**Lab 4-2: OptiTrack Signature/Drawing Exercise**

**Introduction:**

For this lab, we used Optitrack to examine fine motor control differences between the dominant and non-dominant hand in a series of writing tasks, in order to test Fitt’s Law.

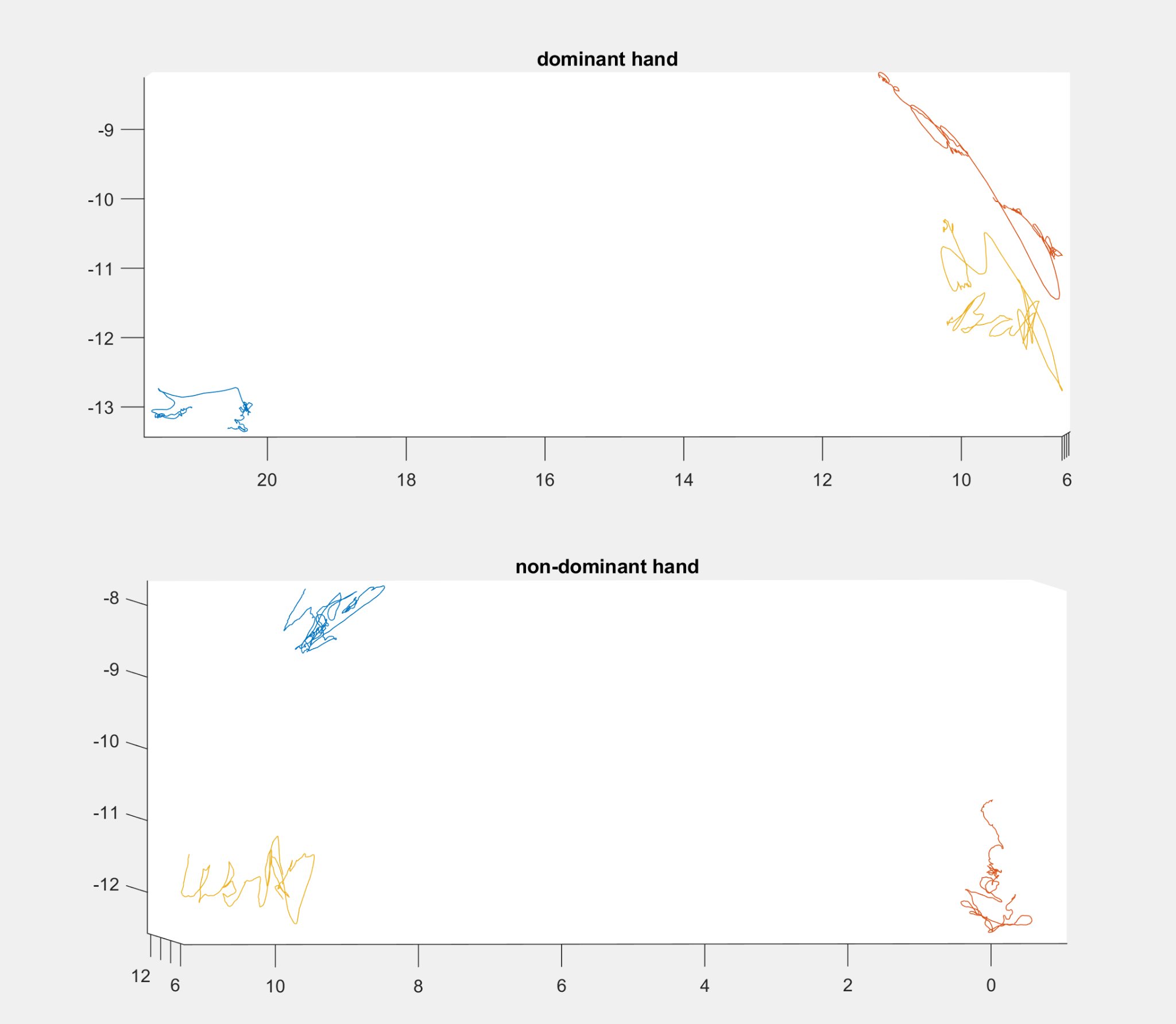
**Materials:**

* Optitrack Motion Capture System
* 3 Optitrack Markers
* 1 Pen
* Medical Tape

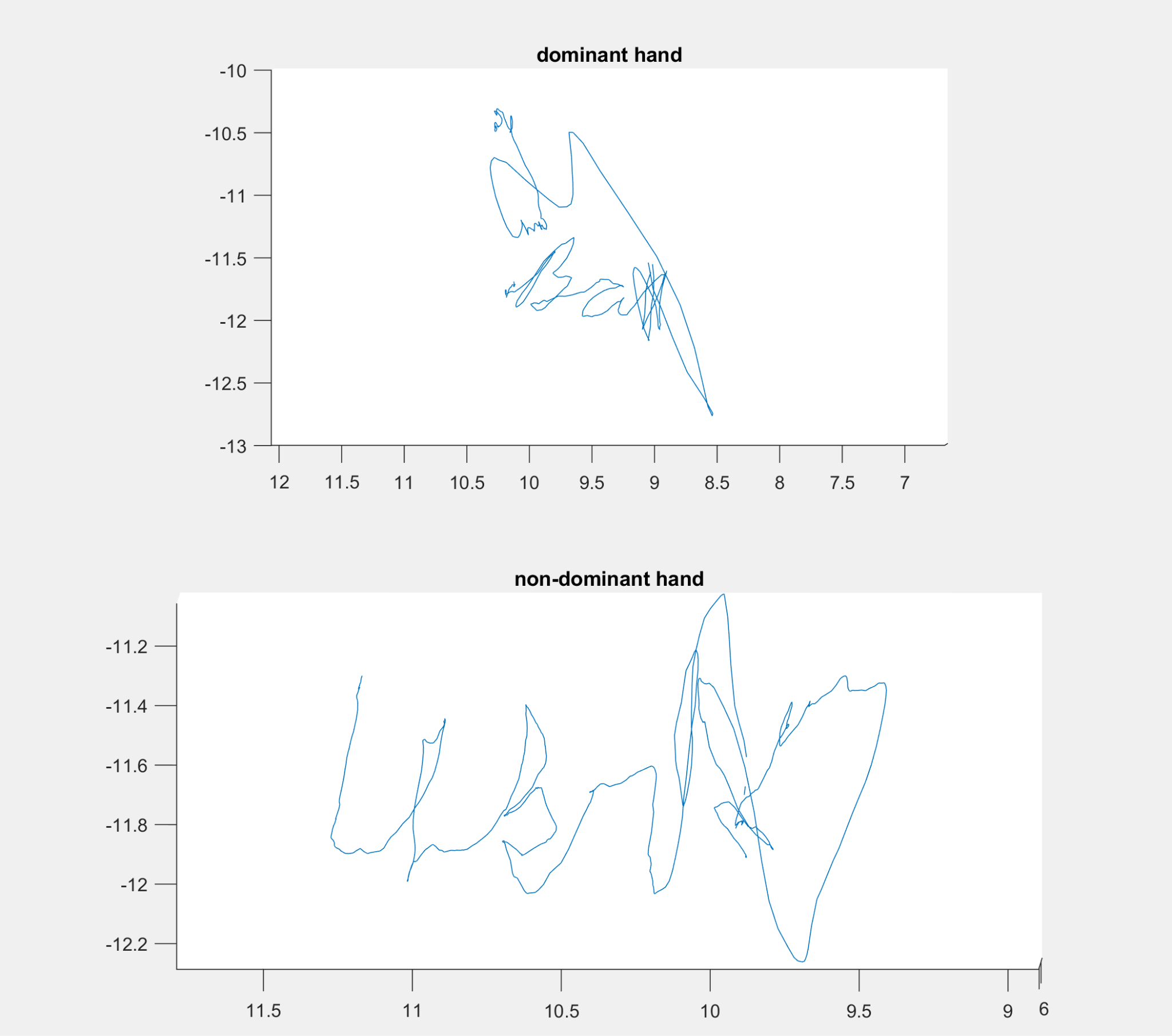
**Methods:**

**Part 1: Optitrack Signature Exercise**

For this part of the lab, we attached Optitrack reflective markers to the pen, the first knuckle of the thumb and the wrist, using medical tape as necessary to secure them. We then recorded tracing our signatures onto the piece of paper attached to the table. This was first performed with the dominant hand, then non-dominant.

****

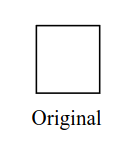
**Figure 1.** The signature, BrX, in yellow. It can be seen that the nondominant hand’s signature is less clear than that of the dominant hand.



**Figure 2.** The isolated signature for both the dominant and nondominant

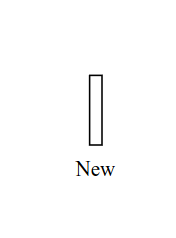
**Part 2: Dominant versus non-dominant arm: comparing the accuracy of reaching**

For this part of the lab, we attached the Optitrack sensors to the pen tip, first knuckle of the thumb, and wrist. We then drew two square targets approximately 3 inches apart (Figure 1).



**Figure 1.** The original target size.

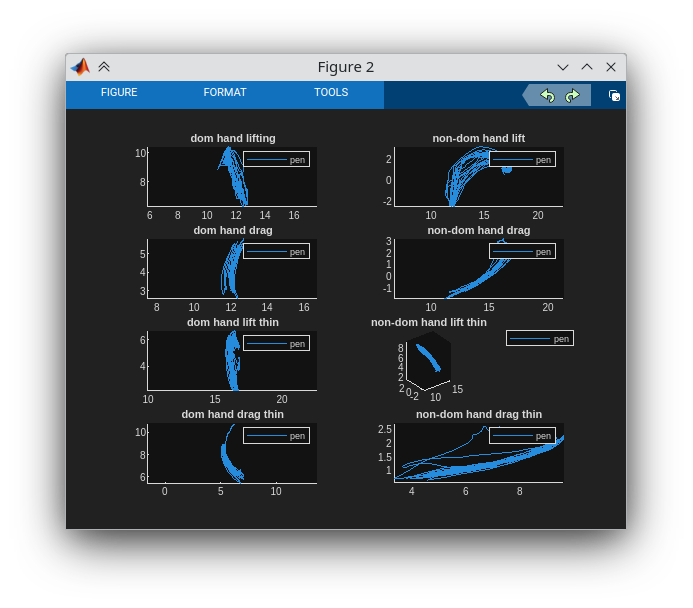
Markers were placed on the paper so that we were able to get the coordinates of the targets. Before proceeding with the trials, the markers on the paper were removed. For task A, we moved the pen from the first target to the second target by lifting it, going back and forth as fast and accurately as possible, 10 times. For task B, we moved the pen from the first target to the second target, this time dragging the pen on the paper rather than lifting it. For task C, we drew narrow targets (Figure 2), and repeated task A and B with these new targets.



**Figure 2.** The new, narrow target.

Task D required us to repeat tasks A, B, and C with our non-dominant hands.

**Results:**

****

Calculate the mean and standard deviation of the target to target movement duration for

each task:

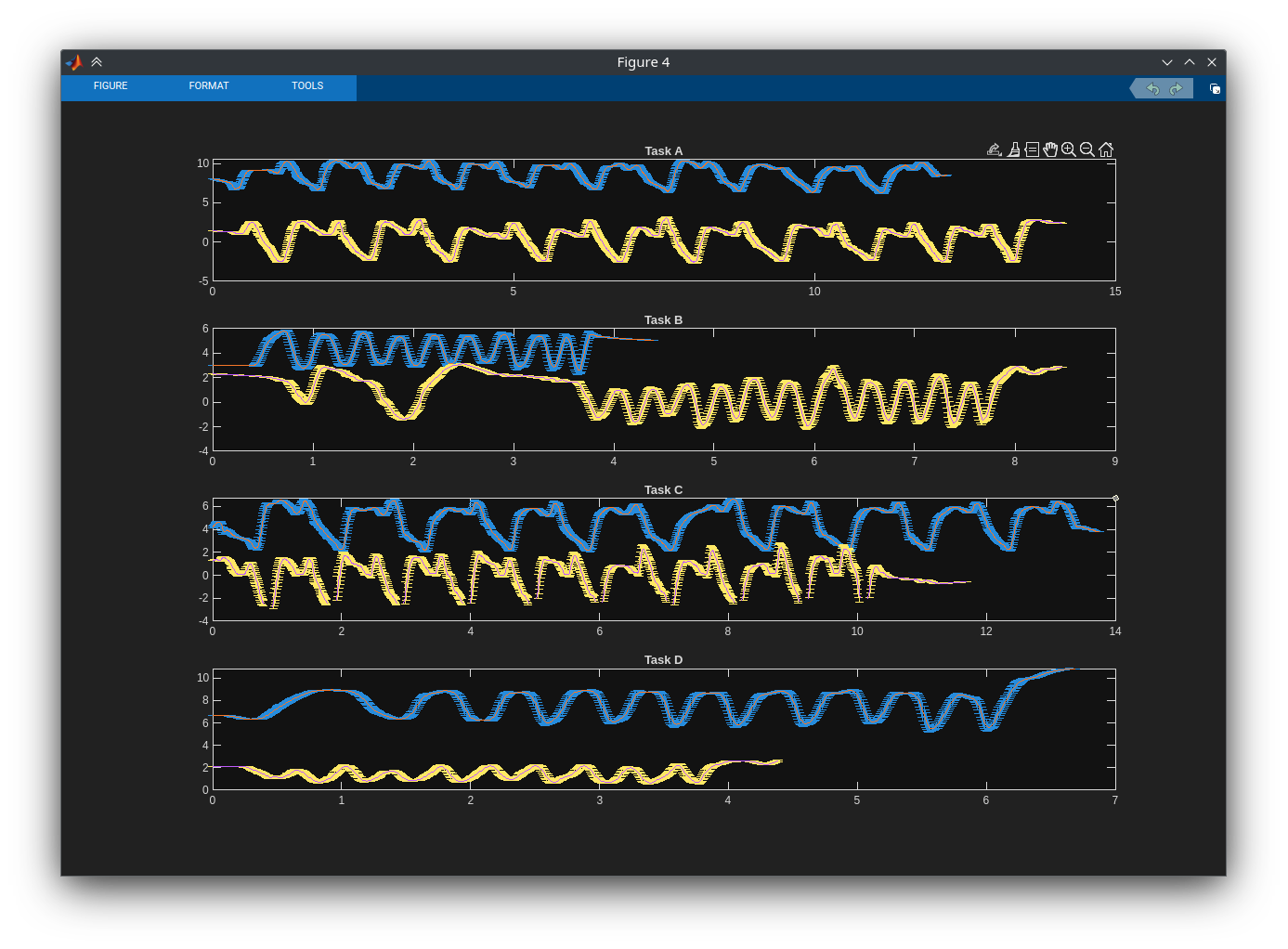
| **Dominant Hand** | Task A | Task B | Task C | Task D |
| --- | --- | --- | --- | --- |
| Mean (s) | 0.563596 | 0.171032 | 0.659649 | 0.262719 |
| Standard Deviation (s) | 0.117825 | 0.055253 | 0.115813 | 0.064929 |
| Spatial Variation (cm) | 2.539303 | 2.238740 | 3.064454 | 2.499342 |

| **Non-dominant Hand** | Task A | Task B | Task C | Task D |
| --- | --- | --- | --- | --- |
| Mean (s) | 0.685648 | 0.176316 | 0.696154 | 0.191204 |
| Standard Deviation (s) | 0.108269 | 0.027814 | 0.219680 | 0.035294 |
| Spatial Variation (cm) | 1.337261 | 0.549296 | 1.832493 | 0.636030 |

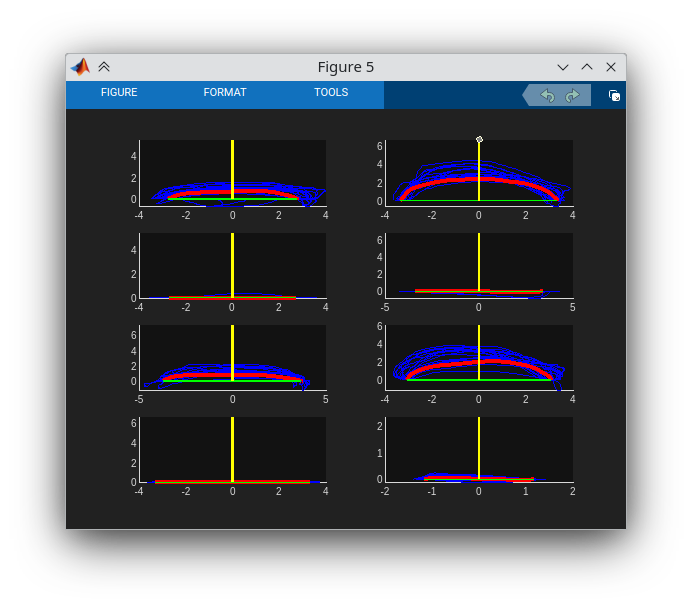
The dominant hand was faster and more accurate than the nondominant hand, except for the last case, because the nondominant hand was inaccurate in exchange for speed.



This figure illustrates the velocities over time, which were used to determine when the direction change occurred, meaning when the movement from one target to another changed. This allowed us to determine the amount of time taken for each movement.



This figure illustrates the means and standard deviations vs time for Tasks A through D.



This figure shows the subject’s path in blue, with the mean path in red.

**Matlab code:**

**Main File:**

clc

clear vars

close all

%% Part 1 Stuff

% This is Bryan's signature with dominant hand (left)

figure(1)

subplot(2,1,1)

handcoords = readtable("bryan dominant.csv");

handcoords(1:4,:)=[]; % make top 4 rows empty

data=table2array(handcoords);

data = data\*100; % converts to centimeters

pen = [data(:,3),data(:,4),data(:,5)];

plot3(pen(:,1),pen(:,2),pen(:,3)); % wrist

hold on

wrist = [data(:,6),data(:,7),data(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % thumb

thumb = [data(:,9),data(:,10),data(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % pen

legend("wrist","thumb","pen");

hold off

axis equal

subplot(2,1,2)

% This is Bryan's Signature with nondominant hand (right)

handcoords = readtable("bryan non-dominant.csv");

handcoords(1:4,:)=[]; % make top 4 rows empty

data=table2array(handcoords);

data = data\*100; % converts to centimeters

pen = [data(:,3),data(:,4),data(:,5)];

plot3(pen(:,1),pen(:,2),pen(:,3)); % thumb

hold on

wrist = [data(:,6),data(:,7),data(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % wrist

thumb = [data(:,9),data(:,10),data(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % pen

legend("thumb","wrist","pen");

hold off

axis equal

%% PART 2 STUFF

%% TASK A

% Lifting Dominant Hand

clc

global subplot\_position;

subplot\_position = 1;

% clear vars

% close all

liftcoords = readtable("jon dominant lifting.csv");

liftcoords(1:4,:)=[]; % make top 4 rows empty

liftdata = table2array(liftcoords);

time = liftdata(:,2);

liftdata = liftdata\*100; % converts to centimeters

pen = [liftdata(:,3),liftdata(:,4),liftdata(:,5)];

figure(2);

subplot(4,2,1)

plot3(pen(:,1),pen(:,2),pen(:,3)); % thumb

hold on

wrist = [liftdata(:,6),liftdata(:,7),liftdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % pen

thumb = [liftdata(:,9),liftdata(:,10),liftdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % wrist

legend("thumb","pen","wrist");

title('dom hand lifting')

hold off

axis equal

[meanTime, stdTime] = testy(wrist(:,2), time, 50, 0, 11.5, 'Lifting Domininant Hand', "Task A");

% Lifting NONDominant Hand

figure(2)

% clear vars

liftcoords = readtable("jon non dominant lifting.csv");

liftcoords(1:4,:)=[]; % make top 4 rows empty

liftdata = table2array(liftcoords);

time = liftdata(:,2);

liftdata = liftdata\*100; % converts to centimeters

pen = [liftdata(:,3),liftdata(:,4),liftdata(:,5)];

subplot(4,2,2)

plot3(pen(:,1),pen(:,2),pen(:,3)); % wrist

hold on

wrist = [liftdata(:,6),liftdata(:,7),liftdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % thumb

thumb = [liftdata(:,9),liftdata(:,10),liftdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % pen

legend("wrist","thumb","pen");

title('non-dom hand lift')

hold off

axis equal

[meanTime, stdTime] = testy(thumb(:,2), time, 140/2, 0.3, 13, 'Lifting NonDomininant Hand', "Task A");

%% TASK B

% Drag Dominant Hand

figure(2);

% clear vars

dragcoords = readtable("jon dominant dragg.csv");

dragcoords(1:4,:)=[]; % make top 4 rows empty

dragdata = table2array(dragcoords);

time = dragdata(:,2);

dragdata = dragdata\*100; % converts to centimeters

pen = [dragdata(:,3),dragdata(:,4),dragdata(:,5)];

subplot(4,2,3)

plot3(pen(:,1),pen(:,2),pen(:,3)); % thumb

hold on

wrist = [dragdata(:,6),dragdata(:,7),dragdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % pen

thumb = [dragdata(:,9),dragdata(:,10),dragdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % wrist

legend("thumb","pen","wrist");

title('dom hand drag')

hold off

axis equal

[meanTime, stdTime] = testy(wrist(:,2), time, 10, 0.5/2, 4, 'Drag Dominant Hand', "Task B");

% Drag NONDominant Hand

figure(2)

% clear vars

dragcoords = readtable("jon non dominant dragging.csv");

dragcoords(1:4,:)=[]; % make top 4 rows empty

dragdata = table2array(dragcoords);

time = dragdata(:,2);

dragdata = dragdata\*100; % converts to centimeters

pen = [dragdata(:,3),dragdata(:,4),dragdata(:,5)];

subplot(4,2,4)

plot3(pen(:,1),pen(:,2),pen(:,3)); % wrist

hold on

wrist = [dragdata(:,6),dragdata(:,7),dragdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % thumb

thumb = [dragdata(:,9),dragdata(:,10),dragdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % pen

legend("wrist","thumb","pen");

hold off

axis equal

[meanTime, stdTime] = testy(thumb(:,2), time, 20/2, 4, 7.4, 'Drag NonDominant Hand', "Task B");

%% TASK C

% Lifting Dominant Hand Thin

figure(2)

% clear vars

liftcoords = readtable("jon dominant lifting thin.csv");

liftcoords(1:4,:)=[]; % make top 4 rows empty

liftdata = table2array(liftcoords);

time = liftdata(:,2);

liftdata = liftdata\*100; % converts to centimeters

pen = [liftdata(:,3),liftdata(:,4),liftdata(:,5)];

subplot(4,2,5)

plot3(pen(:,1),pen(:,2),pen(:,3)); % thumb

hold on

wrist = [liftdata(:,6),liftdata(:,7),liftdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % pen

thumb = [liftdata(:,9),liftdata(:,10),liftdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % wrist

legend("thumb","pen","wrist");

title('dom hand lift thin')

hold off

axis equal

[meanTime, stdTime] = testy(wrist(:,2), time, 120/2, 1, 14, 'Lifting Dominant Hand Thin', "Task C");

% Lifting NONDominant Hand Thin

figure(2)

% clear vars

liftcoords = readtable("jon non dominant lifting thin.csv");

liftcoords(1:4,:)=[]; % make top 4 rows empty

liftdata = table2array(liftcoords);

time = liftdata(:,2);

liftdata = liftdata\*100; % converts to centimeters

pen = [liftdata(:,3),liftdata(:,4),liftdata(:,5)];

subplot(4,2,6)

plot3(pen(:,1),pen(:,2),pen(:,3)); % pen

hold on

wrist = [liftdata(:,6),liftdata(:,7),liftdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % thumb

thumb = [liftdata(:,9),liftdata(:,10),liftdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % wrist

legend("pen","thumb","wrist");

hold off

axis equal

[meanTime, stdTime] = testy(pen(:,2), time, 50, 0.3, 10, 'Lifting NonDominant Hand Thin', "Task C");

%% TASK D

% Drag Dominant Hand Thin

figure(2);

% clear vars

dragcoords = readtable("jon dominant dragg thin.csv");

dragcoords(1:4,:)=[]; % make top 4 rows empty

dragdata = table2array(dragcoords);

time = dragdata(:,2);

dragdata = dragdata\*100; % converts to centimeters

pen = [dragdata(:,3),dragdata(:,4),dragdata(:,5)];

subplot(4,2,7)

plot3(pen(:,1),pen(:,2),pen(:,3)); % thumb

hold on

wrist = [dragdata(:,6),dragdata(:,7),dragdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % pen

thumb = [dragdata(:,9),dragdata(:,10),dragdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % wrist

legend("pen","thumb","wrist");

title('dom hand drag thin')

hold off

axis equal

[meanTime, stdTime] = testy(wrist(:,2), time, 40/2, 1, 6.2, 'Drag Dominant Hand Thin', "Task D");

% Drag NONDominant Hand Thin

figure(2)

%clear vars

dragcoords = readtable("jon non dominant dragg thin.csv");

dragcoords(1:4,:)=[]; % make top 4 rows empty

dragdata = table2array(dragcoords);

time = dragdata(:,2);

dragdata = dragdata\*100; % converts to centimeters

pen = [dragdata(:,3),dragdata(:,4),dragdata(:,5)];

subplot(4,2,8)

plot3(pen(:,1),pen(:,2),pen(:,3)); % thumb

hold on

wrist = [dragdata(:,6),dragdata(:,7),dragdata(:,8)];

plot3(wrist(:,1),wrist(:,2),wrist(:,3)); % pen

thumb = [dragdata(:,9),dragdata(:,10),dragdata(:,11)];

plot3(thumb(:,1),thumb(:,2),thumb(:,3)); % wrist

legend("pen","thumb","wrist");

hold off

axis equal

[meanTime, stdTime] = testy(wrist(:,2), time, 17, 0.15, 3.7, 'Drag Dominant Hand Thin', "Task D"); % using 17

%instead of 17.5 here because 35/2 causes an error (can't have half a frame)

**Testy function:**

function [domliftmean, domliftstd, ymean, zmean, ys, zs, turningPoints, dy] = testy(Position, time, minDelay, timeMin, timeMax, descriptor, taskname, Position2)

%for one of our trials, out position was filled with part with Data that

%wasnt picked up, so interpolation was needed to fill in the gaps

emptydata = isnan(Position);

if any(emptydata)

validdata = ~emptydata;

validTime = time(validdata);

validPosition = Position(validdata);

filledPosition = interp1(validTime, validPosition, time, 'pchip', 'extrap');

Position = filledPosition;

end

dy = diff(Position); % Change in Z

dt = diff(time); % Change in time

global subplot\_position; % loads the variable for the subplot position

% Calculate Y-velocity

vy = dy ./ dt;

turningPoints = [];

i = 1;

while i <= length(vy) - 1

% Check if current time is within desired range before evaluating vy

if time(i) > timeMin && time(i) < timeMax

if (vy(i) <= 0 && vy(i + 1) > 0)||(vy(i) >= 0 && vy(i + 1) < 0) % Condition for turning point

turningPoints = [turningPoints; i + 1]; % Collect turning point

i = i + minDelay; %Search for turning Points in set interval

continue;

end

end

i = i + 1; % Otherwise, proceed to the next index

end

% Extract every other value

everyOther = Position(turningPoints(1:2:end));

everyOther2 = Position2(turningPoints(1:2:end));

% Calculate the average

average = mean(everyOther);

average2 = mean(everyOther2);

% Extract every other value starting from the second element

everyOtherOpposite = flip(Position(turningPoints(2:2:end)));

everyOtherOpposite2 = flip(Position2(turningPoints(2:2:end)));

% Calculate the average

averageOpposite = mean(everyOtherOpposite);

averageOpposite2 = mean(everyOtherOpposite2);

TY = [average, average2];

TZ = [averageOpposite, averageOpposite2];

% turningPointsmean = plot(TY,TZ,'Color','g');

% Calculate the slope of the original line

slope = (TZ(2) - TZ(1)) / (TY(2) - TY(1));

% Calculate the slope of the perpendicular line

slope\_perp = -1 / slope;

% find perpendicular line

Yint = (TZ(2)-TZ(1))/2 - slope\_perp \* (TY(2)-TY(1))/2;

timeForVy = time(2:end); % Drop the first time value

% Plot Y-velocity over time

figure(3);

subplot(4,2,subplot\_position)

plot(timeForVy, vy, 'LineWidth', 2);

xlabel('Time');

ylabel('Velocity in Y');

title('Y-Velocity Over Time');

hold on;

% Highlight turning points on the plot

figure(3);

subplot(4,2,subplot\_position)

scatter(timeForVy(turningPoints), vy(turningPoints), 'r', 'filled');

turningPointsX = timeForVy(turningPoints(1:end));

turningPointsX = diff(turningPointsX);

domliftmean = mean(turningPointsX);

domliftstd = std(turningPointsX);

fprintf('Mean of time at turning points for %s: %f seconds\n', descriptor, domliftmean);

fprintf('Standard deviation of time at turning points for %s: %f seconds\n', descriptor, domliftstd);

% labeling

legend('Y-Velocity', 'Turning Points');

% moving mean

% figure(4);

% subplot(4,2,subplot\_position)

movingmean = movmean(Position,5);

% Labeling plot elements:

legend('error bars','moving mean');

% iterate subplot\_position

subplot\_position = subplot\_position + 1;

% Plots moving standard deviation

figure(4);

subplot(4,1,floorDiv(subplot\_position,2))

movingstd = movstd(Position,10);

errorbar(time,movingmean,movingstd)

hold on

plot(time,movingmean) % plotting moving mean on top so it's visible

title(taskname)

% midpointsPosition = [];

% for i = 1:length(turningPoints)-1

% startPoint = turningPoints(i);

% endPoint = turningPoints(i+1);

% midPoint = floor((startPoint + endPoint) / 2);

%

% midpointsPosition(end+1) = Position(midPoint);

%

% end

% for i = 1:numel(everyOtherOpposite)

% avgTime = floor(everyOtherOpposite(i) + everyOther(i)) / 2;

%

% % Calculate spatial distance

% spatialDistances(i) = Position2(avgTime) - mean(TZ);

% end

%plotting trials in appropiate subplot

figure(5);

subplot(4,2,subplot\_position-1)

hold on

%finds the max amount of data between turning points

sections = diff(turningPoints);

maxlength = max(sections);

ys = NaN(maxlength, length(turningPoints)-1);

zs = NaN(maxlength, length(turningPoints)-1);

%puts data between turning points into columns to find mean lines

for i = 1:length(turningPoints)-1

a = 1:sections(i);

b = linspace(1, sections(i), maxlength);

ySection = Position(turningPoints(i):turningPoints(i+1)-1);

zSection = Position2(turningPoints(i):turningPoints(i+1)-1);

% if odd, puts the data normally, if even, puts data in reverse, so

% mean calculation is possible

if mod(i,2) == 1

%interpolates data so all data points are the same size

ys(:,i) = interp1(a, ySection, b, 'pchip', 'extrap');

zs(:,i) = interp1(a, zSection, b, 'pchip', 'extrap');

else

yflipped = flip(ySection);

zflipped = flip(zSection);

ys(:,i) = interp1(a, yflipped, b, 'pchip', 'extrap');

zs(:,i) = interp1(a, zflipped, b, 'pchip', 'extrap');

end

end

% Assume TY, TZ define the reference line

slope = (TZ(2) - TZ(1)) / (TY(2) - TY(1));

intercept = TZ(1) - slope \* TY(1);

numColumns = size(ys, 2); % Number of columns in ys and zs

allDistances = [];

for col = 1:numColumns

for row = 1:size(ys, 1)

x0 = ys(row, col); % x-coordinate of the point

y0 = zs(row, col); % y-coordinate of the point

% Skip NaN values

% Distance from point to line

distance = abs(-slope \* x0 + 1 \* y0 - intercept) / sqrt(slope^2 + 1^2);

allDistances = [allDistances, distance]; % Append distance

end

end

% calculate the standard deviation of all distances

SpacialVar = std(allDistances);

fprintf('Spatial Variation: %f\n', SpacialVar);

x = linspace(TY(1),TY(2),maxlength);

Perp = slope\_perp\*x+Yint;

Perp = Perp';

% Perpen = plot(x,Perp,"Color",'y');

%zeros the points

for i = 1:size(ys, 2)

yzeroed = ys(:,i) - ys(1,i);

zzeroed = zs(:,i) - zs(1,i);

ys(:,i) = yzeroed;

zs(:,i) = zzeroed;

end

%rotating the polot

for i = 1:size(ys,2)

ySection = ys(:,i);

zSection = zs(:,i);

diffy = ySection(end) - ySection(1);

diffz = zSection(end) - zSection(1);

angle = atan2(diffz, diffy);

rotation = -rad2deg(angle) + 180;

midPoint = [(ySection(1) + ySection(end))/2, (zSection(1) + zSection(end))/2];

ySection = ySection - midPoint(1);

zSection = zSection - midPoint(2);

h = plot(ySection, zSection, 'b');

rotate(h, [0 0 1], rotation, [0 0 0]);

end

%rotating the mean line

ymean = mean(ys, 2);

zmean = mean(zs, 2);

mdiffy = ymean(end) - ymean(1);

mdiffz = zmean(end) - zmean(1);

angle = atan2(mdiffz, mdiffy);

rotation = -rad2deg(angle) + 180;

midPoint = [(ymean(1) + ymean(end))/2, (zmean(1) + zmean(end))/2];

ymean = ymean - midPoint(1);

zmean = zmean - midPoint(2);

h = plot(ymean, zmean, 'LineWidth', 3, 'Color', 'r');

rotate(h, [0 0 1], rotation, [0 0 0]);

% TY = TY - TY(1);

% TZ = TZ - TZ(1);

TY = [mean(ys(1,:)), mean(ys(end,:))];

TZ = [mean(zs(1,:)), mean(zs(end,:))];

%rotating turningPoint line

mdiffy = TY(2) - TY(1);

mdiffz = TZ(2) - TZ(1);

angle = atan2(mdiffz, mdiffy);

rotation = -rad2deg(angle) + 180;

midPoint = [(TY(1) + TY(end))/2, (TZ(1) + TZ(end))/2];

TY = TY - midPoint(1);

TZ = TZ - midPoint(2);

h = plot(TY, TZ, 'LineWidth', 1, 'Color', 'g');

rotate(h, [0 0 1], rotation, [0 0 0]);

TY = TY - TY(1);

TZ = TZ - TZ(1);

%rotating Perp Line

% mdiffy = x(end) - x(1);

% mdiffz = Perp(end) - Perp(1);

% angle = atan2(mdiffz, mdiffy);

rotation = -rad2deg(angle) + 90;

% midPoint = [(x(1) + x(end))/2, (Perp(1) + Perp(end))/2];

% x = x - midPoint(1);

% Perp = Perp - midPoint(2);

h = plot(TY, TZ, 'LineWidth', 2, 'Color', 'y');

rotate(h, [0 0 1], rotation, [0 0 0]);

hold off

**Discussion and Conclusion:**

Overall, the results were mostly in line with our expectations; the dominant hand was faster overall, and generally accurate, but surprisingly, although our subject was right handed, their left hand exhibited superior performance in certain scenarios, and showed lower spatial variance compared to their right hand. This was likely due to the subject taking their time with the movements, which would lead to higher accuracy and more consistent results. Our results were slower with the thinner targets, which matches Fitt’s law, that as target size decreases, the difficulty of reaching the target is more difficult. Except for the very last nondominant hand’s trial, our results were consistent with established theories.