Movement Prediction

The Goal

In the previous week, I was able to get a proper representation of the ball's movement trajectory.

For this week, I thought that it would be interesting to predict the moving ball's trajectory by calculating the next predicted coordinates in the current location of the ball, and comparing the predicted coordinates to the actual coordinates and seeing how close together they are.

The Calculations

In the previous weeks, we used the distance between the coordinates of the balls throughout simultaneous frames to calculate speed. However, in order to find a prediction of the next point throughout the trajectory, we need to consider velocity(which is speed, also considering direction) for each of the X,Y,Z axis' separately, and then predict the coordinates of each of the axis' independently.

The two main formulas we'll be looking at are the following:

$$v = \frac{x_1 - x_0}{\triangle t}$$

$$x_2 = x_1 + v \times \triangle t$$

Now, considering we are taking shots at a constant 30FPS rate, we have the following(note that the deltaT's are the same for both):

$$v = \frac{x_1 - x_0}{\frac{1}{30}} = 30 \times (x_1 - x_0)$$

$$x_2 = x_1 + v \times \frac{1}{30}$$

Now we can simplify the second formula with the help of the first formula:

$$x_2 = 2x_1 - x_0$$

However, the above formula is without considering acceleration. By adding it we have:

$$x_2 = 2x_1 - x_0 + a_x$$

since we are moving in a 3D space, we will have gravity acceleration on the Y axis. Considering this all, we have the following formula:

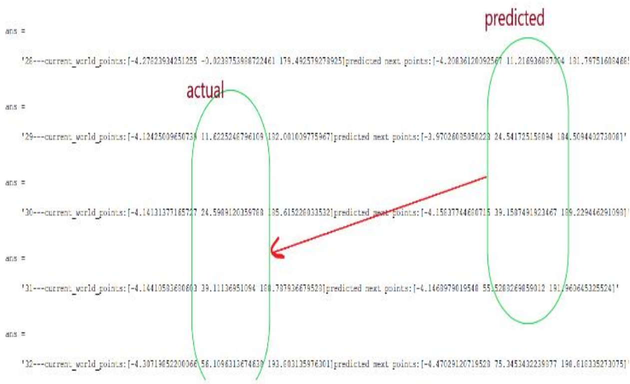
$$\left[x_2,\ y_2,\ z_2\right] = 2\left[x_1,\ y_1,\ z_1\right] - \left[x_0,\ y_0,\ z_0\right] + \left[a_x,\ a_y,\ a_z\right]$$

$$\text{where } \left[a_x,\ a_y,\ a_z\right] \textit{ is equal to } \left[0,\ -\frac{1}{2}\times 9.81\times t^2,\ 0\right]$$

Note that these equations do not take into consideration some world factors such as wind (which may cause slight difference between prediction and the actual value)

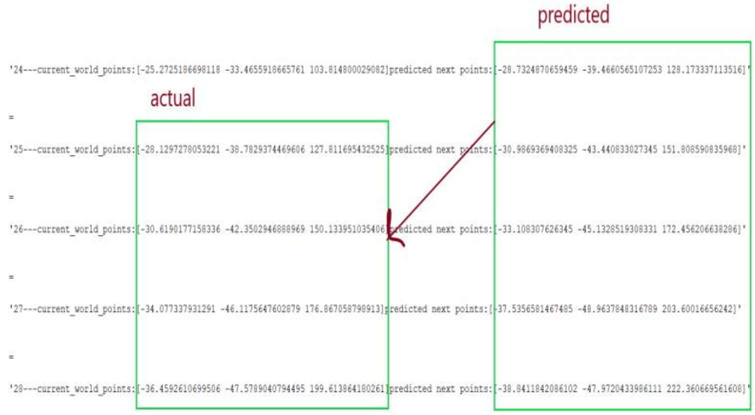
The Results:

Considering the free-fall scenario we had in the previous week, we expect the y axis to change over each frame, and we are looking to predict the next frame for each of the frames.



As it can be seen, we are getting great prediction results which are almost exactly identical to the actual location.

For the second scenario, we consider the videos that I threw the ball as fast as I could, away from the cameras:



Here we are also getting predicted values which are very close to the actual values.

From the results we obtained, we can conclude that my code is tracking the ball’s trajectory in a proper and acceptable way, and it seems as if our ping-pong playing robot will not have problems seeing the trajectory.