Team 3-Core CPU

Devon Delgado

Beza Desta

Isaaq Khader

(Questions 1 & 2 answered in the code)

3. Use the previous functions to determine attribute trends across all data samples for a given activity. Which attributes best help distinguish the sensor traces of one activity compared to another?

Overview	
Standing	Headset_pos(X,Y,Z), Left and right controller (x,y,z)
Sitting	Headset_pos(X,Y,Z), Left and right controller (x,y,z)
Jogging	Left and Right Controller Velocity (X, Y, Z) Headset Position in Y-Direction
Arms Stretching	controller_left_angularVel.y, controller_right_angularVel.y Controller_right_pos.x, controller_right_pos_z, controller_left_pos.x, controller_left_pos.z
Overhead	Controller Velocity in Y-Direction Controller Position in Y-Direction
Twisting	headset_angularVel.y, headset_rot.y, controller_left_angularVel.y, controller_right_angularVel.y, controller_left_pos_z, controller_right_pos.z

4. Which attributes are less useful for distinguishing the differences between activities? Why?

The headset velocities and angular velocities in the X and Z directions were particularly unhelpful for distinguishing between activities. Granted, there are some activities that share similar patterns in some of the measurements (controller positions and angular velocities for twisting and stretching), but there are some ways to differentiate between them. The headset measurements in these directions are unhelpful because no activity specifically requires movement in these directions. We aren't moving from left to right or back and forth and aren't rotating along those axes either. So, the measurements for them should be roughly the same for all activities.

5. Compute the mean and variance of the significant attributes across all data samples, then present them in a table for all six activities. Also, include visualizations that highlight how motion patterns differ between activities for certain attributes. Explain how the statistics and visualizations support your answer to #3.

Means and Variance Per Activity

Standing

Attribute	Average	Variance
Headset pos x	0.090	3.005
Headset pos y	0.008	0.009
Headset pos z	0.113	0.054
Left_controller_pos x	-0.026	0.004
Left_Controller_pos y	-0.712	0.028
Left_Controller_pos z	0.203	0.061
Right_Controller_pos x	0.209	0.001
Right_Controller_pos y	-0.714	0.028
Right_Controller_pos z	0.211	0.072

Sitting

Attribute	Average	Variance
Headset pos x	0.164	0.011
Headset pos y	-0.319	0.009
Headset pos z	-0.166	0.011
Left_controller_pos x	0.073	0.008
Left_Controller_pos y	-0.886	0.006
Left_Controller_pos z	0.033	0.016
Right_Controller_pos x	0.261	0.012
Right_Controller_pos y	-0.889	0.006
Right_Controller_pos z	0.023	0.014

Arms Stretching

Attribute	Average	Variance
controller_left_angularVel.y	0.237	2.781
controller_right_angularVel.y	-0.379	0.045
Controller_right_pos.x	-0.232	0.660
controller_right_pos_z	0.597	0.794
controller_left_pos.x	-0.379	0.071
controller_left_pos.z	0.183	0.167

Jogging

Attribute	Average	Variance
Headset Position X	0.2185	0.0236
Headset Position Y	0.0019	0.0099
Headset Position Z	0.1523	0.0639
Left Controller Velocity X	-0.00025	0.1018
Left Controller Velocity Y	0.00287	0.8383
Left Controller Velocity Z	-0.0114	0.1782
Right Controller Velocity X	0.0038	0.09378
Right Controller Velocity Y	0.00099	1.0722
Right Controller Velocity Z	-0.0132	0.1927

Overhead

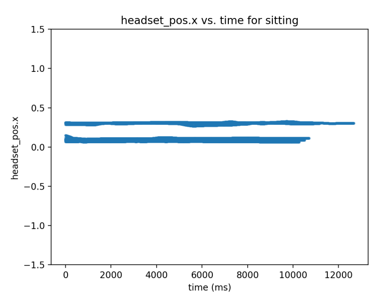
Attribute	Average	Variance
Left Controller Velocity Y	0.008	0.8061
Right Controller Velocity Y	0.0096	0.8132
Left Controller Position Y	0.0282	0.1221
Right Controller Position Y	0.2601	0.0034

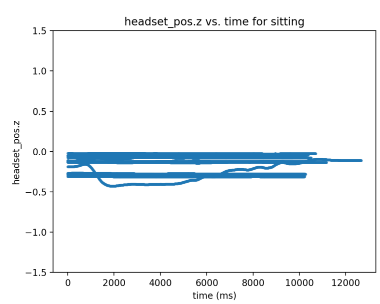
Twisting

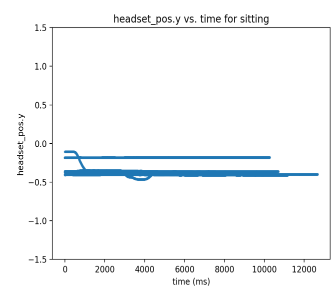
attribute	average	Variance
headset_angularVel.y	-0.001	0.09
headset_rot.y	182.568	15604.299
controller_left_angularVel.y	-0.004	4.703
controller_right_angularVel.y	-0.031	0.560
controller_left_pos_z	0.127	0.065
controller_right_pos.z	0.261	0.013

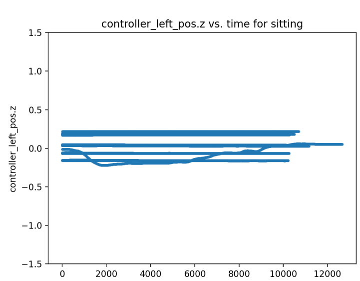
Plots

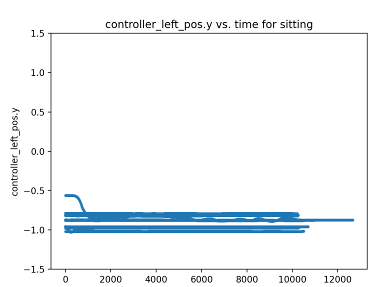
Sitting

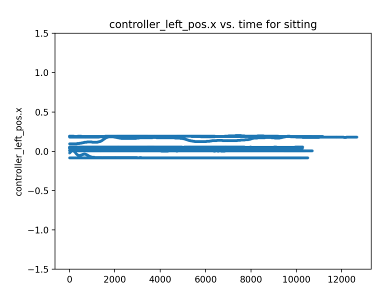


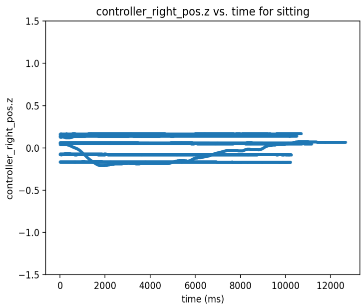


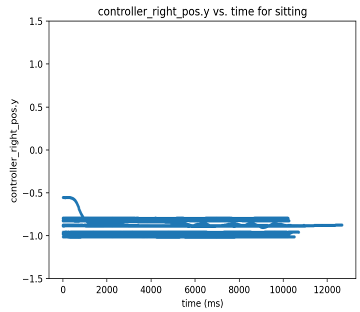


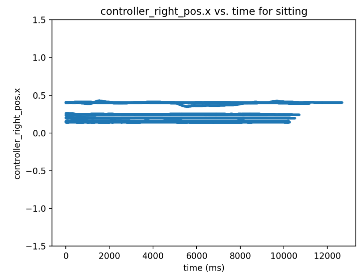




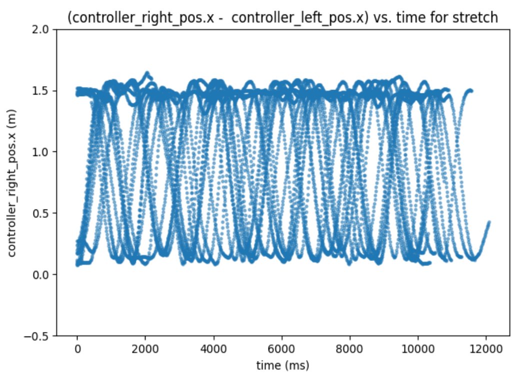


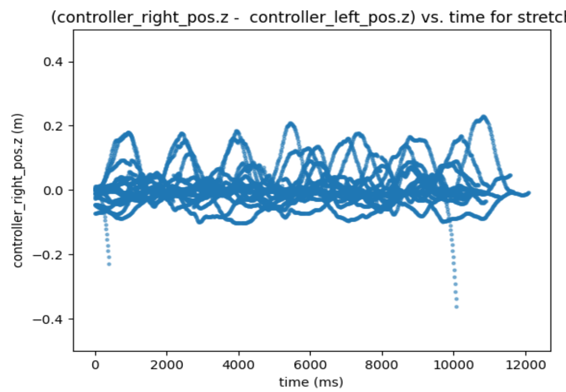


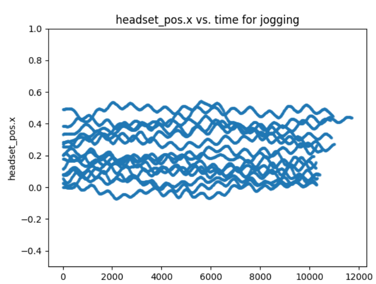


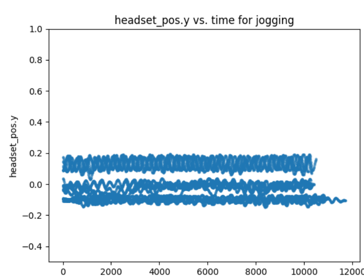


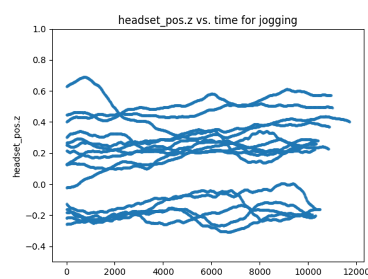
The graphs show us that position is an important attribute when looking at a static activity like sitting. Especially when compared to activities like stretching or jogging which have varying positions for both headset and controllers. By looking at the fairly quiet graphs produced from the data collected by sitting, we can say that this activity is sitting.



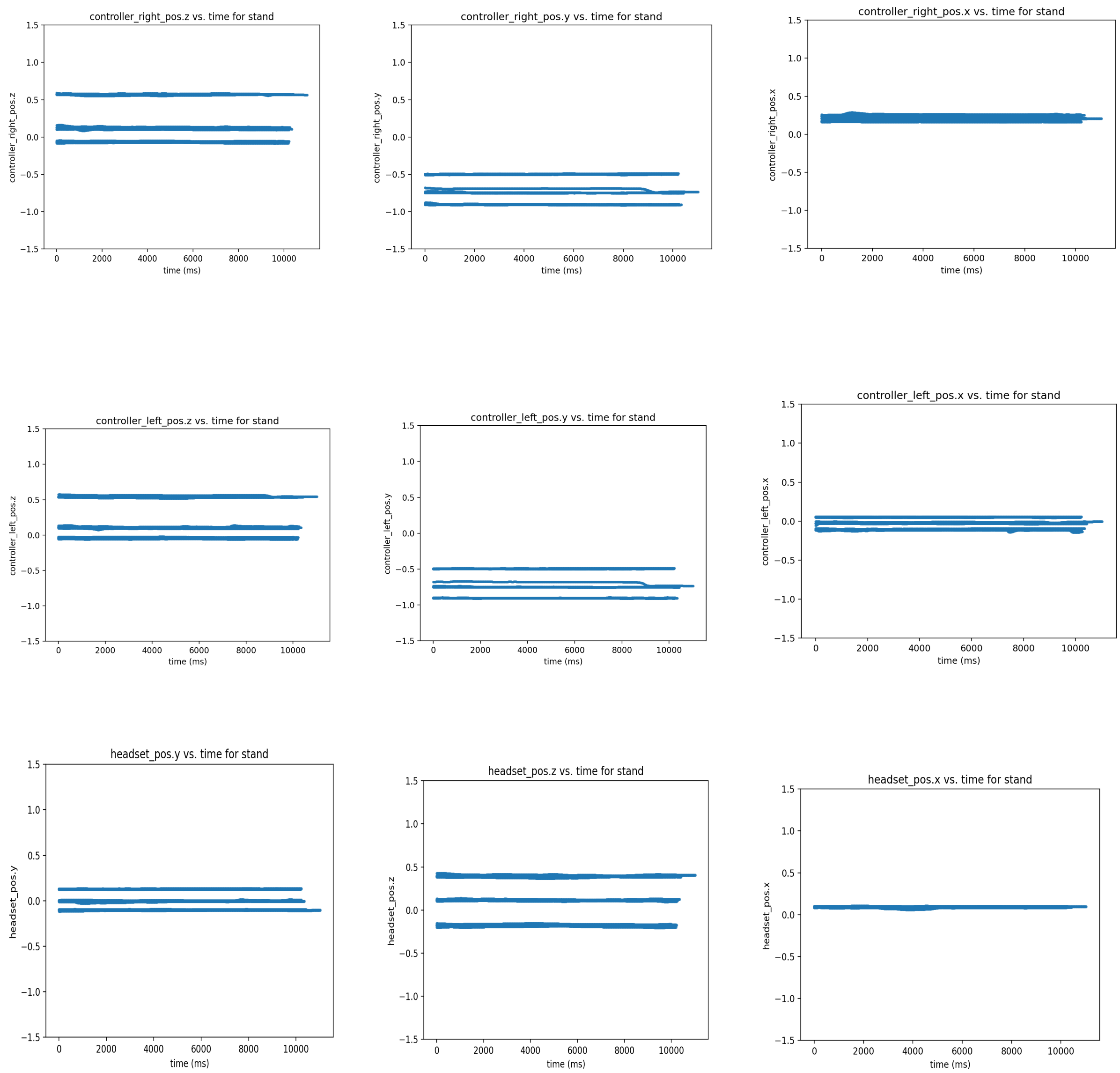






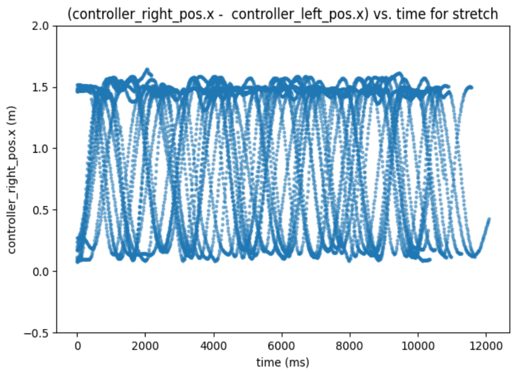


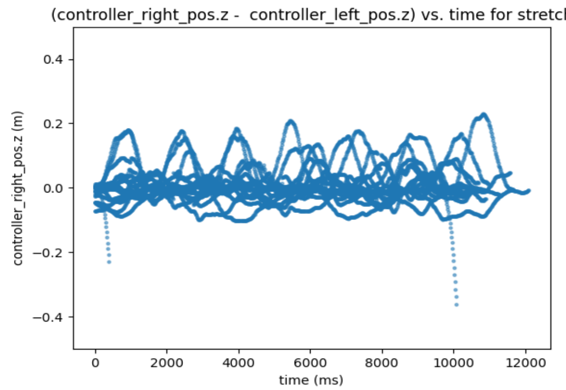
Significant attributes for standing

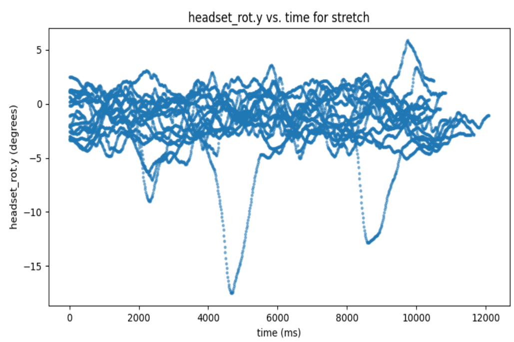


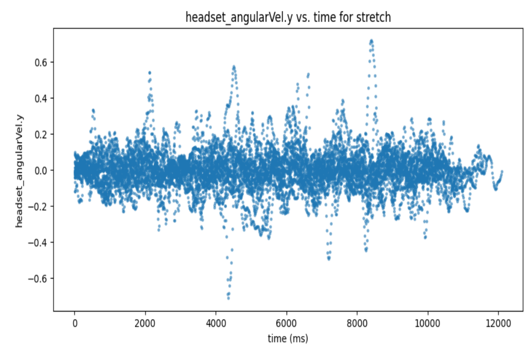
While sitting and standing can be easily differentiated from all the other activities since they are both static activities and yield quiet graphs, sitting and standing can be fairly difficult. In order to differentiate between sitting and standing, the most important thing is to look at the y position of both the headset and both controllers. The y position and for sitting as can be expected is much lower than that of standing.

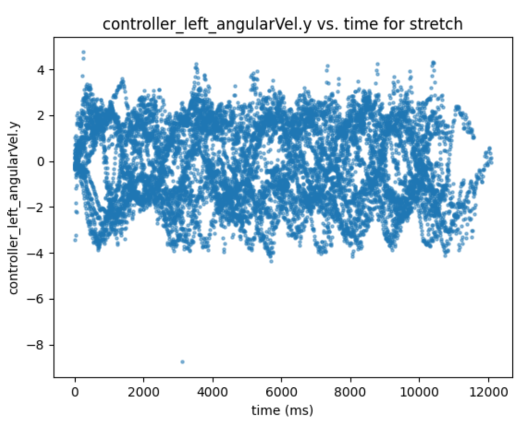
Arms Stretching:

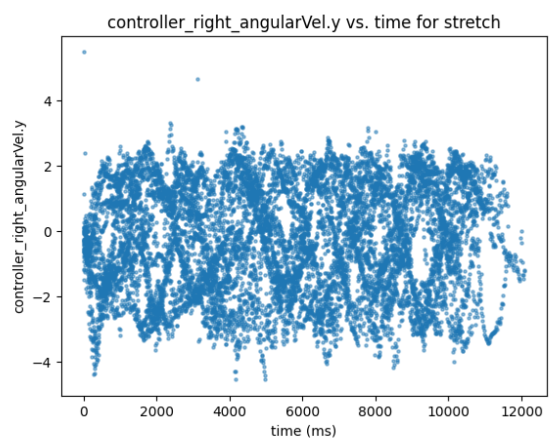






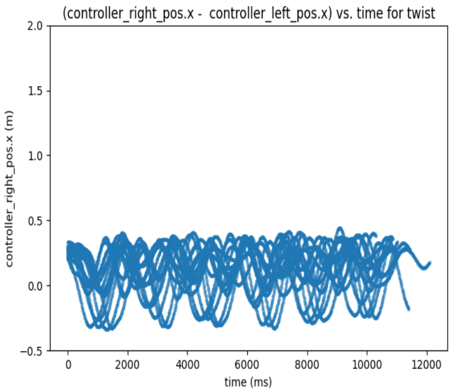


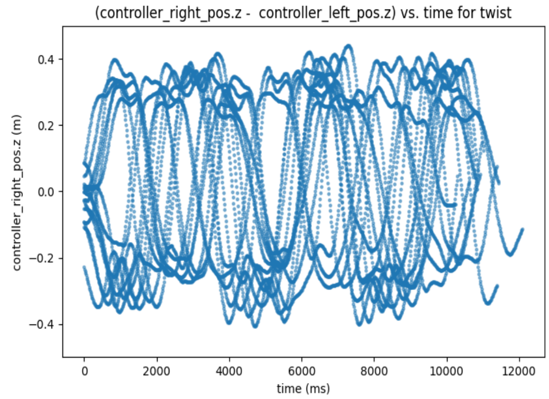


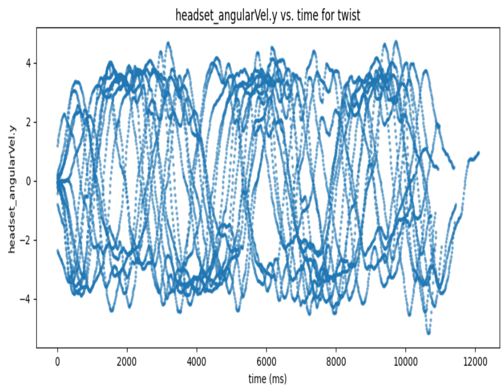


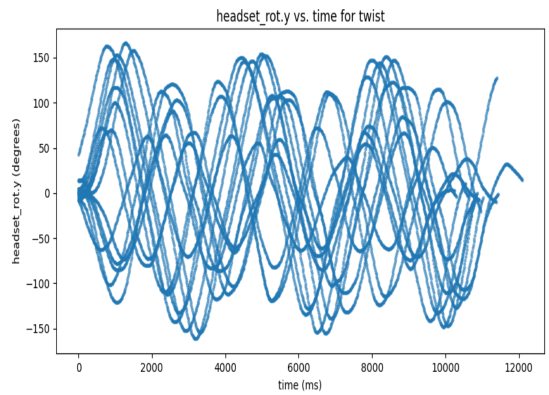
Arms stretching involves both arms moving in contrary motion, so hence the sinusoidal trends in the difference between their positions in X and Z. The headset does not move much at all when stretching, but this is a distinguishing factor to decipher it between this and twisting, so it is worth noting. The angular velocities also vary in a sinusoidal pattern as well, noted above.

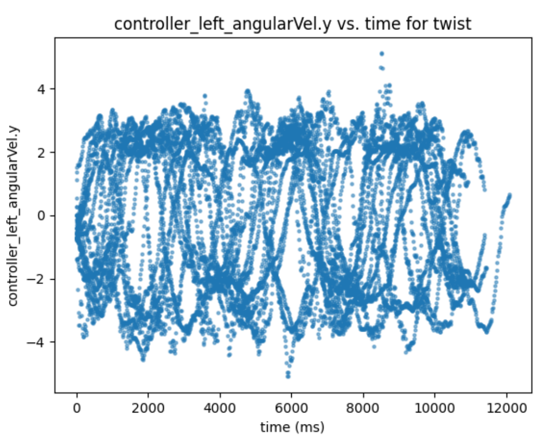
Twisting:

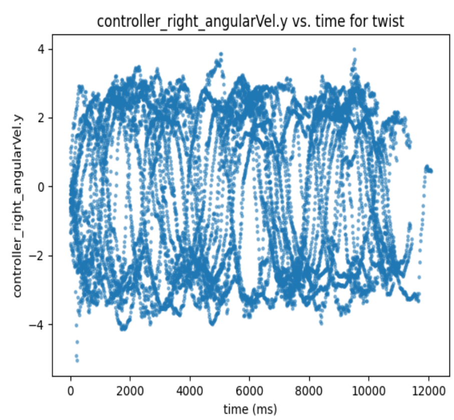










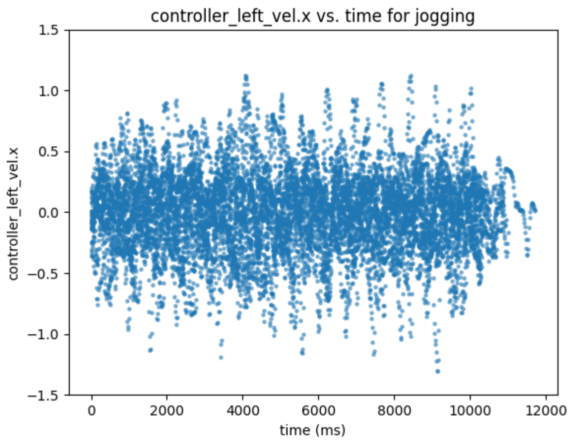


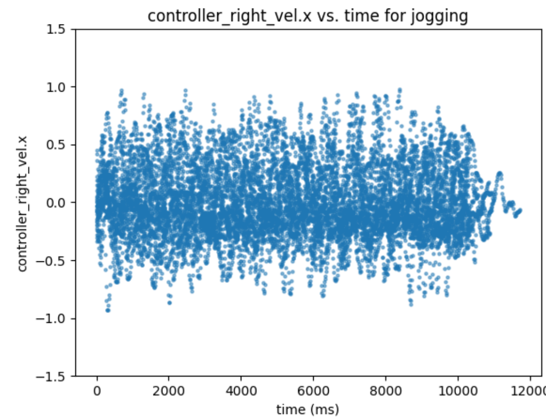
Twisting involves rigid movement between the arms and the head, so all the plots are sinusoidal. This is the only activity that involves the headset moving in the y direction a lot, so that is an important feature. Likewise for the angular velocities for the controllers. As stated above, the main difference between arms stretching and twisting is that the controllers move differently relative to one another, so the position difference graph is worth noting above.

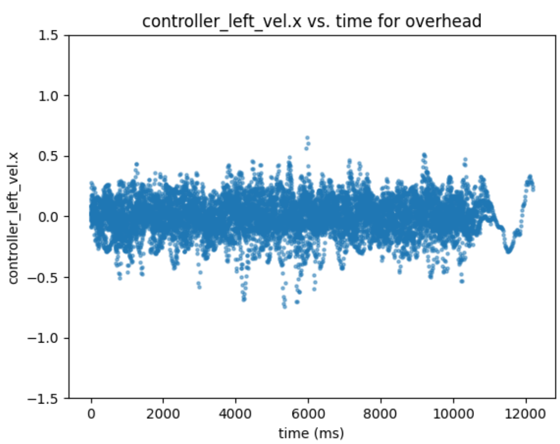
Jogging

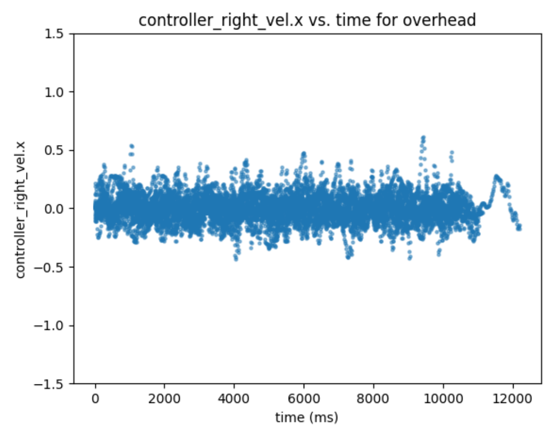
Comparisons in velocities for both controllers in x, y and z directions

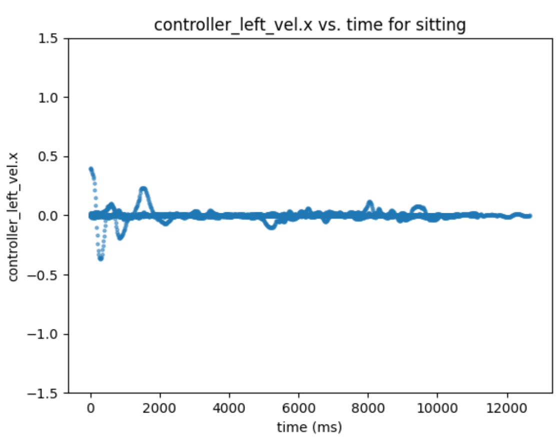
X direction

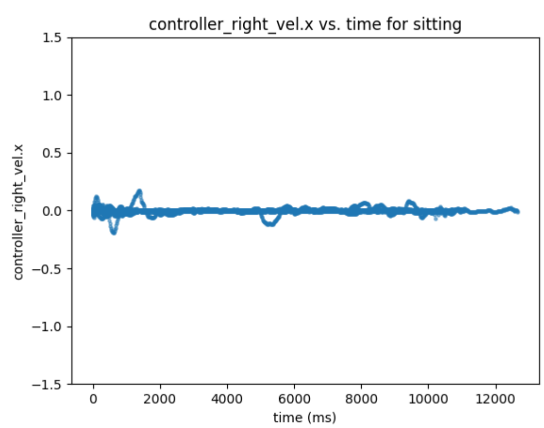




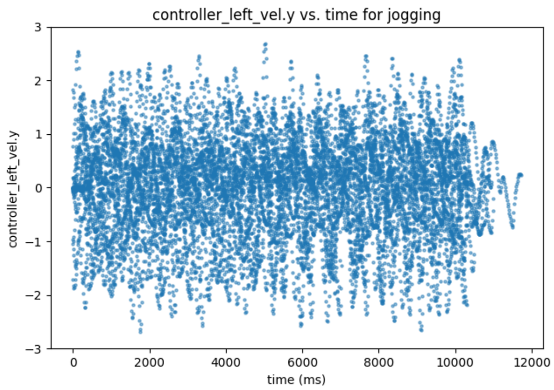


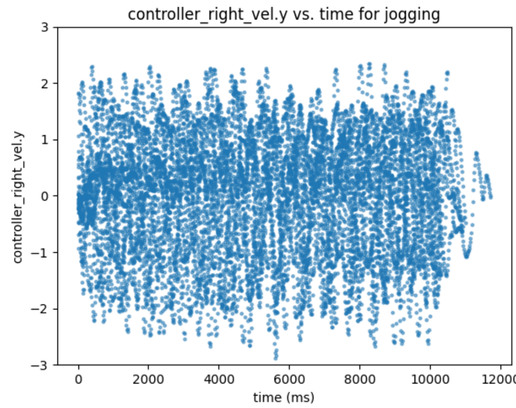


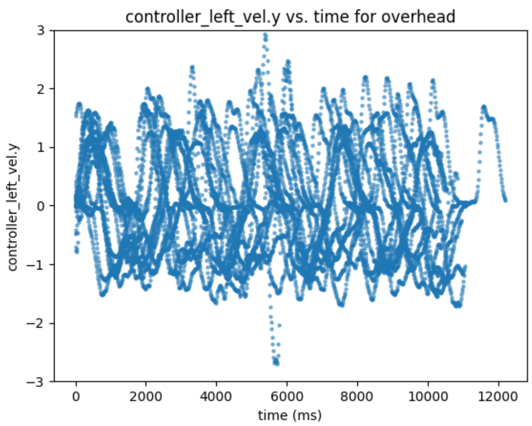


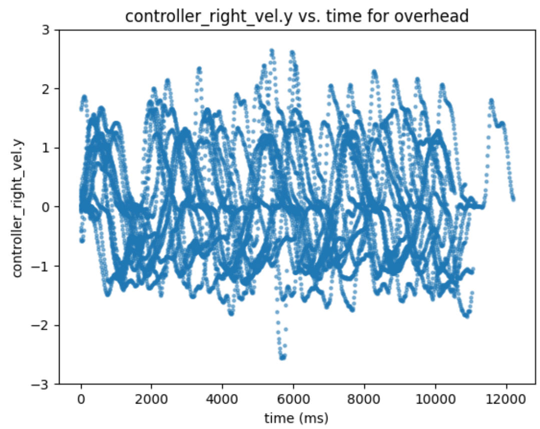


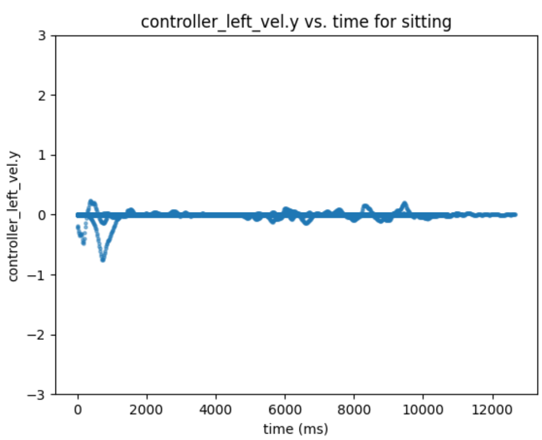
Y Direction

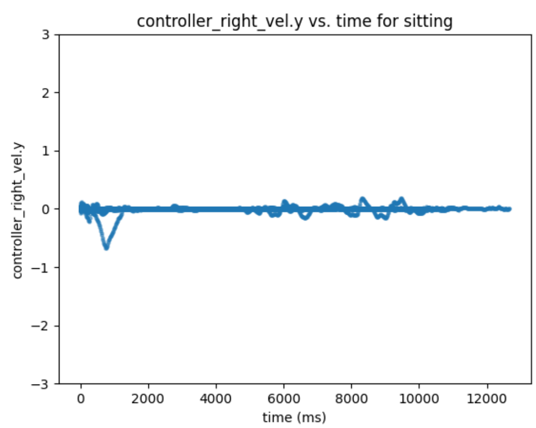

controller_left_vel.y vs. time for jogging
controller_left_vel.y
time (ms)


controller_right_vel.y vs. time for jogging
controller_right_vel.y
time (ms)

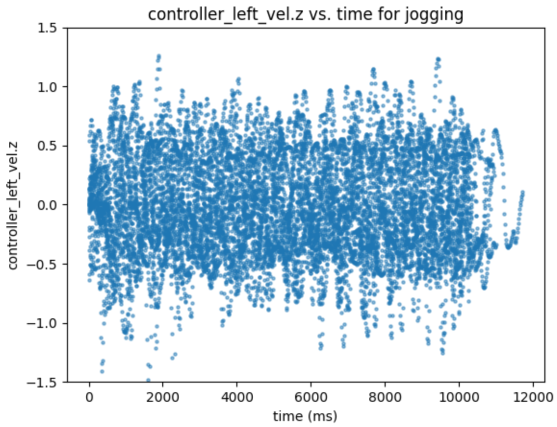

controller_left_vel.y vs. time for overhead
controller_left_vel.y
time (ms)

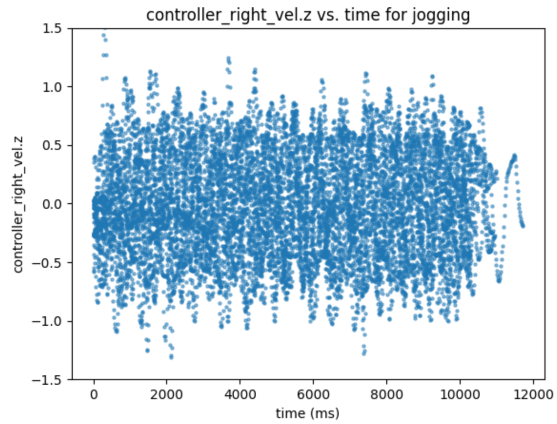

controller_right_vel.y vs. time for overhead
controller_right_vel.y
time (ms)

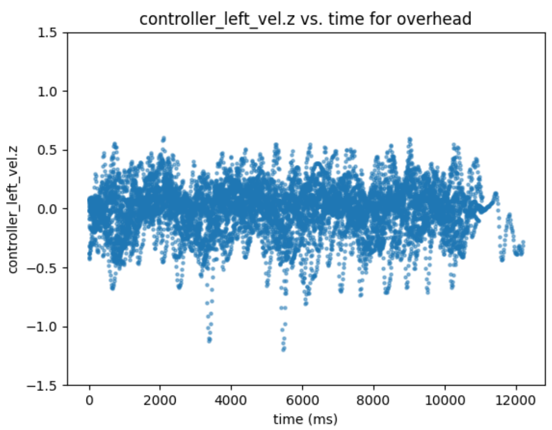

controller_left_vel.y vs. time for sitting
controller_left_vel.y
time (ms)

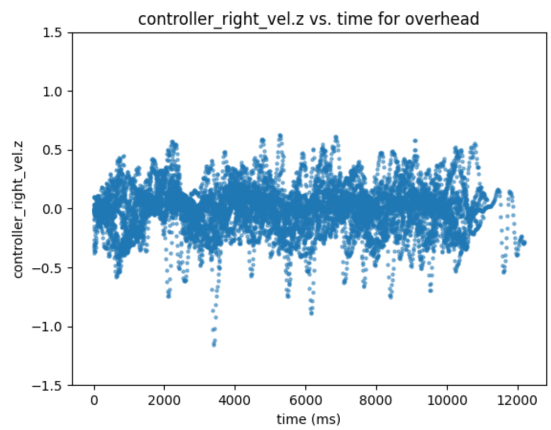

controller_right_vel.y vs. time for sitting
controller_right_vel.y
time (ms)

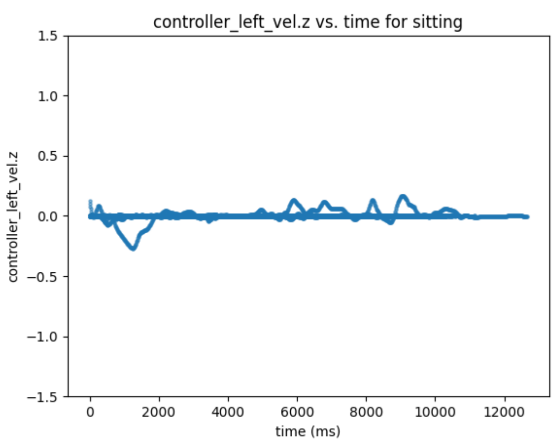
Z Direction

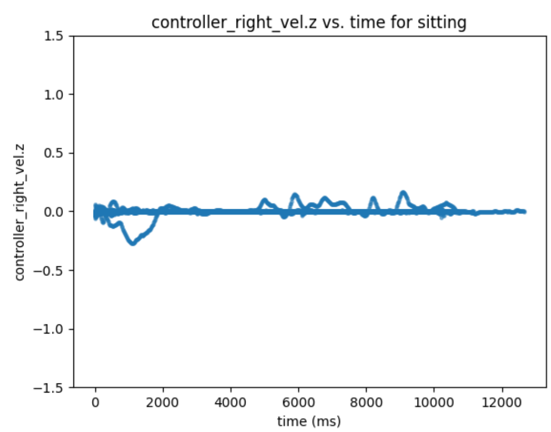






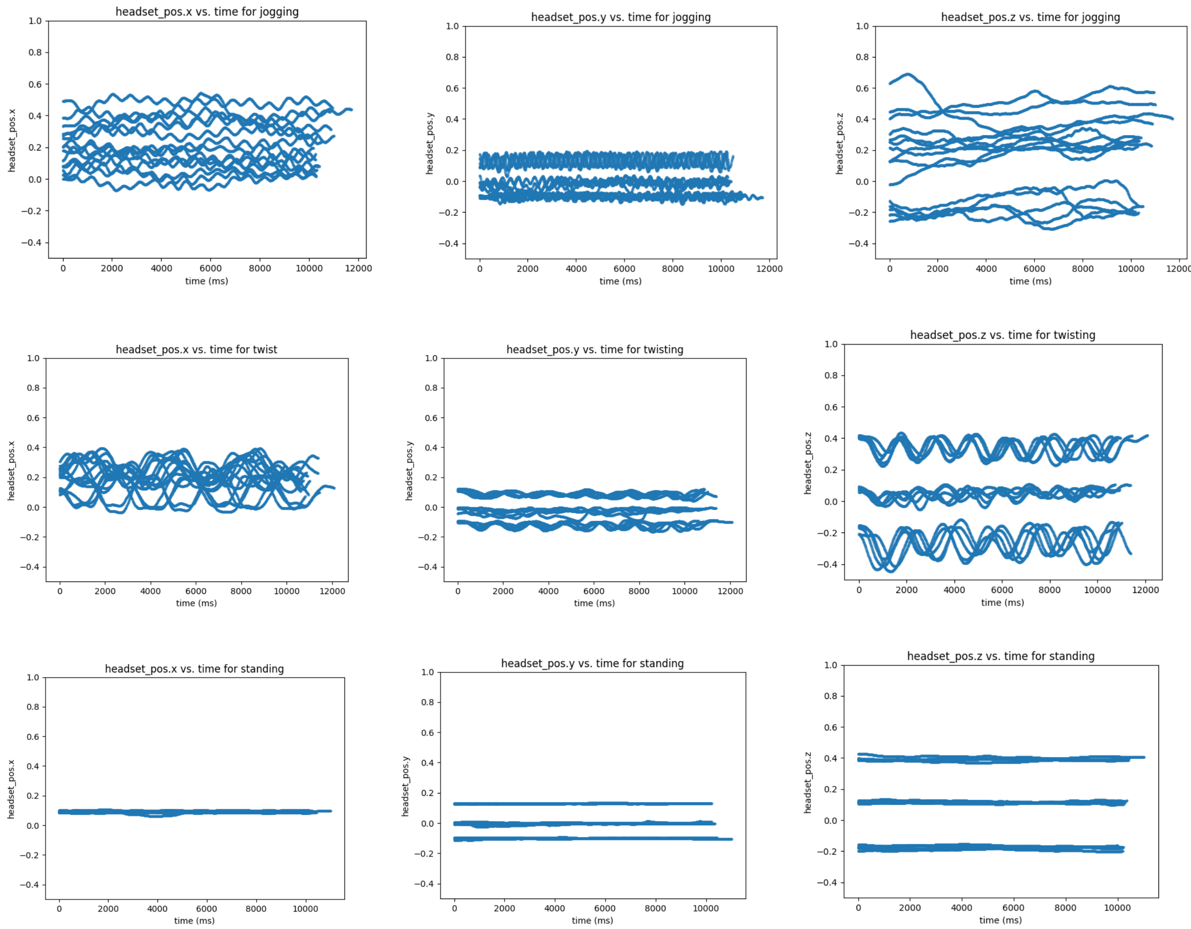






As seen by the graphs above, the velocity in all directions for jogging is a significant attribute because each direction has the largest range and is the most scattered. Compared to activities like overhead and sitting, jogging does not follow any statistical pattern for velocity. Jogging has a large variance compared to overhead and sitting.

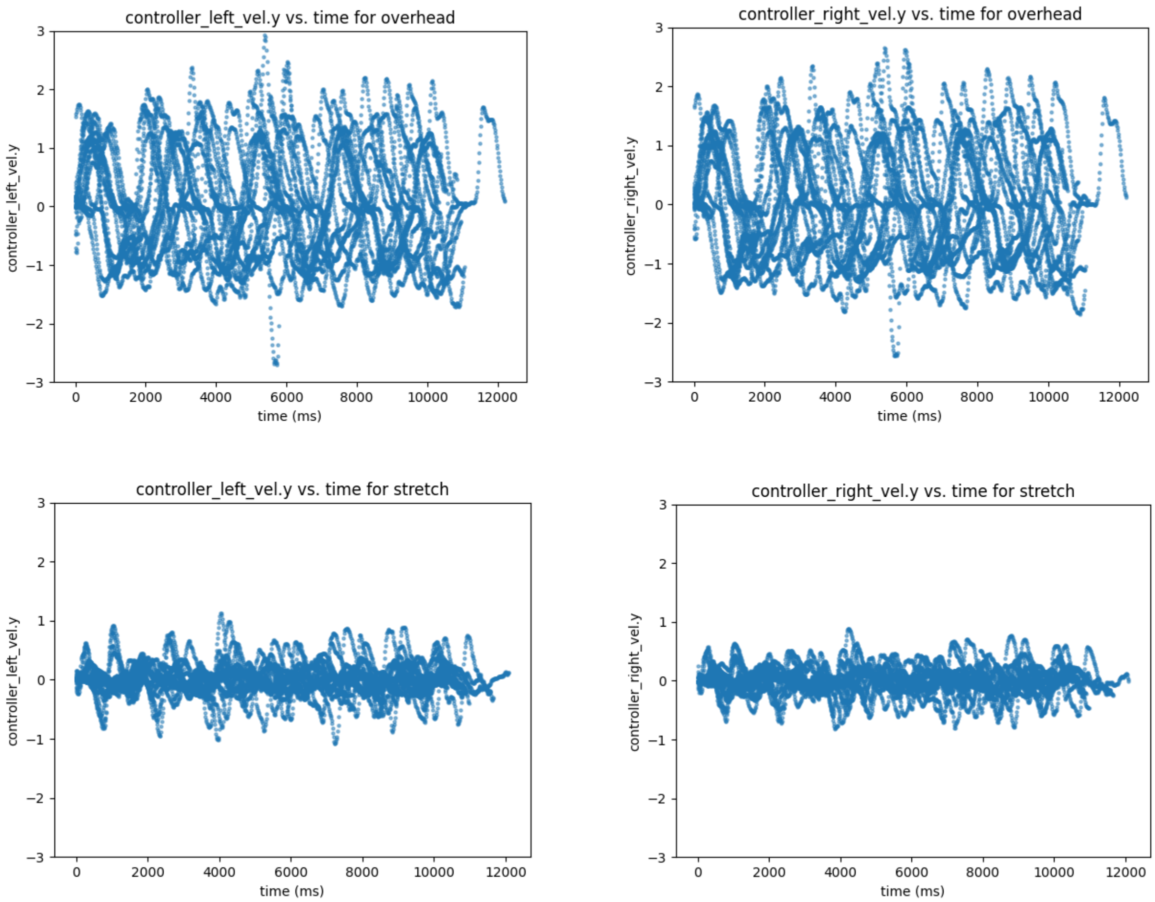
Headset Positions (x, y, z directions)



When comparing headset positions, we see different patterns in the data for each direction. This data is a compilation of all the participants data so for standing, this is why we see three different straight lines. We see that jogging begins to follow a pattern and notice the frequency of each axis direction is different depending on the activity. The headset position in the y-direction has the largest frequency and this can be used to determine if we are jogging versus twisting. We see that for jogging, the z-direction does not follow a clear pattern but for twisting it does.

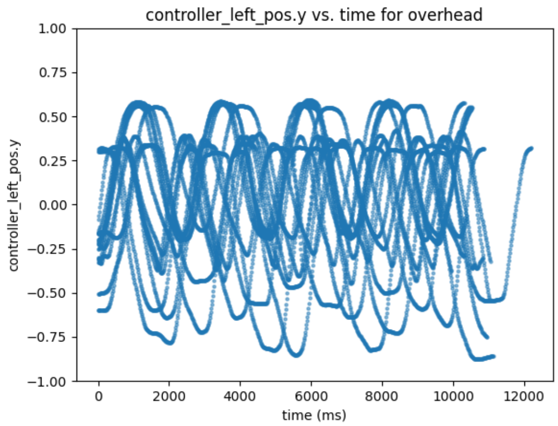
Overhead

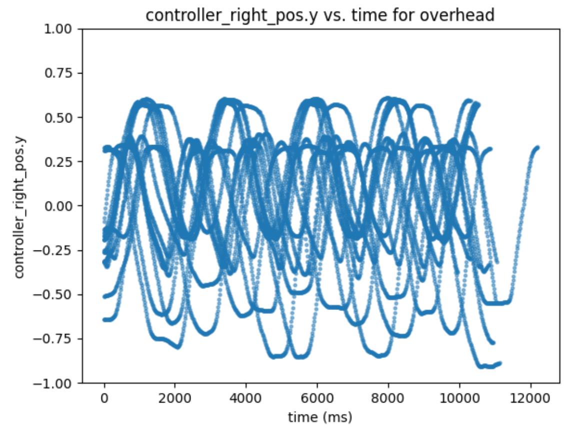
Y-direction Controller Velocity

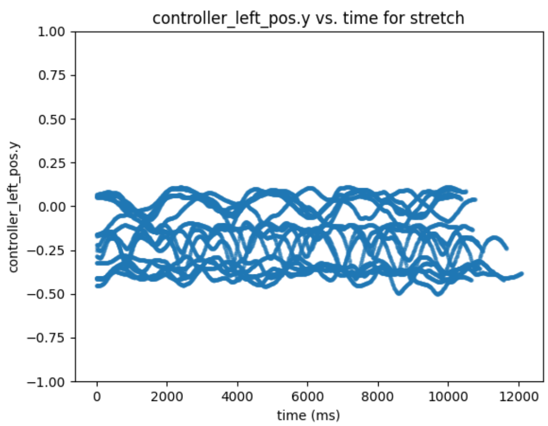


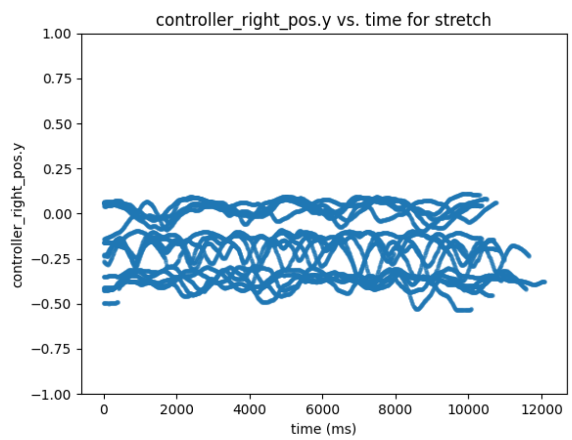
In regard to the velocity in the y-direction, we find for overhead that the amplitude of the graph is larger than that of stretch. The wavelength appears to also be slightly shorter than stretching but we find that the amplitude is an important feature for the controller's y-direction velocities.

Y-direction Controller Position









Similar to the controller's y-direction velocity, the y-direction position has a high amplitude and follows a wavelength function. The wavelength is more defined in the position and the frequency is clearer. This makes sense based on the direction the overhead activity takes going from high to low constantly. As opposed to stretch, it does not move much in the y-direction or shows a definite pattern.

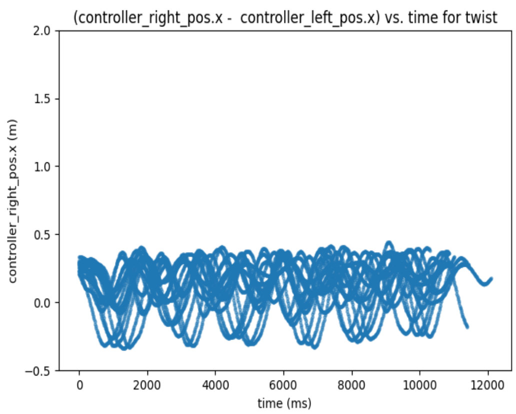
6. Using the significant attributes found above, describe how you would design and implement a simple algorithm (e.g. using statistical thresholds) to determine which activity was performed for a given sensor trace. Your algorithm should work regardless of how long a data sample is.

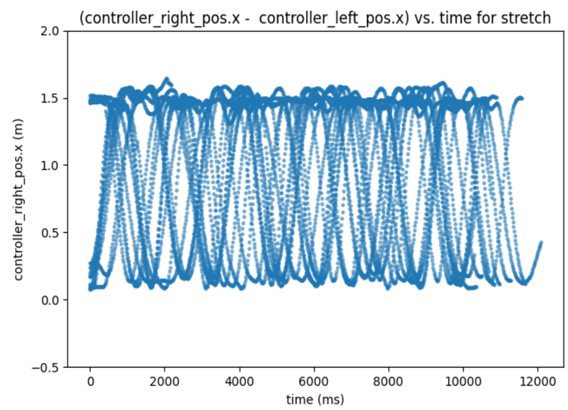
```
#the dictionary would contain a list of attributes that we have deemed to be important for each activity and each element in the dictionary would have a ky of the activity the attribute is associated with and the mean and variance we have calculated using our sample data
Imprt_attrbt = Dictionary of important attributes compiled from our list above with mean, variance, and activity keys
#global variables to store differences in for later
Temp_x = 0
Temp_y = 0
For attribute in list of attributes of an activity: #goes through each attribute of the activity a user just recorded
        If attribute is in Imp_att:#checks if the attribute is one of the attributes we think it's important
                Calculate mean and variance of the attribute
                New_dict = Store to a dictionary with attribute, mean and variance
                #store in a new dictionary this attribute along with its mean and variance
For each new_key in New_dict:# for each of the keys we just stored
        For key in imp_att: #we will see how close of a difference that the avg and var of the attribute we just calculated is to the avg and var of our sample data
                If key.attribute == new_key.attribute:
                        X = New_key.mean - key.mean and Y = New_key.var - key.var
                        If temp_x > |X| :
                                 temp_x = X
                                Closest_activity_according_to_x = key.activity
                        If temp_y > |Y|
                                Temp_y = Y
                                Closest activity_according_to_y = key.activity

# so if mean and var have the closest difference for the same activity then that means we have the best possible prediction of the activity we are tracing
If Closest_activity_according_to_x  == Closest activity_according_to_y:
        Most probable activity = Closest_activity_according_to_x
        Return Most probable activity
#if not we will go with the activity that has the smallest difference to the sample data
Else:
        If temp_y < temp_x:
                Return Closest activity_according_to_y
        Else:
                Return Closest_activity_according_to_x
```

7. What additional features you could derive from the provided raw data that would be helpful for distinguishing between activity types? Compute at least one feature, then graph it and show how it can be used to distinguish between two different activities. (hint: think of attributes used in physics that can be derived from the raw data)

For some activities, the motions for controllers are very similar and hard to distinguish the different activities. The main example is stretching vs. twisting. Angular velocities and velocities are very similar between them. However, we can use the fact that the arms stretch outwards for stretching while the arms are rigid in twisting to our advantage. By using the difference between the positions and rotations, we note that it is very close to 0 for twisting and is sinusoidal for stretching. This fact can also be applied to other activities, as the arms are very close together for overhead movement and follow a similar sinusoidal trend in the Y direction. For standing or sitting, movement between controllers is minimal. Below are some examples.





8. Examine the data traces between group members for the same activity. Do you think individual group members can be identified by their data traces? Why/why not? What implications does this have on the security of VR systems?

Yes, individual group members can be identified by the data traces. For example, the headset position in the Y direction for standing gives a user's height, likewise for jogging. For stretching, looking at the maximum distance between the controllers can give a user's maximum arm width which can also contribute to height. This is also the case for overhead stretching. On security for VR systems, as we saw in "Personal identifiability of user tracking data during observation of 360-degree VR video, Nature 2020," this becomes a problem when applications require you to sign up and provide some information about your body to better improve your experience. For example, in the VR game "Beat Saber," you need to provide a height so that you will be able to hit the blocks coming at you without having to stretch up or bend down to hit them. Analyzing the average height at which you stand and how your arms move in the app can allow for data harvesting from other applications you use and can trace it back to you. In "Personal identifiability of user tracking data during observation of 360-degree VR video, Nature 2020," they were able to identify a user with up to 95% accuracy based on just a very short amount of time in a VR setting. With this, if someone were to look at the data we studied in this lab, there is a strong possibility the data can correlate back to the user which presents issues of user privacy. If someone were to misplace their VR headset, this can cause personal information to be released beyond information left on the device. This can lead to impersonation as the data presents information about the user.

9. **Individual Contributions**

Devon:

I worked with everyone to make the basic functions to get means and variance and plot specific attributes with respect to time. I also made helper functions to process csv files as well as concatenating them. Using these functions, I focused on plotting the data traces with arms stretching and twisting. I also made tables for these for means and variance. I also described the statistical trends in these plots. I provided an idea for an additional feature for part 7 and wrote the first half of part 8 with an example from a previous paper we wrote.

Isaaq:

I worked on questions 1 and 2 with the group as we all coded together. I assisted in setting up the VR headset and collecting the data for our lab. From there, I uploaded it to the repository. Using the functions created in questions 1 and 2, I plotted data for the jogging and overhead activities and provided analysis using the tables for question 5. I worked on the second half of question 8 describing the implications of security in VR systems. To show what I completed, I created a separate branch in the repository so that nothing conflicted with the main code that Devon pushed.

Beza:

I also helped with writing up the code mostly for summarizing the code part on question 1. I plotted the data for sitting and standing as well as analyze the important attributes and compare them with other attributes for other activities. I analyzed the attributes of sitting and standing by summarizing them onto a table. I helped by giving ideas and comments. I also talked through with everyone and wrote the pseudo code for the new activity tracing using statistical thresholds.