Kangsan Kim
Kevin Yee
05/06/2016
CIS II

Project #10: Real Time Motion Reflexes for Robotic Hip Surgery

Background

This project focuses on improving the performance of the Robone robotic surgery system. Upon the system's completion, Robone will be used to automate aspects of hip replacement surgery including the femur milling. Although robotic milling is inherently more accurate than the milling process in traditional surgery, one drawback of using a robot for this process is that it takes longer time to complete. Longer surgery durations lead to increased hospital costs because fewer patients can be treated per unit time. Shorter surgeries are also advantageous for patients because they will spend less time in the hospital.

Problem

The current system implementation provides a constant tool speed and the tool traverses the cut path at that speed without modification. Decreasing the operation time is not as simple as increasing tool speed because when dense bone is cut too quickly, adverse effects result. *Tool chattering*, undesired vibrations at the tool cutting surface, may occur which can be dangerous to the patient and surgical team as well as damage the tool itself. Additionally, the cut may not be as accurate when it is performed too fast. That being said, less dense bone can be safely and accurately milled more quickly than denser bone. It would be useful to map the density of the bone preoperatively so that tool velocity could be incorporated into the path planning, but this method is unrealistic and not worth as many resources as it requires. The alternative is to measure or approximate the density of bone during the operation itself.

Fortunately, the force on the tool tip is a good approximation of the density of bone and this project intends to leverage that data to address this problem.

The primary hardware component of Robone is the Kuka LBR iiwa robotic arm. In each joint of the arm is a torque sensor, but the data from these sensors is not currently used by the system in any capacity. If the torque sensor data is retrieved from the robot arm, it could be processed and interpreted in a *force-controlled velocity* (FCV) algorithm which could update the tool speed in real time. This algorithm would increase the time efficiency of the surgery without compromising safety or accuracy.

Approach

There were three independent components required implement the FCV algorithm:

1) Read the raw data from the each joint's torque sensors and process the data to calculate the force on the tool tip
2) Devise a method to traverse the cutting path at variable speeds as opposed to the constant speed in the current system
3) Develop an appropriate model between tool tip force and cutting velocity such that the tool moves at some user-defined maximum velocity when no force is present and does not move at all when the tool experiences some maximum force

The first two components were implemented and tested in the Virtual Robot Experimentation Platform (V-REP). To pass the torque data from the Kuka machine to V-REP, the data was passed to a plug-in. The plug-in serialized this torque data, and wrote the external torques into the respective joints in the simulated arm. The external torques were then converted back to numbers. To calculate the tool-tip force, the jacobian of the robot arm was computed using a V-REP function. Then, the jacobian was multiplied by the external torques in each joint to calculate the x,y,z tool-tip forces, in which we can calculate the magnitude of the tool-tip force.

The previous implementation of the system used a command in the V-REP API that would take in a path object and a velocity and move the tool along that path at the given velocity to completion. This function call should no longer be replaced because there is no way to interrupt it while it is completing its path. The alternative is to break up the path into smaller pieces and move in discrete increments of varying length. Because velocity is the time derivative of position, modifying distance while keeping time constant can approximate the derivative. Therefore, instead of actually determining a relationship between force and velocity, it would be more appropriate to determine a relationship between force and offset. When zero force is applied, maximum velocity should be achieved so the distance the arm travels in a given time step should be at its maximum. Conversely, when a maximum force is applied, velocity should be zero so the distance traveled is also zero.

The workflow described is made possible by making use of the path length property of a path. In V-REP there are two ways to define a position on a path: absolute distance and relative distance. The absolute distance is on a scale where zero is the beginning of the path to the path length at the end of the path. The relative distance is on a scale from zero to one. Converting between the two is determined by the path length proportionality constant. Adding the offset determined by the current velocity to the absolute path distance and converting it to the relative path distance and then to [x, y, z] coordinates, the next robot arm location is determined.

In order to define the ideal force-offset relationship, extensive testing must be performed on cadavers. With the assistance of surgeons, who will evaluate the cut performance, and engineers, who will evaluate the effects on the tool, a more appropriate model can be determined. In the meanwhile, there were a few models that were experimented with. In addition to a linear model, square root and logarithmic functions were used. The reasoning behind these models was to decrease the sensitivity to noise at low force value readings. The concave down nature of these models achieves this. At low forces, a increment in force is does not result in as large a change in offset as it does at higher forces. The expectation is that the final model would incorporate this feature.

Results

After implementation of the FCV algorithm, the informal tests appeared to pass without issues. When the robot path was impeded enough, it stopped until the external force was reduced below a defined threshold. Once the external force was released, the robot resumed along the planned path. Once more formal testing was about to begin, irregularities started to present themselves. For example, when a mass approximately 5 lbs was attached to the robot arm, the measured force on the arm was over double the actual force (50 N ≈ 11 lbs). After several similar discrepancies in informal tests, the robot arm manufacturer was contacted. The manufacturer tried to replicate the conditions under which the data was received, but upon failure informed the team that the feature was unsupported on the Fast Robot Interface (FRI).

The robot has two means to communicate with its surroundings: FRI and Java. The intention was to use FRI because it has a much shorter delay (~ 5ms) as compared to the Java interface (~ 300 ms). The manufacturer suggested using the Java interface to proceed which created complications because the existing structure for communicating with the robot over FRI is not nearly as built up as it is for the Java interface. In order for the data to be used, that communication link must be developed.

Significance

Giving the robotic arm the capability to update its cutting speed in response to the torques it experiences in real time allows for a time-efficient yet safe milling operation. Using our force controlled velocity algorithm, the robot is able to cut at the limit of safe speed such that surgical costs can be lowered by decreasing the time of surgery without hurting the safety of the procedure.

Although we were not able to reach all of our goals, our project laid a solid foundation for a force controlled milling operation velocity. If a method of accurately calculating the external torques were implemented, it could easily be plugged into our code to yield the tool tip force. Also, a model for force to velocity that was more fine tuned to the surgical applications could also be plugged in easily to our code.

Who did what

Kangsan was responsible for reading the torque sensor data and processing it to calculate a force on the tool tip. He also worked on smoothing the arm motion jerkiness caused by writing the arm to discrete positions as opposed to continuous path.

Kevin was responsible for devising the method to traverse the path at different speeds and developing the initial force-offset models.

What was accomplished vs. planned

When the project began, there were two parts of the project: the force-controlled velocity algorithm and *null space compliance*. Null space compliance is when the robot allows joints to be physically manipulated by external forces with the constraint that the tool position and orientation do not change. This is advantageous for a surgical system because it allows the surgeon and assistants to push the robot out of the way if it obstructs their view or inhibits their ability to operate on the patient. This half of the project was abandoned after the torque sensor was found to be unreliable. It remains a valuable future project should the torque sensor feature become supported in the near future.

However, the force-controlled velocity half of the project was almost fully implemented. Because the external torque values were inaccurate, we could not accurately calculate the tool-tip force, a vital component of our force controlled velocity algorithm. However, our method of calculating the tool-tip force given external torque values is mathematically sound. Once accurate external torque values are obtained, calculating tool-tip force should be trivial. Rigorous testing was included in the original project plan, but it became apparent that testing would be an unproductive use of time because of the wildly erratic observations we made even before testing in a controlled. For example, when a tool that weighed approximately 5 lbs was attached to the robot, the sensor readings corresponded to a weight of 10 lbs (50 N).

Discuss what might be next

Although the FRI torque data for the robot unreliable, the robot manufacturer claims that the torque data accessed through the Java interface should be accurate. A future project could include setting up a method of communication between the robot and V-REP via the Java interface so that the torque data can be used in the same was that the FRI data was intended to be. Independently, a more appropriate model for the force-velocity relationship can be determined with the assistance of doctors, healthcare professionals and engineers with experience in experience in the field.

What you learned

The most valuable lesson we learned by far was to not make any assumptions about someone else's code, even if it's the code of a well established company. We assumed that the external torque calculations obtained from a function from Kuka was correct. We used this external torque to calculate our tool tip force. Clearly, an accurate tool tip force value is of utmost importance to milling operation velocity controlled by the force the tool tip experiences. This also interfered with out expected deliverable of accounting for the end-effector mass in the external torque values. Finding a workaround to obtain accurate values became a priority and set back our planned progress.

In addition, this project taught us how to code in Lua, how to work with the V-REP program, and familiarity with how to pass messages between different programs and machines.

Technical Appendices

Code can be found on our wiki page with comments explaining how to use the function and an explanation of the steps.