Projectile Trajectory Maker

With Air Resistance & Spin Resistance

Sunwu Choi

Introduction

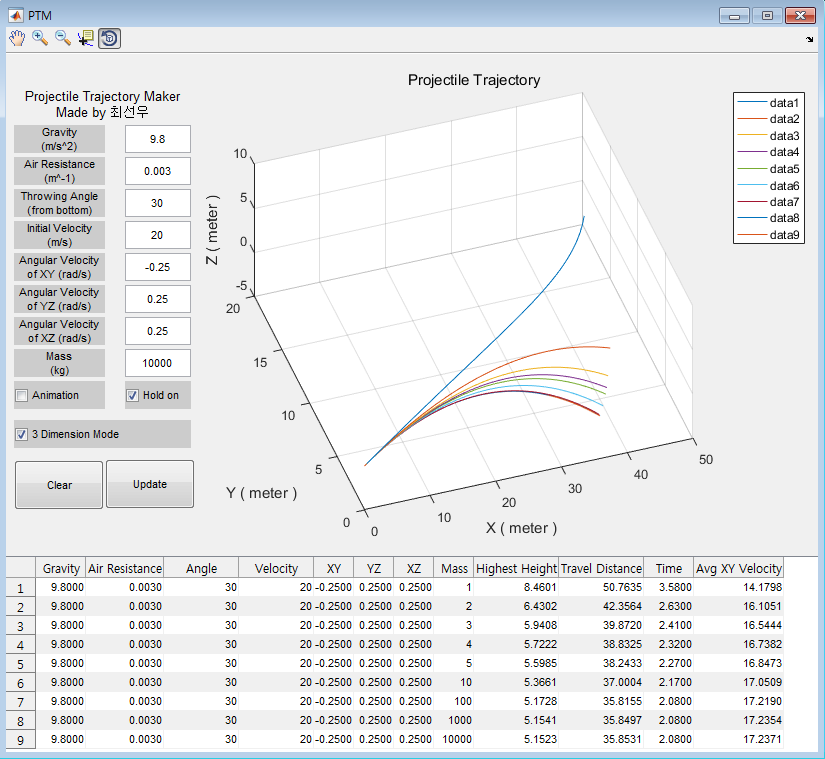
This program generates a two-dimension, or three-dimension trajectory of a projectile based on the throwing angle, air resistance, magnus effect generated by the spin, mass, gravity, and the initial velocity. This program can generate an animation and record the time consumed to hit the ground, the highest height, distance traveled, average speed on the XY plane.

How to use

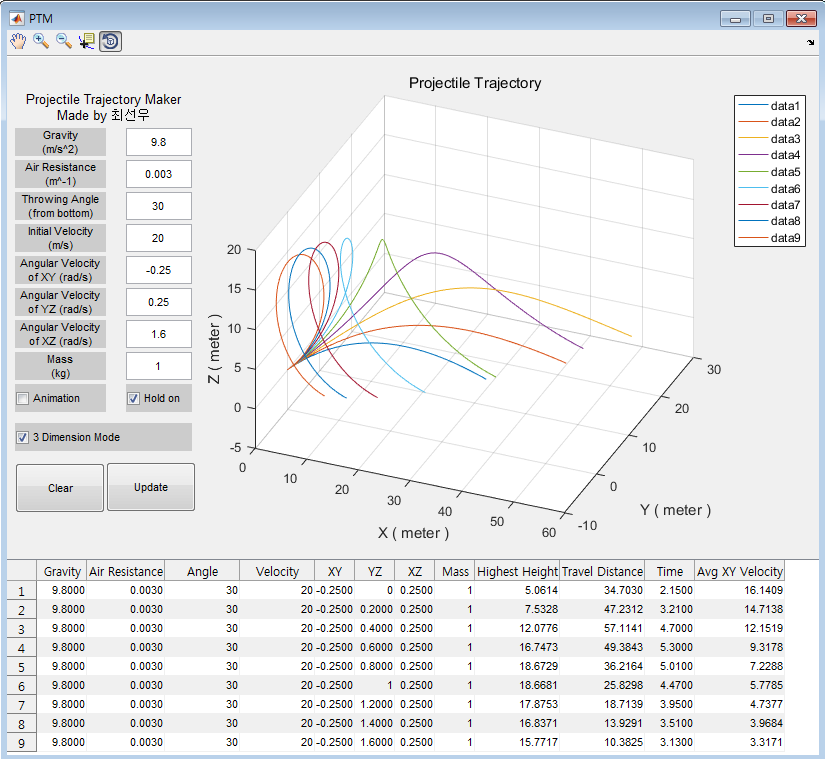
There are many white boxes that are available to edit. The throwing angle is the angle from the ground, not from the Z axis and this is for the ease of understanding. Animation check box enables real time animation until the projectile hits the ground. Hold on check box make sure that the current run is stored in the history. 3-dimension mode is used to generate a 3-dimension projectile trajectory. If the animation check box is checked, the user cannot press clear or update until the projectile hits the ground.

Algorithm

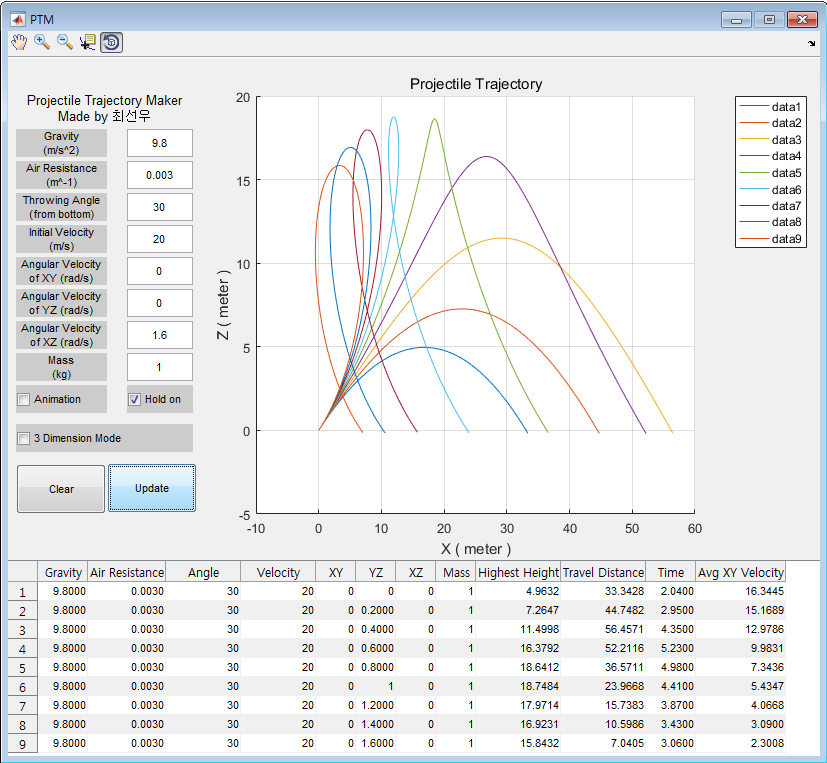
This program is made with some object-oriented design knowledge from my previous semesters and the basic GUI provided by MATLAB which is guide. Main file to run the program is the PTM and drawer, drawer3d is used for 2-dimension and 3-dimension plotting. Drawer function calculates the trajectory by intaking parameters such as gravity, air resistance, throwing angle, initial velocity, XZ angular velocity, mass, animation on or off. Drawer3d function calculates the trajectory by intaking parameters such as gravity, air resistance, throwing angle, initial velocity, XZ angular velocity, YZ angular velocity, XY angular velocity, mass, animation on or off. The 2d drawer function ignores other angular velocity except XZ so even if there is a value, it reads it as 0 but in 3d mode, it does not ignore the YZ and XY angular velocity.

Results

As seen in the above screenshot, when every parameter is fixed but the mass, if the mass goes above a certain point, the trajectory did not really change that much. This is because the initial velocity is same but mass changes, which means the air resistance becomes meaningless as the mass increases.

Next case was tinkering the XY YZ and XZ axis angular velocity to replicate a trajectory of a professional baseball pitchers throw.

When every parameter was the same but the XZ angular velocity, the projectile did a spin in the air and came down to the ground. But this program is simulating an ideal environment which means reality is quite different than this extremely ideal setting.

Travel distance here is not the pure distance traveled in the air, but it is the distance from the starting point to the point where the projectile hit the ground, which is the XY plane travel distance. Using this, I was able to find which parameters affect the travel distance the most. For example, tinkering the XZ axis spin so that it spins counterclockwise while kicking to the right, it is able to fly like a soccer ball a goalie kick.

Algorithm and further improvement idea

First, I wanted to make the animation to be the same as its real motion, not just ticks of calculation, but this was hard to achieve at my knowledge. Also, I was wondering who would watch a 10 second or longer animation until it finishes its flight.

Second, I wanted to create a mode that two or more projectile can be throwed at the same time, but then when the parameters are different, the array size can be different, which means it makes a dimension error when rendering both at the same time. So, I tried to put NaN for every non applicable array index, but this didn’t work out yet.

Finally, I tried to implement multi-threading, but I couldn’t find any good documentation about it in MATLAB system.