MAE 8 - Winter 2019 Project
Simulation of Free Kicks in a Soccer Game

In this project, you are to simulate the motion of a soccer ball during a free kick. The ball is kicked from an initial position in the field at an initial speed along a given direction. In the field, there are four defenders and a goalkeeper who attempt to block the ball from going into the goal. In flight, the motion of the ball is influenced by gravitational, frictional drag and Magnus forces. The Magnus force which results from the spinning motion causes the ball trajectories to bend. You will analyze the trajectories of the ball for seven different sets of parameters.

The trajectory of the soccer ball can be described by the following equations:

$$\begin{aligned}
\frac{\partial U}{\partial t} &= -C_d\frac{\rho A}{2m}U\sqrt{U^2+V^2+W^2} - C_m\frac{\rho A r}{2m}(\omega_y W - \omega_z V),\\
\frac{\partial V}{\partial t} &= -C_d\frac{\rho A}{2m}V\sqrt{U^2+V^2+W^2} - C_m\frac{\rho A r}{2m}(\omega_z U - \omega_x W),\\
\frac{\partial W}{\partial t} &= -C_d\frac{\rho A}{2m}W\sqrt{U^2+V^2+W^2} - C_m\frac{\rho A r}{2m}(\omega_x V - \omega_y U) - g,\\
\frac{\partial X}{\partial t} &= U,\\
\frac{\partial Y}{\partial t} &= V,\\
\frac{\partial Z}{\partial t} &= W. \qquad (1)
\end{aligned}$$

The parameters are defined as follows:

- $m = 0.4\,(kg)$: mass of the ball
- $r = 0.11\,(m)$: radius of the ball
- $A = \pi r^2\,(m^2)$: cross-sectional area of the ball
- $\rho = 1.2\,(kg/m^3)$: density of air
- $g = 9.81\,(m/s^2)$: gravity
- $C_d = 0.3$: drag coefficient
- $C_m = 0.6$: Magnus coefficient
- $t\,(s)$: time
- $X, Y, Z\,(m)$: position of the ball in rectilinear coordinate
- $U, V, W\,(m/s)$: velocity components of the ball in the x, y, and z directions, respectively
- ω_x, ω_y and $\omega_z\,(rad/s)$: angular velocity around the x, y and z axis, respectively
- $U_{mag,0}\,(m/s)$: initial speed of the ball
- ϕ and $\theta\,(^\circ)$: kick angles with respective to the positive x-axis and z-axis, respectively

During the analysis, the following parameters are to be computed:

$V_{mag} = \sqrt{U^2 + V^2 + W^2}\,(m/s)$: speed of the ball
$Acc = dV_{mag}/dt\,(m/s^2)$: acceleration of the ball
$KE = 0.5mV_{mag}^2\,(J)$: kinetic energy of the ball
$PE = mgZ\,(J)$: potential energy of the ball
$max\_height\_location$: location where the absolute value of vertical velocity W is smallest
$final\_location$: final position of the trajectory

You are to simulate seven different free kicks. The initial position of the ball ($X0$, $Y0$, $Z0$), the initial speed ($U_{mag,0}$), the direction angles (θ, ϕ), and the three components of the angular velocities (ω_x, ω_y, ω_z) for the seven simulations are given in the file **input_parameter.txt**. Download the file from TritonED.

The geometries of the field and the goal are given **field.mat** and **goal.mat**, respectively. File **field.mat** contains a structure **field** with 3 fields (X, Y, Z) which give the coordinates of the points used to render the field. File **goal.mat** contains a structure **goal** with 3 fields (X, Y, Z) which give the coordinates of the points used to render the goal. The surface render of the four defenders and the goalkeeper (i.e., the fifth defender) are produced by the function **defender.m**. Note that the surfaces vary in time as the defenders and the goalkeeper try to block the ball. The file **plot_layout.m** provides an example of how to plot the layout of the soccer field. Download these four files from TritonED.

Using Euler-Cromer method, equations 1 can be transformed into the following algebraic form:

$$\begin{aligned}
U_{n+1} &= U_n - \left[C_d\frac{\rho A}{2m}U_n\sqrt{U_n^2+V_n^2+W_n^2} + C_m\frac{\rho A r}{2m}(\omega_y W_n - \omega_z V_n)\right]\Delta t,\\
V_{n+1} &= V_n - \left[C_d\frac{\rho A}{2m}V_n\sqrt{U_n^2+V_n^2+W_n^2} + C_m\frac{\rho A r}{2m}(\omega_z U_n - \omega_x W_n)\right]\Delta t,\\
W_{n+1} &= W_n - \left[C_d\frac{\rho A}{2m}W_n\sqrt{U_n^2+V_n^2+W_n^2} + C_m\frac{\rho A r}{2m}(\omega_x V_n - \omega_y U_n) + g\right]\Delta t,\\
X_{n+1} &= X_n + U_{n+1}\Delta t,\\
Y_{n+1} &= Y_n + V_{n+1}\Delta t,\\
Z_{n+1} &= Z_n + W_{n+1}\Delta t.
\end{aligned} \tag{2}$$

Implement the equations 2 above with $\Delta t = 1/1000\,s$ to obtain the trajectories of the ball. Use the following conditions to end the trajectories:

- The ball needs to be above the ground, e.g. $z = 0\,m$, at all times.
- The ball needs to be inside the perimeter of the field at all times except when it is inside the goal.
- The ball can not be in contact with the defenders or the goalkeeper at any time.

You are to write three MATLAB files: **soccer.m**, **read_input.m** and **project.m**. The descriptions of these files are given below.

- File **soccer.m**: This is the function which solves the equations 2 for the trajectories of the ball for a given set of parameters. The function should have the following header: **function [T, X, Y, Z, U, V, W] = soccer(X0, Y0, Z0, Umag0, theta, phi, omgX, omgY, omgZ)** where the inputs and outputs are defined above. All inputs are scalars while all outputs are vectors.

- File **read_input.m**: This is a function that reads the parameters stored in the text file **input_parameter.txt** into MATLAB. The function should have the following declaration: **function [X0, Y0, Z0, Umag, theta, phi, omgX, omgY, omgZ] = read_input(input_filename, kick_id)** where **input_filename** is a string denoting the name of the file to be read and **kick_id** is an integer indicating the kick ID number. The outputs are the initial position ($X0$, $Y0$, $Z0$), the initial speed (**Umag0**), the direction angles (**theta**, **phi**), and the components of the angular velocity (**omgX**, **omgY**, **omgZ**). When the input **kick_id** is not available in the file, the function should set all outputs to **NaN** and display a warning to screen.

- File **project.m** is a script that includes the results of the tasks (and subtasks) described below. Make sure that the two figures are plotted and a text file **report.txt** is generated when your **project.m** is executed.

Task 1: To simulate the trajectories, call function **read_input** to get the necessary parameters and then call function **soccer** to obtain the trajectories. For each simulation, the outputs should include time (T), three components of position (X, Y, Z) and three components of velocity (U, V, W). All of these variables should be vectors with the same length. The project will be graded based on the following 3 tasks.

Task 2: You will create 2 figures based on the simulation results. All figures should have title, axis labels with correct units and legends. Use different colors to indicate the different trajectories. Please make sure that all 2 figures are plotted when your **project.m** is executed.

- Create **figure 1** to plot the seven trajectories, the seven final locations of the ball, the field, the goal, the defenders and the goalkeeper. The surface of the defenders and the goalkeeper should be computed using the longest time among the seven trajectories. Use function **plot3** and **surf** as demonstrated in **plot_layout.m**.

- Create **figure 2** which includes 2 panels, one on top of the other. On the top panel, plot the time on the horizontal axis and the speed of the ball on the vertical axis for all trajectories. On the bottom panel, plot the time on the horizontal axis and the ball acceleration (Acc) on the vertical axis for all trajectories.

Task 3: Create a 7-element data structure named **sim_res** with the following fields:

- **kick_ID**: to include the kick ID number.

- **final_time**: to include the total travel time of the ball.

- **max_height_location**: to include a 3-element vector indicating the location of the ball when it is at the maximum height.
- **final_location**:to include a 3-element vector indicating the final location of the ball.
- **final_speed**: to include the acceleration at final location.
- **final_acceleration**: to include the acceleration at final location.
- **travel_distance**: to include the total travel distance that the ball makes along the trajectories

Task 4: Use function **fprintf** to create a text file named **report.txt**. The text file should include the following lines:

- Your name on the first line.
- Your PID on the second line.
- A string '**kick_ID, final_time (s), final_speed (m/s), final_acceleration (m/s^2), travel_distance (m)**' on the third line.
- Corresponding values of kick ID number, travel time, final speed, final acceleration, and travel distance for each of the seven trajectories from the fourth to the tenth lines. Use single digit for the launch number and format %15.9*e* for others.

At the end of your **project.m** script, set the following:

p1a = evalc('help read_input');
p1b = evalc('help soccer');
p2a ='See figure 1';
p2b ='See figure 2';
p3a = sim_res(1);
p3b = sim_res(2);
p3c = sim_res(3);
p3d = sim_res(4);
p3e = sim_res(5);
p3f = sim_res(6);
p3g = sim_res(7);
p4 = evalc('type report.txt');

Submission instructions: Follow the homework solution template. Remember to **clear all**, **close all**, **clc**, and fill in your name and PID. Set **hw_num = 'project'**. Create a zip archive named **project.zip**. The zip archive should include the following files: **project.m**, **soccer.m**, **read_input.m**, **defender.m**, **input_parameter.txt**, **field.mat**, **goal.mat** and any other scripts / functions that you have written for the project. Make sure that you include all necessary files so that your **project.m** will execute properly. Submit **project.zip**

through TritonED before 10 PM on 03/17/2019. Use double precision unless otherwise stated.