Anatomical virtual fixture assisted mastoidectomy

I. Basic Info (->Summary)

- Student Name: Yiping Zheng
- Mentor: Max Li, Dr. Taylor
- Size group: 1
- Skills: CISST code library, C++, motion planning, nonlinear control

- Goal: Create a motion planning demo for Galen robot to perform robot-assisted mastoidectomy task

II. Background, Specific Aims, and Significance

Mastoidecotmy is a deliberate surgical procedure which is important to the treatment of diseases such as cochlear implant, acoustic neuroma etc. Surgeons have to mechanically drill a hole on patient's skull and all the way down to the meningeal, meanwhile carefully avoiding sensitive anatomy structure such as facial nurve, sigmoid sinus, and arteries. The drilling process is a challenging process to surgeons which often lasts 8 hours.

Surgical robots such as da Vinci Surgical System can mitigate the challenges by extending human capabilities. However, mainly because of the precision requirement of mastoidectomy is very high, so far there hasn't been any application or attempt of robot assisted mastoidectomy to our best knowledge.

Virtual fixture is software motion constraints, can further reduce the operational difficulties by allowing the surgeon and robot to work together to complete the surgical task with improved stability, reliability and precision. By tracking the relative position of the surgical tool with regard to patients body, it can stop the surgeon from making sudden motions and accidentally damaging critical anatomies and have the potentiality to make the robot assisted mastoidectomy possible.

Recently, a new virtual fixture generation algorithm was proposed which fits the scenario of mastoidectomy very well. By obtaining the 3D data of patient's skull, either pre-operatively via CT scan or intra-operatively via 3D ultrasound, it can generate virtual fixtures online from polygon mesh representations of complex anatomical structures and provide dynamic constraint formulation for the planning algorithm. The algorithm has been testified through validation and runtime experiments.

In this project, I'm going to integrate this anatomical virtual fixture generation algorithm with Galen surgical robot and perform a demo of robot-assisted mastoidectomy, which may be very useful to avoid touching patient's critical anatomy structure and mitigate surgeons' tension in the procedure.

III. Technical Approach

A. Constraint Optimization

Constraint optimization approaches are well-established methods to implement virtual fixtures. In this project, we formulate robot kinematic motion control as a quadratic optimization problem with linear constraints. Objective function solved for the desired motion:

$$\underset{\Delta \boldsymbol{q}}{\arg\min} \|\Delta \boldsymbol{x} - \Delta \boldsymbol{x}_d\|_2$$

$$\textit{subject to} \quad \boldsymbol{A}\Delta x \geq \boldsymbol{b}, \quad \Delta \boldsymbol{x} = \boldsymbol{J}\Delta \boldsymbol{q}$$

where $\Delta \boldsymbol{x}$ and $\Delta \boldsymbol{x}_d$ are the computed and desired incremental Cartesian positions, and $\Delta \boldsymbol{q}$ is the incremental joint position. $\boldsymbol{J}$ is the Jacobian matrix relating the joint space to the Cartesian space. $\boldsymbol{A}$ and $\boldsymbol{b}$ are matrix and vectors necessary to describe the linear constraints.

In a telemanipulated control environment, $\Delta \boldsymbol{x}$ in the objective function is computed from the motion of the master manipulator. Additional objectives may be added to express additional desired behaviors. Otherwise, in cooperative control setting, $\Delta \boldsymbol{x}$ is computed from the force sensor input.

Inequality constraints can be used to impose motion constraints. For example, a virtual forbidden wall for tool tip x can be defined by a hyper-plane with normal n and point p , i.e. tool tip can only move on the positive side of the hyperplane as shown in **Fig. 1**. This can be achieved by forcing the signed distance d_t from the tool tip to plane at any time t to be positive, i.e.,

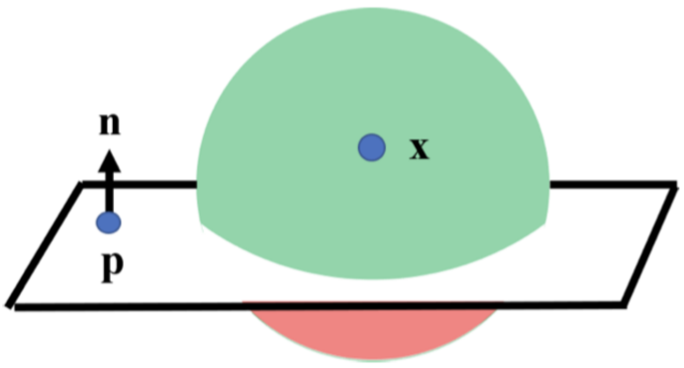

Fig. 1: Illustration of plane constraint. The radius of the sphere is the maximum motion capable of the robot in one control iteration. Green zones denote the allowable region and red zones denote the forbidden region.

$$\begin{aligned} d_{t-1} &= \boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{p}) \\ \Delta d &= \boldsymbol{n}^T \Delta \boldsymbol{x} \\ d_t &= d_{t-1} + \Delta d \geq 0 \\ \boldsymbol{n}^T \Delta \boldsymbol{x} &\geq -\boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{p}) \end{aligned}$$

where Δd is the change of the signed distance. The constrained motion can be realized by setting $\boldsymbol{A} = \boldsymbol{n}$ and $\boldsymbol{b} = -\boldsymbol{n}^T(\boldsymbol{x} - \boldsymbol{p})$. The constrained least-squares problem may then be solved to produce the desired motion. Additional terms may be added to further constrain the tool motion.

B. Polygon Mesh

In this work, polygon meshes consisting of triangles are used to represent anatomical surfaces. A locally concave surface (Fig. 2a) produces a convex set of linear constraints. It is safe to include all triangles as plane constraints (Fig. 2b). A locally convex surface (Fig. 2c) produces a non-convex set of linear constraints. Naively adding all triangles as plane constraints will rule out many allowable regions (Fig. 2d). This necessitates an approach to dynamically activate and deactivate the constraints based on the local convexity and concavity of the anatomical surfaces.

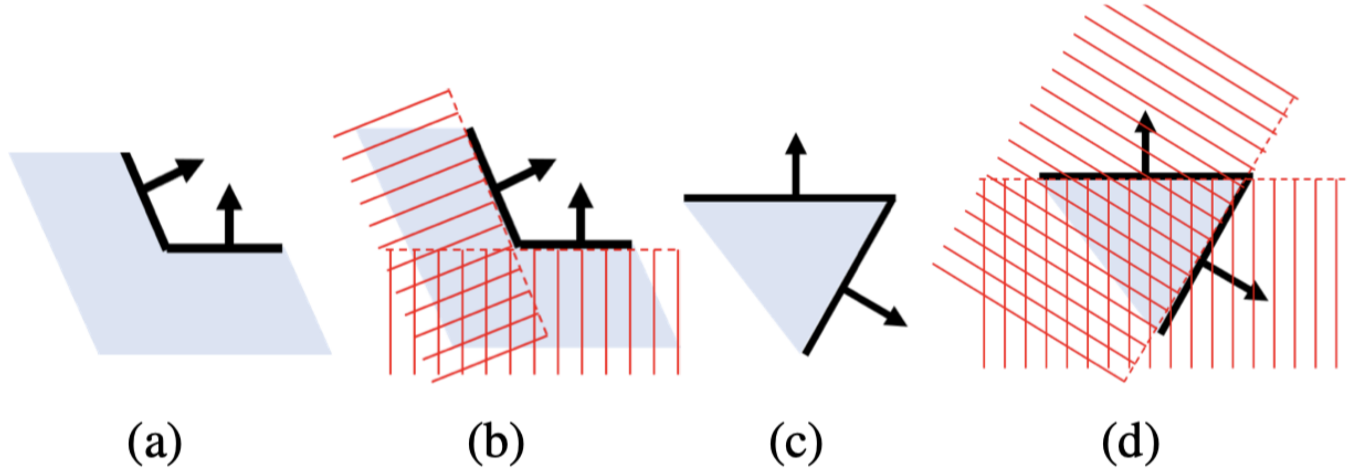

Fig. 2: 2D illustration of (a-b) a locally **concave surface**, and (c-d) a locally **convex surface**. Red crosshatched zones denote forbidden region. Note in (d) many allowable regions are ruled out. Arrows denote the face normals. Light blue zones denote the patient's anatomy.

C.Virtual Fixture for Polygon Mesh

We use the Polygon Mesh Constraint Algorithm developed in [1] to generate proper anatomical constaints for the motion planner. And we use constraint optimization algorithm to command the robot to stop in close proximity to the patient's tissue.

The algorithm is described as follows,

Algorithm 1: Polygon Mesh Constraint

```
Input:  PD-Tree, Current Position $x$
Result: List of Active Plane Constraints $\mathcal{L}$
Find intersected triangles $\mathcal{T}$, corresponding closest
  points $\mathcal{CP}$ and face normals $\mathcal{N}$ ;
for triangle $\mathcal{T}_i \in \mathcal{T}$ do
    if $\mathcal{CP}_i$ in-triangle & $\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$ then
        add $\{\mathcal{N}_i, \mathcal{CP}_i\}$ to $\mathcal{L}$ ;
    else if $\mathcal{CP}_i$ on-edge then
        Find adjacent triangle(s) $\mathcal{T}_{i,a}$ ;
        if $\mathcal{CP}_i == \mathcal{CP}_{i,a}$ & locally convex then
            add $\{x - \mathcal{CP}_i, \mathcal{CP}_i\}$ to $\mathcal{L}$ ;
        else if $\mathcal{N}_i^T(x - \mathcal{CP}_i) \geq 0$ & locally concave
          then
            add $\{\mathcal{N}_i, \mathcal{CP}_i\}$ to $\mathcal{L}$ ;
end
```

IV. Steps and Milestones

Below two versions of steps and milestones are kept. One original version, representing the idealistic plan. One revised version, representing the compromised plan due to 2019-nCov situation.

A. Original Version

Step 1: Getting started

- Write proposal, give presentation and construct the wiki page.
- Get familiar with the CISST code base (focused on numerical environment)

  Milestone: Complete the task of loading mesh stl binary file and add it to the library. (ASCII file version is already there)

Date: Mar. 5 ~ Mar. 15 (10 days)
Status: 30%

Step 2: Integrate Mesh Constraint to Galen Robot

- Learn the usage of Galen Robot.
  Milestone: Test the existing code examples on Galen Robot.
  Date: Mar.16 ~ Apr. 5 (20 days):
  Status: 0%

Step 3: Test the constraint formulation

- Integrate the constraint formulation with simple geometry obstacle
- Integrate the constraint formulation with 3D phantom of patient anatomy.
- Complete the user study (if possible)
  Date: Apr. 6~ May 1 (25 days)
  Milestone: a video to demonstrate the Galen robot can actually perform the above tasks.
  Status: 0%

B. Revised Version

The focus has shifted from creating the real-world demo to strengthening the robustness of the mesh constraint algorithm and creating a demo in simulation.

Step 1: Getting started

- Write proposal, give presentation and construct the wiki page.
- Get familiar with the CISST code base (focused on numerical environment, and the Constraint Controller part)
  Milestone: Complete the task of loading mesh stl binary file and add it to the library.
  Date: Mar. 5 ~ Mar. 19 (14 days)
  Status: 90%

Step 2: Integrate Mesh Constraint Formulation to Galen Robot Controller Code

- Get familiar with the Galen Robot code base
- Adding slack variable of soft constraints to the optimal controller. (This is also the minimum deliverable)
  Milestone: Pass compilation of the integrated Galen Robot cotroller.

Date: Mar.20 ~ Apr. 5 (14 days):
Status: 0%

Step 3: Test the integrated controller in simulation, using simple robot model

- Test the controller with a simple robot model (eg. UR5 robot arm)
- Test the controller with 3-Dof translation-only motion.
- Test the controller with simple geometry obstacle.
- Test the controller with 3D phantom of patient anatomy. (online data source)
- Improving the mesh-constraint algorithm with respect to the whole surface (use sphere model) of the end-effector, rather than modeling it with a point. (This is also the expected deliverable)
- Test the controller with 6-Dof motion.
  Date: Apr. 6 ~ Apr. 20 (14 days)
  Milestone: a video to demonstrate a simple robot can actually perform the above tasks in simulation.
  Status: 0%

Step 4: Test the integrated controller in simulation, using Galen Robot model

- Integrating the tested controller into the Galen Robot code base.
- Test all the above tasks with respect to the Galen Robot's forward kinematics and Jacobian matrices.
  Date: Apr. 21 ~ Apr. 30 (10 days)
  Milestone: a video to demonstrate the Galen robot can actually perform the above tasks in simulation.
  Status: 0%

V. Deliverables

Likewise, here we kept 2 versions of deliveralbes, idealistically original plan A and relistically revised plan B.

A. Original Plan

- **Minimum**: Simple geometry code integration
- **Expected**: Patient anatomy code integration(3D phantom)
- **Maximum**: Complete the user study.

B. Revised Plan under 2019-nCov

Since the laboratory access is not achievable any more, the goal for the CIS2 project has been shifted to simulation environment accordingly, and improving its robustness, rather than creating a demo in real world.

- **Minimum**: Get slack formulation integrated into the optimal controller.
- **Expected**: Perform the surgery task simulation with a simple robot model, with respect to the whole surface of the end-effector.
- **Maximum**: Perform the surgery task simulation with the Galen robot model, with respect to the complexity of its forward kinematics and Jacobian.

Dependencies

	Contact	Alternatives	Status	Notes
Galen Robot Access	Max	None	Not possible for now	Hardware
A computer	My PC	Dr. Taylor	PlanA succeeded	Hardware
Code Base Access	Max	None	PlanA succeeded	Software
Preoperative CT scan model	Pete	CT scan model from Internet	PlanB succeeded	Testing Data (facial nerve, artery)
Separate Code repo	Max	None	Pending	Visualization tools (like rViz)

Project Bibliography

- Zhaoshuo Li et al. Anatomical Mesh-Based Virtual Fixtures for Surgical Robots (unpublished by Mar. 2020)
- Funda, J., Taylor, R. H., Eldridge, B., Gomory, S., & Gruben, K. G. (1996). Constrained Cartesian Motion Control for Teleoperated Surgical Robots. Robotics, 12(3).
- Xia, T., Kapoor, A., Kazanzides, P., & Taylor, R. (2011). A constrained optimization approach to virtual fixtures for multi-robot collaborative teleoperation. IEEE International Conference on Intelligent Robots and Systems, 639–644. https://doi.org/10.1109/IROS.2011.6048816

- Li, M., Ishii, M., & Taylor, R. H. (2007). Spatial Motion Constraints Using Virtual Fixtures Generated by Anatomy. 23(1), 4–19.
- Kapoor, A. (2008). Motion constrained control of robots for dexterous surgical tasks.

Reports and Presentations

(under costruction)

Log

(under costruction)

Other Resources and Project Files

(under costruction)