

16-741 Mechanics of Manipulation

Project Report

# Mechanics of Frisbee Throwing

Debidatta Dwibedi (debidatd)

Senthil Purushwalkam (spurushw)

# Introduction

Frisbee is a popular recreational and professional sport. There has been a growing interest in the game of frisbee. The game basically involves passing around a flying disc to your teammates while avoiding the enemy. Wham-O.com presents an interesting statistic which shows that there are more frisbees sold every year than baseballs, footballs and basketballs combined.

While the game is really simple, skilled players have mastered the art of throwing the frisbee along interesting trajectories. Depending on the wrist and arm action of the player, the trajectory of the frisbee can be varied significantly. The internet contains a large number of interesting videos on the internet of people performing advanced moves using a frisbee. Some of them include using the backhand, making the frisbee come back like a boomerang and making the frisbee bounce off the ground like a skipping stone. Some examples of the above can be seen in the videos can be seen: [here](#) and [here](#).

Humans rely entirely on intuition to execute the advanced moves. Consider the task of building a frisbee throwing robot. Given a trajectory, how does a robot figure out the similar wrist and arm motions to make the frisbee follow a similar trajectory? A lot of studies have attempted to model the dynamics of a frisbee. In this work, we investigate if it is possible to leverage the knowledge of frisbee dynamics to infer the initial throwing conditions given different goals.

The advanced shots in particular are very helpful when a player is trying to complete a pass without the frisbee being intercepted by an opponent. In order to imitate such an ability, we formulate 3 goal tasks: follow a given trajectory, reach a teammate, reach a teammate while avoiding an opponent.

In summary, in the project we tried to achieve the following:

1. Understand the physics involved in throwing a frisbee.
2. Use a frisbee simulator to produce different trajectories based on the initial throw conditions
3. Infer the initial throw conditions for the frisbee to follow a target trajectory
4. Infer the initial throw conditions for the frisbee to reach a teammate
5. Infer the initial throw conditions for a pass in the game of ultimate frisbee that tries to throw the frisbee to a player on the same team while throwing it away from an opposition player.

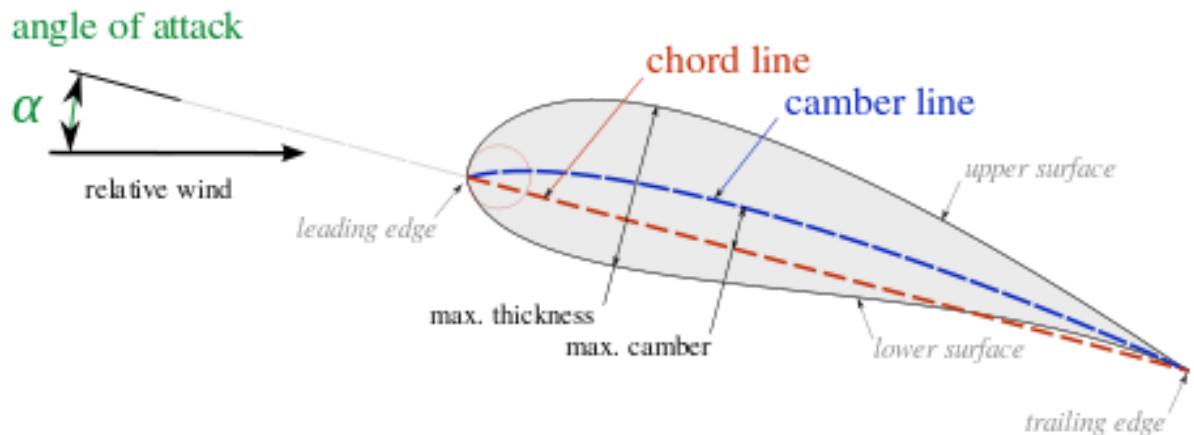
# Dynamics of a frisbee

The flight of a frisbee involves rotation and velocities about the three axes. In order to understand the flight of the frisbee, we can break down its dynamics into two components<sup>[1]</sup>:

1. an airfoil
2. A gyroscope

The interaction between these two systems results in the flight of a frisbee.

## Frisbee as an Airfoil



An airfoil is a structure shaped like the wing of an airplane. The movement of an airfoil shaped body through a fluid produces aerodynamic forces. *Lift* is the component of the force acting perpendicular to the motion. *Drag* is the component of the force acting perpendicular to the motion. When the airfoil moves a suitable angle of attack, it deflects the fluid which gives rise to a force in the opposite direction.

There are essentially four components that affect the flight of a frisbee as an airfoil:

1. **Gravity:**  
The gravitational force causes the frisbee to eventually land on the ground. Gravity is a constant vertical force that acts at the Centre Of Mass(COM) of the frisbee.
2. **Drag:**  
As explained above, drag acts in the a direction parallel to the direction of motion. Generally, drag causes the frisbee to decrease velocities in the direction of motion. It is important to not that the drag acts on the frisbee at the Center of Pressure (COP).

The magnitude of the drag can be written as

$$D = \frac{1}{2} \rho v^2 A C_D$$

where

- a.  $\rho$  is the density of the fluid
- b.  $v$  is the velocity of the frisbee
- c.  $A$  is the area of the frisbee
- d.  $C_D$  is the drag coefficient

The drag coefficient can be known to be quadratic in the angle of attack:

$$C_D = C_{D0} + C_{D\alpha} (\alpha - \alpha_0)^2$$

where  $\alpha_0$  is the angle of attack at which drag is minimum

### 3. Lift:

As explained above, lift acts in the a direction perpendicular to the direction of motion. Lift is the cause behind the rise in height of a frisbee even when it is thrown with no vertical velocity component. Lift also opposes the force of gravity. It is important to not that the lift also acts on the frisbee at the Center of Pressure (COP).

The magnitude of lift can be written as:

$$L = \frac{1}{2} \rho v^2 A C_L$$

Where

- a.  $\rho$  is the density of the fluid
- b.  $v$  is the velocity of the frisbee
- c.  $A$  is the area of the frisbee
- d.  $C_L$  is the lift coefficient

The lift coefficient can be known to be linear in the angle of attack:

$$C_L = C_{L0} + C_{L\alpha} \alpha$$

It is found that when the angle of attack is equal to  $\alpha_0$ , the lift is zero.

- ### 4. Aerodynamic Moments:
- Since the Centre Of Pressure(COP) does not coincide with the COM, this causes the following moments about the COM:
- a. Roll Moment (R)
  - b. Pitch Moment (M)
  - c. Spin down moment (N)

## Frisbee as a Gyroscope

The spin of the frisbee provides stability in flight due to the angular momentum and gyroscopic precession. These are essentially the two forces components that affect the frisbee as a gyroscope:

1. **Angular Momentum**

Angular momentum,  $\mathbf{H}$ , is the product of the spin angular velocity vector of the Frisbee in inertial space,  $\boldsymbol{\omega}$ , and the inertia matrix,  $\mathbf{I}$

$$\mathbf{H} = \mathbf{I}\boldsymbol{\omega}$$

2. Wobble: Two sources of wobble:
  - a. Induced by the thrower
  - b. Aerodynamic forces

# Simulating the Frisbee

For our experiments, we use [FrisPy](#) which is an open-source frisbee simulator. It is based on the physics model presented in the above section. The system takes as input a vector of initial conditions and outputs a trajectory taken by frisbee when thrown with the initial conditions. The vector of initial conditions is given by the following quantities:

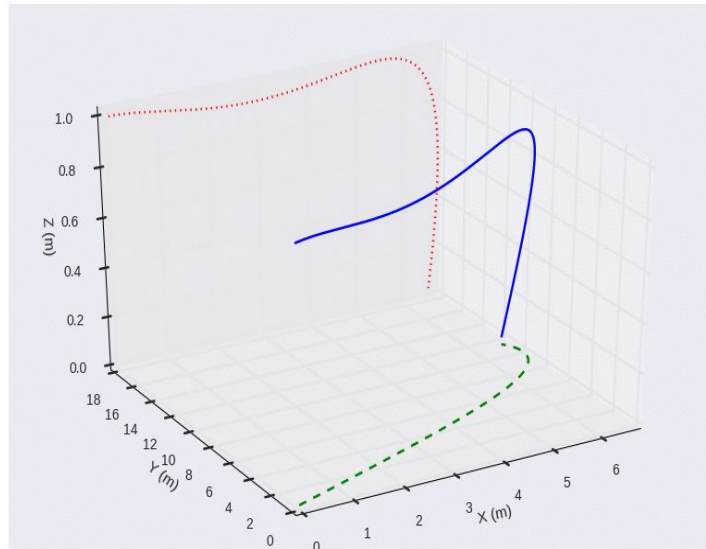
1. X - X coordinate of the center of frisbee
2. Y - Y coordinate of the center of frisbee
3. Z - Z coordinate of the center of frisbee
4.  $V_x$  - Velocity of frisbee in X direction
5.  $V_y$  - Velocity of frisbee in Y direction
6.  $V_z$  - Velocity of frisbee in Z direction
7. Phi - Angle about the X axis
8. Theta - Angle about the Y axis
9. Gamma - Angle about the Z axis
10.  $\phi\text{Dot}$  - Angular Velocity of about the X axis
11.  $\theta\text{Dot}$  - Angular Velocity of about the Y axis
12.  $\gamma\text{Dot}$  - Angular Velocity of about the Z axis

The frisbee is able to take on a variety of trajectories based on these initial throw condition.

We try to simulate two basic trajectories which can be seen as a result of what are colloquially called flat backhand and flat forehand. The frisbee rises slightly due to lift in both these scenarios but slowly falls down due to gravity. Initially, the trajectory for both these throws are the same because the velocities at which they were both thrown were the same but the angular velocities were in opposite directions. Because of the difference in sign, the trajectories curve in opposite directions before falling. More details of the trajectory can be found below:

## Flat Forehand

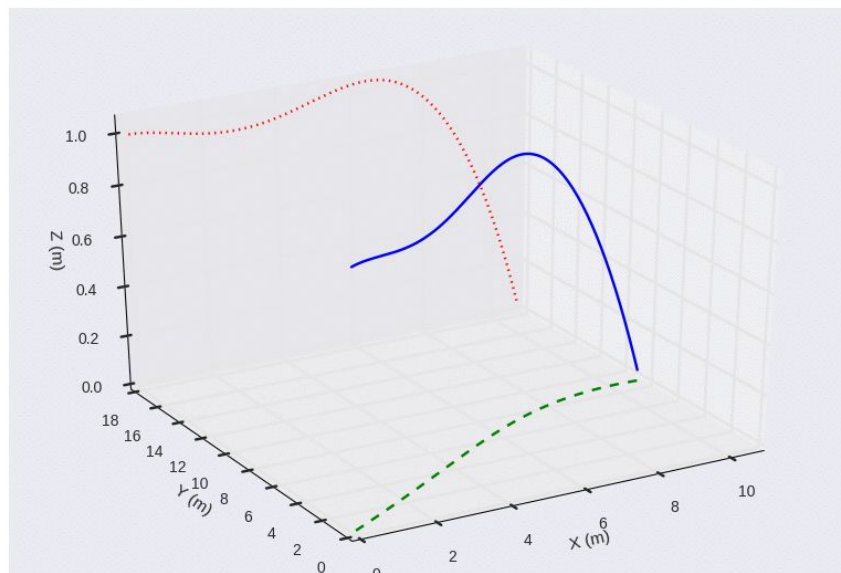
**Initial conditions:** Height of throw( $Z$ )= 1m,  $V_x = 9$  m/s,  $V_y = 9$  m/s,  $\text{gammaDot} = 2864.7889$  deg/s, Other initial conditions = 0



Trajectory for a flat forehand throw

## Flat Backhand

**Initial conditions:** Height of throw( $Z$ )= 1m,  $V_x = 9$  m/s,  $V_y = 9$  m/s,  $\text{gammaDot} = -2864.7889$  deg/s, Other initial conditions = 0



Trajectory for a flat backhand throw

# Optimization using the Frisbee Simulator

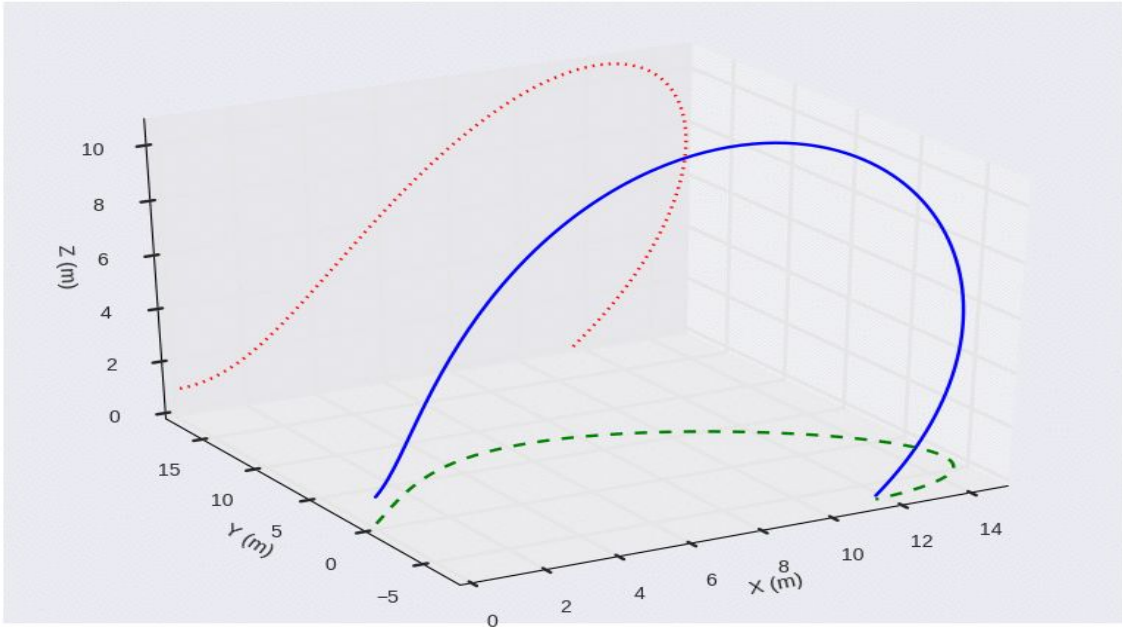
Having learnt how to use FrisPy, we moved on to the problem of finding the initial conditions that allow us to find a desired trajectory. We do that by framing the above problem as an optimization problem. If the trajectory of the frisbee was a closed form function we could have used the gradient of the trajectory with respect to the input parameters by analytically differentiating the function we are optimizing. But we do not know the closed form equation for the Frisbee's trajectory. Instead, we use the simulator as a function that takes the initial conditions and outputs a trajectory. Instead of analytically differentiating the function we use the numerical derivative of the simulator by varying its parameters (i.e. Initial conditions) by small values and seeing how the trajectory varies. To keep the optimization simple, we keep the size of the trajectory fixed that is we simulate for a fixed number of time steps. We do the minimization using Sequential Least Squares Programming (SLSQP) which is available as a solver in the minimize routine of SciPy.

One interesting thing is that if we optimize as mentioned above, the solver gives us initial conditions that are not humanly possible to attain due to constraints on the angles and speeds at which a person can throw a frisbee. Also the speed at which the solution is found might not lie in the region where all physics equations actually hold. Hence, we add certain bounds on the initial conditions so that the optimization actually returns an initial condition at which a human can throw a frisbee. This also makes the optimization stabler. We extract these constraints from Sarah Hummel's thesis [\[1\]](#) who has experimentally stated these constraints.

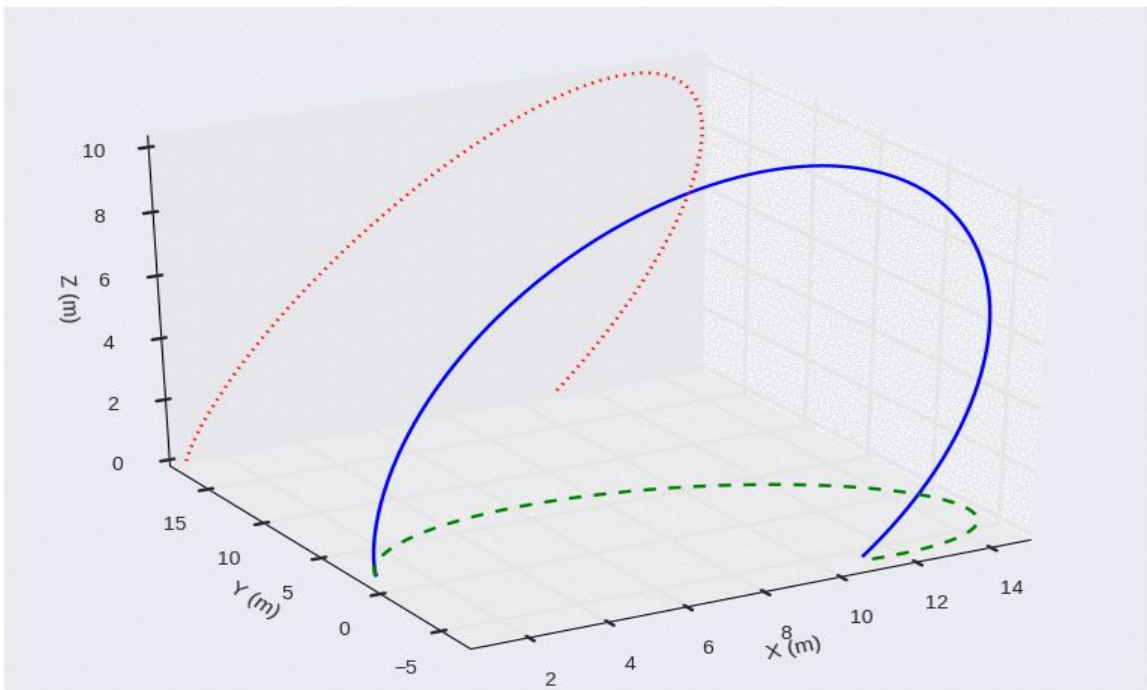
## Imitate a Target Trajectory

In this part, we want the frisbee to follow a desired trajectory by providing a target trajectory. The target trajectory is a fixed-length vector of 3-element tuples representing the (X,Y,Z) coordinates of the frisbee's trajectory. We want to minimize the L2 error between the simulated trajectory and the target trajectory. The solver is able to find solutions or proper initial conditions that produce simulated trajectories very close to the given trajectory.





Target trajectory



Predicted Trajectory

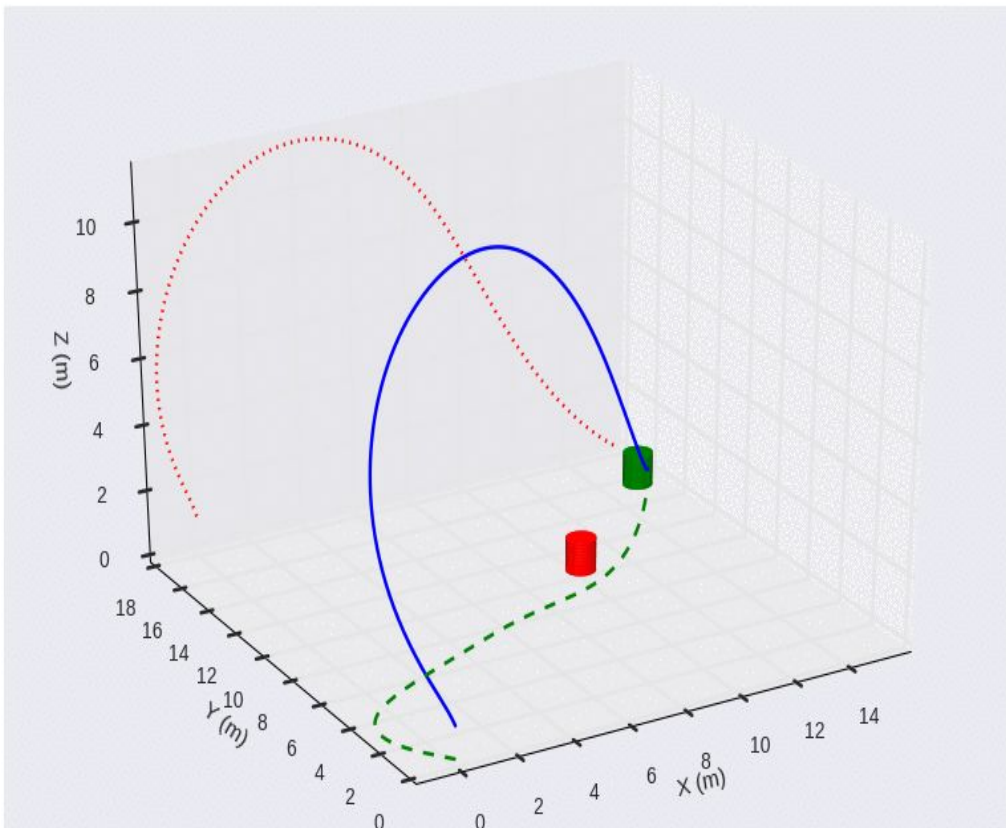
## Avoid an enemy and pass to a teammate

It is unlikely that humans are optimizing over full trajectories when pass the frisbee. What is more likely, humans have an intuitive objective like throwing as far as possible from the enemy while still being close to the teammate. We wanted to see if we can come up with trajectories for passes which one sees in a game of ultimate frisbee. The function which is minimized is the following:

$L = \text{Least distance from Team Position} - \text{Least distance from Enemy Position}$

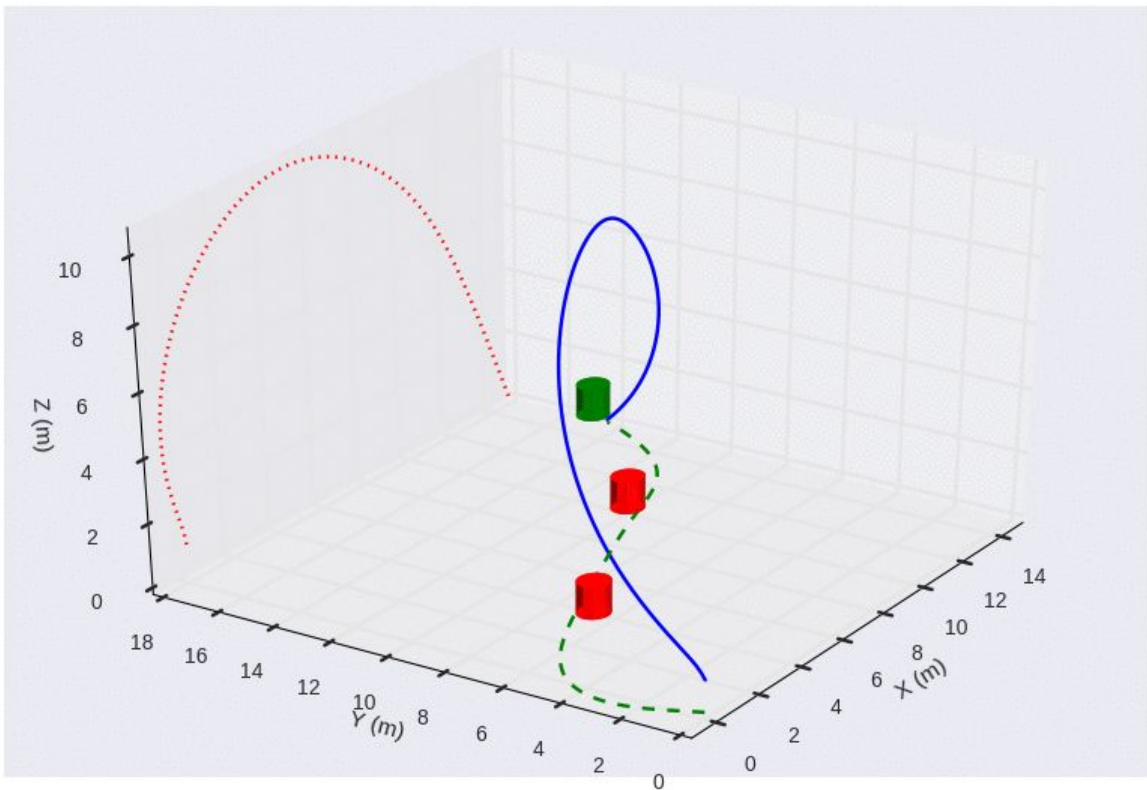
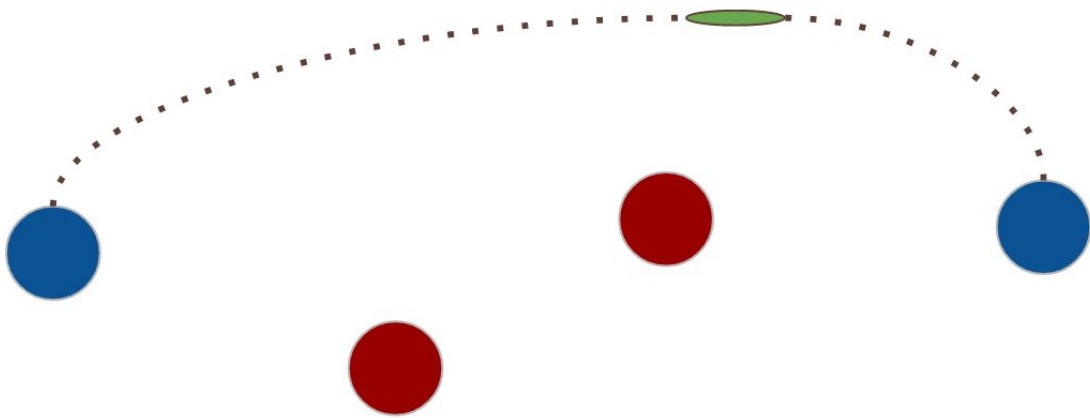
The following trajectories are obtained for two different kinds of scenarios:

i) One enemy:



Trajectory with one teammate and one enemy

ii) Two enemies:



Trajectory with one teammate and two enemies

## References

1. Hummel, Sarah Ann. *Frisbee flight simulation and throw biomechanics*. Diss. University of California Davis, 2003.