Introduction

This document will cover the design, construction, and operation of an automated craniotomy robot, of the 'homebuilt system' architecture of the 2015 Journal of Neurophysiology paper. This will include information on the basic operating principles, required parts, and a description of the LabVIEW code used to run the machine.

Automated Craniotomy Principles

The automated craniotomy robot uses a signal to automatically detect when the drill bit has gone through the skull so that it can stop drilling and not damage the brain. A small AC voltage (1 mV) is sent through the drill bit and the body of the mouse and the amplitude of this signal is measured by a computer program. Bone is a very good electrical resistor, so that when the drill bit pierces through the skull, a noticeable increase in conductance signifies when to stop the robot from drilling further down.

Parts

An automated craniotomy robot consists of a voltage detection circuit, a drill, and actuators. The parts listed below are the ones currently used on the automated craniotomy robots in the Boyden lab.

Part: DAQ board
Description: The data acquisition board is used to interact with LabVIEW software. Specifically, an analog output is used to generate the AC signal for the detection circuit. An analog input is used to measure this signal, and a digital output is used to turn the drill on and off with a solenoid valve.
In the above photograph, the sense resistor is connected directly on the DAQ board to minimize noise.

Part: 10 MΩhm sense resistor
Description: The sense resistor is large so that the maximum current remains low.

Part: Shielded USB cable
Description: A shielded USB cable is used to send the AC signal to and from the mouse. Without a shielded cable, environmental noise would dominate the signal since a very small amount of current (∼10 pA) is used in the detection circuit. The ground of the USB cable is connected to earth ground as shown in the picture below.
Minigrabbers are used to connect to the drill and to the body of the animal. These have been modified to decrease their applied force by stretching their springs. The unshielded wire distance is minimized for noise purposes as shown below.

Part: dental drill  
Model: NSK Presto (http://www.nskamericacorp.com/prod_handtools_presto.aspx) 
Description: This is an air-powered dental that has the drill axis aligned with the body of the dental drill. Unlike most dental drills that have a 90° bend between the handle and the
drill axis, this drill does not get in the way of the stereotaxic frame. There are other dental drills that could potentially work. However, it should be noted that this dental drill has almost no run-out, unlike the first dental drill that was tried, the Buffalo No. 220 (http://www.buffalodental.com/handpieces.html).

Part: electrically insulated dental drill holder
Description: A custom 3D printed part that secures the dental drill to the three-axis stage. It is made of plastic (ABS), which prevents the detection signal from travelling through the stage. A Solidworks model of this part can be found at http://www.autosurgery.org/. It should be noted that this part was initially designed for the first dental drill tried (Buffalo No. 220), but can also secure the NSK Presto.

Part: three-axis stage with linear servomotors
Description: A three-axis stage with linear servomotors that has 1” of travel in each direction. These motors have a repeatable step size of 0.2 µm. The program steps down in 5 µm steps so this level of resolution is most likely unnecessary. These motors are also controllable through LabVIEW.

Part: motor controller
Model: TDC001 (http://www.thorlabs.com/thorproduct.cfm?partnumber=TDC001)
Description: Each motor needs a motor controller.

Part: power supply for motors
Model:
cheap option: (3) of TPS001 (http://www.thorlabs.com/thorproduct.cfm?partnumber=TPS001)
medium option: (1) of TPS008 (http://www.thorlabs.com/thorproduct.cfm?partnumber=TPS008)
expensive option: (1) of TCH002 (http://www.thorlabs.com/thorproduct.cfm?partnumber=TCH002)
Description: Each motor controller needs to receive power and send/receive signals from the computer. The cheap option is an individual power supply for each motor controller. In this case, each motor controller will also have to be connected to the computer through a USB cable. The medium option is a single power supply for up to (8) motor controllers. In this option, each motor controller still needs to be connected to the computer through a USB cable. The expensive option (and the one I prefer since it is less cluttered) is a power supply and base for up to (6) motor controllers. The base connects all of the motor controllers to the computer through a single USB cable.

Part: stereotaxic
Description: A stereotaxic is used to secure the animal during surgery, as in a normal surgery setup. Unlike some other models, the Kopf Model 900 has a lot of space for the drill to fit.
Part: solenoid valve  
Model: Clippard EV-2-6 (http://www.clippard.com/part/EV-2-6)  
Description: A solenoid valve is used to turn the dental drill on and off through computer commands.

Part: transistor+power supply  
Description: Since the DAQ cannot provide enough current to activate the solenoid valve, a transistor and DC power supply are required.

Part: drill bit/end mill  
Model:  
Drill bit: 200 μm drill bit from McMaster 8904A27 (http://www.mcmaster.com/#8904a27/=t2ezmo)  
Description: Commercially available dental burrs come in various shapes and sizes but only go down to 500 μm in diameter. We use miniature drill bits and end mills to create craniotomies of 200 μm diameter (both the drill bits and end mills come in various sizes, starting at 50 μm in diameter). Unfortunately, dental drills come with a standard chuck size that will only accept 1/16” diameter shanks. These drill bits have 1 mm shanks, so we machine custom aluminum sleeves with a 1 mm ID and a 1/16” OD. The end mills come with a 1/8” shank. We have these bits ground down by another vendor, Contour 360 (http://contour360.com/). This needs to be done with special tooling since the bits are made of tungsten carbide, and it is very important that the concentricity of the bit shank and cutting surfaces is maintained when the shank diameter is reduced. Also, a wire is required to make direct contact with the drill bit for the detection circuit, as shown below.
Other parts: computer with Labview software, wall air supply, standard surgery equipment, a secure mount for 3-axis stage, stereomicroscope

The above photograph shows one way of securing the three axis stage.

Full setup, with microscope.

LabVIEW Program Description/Walkthrough
This is a description of how the automated craniotomy LabVIEW program works. Specifically, this is for version 2.8, but other versions are similar. The program works in the following manner:

Before running the program, the user inputs how many craniotomies are to be drilled by entering the number of rows and columns. Also, the spacing between craniotomies in the rows and columns is entered. Finally, a file path for a data log is entered if it is desired. The data log creates a text file for each craniotomy that indicates the voltage amplitude at each drill depth.

This figure shows the front panel of the LabVIEW VI.

Here, the user enters the number of rows and columns, and the spacing between each craniotomy. Also, the user enters where the data file will be stored.
When the program is started, it takes a few seconds for the motors to initialize. After they are initialized, they can be moved into position by using the buttons on the motor drivers or by entering a number for the displacement of each motor in the LabVIEW window.

This is the interface for the servomotors. When they are initiated, the fields are filled in with numbers. After initialization, numbers can be entered to specify the location of the motors in the field under the label “apt motor controller.”

Once the drill is positioned in the location for the first craniotomy, the user presses the “Done Positioning” button and the automated craniotomy robot begins drilling the first hole. It will proceed to drill all the holes, and when it finishes, it will retract the drill to the maximum height and center the other two axes to prepare for the next procedure.

The automated craniotomy LabVIEW program is a fairly simple VI. The following section will describe how it works in a manner that should be simple enough so that a user not experienced in LabVIEW programming can understand and make modifications to the program.

The VI is based on a flat sequence structure. This simple means that it operates in a linear manner where each frame of the program executes one after the other, from left to right.
This first frame initializes the motors and then waits for the user to position the drill in the correct location. The blocks labeled “MG17MOTOR” are used to interface with the motors. First, the motors are sent a serial number that corresponds to each different motor. Next, they receive a start command. The final motor block in this frame is the “GetPosition” block. This gets the position of each motor and stores it in two local variables: position and origin. Since these motor blocks are in a while loop, as the user moves the motors the position is constantly being updated. Only after the “Done Positioning” button is pushed, do the position values get stored in the two previously mentioned local variables. Another item to note is the reference out local variables. These are used throughout the program whenever a “MG17MOTOR” block is used. These store information that the motor controllers need so it is important that they are passed between the different motor blocks.
Next, there are several nested loops that move the drill to each position, and automatically drill down through the skull. The outermost loop is a for loop that does the columns, or x coordinates for the craniotomies. As seen in the above image, it moves the x motor the distance “delta x,” unless it is the first column. If it is the first column, the conditional statement tells the motor to move a step size of zero since the drill is already in the desired location for the first column of craniotomies. The “MoveJog” command is used for the first time here. This gets an input of either 1 or 2 for the direction (in this case a 2).
After the drill is moved to the column location, it is next moved to the row location, as seen above. As before, if it is the first row, the motor does not move, since the drill is already in the correct location. Otherwise, the drill moves over the distance “delta y.” Additionally, instead of moving in the same direction each time, the y motor changes direction for each column.

Instead of moving as shown in A) in the above figure, the drill moves as in B). This saves a small amount of drilling time because the skull curvature in the x direction is greater than that in the y direction. A greater curvature in the x direction means that the drill must be retracted a large distance in order to safely move in the x direction without hitting the skull. Retracting the drill more means that the drill may have to take more steps before creating a craniotomy, which leads to a longer drilling time. By moving the robot as in B), a smaller z offset is used between the craniotomies in the y direction. This is just a minor feature, but does save some time, especially for large numbers of craniotomies.
The next frame in the sequence is another small feature to try to save some drilling time. After a craniotomy is made, the depth at which the drill stopped is recorded and the next craniotomy is drilled to this depth plus an offset of either 300 or 400 µm. The skull is only about 150 µm thick, but due to the curvature of the skull, a 300-400 µm offset is required to ensure that the drill does not go too far on this step. For this step, a digital on signal is sent to activate the drill, then the z motor moves the drill down, and finally the drill is turned off. There are pauses in each of the frames in this sequence to ensure that each step is finished before the next one starts.

The next part of the code is the actual automated drilling section. The first thing that happens is a measurement of the voltage amplitude is made. This is done using a discrete Fourier transform (http://en.wikipedia.org/wiki/Discrete_Fourier_transform), which allows the detection signal to be a very low current signal by extracting the information from a noisy signal. The messy looking for loop in the above image is executing the discrete Fourier transform. The voltage amplitude obtained from this is compared to the threshold value, and this is used to make a decision in the next step.
If the measured amplitude is greater than or equal to the threshold, this means that the increase in conductance associated with breaking through the skull has not occurred, and the robot must drill further. This sequence is similar to the one previously mentioned. The drill is turned on, the z motor moves down 5 µm, and then the drill is turned off.

If, however, the measured voltage amplitude is below the threshold, the drill is not turned on, and the z motor retracts to a safe distance above the skull to move the drill to the next position. This stops the while loop that contains the amplitude measurement and drilling sequence, and allows the program to move to the next drilling position.

Each time a voltage amplitude measurement is made, it is recorded along with the depth at which it was made. This is stored in a .txt file.
After all the craniotomies have been made, the drill is retracted in the z direction up to the maximum distance. Then, the x and y motors are centered. Finally, all three motors are stopped. This completes the automated craniotomy program.