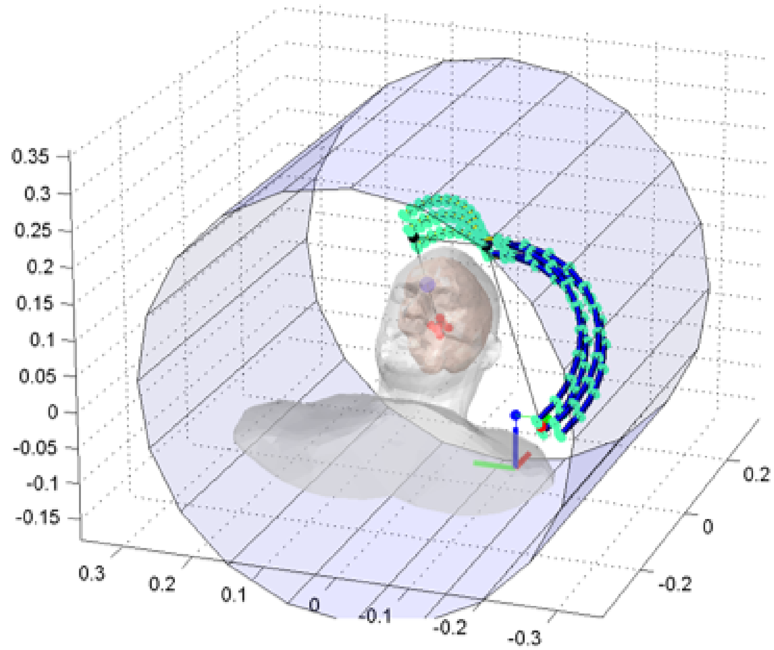
[2] Jonathan D Cohen, Ming C Lin, Dinesh Manocha, and Madhav Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 189–ff. ACM, 1995.

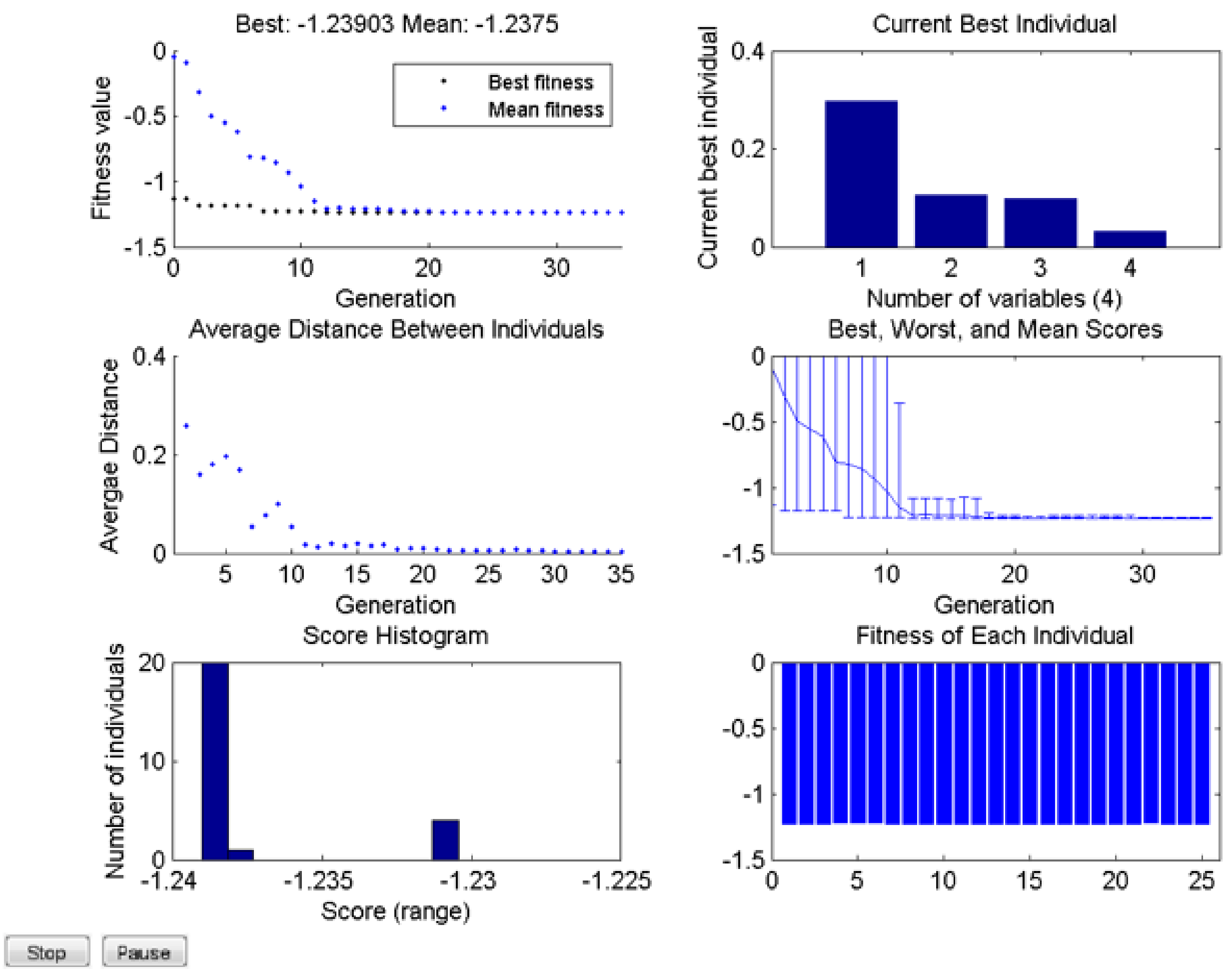
[3] Yuval Davidor. *Genetic Algorithms and Robotics: A heuristic strategy for optimization*, volume 1. World Scientific, 1991.

[4] Pablo Jiménez, Federico Thomas, and Carme Torras. 3d collision detection: a survey. *Computers & Graphics*, 25(2):269–285, 2001.

APPENDIX
REFERENCE GRAPHICS






Best: -1.23903 Mean: -1.2375
Best fitness
Mean fitness
Fitness value
Generation
Current Best Individual
Current best individual
Number of variables (4)
Average Distance Between Individuals
Avergae Distance
Generation
Best, Worst, and Mean Scores
Generation
Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Stop
Pause