Motion Planning for Needle Steering*

RBE 550 Group Project Proposal

Farid Tavakkolmoghaddam Emmanuel Jayaraju Ryan Carnemolla Febin Fredi

***Abstract*— This paper presents a method for motion planning for a surgical needle robot. Through the use of a bicycle kinematic model and an RRT-based motion planning algorithm, the planner was able to generate paths for the robot to follow quickly and efficiently while adapting to obstacles as they are discovered. Obstacle is especially important in surgical robotics as obstacles represent potentially critical body tissue. The needle is bevel-tipped and non-holonomic, which creates a unique method of motion. Even with this unique mode of motion, the method presented in this paper was capable of reliable motion planning for the needle.**

***Index Terms*—motion planning, RRT, bicycle model, non-holonomic, surgical robot**

I. INTRODUCTION

The advancement of technology in medical imaging have opened up new avenues towards image-guided therapy (IGT) procedures where an imaging modality is used to capture images of desired anatomy and surgical tool and provides them as feedback to the surgeon to help with the navigation of the tool to a desired target region within the anatomy. Several imaging modalities are developed and are currently used for a wide range of therapeutic and diagnosis applications. Each imaging modality have unique characteristics which makes them suitable towards specific applications and are generally selected based on the anatomy of interest, surgical procedure, imaging resolution, image acquisition time, cost of imaging, ionizing radiation, ease of use, and portability. Among imaging modalities used ultrasound (US), magnetic resonance imaging (MRI), computed tomography (CT), X-ray, fluoroscope are among the most frequently used modalities for diagnosis or IGT procedures [1].

One area where IGT has gained footing is prostate biopsy. On average, one out of eight men are diagnosed with prostate cancer at some point in their lives, which translates to about 170,000 new cases in the US every year as of 2018 [2]. The best way to lower the mortality rate is through early diagnosis [2], which is typically done through collecting samples of the prostate by steering a biopsy needle in conjunction with an image guidance into a desired region in prostate. US is the most common imaging modality used for this procedure which provides a 2D image of the anatomy and needle. Although use of US has benefits such as low operation cost, no ionizing radiation, portability, and can be generally done in a clique, however, low tissue resolution and contrast makes it challenging to ensure proper needle navigation into the desired prostate region which typically results in multiple needle insertions to ensure adequate sampling and avoiding false negatives [3]. Magnetic Resonance Imaging (MRI) provides 2D or volumetric (3D) imaging with excellent tissue visualization, no ionizing radiation, and close to real-time imaging which makes them an excellent modality for prostate biopsy [3].

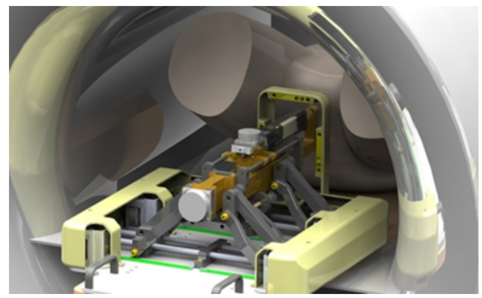

Fig. 1. Render of AIM labs prostate biopsy robot. The robot has 5 degrees of freedom including needle rotation and insertion.

Robots have been proposed for use in medical applications almost around the same time of their first introduction to industry [3]. The positional accuracy of robots along with ease of control integration with other systems made them suitable candidates for incorporation into IGT. This coined a new term robotic image-guided therapy or RIGT in short, where a robot is used along with a form of image feedback to perform therapeutic or diagnosis. One procedures where RIGT has been used is prostate biopsy where the robot holds and orients a straight or beveled tip biopsy needle while simultaneously providing insertion and rotation degrees of freedom (DoF) to enable close-loop feedback control to account for any needle misalignment in its path to the target region [4]. A six DoF MRI-compatible (safe for use within MR environment as it has no ferro-magnetic components), robotic system for prostate biopsy was developed at WPI and presented in [3], which enabled fully automated or a cooperative biospy under MRI guidance. This robotic system provided initial alignment of the needle for insertion as well as two DoF including needle rotation and insertion to steer a beveled tip needle to a desired target.

The beveled tip nature of these needles cause the needle to deflect and follow an arc path upon insertion. Rotation of the needle along its main axis can effectively provide steering capabilities which can be beneficial in avoiding obstacles or compensating for any unwanted error between the tracked needle location in MR image and its desired path.

In this paper, we investigate motion planning for a beveled tip needle that can be used for the robotic system described in [5]. Through specifying the needle rotation and insertion distance, the path of the insertion can be controlled.

A. Problem Statement

In this project we aim to implement a motion planning scheme to enable steering of a bevel tip prostate biopsy needle along a generated path from an entry point toward a goal while avoiding collision with obstacles. To achieve this task, three primary challenges are addressed in this work and are as followed:

1) Implement kinematic model of a beveled tip needle.
2) Investigate motion planning schemes and implement the one most suitable for our application
3) Refine the motion planner to generate a collision free path.
4) Implement a trajectory generation algorithm for the needle to follow the generated planned path.

Based on the above challenges we present our contributions and approach for this project:

- Implement and validate the non-holonomic kinematic model of needle in MATLAB.
- Implement Spline-RRT* non-holonomic motion planner to plan a collision-free path from the start to goal point.
- Implemented a variable-curvature trajectory generation algorithm based on duty-cycling of the needle rotation to generate a desired trajectory based on the provided planned path for the needle to follow.

The organization of the rest of this manuscript is as follows: In the sectionII we provide an overview of the related works and the robotic system this project was inspired of, in the section III we explain the methods used in this project, in sectionIV we present our simulation results and finally, in section V we discuss our findings and our future work directions.

II. Background

As we are working with a flexible needle with a bevel tip the symmetry at the tip of the of the needle causes it to bend when it's pushed through a soft tissue. The needle must be inserted through potentially critical body tissue, so improving the needle's motion planning is very important. This can help improve safety and patient experience. There are several sources of error that make controlling the needle difficult. These are target shift due to patient movement, anatomy shift due to needle insertion, registration error, and needle deflection due to bevel angle. To steer this needle through the tissue to reach a 3D target a non-holonomic model is generated for the needle [6]. The model introduced in [7] generalizes the standard three degree-of-freedom (DOF) nonholonomic unicycle and bicycle models to 6 DOF using Lie group theory. This model can be further used in path planning and control of the bevel tip needle. The model parameters are fit using experimental data. The model in this paper is used to find the behavior of needle when it's inserted into tissue and thus in finding the trajectory of the needle. Once the non-holonomic modelling the bevel tip is done, we need to design a motion planner which can be used to obtain the motion plans which will be used to steer the needle in 3D complex environments. The planner needs to be fast as the human body is not a static environment and thus should be able to find new motion plans quickly. RRT is a fast algorithm capable of satisfying these requirements [8]. This makes it extremely adaptable for many different uses [9]. The [**?**] uses RRT algorithm with new extensions and customizations necessary for fast planning of steerable needles. A reachability-guided sampling heuristic is used that alleviates the sensitivity of the RRT planner to the choice of the distance metric and yields significant improvement in performance of the planner, but the authors relax the restrictive constraint of constant-curvature needle trajectories by using duty cycling as introduced by Minhas at al. [10] to realize bounded- curvature trajectories. Other studies have similarly implemented algorithms similar to RRT, such as Bernardes et al. [2], which used the Arc-RRT algorithm to successfully generate dynamic paths for a comparable bevel-tipped, non-holonomic needle. This approach attempted to connect randomly sampled positions with their nearest node through arc links, allowing it to generate more reasonably shaped paths for the robot at a faster rate than standard RRT. Additionally, this approach used an image-guided, closed-loop feedback controller to ensure accuracy and obstacle avoidance. These insights provided crucial inspiration for the direction of this paper. The use of image data for validation and control are not limited to one example. Abayazid et al. used ultrasound (US) images for use in a similar controller [3]. The image data was taken perpendicular to the bevel tip and was used to find the difference between the expected and actual tip centroid location. Like the project of this paper and the previously mentioned related work, this study used an RRT based algorithm for motion planning through the surgical environment. In doing so, the resulting system could plan paths capable of reacting to obstacles and real-world uncertainties. Its results were recorded both in MATLAB simulation and with experimental validation on the robot. Previous work has shown that it is possible to steer the surgical needle using a bicycle model and an RRT-based algorithm.

III. Methodology

Our approach on solving the problem at hand was to break it down into 3 phases. Firstly, we implemented the kinematic model of the needle on MATLAB to verify the curvature that the needle could trace. The second phase was to develop a path planning algorithm to generate a smooth path from the start to target location which the needle could traverse considering it's non-holonomic constraints. Finally, as the third phase, the generated path is then given as an input to the duty cycling method, which does the trajectory planning, i.e. it encodes the insertion velocity and angular velocity of the joints that is needed for the needle to go through the planned way points without hitting the obstacles. This section contains further details on each of these phases.

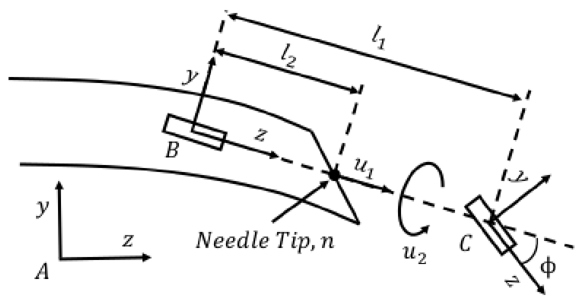

Fig. 2. Bicycle kinematic model as described in [7]. In this model the needle is approximated as a bicycle which has a fixed steering angle of ϕ, distance between two wheels of l_1, and a distance of l_2 which is between the back wheel and the needle frame. Two control inputs namely u_1, and u_2 that are insertion and rotation speed respectively, control the shape of the needle.

A. Kinematic Modelling

Kinematic model of bevel tip needle is required to accurately simulate and predict the motion profile of a bevel tip needle. Webster et.al. [7] proposed two kinematic models based on a simplified bicycle model called a unicycle and bicycle models, to model the kinematics of a bevel tip needle inserted into a tissue. This model approximated the needle as a bicycle with fixed front wheel angle ϕ which follows a circular arc once inserted into a tissue much stiffer than the needle stiffness. The other parameters in this model consist of l_1 which is the distance between the two wheels of the bicycle, l_2 the distance between the back wheel and the needle tip frame, and two control input parameters corresponding to insertion velocity and rotation speed u_1 and u_2 respectively. The discretized formulation of this model is presented in equation 1. Fig.**??** demonstrates the parameters used in this model. with non-holonomic constraints to model the position and orientation of a bevel tip needle [7].

$$\begin{aligned} g_{ab}(k+1) &= g_{ab}(k)e^{(u_1(k)V_1+u_2(k)V_2)T} \\ n(k) &= R_{ab}(k)l_2e_3(u_1(k)V_1+u_2(k)V_2)T \end{aligned} \tag{1}$$

In this model V_1 and V_2 are the discritized insertion and rotation amounts. In this project we aim to use the discritized version of this model with constant parameters such l_2 and κ based on the findings in [7]. It is worth noting that both l_2 and κ have direct correlation with the steering angle ϕ.

B. Path Planning in 3D space

We chose RRT-based algorithm because it's computation speed provides the fast, on-the-go planning needed when new obstacles are discovered during planning, allowing for more ready path finding in dire situations. Additionally, it is able to account for the non-holonomic constraints of the robot, whereby the needle must follow a specific arced path based off of the path it took to reach its position.

To review, standard RRT is extremely fast, even in high dimensional space. It can quickly branch out and find a solution within a 3D space. However, it is truly random and therefore provides no optimality. Jagged paths are more difficult to smooth and less practical for operation. RRT*, on the other had is still fast, but is more optimal through it's rewiring function. While this costs time, it is not enough to outweigh the benefits of its optimality. Spline RRT* combines the benefits of RRT* on its own, with the ability to account for the constraints of the needle and generate a smooth path that is feasible for the needle to follow.

The Spline based RRT* algorithm smooths the raw path generated from a traditional RRT* algorithm by using the spline interpolation method. Spline interpolation method is common used to generate curves in the computer graphics for animation and game development. In addition, it is increasingly being used in the self-driving cars to generate a smooth curve. The spline that we have chosen in the cubic Beizer function. We did explore various other splines like Linear Beizer curve, Quadratic Beizer curve, but found the cubic Beizer function to be giving the best results for smoothing the path. By using the cubic Beizer curve, the path planne we have developed generates a curve path feasible to traverse using duty cycling method. The SmoothPath method applies the smoothing process to the raw path generated. The equation of quadratic Beizer equation is shown in 2 and that of a cubic Beizer equation is shown in 3.

$$Q(t) = (1-t)^2P_0 + 2(1-t)tP_1 + t^2P_3 \tag{2}$$

$$C(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2P_2 + tP_3 \tag{3}$$

Now, let us understand the finer details of our proposed method. This algorithm involves 2 phases. In the first phase, it generates a geometric trajectory at each node in the RRT tree by exploring the cartesian space using the RRT*-based planner. The planner gives a tree G which consists of vertices and edges. The path consisting of discrete x, y and z coordinates is generated using this tree which is jagged and cannot to traversed by using duty cycling method. In the second phase, we generate a smooth curve trajectory using Brezzier Function which generates a smooth curve trajectory between any two points. So, the jagged trajectory consisting of discrete points is converted to a curved trajectory and then is used in duty cycling method to generate motion plans encoding control inputs.

In addition to Spline-based RRT*, we compared our results with Spine-based RRT algorithm too. The algorithms were implemented using C++ and the results are shown in the Results section. The Fig.III-B shows the class diagram of our implementation at the lower level. The code is designed using standard object oriented principles for good modularity and ease of extending the functionality in future. The Environment class contains the information about the phantom dimensions and obstacles from MATLAB. The planners are designed to be sepearate classes with IPlanner as the interface. In the future, new planning algorithms can be implemented as child classes

Algorithm 1 Modified RRT*

$V \leftarrow \{x_{init}\}; E \leftarrow \phi$
for $i = 1, ..., n$ **do**
  $x_{rand} \leftarrow SampleFree_i;$
  $x_{nearest} \leftarrow Nearest(G = (V, E), x_{rand});$
  $x_{new} \leftarrow Steer(x_{nearest}, x_{rand});$
  if $ObstacleFree(x_{nearest}, x_{new})$ **then**
    $X_{near} \leftarrow Near(G = (V, E), x_{new}, min(\delta_{RRT*(log(card(V))/card(V))^(1/d), \eta});$
    $V \leftarrow V \cup x_{new};$
    $x_{min} \leftarrow x_{nearest}; c_{min} \leftarrow Cost(x_{nearest}) + c(Line(x_{nearest}, x_{new}));$
    for all $x_{near} \in X_{near}$ **do**
      if $Collision-Free(x_{near}, x_{new}) \wedge Cost(x_{near}) + c(Line(x_{nearest, x_{new}})) < c_{min}$ **then**
        $x_{min} \leftarrow x_{near}; c_{min} \leftarrow$
        $c_{min} \leftarrow Cost(x_{near} + c(Line(x_{near}, x_{new})))$
      end if
    end for
    $E \leftarrow E \cup (x_{min}, x_{new});$
    for all $x_{near} \in X_{near}$ **do**
      if $Collision-Free(x_{near}, x_{new}) \wedge Cost(x_{near}) + c(Line(x_{nearest, x_{new}})) < Cost(x_{near})$ **then**
        $x_{parent} \leftarrow Parent(x_{near});$
        $E \leftarrow (E \backslash (x_{parent}, x_{near}) \cup (x_{new}, x_{near})$
      end if
    end for
  end if
end for
$T = (V, E)$ = $SmoothPath$(G=(V,E))
return $T = (V, E)$

of IPlanner and the same existing Environment class can be used for testing with very minimal changes.

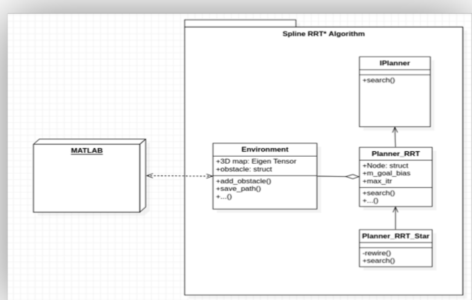

Fig. 3. Class diagram of implementation of spline based RRT algorithm.

C. Duty cycling

As previously discussed in sec III-A, kinematics of a bevel tip needle is non-holonomic, meaning that the needle is constrained from moving sideways, and it follows a circular arc along its insertion axis. This arc is denoted as the natural curvature of the needle and the radius corresponding to it is the smallest radius the needle can track. This inherently would limit the motion planning algorithm to effectively navigate the needle among the obstacles and would severely lower the maneuverability of the needle. Therefore, by relaxing the constant curvature constraint of the needle in a way that it allows the needle to assume variable curvatures between its natural curvature and zero curvature i.e, straight line, the planner algorithm is able to sample more points from the space and find better paths from the start to target point. Several methods have been proposed which enable bevel tip needles to follow variable curvature paths, namely the duty-cycling method as originally described in [11] and its modified variants as presented in [12], and more recently CURV steering method as described in [13] which enables variable curvature profiles by continuously changing the rotational speed of the needle. In this work we have implemented the duty-cycling approach to achieve variable curvature path profile.

During duty-cycled spinning, the needle path varies between a maximally curved path (resulting from pure needle insertion) and a minimally curved path (resulting from needle insertion with continuous axial rotation, ideally zero curvature). In duty-cycled spinning, a full needle rotation must be achieved at each duty cycle step to prevent planar deviation of the needle during insertion. Additionally, the rotation (spin) velocity must be much greater than the insertion velocity ($w(t) >> v(t)$) to prevent helical needle trajectories, though it is possible to exploit these trajectories for needle control [14]. The formulation for the duty-cycling is provided below where t is the current time, j is discrete time intervals, τ is desired duty cycle, T is time period, and is the time increment.

$$v(t) = v_{fixed} \equiv 1, 0 < t \leq T_i \tag{4}$$

$$\omega(t) = \begin{cases} \theta, & if \quad t = 0 \\ 2k\pi/\tau\Delta, & if \quad j\Delta \leq t \leq \Delta(j+\tau) \\ 0, & if \quad \Delta(j+\tau) < t < (j+1)\Delta \end{cases} \tag{5}$$

In our implementation the duty-cycling is used to generate kinematically valid motion profiles between each waypoints provided by the motion planning algorithm. Therefore, the descritized path found by the motion planning algorithm is used as an input to the trajectory generation algorithm which in turn uses the duty cycling method to steer the robot between each consecutive points in the planned path. The final generated trajectory encodes the path from the start to the target point in terms of desired rotational and insertion velocity for the needle. Fig.**??** demonstrates a sample motion profile for a desired curvature path.

Therefore, based on the requirement of faster computation speed and nonholonomic constraints of the steerable needle, it was concluded from our study that the above proposed w5w5

Fig. 4. Three sample motion profiles for zeros curvature i.e., 100% duty-cycle (top), 0.67% duty-cycle (middle), and maximum curvature with 0% duty-cycle (bottom). In the zero curvature scenario the needle will continuously rotate with a maximum rotational velocity and would follow a straight helical path. In the maximum curvature, the needle maintains zero rotational velocity throughout the whole insertion resulting in the natural curvature. In the 67% duty-cycle, the rotational speed alternates between zero and 50 rad/sec.

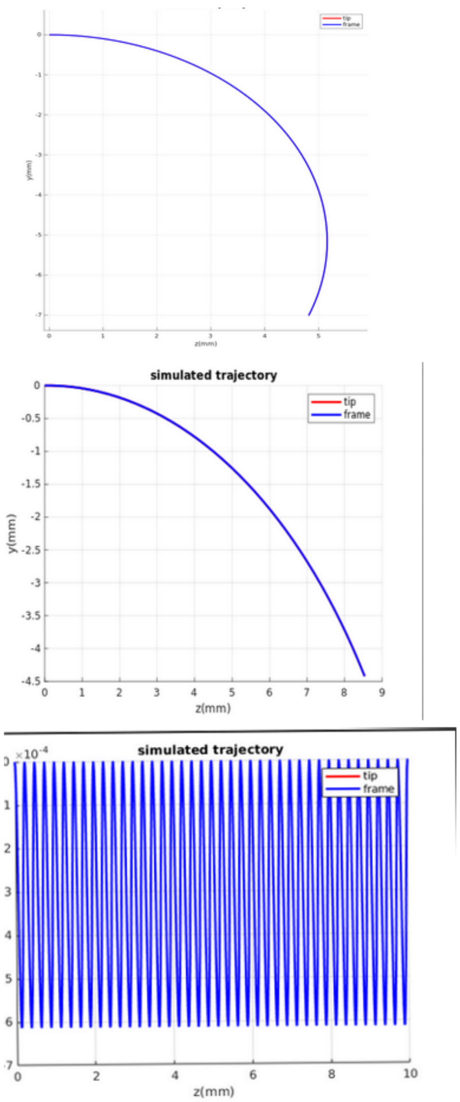

Fig. 5. Simulated needle tip position using duty-cycle method in yz plane. Top image shows the duty-cycle of zero, middle image is the duty cycle of 67% and the bottom figure shows the duty-cycle of 100%.

modified RRT-based planner is one of the best solutions for the critical medical applications involving needle-steering.

IV. Results

A. Kinematic Modelling

We have implemented a discrete model of the bicycle model in Matlab. Some of the primary reasons for choosing Matlab platform for our preliminary studies is because Matlab helps in rapid prototyping of our models under out study. Additionally, it provides excellent data visualization tools to test and verify our understanding of the model and replicate the results in the paper. Moreover, it provides good data manipulation tools which is crucial during the prototyping stage to work with matrices. If time permits, our stretched goal is to implement the algorithm on the AIM lab's prostate robot and run some phantom test to verify our work.

To test the accuracy of this model, we have replicated the tests described in [7] and cross checked our simulation results with their results. The simulation results show similar results which point out that our implemented model works similar to the proposed model in the original article. Below is the simulation result for a double-bend needle insertion compared to the experimental result for the same insertion in [7] which shows the same deflection profile. In Fig.IV-B the needle is first inserted into the tissue with zero rotational velocity for one third of the way, then rotated 180° with zero insertion velocity and then inserted again for the rest of the way with zero rotational velocity.

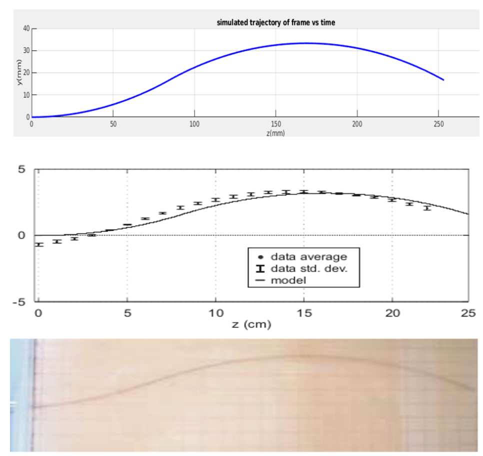

Fig. 6. Top is the simulated bicycle model as implemented in our our work. Middle and bottom images are the simulated and experimental results as seen in [7]. As can be seen, our kinematic model is able to replicate the experimental results accurately.

B. System Integration

We have developed from scratch a RRT-based variant called Spline-based RRT* algorithm for 3D space in C++. At a high level, our work is interfaced between MATLAB and C++. We used MATLAB as it offers easy and powerful simulation tools. We used C++ as it is efficient and more portable to the robot in the AIM lab for future work. To begin, an environment space similar to that of a physical phantom that is typically used for experiments in a real setting is set up in MATLAB, the size of which is communicated to C++. From there, the paths are planned using our RRT and RRT* algorithm implementations. The resulting path data is sent back to MATLAB so that the trajectory of the needle can be visualized. the trajectory planning is carried out for the generated way points using the bicycle kinematic model and duty cycling method on MATLAB. In addition, CURV steering method was also explored.

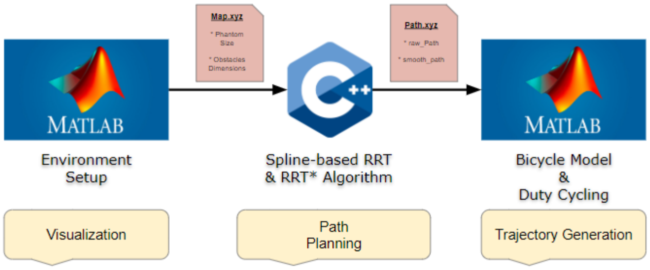

Fig. 7. System integration. Our system integrates functionality in both MATLAB and C++. These platforms were chosen as MATLAB offers powerful simulation and calculation tools and C++ is efficient and more portable to the robot in the AIM lab. To begin, an environment space is set up in matlab, the size of which is communicated to C++. From there, Paths are planned using our RRT and RRT* algorithm implementations. The resulting path data is sent back to MATLAB so it can: Generate smooth, reachable paths using the bicycle kinematic model and duty cycling and Generate a simulation of the needle following the path and avoiding obstacles.

C. Environment setup

The environment is set up to imitate an insertion area with spherical obstacles, like those found in related work. These obstacles are created using a radius and center coordinate within the generated insertion area. This area is comprised of a phantom space, which provides a region to be traversed by the RRT algorithms and the needle itself. To create reliable obstacle avoidance, bounding boxes were created around the obstacles. These were given the same discretized center as the obstacle itself, but better fit the abilities of the RRT algorithms. The bouncing box allowed for a grid check to be performed for collision avoidance ensuring that the needle would not collide with critical body tissue. It is also worth to point out that we have control over the size of the bounding box we want to wrap the obstacle with.

D. Path planning using Spline based RRT and RRT*

The algorithm has been explained in detail in the "Methodology" section. Both the algorithm was implemented on C++. We simulated 4 different cases each increasing in the complexity of the environment by adding more obstacles and placing target point at difficult points. We generated results for all the test cases with both Spline-based RRT and RRT* algorithms. The results obtained are shown below with illustration of each case in their caption. Table 1 compares the results between the Spline-based RRT and Spline-based RRT* algorithm and it was observed that Spline-based RRT is faster compared to Spline-based RRT* but there were cases where Spline-based RRT cannot generate an optimal path with non-holonomic constraints as collision with obstacles were observed. In contrast, Spline-based RRT* took more time to generate path but always gave a feasible solution no obstacle collision. The more time is required in Spline-based RRT* is because it optimizes the path generated and thus gives an optimal and complete path. This is trade off we have to consider but is acceptable as its better to have an optimal path taking even though it takes some more time to achieve. Thus, we concluded that Spline-based RRT* is better choice for our project.

Spline based RRT in 3D		Spline based RRT* in 3D	
Time takes (sec)	Iterations	Time taken (sec)	Iterations
0.1	11	7	7000
0.3	333	7	7000
0.8	817	6	7000
0.7	634	6	7000

TABLE I
COMPARISON BETWEEN RRT AND RRT*

Sampling-based Planning Algorithms	Speed	Optimality	Smooth Paths
RRT	Very Fast	No	No
RRT*	Fast	Yes	No
Spline-based RRT*	Fast	Yes	Yes

Fig. 8. Comparison between RRT, RRT* and spline based RRT*.

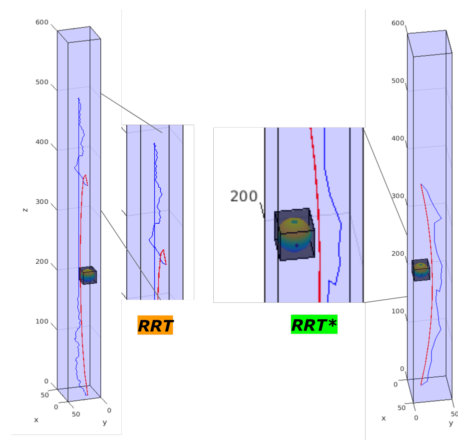

Fig. 9. Path simulation using RRT (left) and RRT* (right) with one obstacle.

V. Conclusion and Future Work

Through the completion of this study, we have shown that it is possible to control a surgical needle robot through motion planning, by creating our own environment and algorithms and replicating various previous methods. The bicycle model successfully accounted for the non-holonomic constraints and represented the kinematics of the robot. Duty cycling allowed for the robot to be steered properly. Both 2D and 3D environments were generated to test the robot's performance. The custom Spline-based RRT* algorithm was developed to generate optimized and smooth paths quickly. All of these capabilities helped to create a process that can help push the study and improvement of prostate biopsy proceedures forward, especially at WPI. Unfortunately, this project did

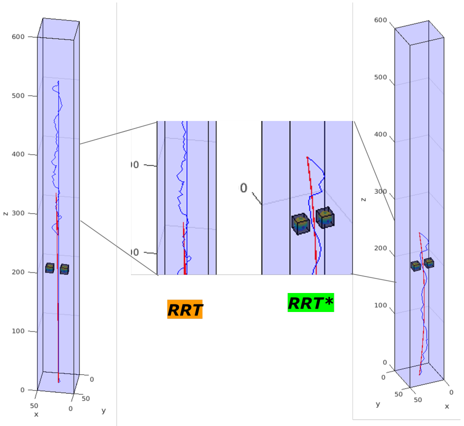

Fig. 10. Path simulation using RRT (left) and RRT* (right) with two obstacles with a narrow passage between them.

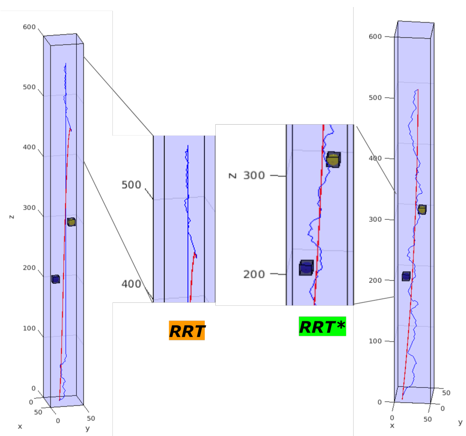

Fig. 11. Path simulation using RRT (left) and RRT* (right) with two obstacles at different height and position.

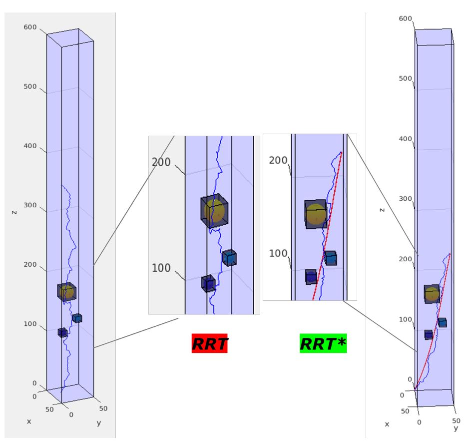

Fig. 12. Path simulation using RRT (left) and RRT* (right) with three obstacles of varying size, placed at different positions with multiple narrow passages.

have some limitations. It was limited to simulated results and was not able to be implemented on the real robot, which is to be desired. Additionally, we use reachability sampling to

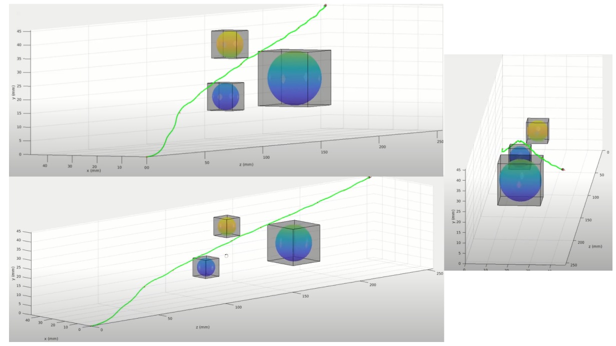

Fig. 13. Successful path traversal in 3D space with multiple obstacles shown from different viewpoints.

handle more challenging cases. This leaves open a few avenues for future research, being Extending our C++ implementation into a multi-threaded application for RRT* algorithm, running physical tests including phantom experiment using AIM lab's prostate robot, and implement a feedback system for real time tracking in MRI or Ultrasound. Overall, this project was intended to prove that advanced motion planning techniques can be applied to beneficial fields, like that of prostate biopsy research. In completing our desired goals, this was achieved.

References

[1] S. M. Neubach Z, "Ultrasound-guided robot for flexible needle steering," *IEEE Trans Biomed Eng*, 2010.

[2] M. Bernardes, B. V. Adorno, P. Poignet, and G. Borges, "Robot-assisted automatic insertion of steerable needles with closed-loop imaging feedback and intraoperative trajectory replanning," *Mechatronics*, vol. 23, no. 6, pp. 630–645, 2013.

[3] G. J. Vrooijink, M. Abayazid, S. Patil, R. Alterovitz, and S. Misra, "Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images," *The International journal of robotics research*, vol. 33, no. 10, pp. 1361–1374, 2014.

[4] K. V. A. R. G. K. C. N. Reed KB, Majewicz A, "Robot-assisted needle steering," *IEEE Automat Magazine*, 2011.

[5] S. Patil and R. Alterovitz, "Interactive motion planning for steerable needles in 3d environments with obstacles," in *2010 3rd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2010, pp. 893–899.

[6] O. M. Okamura A, Simone C, "Force modeling for needle insertion into soft tissue." *IEEE Trans Biomed Engineering*, 2004.

[7] I. Robert J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006. [Online]. Available: https://doi.org/10.1177/0278364906065388

[8] K. A. . H. Z. Noreen, I., "Optimal path planning using rrt* based approaches: a survey and future directions," *Int. J. Adv. Comput. Sci. Appl*, 2016.

[9] C. E. . K. E. Dong, Y., "Faster rrt-based nonholonomic path planning in 2d building environments using skeleton-constrained path biasing," *Journal of Intelligent Robotic Systems*, 2018.

[10] D. S. Minhas, J. A. Engh, M. M. Fenske, and C. N. Riviere, "Modeling of needle steering via duty-cycled spinning," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007, pp. 2756–2759.

[11] ——, "Modeling of needle steering via duty-cycled spinning," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2007, pp. 2756–2759.

[12] A. Majewicz, J. J. Siegel, A. A. Stanley, and A. M. Okamura, "Design and evaluation of duty-cycling steering algorithms for robotically-driven steerable needles," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 5883–5888.

[13] G. Li, “Robotic system development for precision mri-guided needle-based interventions,” *PhD Dissertation, Dept, of Mechanical Engineering*, 2016.

[14] K. Hauser, R. Alterovitz, N. Chentanez, A. Okamura, and K. Goldberg, “Feedback control for steering needles through 3d deformable tissue using helical paths,” *Robotics science and systems: online proceedings*, p. 37, 2009.