Ball Movement Trajectory

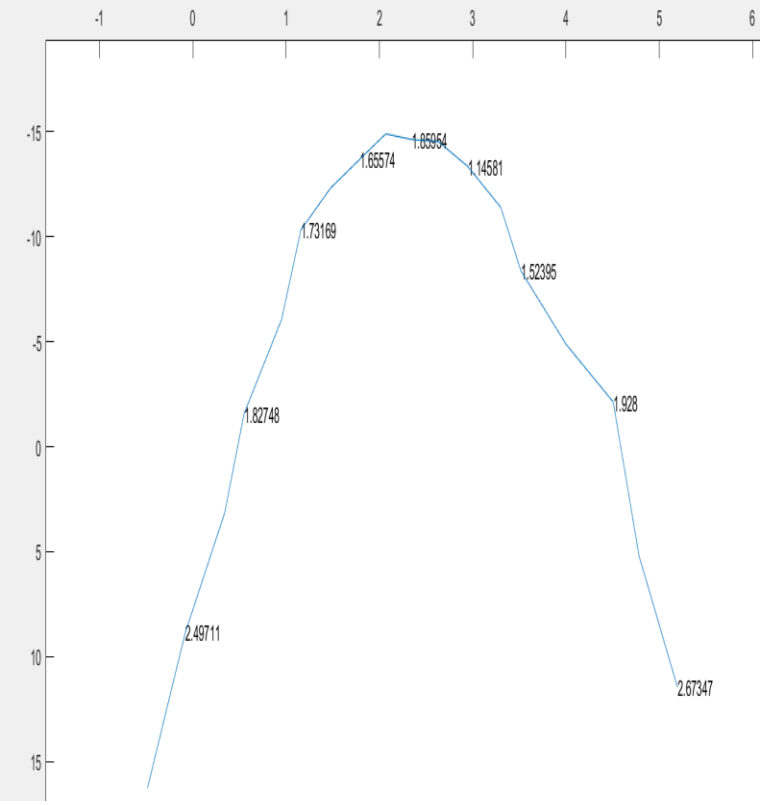
The Goal

In the previous week, I was to research on whether we can properly plot the 3D trajectory of the moving ping-pong ball, or not. I did this by considering two scenarios.

The first was by hitting the moving ball far away from the cameras in the direction towards the cameras. Based on the type of my hit to the ball, I expected to see a parabola graph on the X and Y axis'.

The second scenario was to drop the ping pong ball from my hand without any extra force, so that I would expect to see an almost straight line on the Y axis.

Scenario 1:

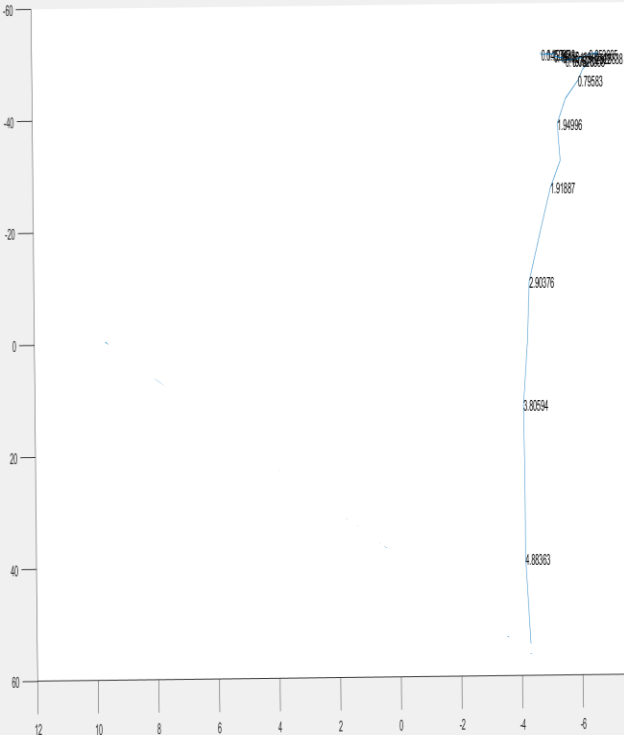


The Result

Considering the above two scenarios, I obtained the following results (note that in order to show speed, I had to show them as numbers on the movement plot).

In this scenario we can see that the trajectory on the X and Y axis is almost similar to how we expect it to be. The speed also seems to be identical to our expectations, Since it is decreasing as the ball rises, and increases as the ball is falling.

Scenario 2:



As can be seen in this scenario, the ball is moving in the Y axis with constantly increasing speed, in an almost straight line (we move less than half a centimeter on the X axis)

Conclusion

The conclusion which can be made is that we are able to get an almost accurate representation of the movement trajectory of the ball. Even though In some cases, there may be a slight noise in the ball's center location, which may impact the speed estimation to not be as close to reality as we expect it to be, this noise can be reduced with better lighting, better backgrounds, and noise removal techniques (which might be a plausible next step to work on).

As a side note, in the work I did for tracking the ball, I noticed that how having a good background for the ball, can make it much easier to track. For example, for the ping-pong ball which I painted blue, I noticed that having a darker background makes it easier to see the movement of the ball, which in turn, leads to less noise in the tracking that my code does.

Additional Work

Added the ability to be able to track speed in my code, even if we have missed frames.

For example, if we have the balls coordinates on frame 1, and the next frame we see the ball is frame 4, the speed will be calculated by finding the distance of the ball in the these two frames, and also considering that we moved 3 frames (from frame 1 to frame 4)