CS 480 Homework 0

0.) NAME: ANSWER KEY

Due 2015-09-11. Standard homework rules apply: your answers should be entirely yours, except for (1) standard reference material such as textbooks or wikipedia, and (2) the sources you explicitly cite.

1.) A cartesian 3D printer's print head, which weighs m=0.5 kilograms, is in the middle of a rapid move at V(0)=100 mm/second (0.1 m/s). Determine how fast can the steppers stop the head's motion, under each of the following assumptions about the stepper's torque.

Reminders:
P(0 seconds) = 0 meters
V(0 seconds) = 0.1 meters/second
V = dP/dt
A = dV/dt
F = mA, A=F/m
We want t and P(t) when V(t)=0 meters/second.

1.a.) Assume the stepper motors can produce a fixed 0.2 Newton-meter of torque (30 oz-in), which with a 10mm radius pulley produces a constant force of F=20 N, and so accelerates the head at A=20N/0.5Kg = 40 m/s^2.

A=-40 m/s^2
V(t) = 0.1 – t*40

Setting this V(t)=0, we can solve for the stop time,
0=0.1 – t*40
0.1=t*40
0.0025=t

Time to stop: 2.5 milliseconds

P(t) = 0.1*t – 0.5*40*t^2
= 1.25e-4 meters

Distance to stop: 0.125 millimeters

1.b.) Now assume the stepper motor torque curve displays an inverse-law fall-off with velocity:
A=-0.5/V (units in meters and seconds)
Assume the stepper can only apply 0.025 Newton-meter of torque when the head is moving at 100mm/second, for an acceleration at this speed of 5 m/s^2. Use calculus to compute an exact answer.

A=*-0.5/V*
dV/dt = *-0.5/V*

If you're concerned about units, you can take the units for the 0.5 constant as m^2/s^3 so if V=m/s, A=m/s^2.

This differential equation has a general solution of the form V(t) = sqrt(k-t) for some constant *k* (see the Wolfram Alpha General Differential Equation Solver).

k has units of seconds. To fix up the units for V, you can treat this as V=sqrt(c*(k-t)), where c=1.0 m^2/s^3, the same units as the constant in the A equation (see general form of A=-c/2/V). This makes the units work out as V=m/s=sqrt(m^2/s^2).

We can find k by substituting V(t) = sqrt(k-t) at t=0:

V(0)=sqrt(k) = 0.1
  k = 0.01 seconds
This immediately gives us the time to stop, when V(t)=0=sqrt(k-t), namely at t==k.

Time to stop: 10 milliseconds

We can now find P(t) when stopped:
V(t) = sqrt(k-t)
P(t) = integral of V(t)
  = -2/3*(k-t)^1.5 + C
C=+2/3*(k)^1.5 to put P(0)=0
  = 6.667e-4 meters

Distance to stop: 0.66 millimeters

1.c.) Do you think these stopping times and distances are important? Why or why not?

For time, 2 to 10 milliseconds is quite a while for a microcontroller—several iterations of the motor control loop. This implies the motion control system needs to look ahead in the command stream to be able to stop on time.

For distance, the 1.a answer of 0.125 milimeters might not be too important, since the nozzle is likely around 0.3 millimeters, and hence plastic blobs might be bigger than that. The 1.b answer of over half a millimeter is definitely important; most printers are more repeatable than this.

(Score: 50 points. Please do show your work!)

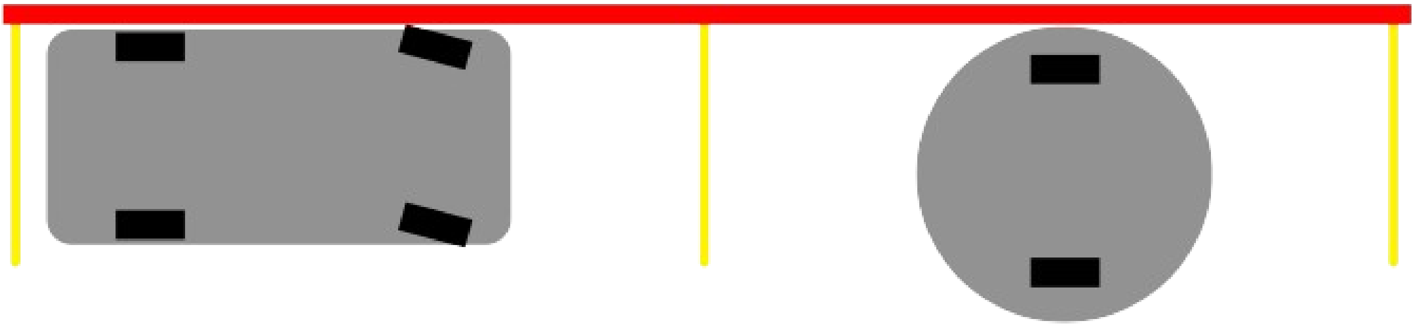
2.) You're about to 3D print a hollow cube with sides of length L and walls of thickness W (and W is much less than L). You'll be printing with an ABS plastic which has a density of 1 gram/cc. How many grams of filament do you need?

The cube has 6 sides that each need printing. Each side has area L^2 and thickness W, for a volume of L^2*W per side. You can ignore the edges and corners of the cube, volumes L*W^2 and W^3 respectively, if W is small. Because this plastic has a density of 1 gram/cc, volume in cc handily equals mass in grams.

Filament usage = 6*L^2*W grams
What might be surprising about this is the non-cubic dependency on size.

If you care about the edges and corners of the cube, assuming L is inside length, you get:
Filament usage = 6*L^2*W + 12*L*W^2 + 8*W^3 grams
If you would like a real algebraic mess, assume L is the outside of the profile and substitute L'=L-2*W here. You can clean up the mess by noting it's also L^3 – $(L-2*W)^3$ =volume of outside – volume of empty middle).
(Score: 10 points.)

3.) This top-down diagram of a street depicts a typical four wheel car-style vehicle on the left, and a Roomba-style two wheel vehicle with differential tank steering on the right. Both are parked at the curb, shown in red.



3.a.) For which vehicle would it be easier to build an autonomous parking system, assuming electric all-wheel drive?
Definitely the Roomba, because it can change direction by spinning in place. Changing the car's direction requires clearance to drive forward and backward.

3.b.) Is either vehicle's control of its 2D position and orientation "holonomic"? Why or why not?
A control system is holonomic if the system's total number of degrees of freedom equals the number of degrees of freedom we can directly control. A system with 2D position and orientation has three degrees of freedom.

For the car, we can drive forward and backward, and turn the steerable wheels left and right, for two degrees of control. For the Roomba, we have one degree of control for each wheel (drive together for forward or backward, drive opposing for turning), for two degrees of control again.

Neither system is holonomic. For example, in this diagram we can't immediately translate either vehicle along the Y axis (perpendicular to the wheels) without affecting other axes.

Here's an example of a holonomic 4-wheel scissor lift: https://youtu.be/IlmKcohyXG0?t=7s

(Score: 15 points)

4.) Write a program in any language to (a) read an STL file (binary or ASCII, your choice), and (b) calculate its volume.

You can calculate the total volume by looping over each of the triangles, and for each triangle with 3D vertices a, b, and c, calculating the signed volume of a tetrahedron formed by adding one vertex at the origin:
    V += dot(a,cross(b,c))/6

To understand the dot cross volume calculation, see this blog post from Jessie at Nervous System:
    http://n-e-r-v-o-u-s.com/blog/?p=4415

The hard part here is reading the STL file! The ASCII version can be read just like a text file, or using any of a variety of libraries (which are perfectly fine as long as you cite them). Here's some fragments I used in reading the files, which rely on my "vec3" 3D vector class https://github.com/olawlor/osl/blob/master/vec4.h

```
/** Binary format triangle: ASSUMES little-endian machine. */
```

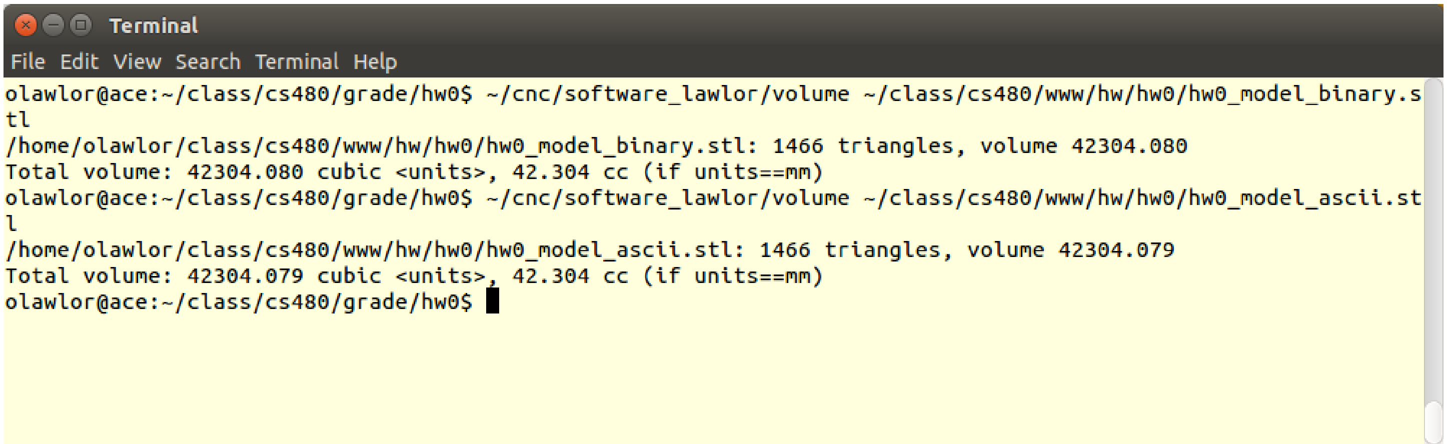
```
struct stl_triangle {
        vec3 normal; // surface normal, or (0,0,0) for right hand rule
        vec3 vtx[3]; // vertex locations

        stl_triangle() :normal(0.0) {}
};

/** Binary format header: ASSUMES little-endian machine. */
struct stl_file_header {
        char comment[80]; // If this starts with "solid", it's ASCII
format
        uint32_t ntri; // triangle count: little endian integer
};


// Calculate volume enclosed by this list of STL triangles.
//  ASSUMES shape is closed, AND triangle orientations are consistent.
double stl_volume(const std::vector<stl_triangle> &tri)
{
        double sum=0.0;
        for (size_t i=0;i<tri.size();i++) {
                const stl_triangle &t=tri[i];
                // This is the volume of tet from origin to triangle:
                sum+=dot(t.vtx[0],cross(t.vtx[1],t.vtx[2]))*(1.0/6.0);
                //  See http://en.wikipedia.org/wiki/Tetrahedron#Volume

        }
        return sum;
}
```



```
olawlor@ace:~/class/cs480/grade/hw0$ ~/cnc/software_lawlor/volume ~/class/cs480/www/hw/hw0/hw0_model_binary.s
tl
/home/olawlor/class/cs480/www/hw/hw0/hw0_model_binary.stl: 1466 triangles, volume 42304.080
Total volume: 42304.080 cubic <units>, 42.304 cc (if units==mm)
olawlor@ace:~/class/cs480/grade/hw0$ ~/cnc/software_lawlor/volume ~/class/cs480/www/hw/hw0/hw0_model_ascii.st
l
/home/olawlor/class/cs480/www/hw/hw0/hw0_model_ascii.stl: 1466 triangles, volume 42304.079
Total volume: 42304.079 cubic <units>, 42.304 cc (if units==mm)
olawlor@ace:~/class/cs480/grade/hw0$ █
```

(Score: 25 points)