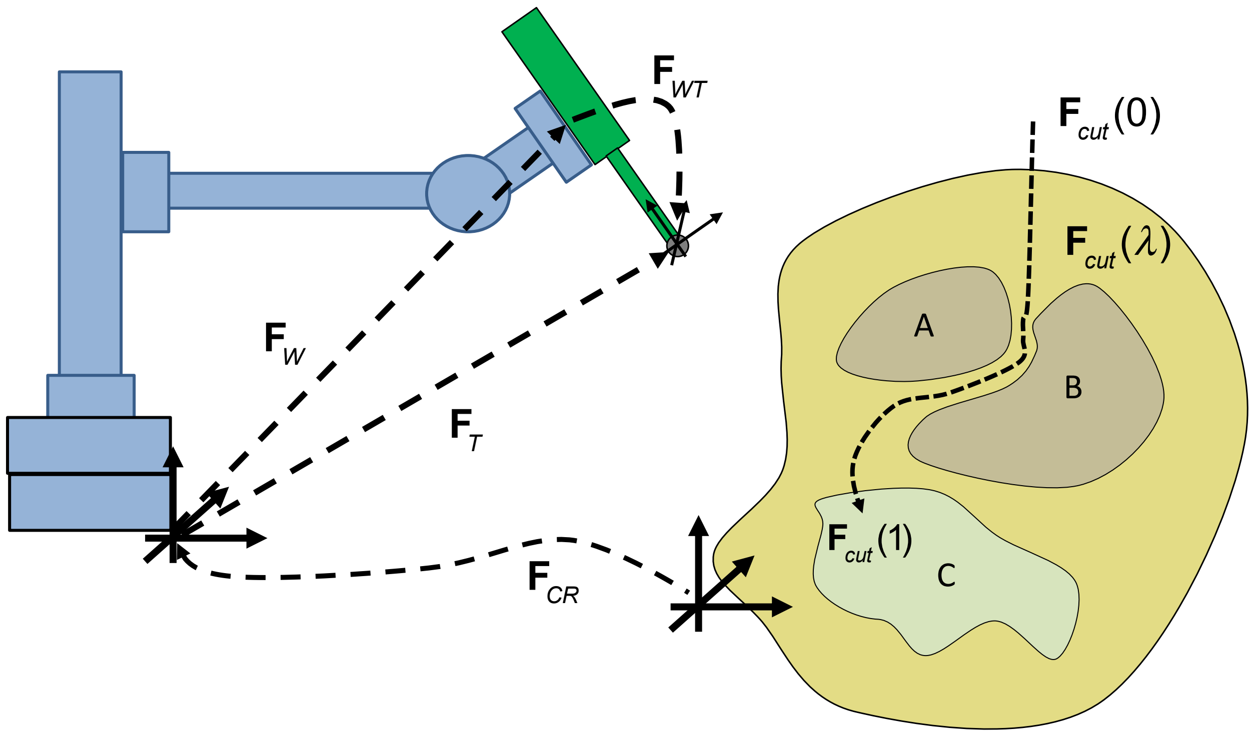
Homework Assignment 4 – 600.455/655 Fall 2024

Name	Name
Email	Email
Other contact information (optional)	Other contact information (optional)
Signature (required) I/We have followed the rules in completing this assignment ______________________________	Signature (required) I/We have followed the rules in completing this assignment ______________________________

1. Remember that this is a graded homework assignment. It is the functional equivalent of a take-home exam.
2. You are to work **alone** or in **teams of two** and are not to discuss the problems with anyone other than the TAs or the instructor.
3. **IMPORTANT NOTE:** If you work in teams of two, you are **not** to split up the questions and each answer a subset individually. You are to work **together**. I encourage teaming on these problems because I believe that it encourages learning, not as a way to reduce the required work for students taking the course. By signing this sheet you are asserting that each of you has contributed equally to each answer and can individually explain the answer as well as if you had answered the question alone. I view this as a question of trust and ethics.
4. It is otherwise open book, notes, and web. But you should cite any references you consult.
5. Please refer to the course organizational notes for a fuller listing of all the rules. I am not reciting them all here, but they are still in effect.
6. Unless I say otherwise in class, it is due before the start of class on the due date posted on the web.
7. Submit the assignment on GradeScope as a neat and legible PDF file. We will not insist on typesetting your answers, but we must be able to read them. We will not go to extraordinary lengths to decipher what you write. If the graders cannot make out an answer, the score will be 0.
8. Sign and hand in this page as the first sheet of your assignment. If you work with a partner, then you both should sign the sheet, but you should only submit one PDF file for both of you, using the GradeScope teaming feature. Indicate clearly who it is from.

This assignment has more than 100 points, but the most that will be applied to your grade is 100

Scenario: Robot Virtual Fixtures



Consider the situation shown in the figure above, in which a robot holds a rotating surgical cutter. The position of the robot's wrist is given by $\mathbf{F}_W(\vec{\mathbf{q}}) = [\mathbf{R}_W(\vec{\mathbf{q}}), \vec{\mathbf{p}}_W(\vec{\mathbf{q}})]$, where $\vec{\mathbf{q}}$ gives the positions of the robot's joints. The effect of a small incremental joint motion is $\mathbf{F}_W(\vec{\mathbf{q}} + \Delta\mathbf{q}) = \mathbf{F}_W(\vec{\mathbf{q}})\Delta\mathbf{F}_W(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}})$ where ,, $\Delta\mathbf{F}_W(\vec{\mathbf{q}}, \Delta\vec{\mathbf{q}}) \approx \Delta\mathbf{F}_W(\vec{\eta}_W) = [\mathbf{I} + sk(\vec{\alpha}_W), \vec{\varepsilon}_w]$ where $\vec{\eta}_W = [\vec{\alpha}_W, \vec{\varepsilon}_w]^T = \mathbf{J}_W(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$.

The cutter tool has a small spherical cutting ball of radius ρ located at the origin of the cutter coordinate system, and the tool shaft is aligned with the z-axis of the cutter coordinate system and emerges from the tool body somewhere along the z-axis, as shown in the figure. The cutter is mounted to the robot rigidly at a transformation $\mathbf{F}_{WT}$, such that its pose in robot coordinates is given by $\mathbf{F}_T(\vec{\mathbf{q}}) = \mathbf{F}_W(\vec{\mathbf{q}})\mathbf{F}_{WT}$.

The robot is equipped with a force sensor that is capable of measuring forces and torques $\vec{\zeta}_W = [\vec{\phi}_W^T, \vec{\tau}_W^T]^T$ in the coordinate system of $\mathbf{F}_W$. **NOTE:** Force-torque vectors like $\vec{\zeta}_W$ are often referred to as "wrenches".

The midlevel controller for the robot executes the following algorithm every Δt seconds:

Step 1: Read the robot state $[\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, \vec{\zeta}_W]$ and perform basic safety checks. Output the previous velocity command $\dot{\vec{\mathbf{q}}}_{cmd}(t)$. Compute $\mathbf{J}_W(\vec{\mathbf{q}})$.

Step 2: $t \leftarrow t + \Delta t$; Define $S(t) = [\vec{\mathbf{q}}, \dot{\vec{\mathbf{q}}}, \mathbf{F}_W, \mathbf{J}_W, \vec{\zeta}_W]$

Step 3: Formulate and solve an optimization problem

$$\Delta\vec{\mathbf{q}}_{cmd} = \arg\min \sum_i w_i E_i(S(t), \Delta\vec{\mathbf{q}}_{cmd}, \vec{\eta}_{cmd}, \mathbf{G}_i, \cdots, \kappa_{i,j}, \cdots)^2$$

Subject to $\mathbf{A}_q \Delta\vec{\mathbf{q}}_{cmd} \le \vec{\mathbf{b}}_q$ and other equality and inequality constraints

where $\mathbf{G}_i, \kappa_{i,j}$ are data and parameters associated with the problem and may vary with $S(t)$ and the w_i are weights associated with how important each term of the objective function is to the desired behavior. If only one term is present, then $w_1 = 1$. It may also be convenient to compute a Jacobean $\mathbf{J}_T(\vec{\mathbf{q}})$ and add equality constraints such as $\vec{\eta}_T^{cmd} = \mathbf{J}_T(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}_{cmd}$ and inequality constraints expressed in terms of $\vec{\eta}_{WT}^{cmd}$. Also, although the form of the objective function terms $E_i(\cdots)$ described above is very general, they typically take a form that looks like

$$E_i(\cdots)^2 = \left\| \vec{\eta}_{something}^{desired} - \mathbf{J}_{something}(\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}_{cmd} \right\|^2$$

where "something" represents some coordinate system and $\vec{\eta}_{something}^{desired}$ is some expression representing a desired incremental motion.

Step 4: $\dot{\vec{\mathbf{q}}}_{cmd} \leftarrow \Delta\vec{\mathbf{q}}_{cmd} / \Delta t$

Step 5: Go to sleep until the next time interval.

The mechanical design of the robot has the following limits: $\vec{\mathbf{q}}_{\min} \le \vec{\mathbf{q}} \le \vec{\mathbf{q}}_{\max}$, $\left|\dot{\vec{\mathbf{q}}}\right| \le \dot{\vec{\mathbf{q}}}_{\max}$, $\left|\ddot{\vec{\mathbf{q}}}\right| \le \ddot{\vec{\mathbf{q}}}_{\max}$, where $\dot{\vec{\mathbf{q}}}$ and $\ddot{\vec{\mathbf{q}}}$ are the joint velocity and acceleration, respectively.

Presurgical planning has been done based on a segmented CT image, and a registration step has been performed such that a position $\vec{\mathbf{p}}_{CT}$ in CT coordinates corresponds to a position $\mathbf{F}_{CR}\vec{\mathbf{p}}_{CT}$ in robot coordinates. Three anatomic structures, labeled as A, B, C have been identified and a desired cutter path $\mathbf{F}_{cut}(\lambda) = [\mathbf{R}_{cut}(\lambda), \vec{\mathbf{p}}_{cut}(\lambda)]$ has been defined in CT coordinates, with $\mathbf{F}_{cut}(0)$ representing an approach pose outside the patient's body and $\mathbf{F}_{cut}(1)$ representing an entry pose to anatomic structure C. Further, the planning software has conveniently computed the parameters $\left[\vec{\mathbf{r}}(\lambda), \vec{\mathbf{n}}(\lambda)\right]$ such that $\Delta\mathbf{F}(\lambda, \Delta\lambda) = \mathbf{F}(\lambda + \Delta\lambda)\Delta\mathbf{F}(\lambda)^{-1} \approx \left[\mathbf{I} + sk(\vec{\mathbf{r}}(\lambda)\Delta\lambda), \vec{\mathbf{n}}(\lambda)\Delta\lambda\right]$.

A function $\lambda \leftarrow FindLambda(\vec{\mathbf{p}})$ is available to compute the value of λ corresponding to the closest point $\vec{\mathbf{p}}_{cut}(\lambda)$ on the cutter path to any point $\vec{\mathbf{p}}$. You can the planning software has software that can return properties associated with every 3D voxel $\vec{\mathbf{c}}$ in the CT image. These properties can be retrieved via a function $propvalue \leftarrow GetMap(\vec{\mathbf{c}}, "name")$. The *Get* function is capable of interpolating values appropriately if $\vec{\mathbf{c}}$ is a real vector. An additional function $PutMap(\vec{\mathbf{c}}, "name", value)$ is available to store additional properties for subsequent retrieval. The following properties have been computed and stored by the planning software:

- *SDF_A, SDF_B, SDF_C* – The signed distance field values for anatomic structures A, B, and C
- *Grad_A, Grad_B, Grad_C* – The SDF gradients for anatomic structures A, B, C

The software also includes a function $\vec{\mathbf{g}} \leftarrow GetGrad(\vec{\mathbf{c}}, SDF)$ for computing the gradient at a point $\vec{\mathbf{c}}$ in a signed distance field.

Questions

NOTE: For the purposes of this assignment, you may assume that $\mathbf{F}_{CR} = \mathbf{I}$ for all questions. Also you can assume that $\mathbf{R}_{cut}(\lambda)$ changes only slowly for $0 \le \lambda \le 1$, i.e., throughout the path for Questions 6-11. Also, you may find it useful to consult the notes on the admittance-style SDF-based virtual fixtures discussed in class, which are relevant for Questions 9-12.

1. (5 points) Explain why it is desirable to output $\vec{\mathbf{q}}_{cmd}(t)$ in Step 1 of the midlevel control loop rather than in Step 4.
2. (10 points) Describe constraints to be added to Step 3 of the midlevel control loop to guarantee that the joint position, velocity, and acceleration constraints will not be violated.
3. (10 points) Give an expression for $\vec{\eta}_T(\vec{\mathbf{q}},\Delta\mathbf{q}) = [\vec{\alpha}_T, \vec{\varepsilon}_T]$ and a corresponding Jacobean $\mathbf{J}_T(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})$ in terms of $\vec{\eta}_W$ and other quantities in the problem scenario, such that $\mathbf{F}_T(\vec{\mathbf{q}}+\Delta\vec{\mathbf{q}}) = \mathbf{F}(\vec{\mathbf{q}})\Delta\mathbf{F}_T(\vec{\eta}_T)$ and $\vec{\eta}_T \approx \mathbf{J}_T(\vec{\mathbf{q}},\Delta\vec{\mathbf{q}})\Delta\vec{\mathbf{q}}$. **Strong Hint (you will need this later on):** Find the formula for $\mathbf{J}_{WT}(\vec{\mathbf{q}})$ such that $\vec{\eta}_T = \mathbf{J}_{WT}(\vec{\mathbf{q}})\vec{\eta}_w$. Then $\mathbf{J}_T(\vec{\mathbf{q}}) = \mathbf{J}_{WT}(\vec{\mathbf{q}})\mathbf{J}_W(\vec{\mathbf{q}})$.
4. (10 points) Suppose one wishes to limit the rate of change of the tool pose such that $\left|\vec{\eta}_T\right| \le \vec{\eta}_T^{\max}$. Describe a set of linear constraints that would accomplish this in terms of $\vec{\eta}_W$ and $\mathbf{F}_W(\vec{\mathbf{q}})$.
5. (15 points) Suppose that the robot is located at an initial pose $\mathbf{F}_W^{init}$, so that the cutter is located at $\mathbf{F}_T^{init} = \mathbf{F}_W^{init}\mathbf{F}_{WT}$. Provide details for Step 3 of the midlevel controller that will cause the cutter pose ($\mathbf{F}_T(t) = [\mathbf{R}_{cut}(0)Rot(\vec{\mathbf{a}}_{move},\theta(t)),\vec{\mathbf{p}}_T(t)]$) to move with uniform angular and translational velocity from $\mathbf{F}_T^{init}$ to $\mathbf{F}_{cut}(0)$ and $\left|\dot{\theta}(t)\right| \le \dot{\theta}_{\max}$ and $\left\|\dot{\vec{\mathbf{p}}}_T(t)\right\| \le v_{\max}$. For this problem, you can ignore any joint velocity constraints. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. **HINT:** You can compute a transformation $\mathbf{F}_{move}$ such that $\mathbf{F}_{cut}(0)\mathbf{F}_{move} = \mathbf{F}_T^{init}$ or (alternatively) $\mathbf{F}_T^{init}\mathbf{F}_{move} = \mathbf{F}_{cut}(0)$. You can then interpolate a trajectory. Here, you can assume that you have available software to compute $[\vec{\mathbf{a}},\theta] \leftarrow AxisAngle(\mathbf{R})$ and $[\theta_x,\theta_y,\theta_z] \leftarrow ExtractAngles(\mathbf{R})$, where $\mathbf{R} = Rot(\vec{\mathbf{x}},\theta_x)Rot(\vec{\mathbf{y}},\theta_y)Rot(\vec{\mathbf{z}},\theta_z)$.
6. (10 points) Suppose that hand-over-hand cooperative control of the robot is desired. Describe an implementation of midlevel controller Step 3 that will yield the behavior $\dot{\vec{\alpha}}_W = \kappa^{rot}\vec{\tau}_W / \Delta t$ and $\dot{\vec{\varepsilon}} = \kappa^{p}\vec{\phi}_W / \Delta t$, where Δt is the sample interval of the midlevel controller, subject to the constraints on joint position, velocity, and acceleration constraints from Question 2. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. Your answer should be expressed in terms of a combined admittance law

$$\dot{\vec{\eta}}_W = \mathbf{K}_W\vec{\zeta}_W = \begin{bmatrix} \kappa^{rot} & \mathbf{0} \\ \mathbf{0} & \kappa^{vel} \end{bmatrix}\begin{bmatrix} \vec{\tau}_W \\ \vec{\phi}_W \end{bmatrix}$$

This form will allow for more sophisticated admittance behavior that may be desired in subsequent questions. Note also that a commanded incremental motion $\dot{\vec{\eta}}_W^{cmd}\Delta t$ corresponds to a commanded pose change $\Delta\mathbf{F}_W^{cmd} = [\mathbf{I} + sk(\vec{\alpha}_W^{cmd}), \vec{\varepsilon}_W^{cmd}]$.

7. (10 points) Now, suppose that you have positioned the robot so that $\mathbf{F}_T = \mathbf{F}_{cut}(0)$. How would you add constraints to your answer to Question 6 to ensure that the cutter tip moves only along the path direction together with an additional constraint that ensures that the orientation of the cutter is at $\mathbf{R}_{cut}(\lambda)$? **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. The added constraints may either be inequalities or equalities.

8. (10 points) One difficulty with the problem statement in Question 7 is that the cutter may stray off the path. How would you modify the constraints from your answer to Question 7 to include a small restoring motion to return the cutter to the path centerline at a speed no greater than $\sigma_{restore} / \Delta t$ while still permitting free motion along the path. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. The added constraints may either be inequalities or equalities.

9. (15 points) Again, suppose that you have positioned the robot so that $\mathbf{F}_T = \mathbf{F}_{cut}(0)$. How would you modify the objective function in your answer to Question 6 to ensure that the cutter tip stays within a distance δ_{cut} of the desired path. If the cutter tip is somehow a distance greater than δ_{cut} from the path then the system will only move in the direction toward the path centerline with a speed at least equal to $\vec{\sigma}_{response} / \Delta t$. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. If desired, you can include some additional computation to be performed by the planning software. Also, you do <u>not</u> need to copy-paste the constraints from your previous answer, I am interested <u>only</u> in the objective function and any calculations that you may need to establish quantities appearing in the objective function. **HINT:** You will find this problem easier to solve in the $\mathbf{F}_T$ coordinate system. If $\vec{\zeta}_W$ is a wrench (i.e., a force torque vector) expressed relative to $\mathbf{F}_W$ then the comparable wrench relative to $\mathbf{F}_T$ is $\vec{\zeta}_T = \mathbf{J}_{WT}^T \vec{\zeta}_W$. Describe the desired motions relative to $\vec{\zeta}_T$. So, the objective function will have the general form $\Delta\vec{\mathbf{q}}_{cmd} = \arg\min \left\| \vec{\mathbf{f}}_T(\vec{\zeta}_T, \mathbf{F}_T, \cdots) - \mathbf{J}_T \Delta\vec{\mathbf{q}}_{cmd} \right\|^2 = \arg\min \left\| \vec{\mathbf{f}}_T(\mathbf{J}_{WT}^T \vec{\zeta}_W, \mathbf{F}_T, \cdots) - \mathbf{J}_T \Delta\vec{\mathbf{q}}_{cmd} \right\|^2$, where $\vec{\mathbf{f}}_T(\zeta, \mathbf{F}, \cdots)$ is a vector valued function, and where you need to specify $\vec{\mathbf{f}}_T$. One possible form is $\vec{\mathbf{f}}_T = \mathbf{K}_T \zeta_T$, but there may be others. Essentially, $\vec{\mathbf{f}}_T$ is specifying a desired direction of motion. Subsequent problems will use similar analysis, although the form of the objective function may be different, and you may need additional terms.

10. (10 points) How would you modify the objective function in your answer to Question 9 to ensure that free motion is permitted as long as the cutter tip stays within a distance δ_{cut}^{free} but that there is increasing resistance to motion away from the path as the distance from the path approaches δ_{cut}. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. If desired, you can include some additional computation to be performed by the planning software. Also, you do <u>not</u> need to copy-paste the constraints from your previous answer, I am interested <u>only</u> in the objective function and any calculations that you may need to establish quantities appearing in the objective function.

11. (15 points) How would you modify your answer to Question 11 to ensure that the cutter ball does not penetrate anatomic structures A and B. There should be increasing resistance to motion toward these

structures whenever any part of the cutter ball comes within a distance $\delta_{approach} = \delta_{cut} - \delta_{free}$ of the structures. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. If desired, you can include some additional computation to be performed by the planning software. Also, you do not need to copy-paste the constraints from your previous answer, I am interested only in the objective function and any calculations that you may need to establish quantities appearing in the objective function. **HINT:** My solution to this problem involved precomputing one or more additional properties for the CT map.

12. (10 points) After the path has been traversed, the next task is to machine out the interior of anatomic structure C. At this point, the cutter orientation is permitted to change, but no part of the cutter ball should penetrate beyond anatomic structure C. There should be increasing resistance to motion toward the exterior of structure C. If (for whatever reason), any part of the cutter ball penetrates into the exterior of structure C, then the only allowed motion should be toward the interior. Describe a Step 3 implementation for this behavior, subject to the joint constraints of Question 2. Here, I am really looking for a specification of the objective function. **NOTE:** Your answer should include formulas and appropriate pseudocode, possibly including conditional statements. If desired, you can include some additional computation to be performed by the planning software. Also, you do not need to copy-paste the constraints from your previous answer, I am interested only in the objective function and any calculations that you may need to establish quantities appearing in the objective function.