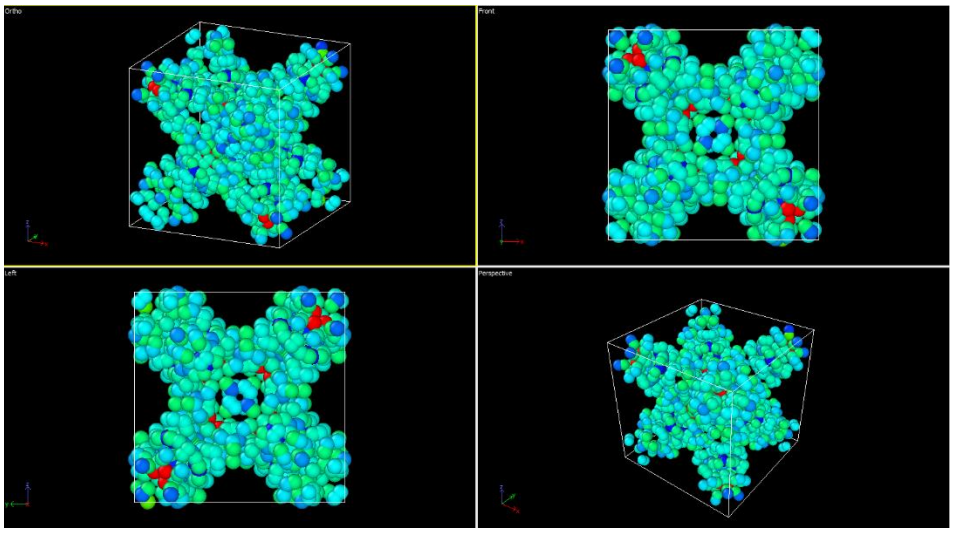
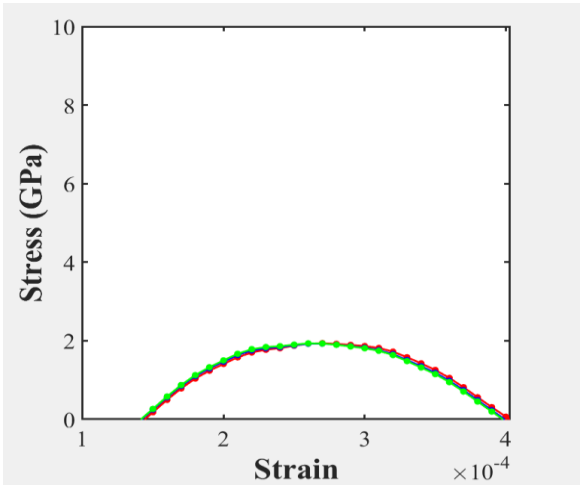
Undergraduate Research (Spring 2020)

For the 2020 spring semester, I have been conducting research under Ioannis Mastorakos on the modeling, manufacturing and testing nanoscale metallic composites and nanofoams. We have investigated numerous types of nanoscale metallic composites/nanofoams and different methods of manufacturing them. We have focused on chemically restructuring MOFs (Metal-Organic Framework). I have used LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator, a molecular dynamics program) to simulate the formation of the nanofoam and calculate the stress-strain curves of the nanofoam under compression and tension. I have been using Ovito (an open source visualization and analysis software for particle simulation data) and MATLAB as post-processing programs; Ovito to visualize the data and export data into usable filetypes, and MATLAB to graph the data.



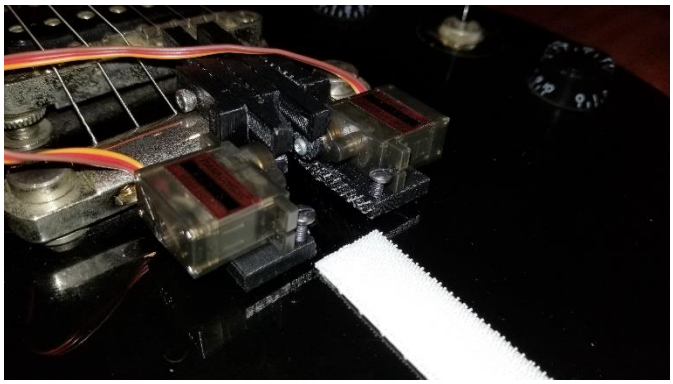

```
read_data   CuC_data.txt

#replicate  5 5 5
replicate   2 2 2

pair_style bop
pair_coeff * * CCu_v2.bop.table Cu Cu Cu C
neighbor    2.0 bin
neigh_modify    every 10 check yes
comm_modify cutoff 12.70

compute csym all centro/atom fcc
compute cpe all pe/atom
compute peratom all pe/atom

fix 1 all nve/limit .1
#fix 1 all npt temp 300.0 300.0 1.0 iso 10.0 10.0 1.0
reset_timestep 0
timestep 0.0001

label loopa
variable a loop 20

#compute csym all centro/atom fcc
#compute cpe all pe/atom
#compute peratom all pe/atom

dump     1 all custom 10 dump.*.* id type xs ys zs c_csym c_peratom fx fy fz

#dump   myDump all atom 5 dump.atoms

thermo 5

run 5

minimize    1.0e-12 1.0e-12 10000 100000

undump 1

next a
jump SELF loopa

unfix 1

reset_timestep 0
#timestep 0.0001

fix 1 all npt temp 300.0 300.0 1.0 iso 1.0 1.0 1.0
thermo 10

dump     1 all custom 10 dump.5..* id type xs ys zs c_csym c_peratom fx fy fz

run 5000
```

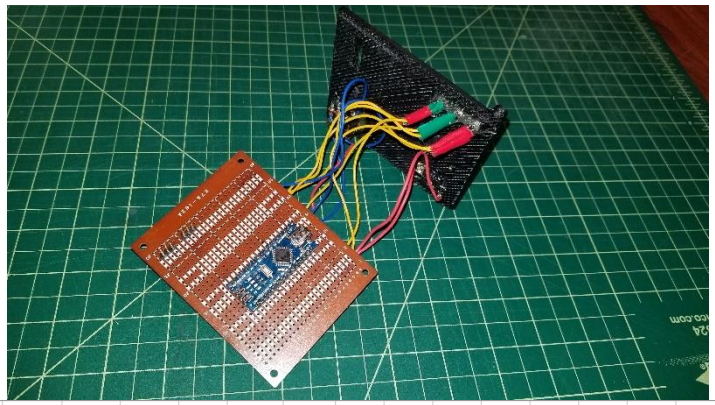



```
C:\Windows\System32\cmd.exe

# Dump to cfg for Ovito post processing
dump     1 all custom 100 dump.*.* id type xs ys zs c_csym c_peratom fx fy fz

# Display thermo
thermo  10
#thermo_style   custom step v_strain temp v_p2 v_p3 v_p4 ke pe press

run             50000
Setting up Verlet run ...
  Unit style    : metal
  Current step  : 0
  Time step     : 0.0001
Per MPI rank memory allocation (min/avg/max) = 7.793 | 7.793 | 7.793 Mbytes
Step Lx Ly Lz Press Pxx Pyy Pzz PotEng Temp
     0    31.72985    31.72985    31.72985    30202.086    30291.448    30059.504    30255.306   -5720.2507    29.108066
    10   31.730167    31.730203   31.730205    29409.021    29497.766    29258.621    29470.677   -5730.9329    109.03573
    20   31.730484    31.731246   31.731255    28106.852    28210.614    27938.029    28171.913   -5745.8096    220.21802
    30   31.730802    31.732946   31.732969    26417.925    26552.204    26224.138    26477.433   -5761.5293     337.5698
    40   31.731119    31.735264   31.735305    24431.722    24609.386    24212.664    24473.115   -5775.0402    438.25447
    50   31.731436    31.738149   31.738215    22226.172    22455.701    21988.009    22234.806   -5784.4015    507.76326
    60   31.731754    31.741549   31.741646    19886.993    20171.404    19640.334    19849.241   -5789.1568    542.71406
    70   31.732071    31.745407   31.745539    17467.081    17804.686    17223.744    17372.813   -5790.2496    550.21335
    80   31.732388    31.749665   31.749837    14975.874    15354.222    14746.882    14824.118    -5789.517    544.06194
    90   31.732705    31.754264   31.754477     12423.22    12834.511    12215.638     12219.51   -5788.9124    538.92796
   100   31.733023    31.759144   31.759397    9780.7421    10214.036    9593.9649    9534.2253   -5789.7382    544.59918
   110    31.73334    31.764241   31.764533    7047.0727    7495.0123    6877.3808    6768.8248    -5792.2     562.64013
   120   31.733657     31.76949   31.769819    4199.5738    4655.7781    4042.8198    3900.1235   -5795.4596    586.78926
   130   31.733975    31.774823   31.775186    1214.7809    1674.6983    1067.9006    901.74378    -5798.094    606.39799
   140   31.734292     31.78017   31.780562   -1883.3377    -1423.313   -2020.8176   -2205.8824    -5798.712    611.05025
```

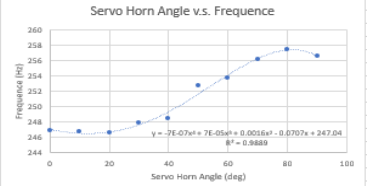
Arduino Controlled Guitar B/G-Bender (Spring 2020)

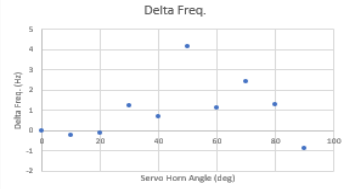
As an independent project I built a b/g-bender for my electric guitar. A b/g-bender is a mechanism that increases the tension in the b and/or g string of the guitar, changing the pitch of the string. I designed and 3D printed a jig that securely grips the bridge of the guitar which has two servos motors mounted to it. The servos turn a small cam, raising one side of a lever arm attached to the jig, causing the other side of the lever press down on the b or g string; therefore, increasing the tension of that string. The servos are controlled by a nano Arduino (microcontroller). The device has three modes: b string, g string, g and b string. The modes can be cycled through by using the push button under the LEDs. The three LEDs indicate the following: power on, b string, and g string. The servos can oscillate their position to give a vibrato effect. The speed that the servos oscillate their position is controlled by the potentiometer located in the opposite corner from the LEDs.

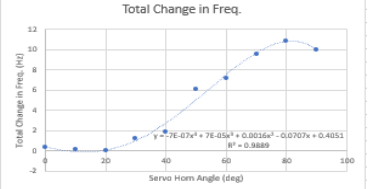




Servo Horn Angle (deg)	b-string Freq change test 1 (Hz)	b-string Freq. change test 2 (Hz)	b-string Freq. change test 3 (Hz)	AVG	Total Change in Freq.	Delta Freq.
0	247.6	246.4	246.9	246.967	0.333333	0
10	246.9	246.4	246.9	246.733	0.1	-0.23333
20	247.1	246.3	246.5	246.633	0	-0.1
30	248.2	247.8	247.5	247.833	1.2	1.2
40	248.8	248.8	248	248.533	1.9	0.7
50	253.1	252.2	252.8	252.7	6.066667	4.16667
60	253.9	253.8	253.7	253.8	7.166667	1.1
70	256.3	256.1	256.2	256.2	9.566667	2.4
80	257.5	258	257	257.5	10.86667	1.3
90	256.8	256.8	256.3	256.633	10	-0.86667
Min Freq.	246.6333					
Max Freq.	257.5					
Max Freq. (	10.86667					





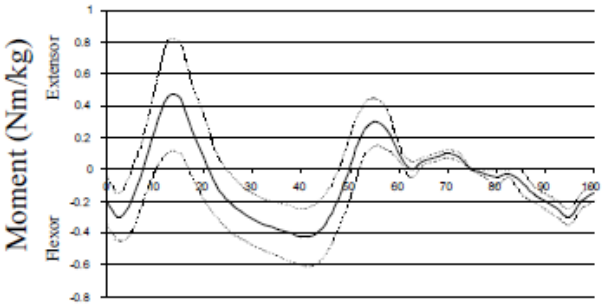


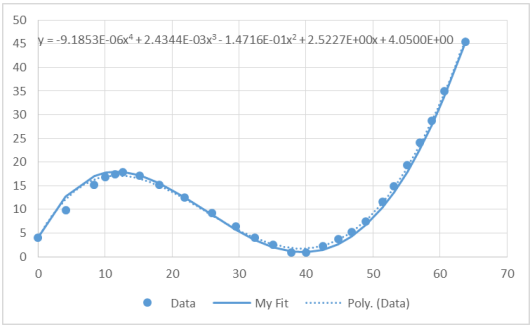
HONORS PROJECT (Fall 2017–Spring 2018)

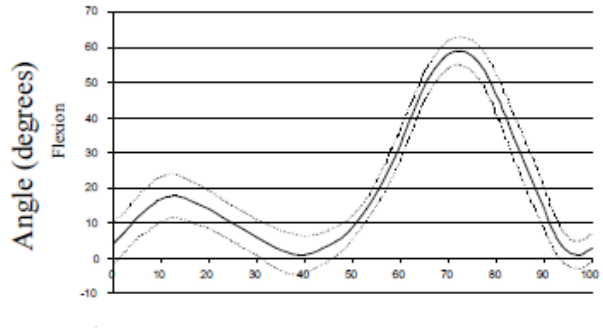
Senior Honors Program is a self-driven course for seniors at Albany Academy where a student learns about something not offered as a class at the school. The student uses their research to make a product for which they give an end of year presentation. Only a select group of applicants can participate in this course. My project was to make a prototype of a mechanically assisted knee device. I read multiple research papers about the kinematics of the leg while walking (an excerpt from the Army Research Lab paper on the kinematics of walking and running, which the main source of information, and some of my data reduction/analysis is shown on the next page). I used Autodesk Inventor to create a model of the frame and then used MATLAB to optimize the moment/torque of the knee. I built and tested multiple designs where I learn how to use an electric actuator, stepper motor, and brushless motor with ESC (electronic speed control).

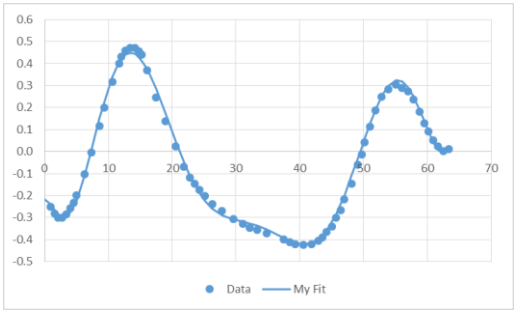
HONORS PROJECT (Fall 2017–Spring 2018)

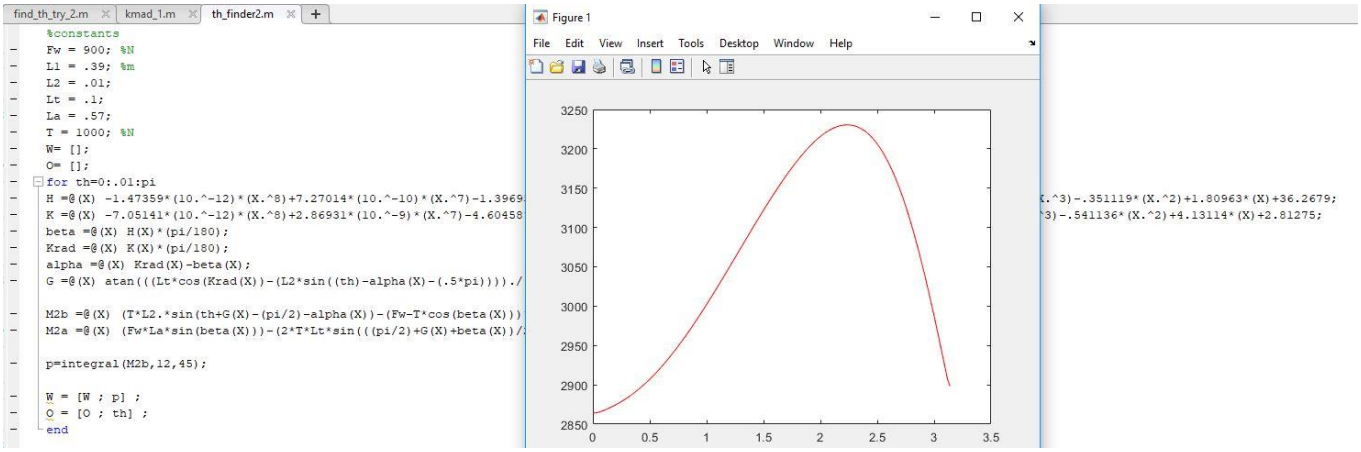
.





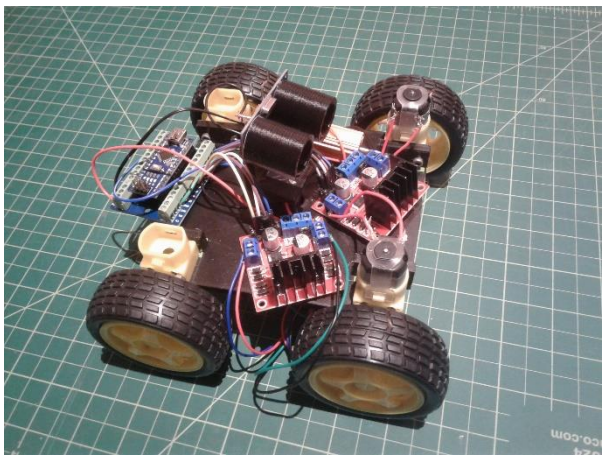






Mobile Arduino Based Scanning Robot (Spring 2019)

As an independent project I decided to learn more about robot solving process; therefore, I chose to design a semi-autonomous robot that could map out obstacles in its environment and plan a route around the obstacles. The robot uses an ultrasonic sensor to measure the distance of obstacle form itself and uses a stepper motor to turn the ultrasonic sensor to measure the distance of objects from itself that aren't right in front of the robot. The robot uses two dual H bridge circuits, one to control the stepper motor another to control two DC motors that move the robot. The brains of the robot is an Arduino Nano (micro-controller). Multiple parts of the robot are 3D printed: the chassis, the blinders for the ultrasonic sensor to focus its range, and the mount for the ultrasonic sensor. The ultrasonic sensor mount was designed to keep the measuring point of the sensor in the center of the robot and not travel in an arc when the stepper motor is turning.



scaner_4 §

```
long duration;
long distance;
float cardist;
float x;
float y;
int a[4] = {1,2,3,4};
int n;
int i;

void setup() {
  Serial.begin(9600);
  pinMode(2,OUTPUT);  //stepper
  pinMode(3,OUTPUT);  //stepper
  pinMode(4,OUTPUT);  //stepper
  pinMode(5,OUTPUT);  //stepper
  pinMode(6,OUTPUT);  //us sensor
  pinMode(7,INPUT);   //us sensor
  pinMode(10,OUTPUT); //motor 1
  pinMode(11,OUTPUT); //motor 1
  pinMode(12,OUTPUT); //motor 2
  pinMode(13,OUTPUT); //motor 2
  /*delay(5000);
  scan();
  delay(20000);*/
  movecar(1);
}

void stp() {
   digitalWrite(2,LOW);
   digitalWrite(3,HIGH);
   digitalWrite(4,LOW);
```