Time to Collision

COURSE 4, MODULE 4, LESSON 3

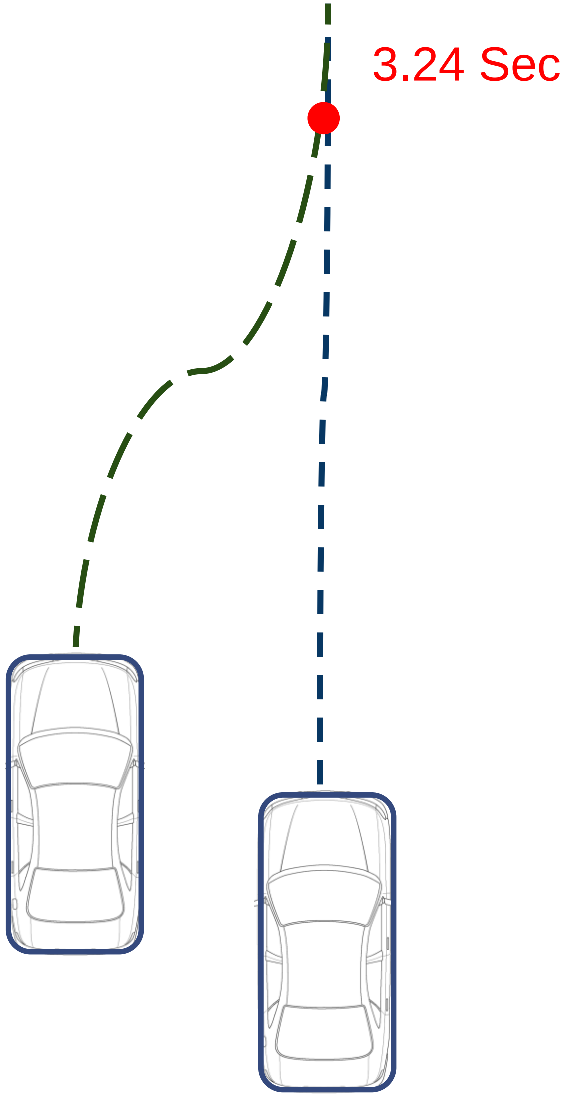
UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Learning Objectives

- Define the concept of time to collision
- Identify two approaches to calculate time to collisions and discuss their strengths and weaknesses
  - Simulation approach
  - Estimation approach
- Outline a simulation based algorithm to calculate time to collision.

Definition of Time to Collision

- Assuming all dynamic object continue along their predicted path:
  - Will then be a collision between any of the objects?
  - If so how far into the future?
- Time to Collision is comprised of:
  - Collision point between the two dynamic objects
  - Prediction of the time to arrive to the collision point
- Requirements for Accuracy:
  - Accurate predicted trajectories for all dynamic objects (position, heading and velocity)
  - Accurate dynamic objects geometries

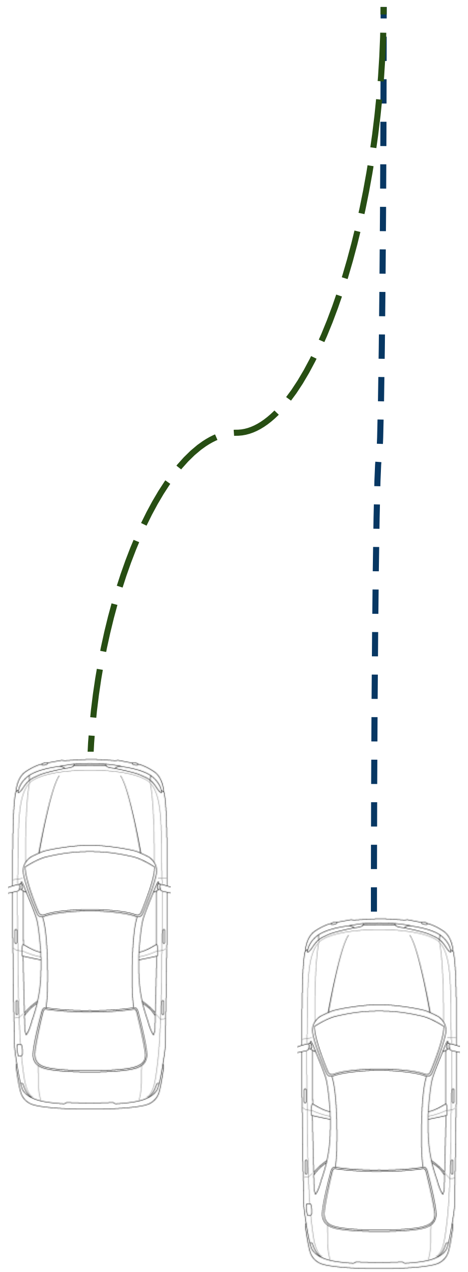


Approaches to calculate time to collision

- Two basic approaches to calculating time to collision:
    - Simulation approach
    - Estimation approach

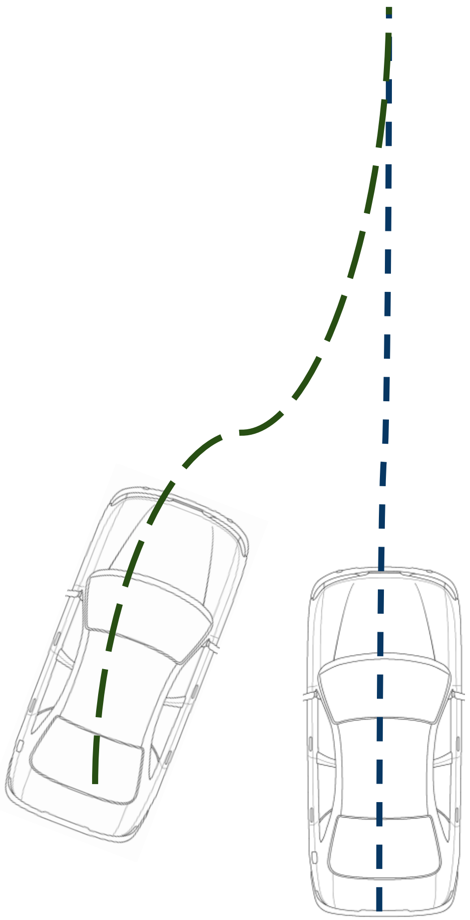
Simulation approach

- Simulate the movement of each vehicle as time passes



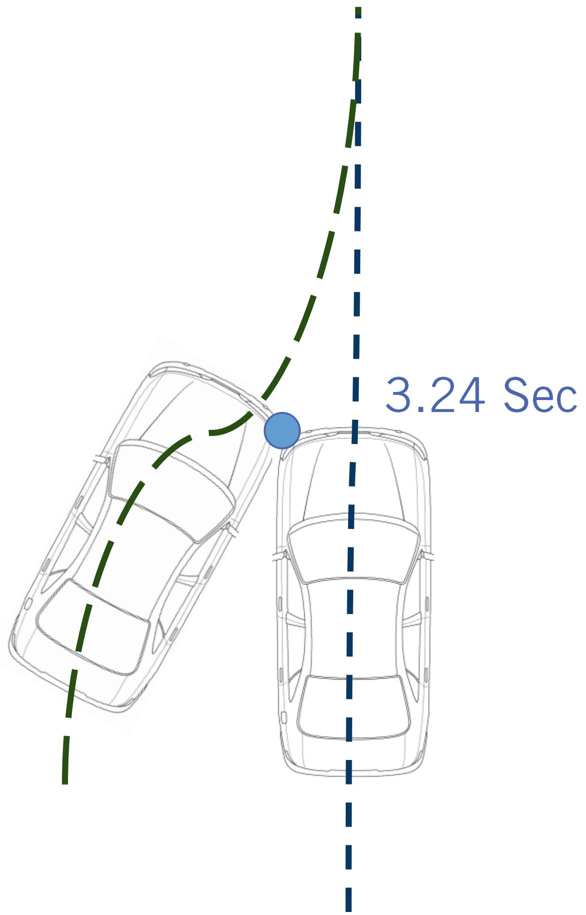
Simulation approach

- Simulate the movement of each vehicle as time passes
- Taking account of the vehicle model over time



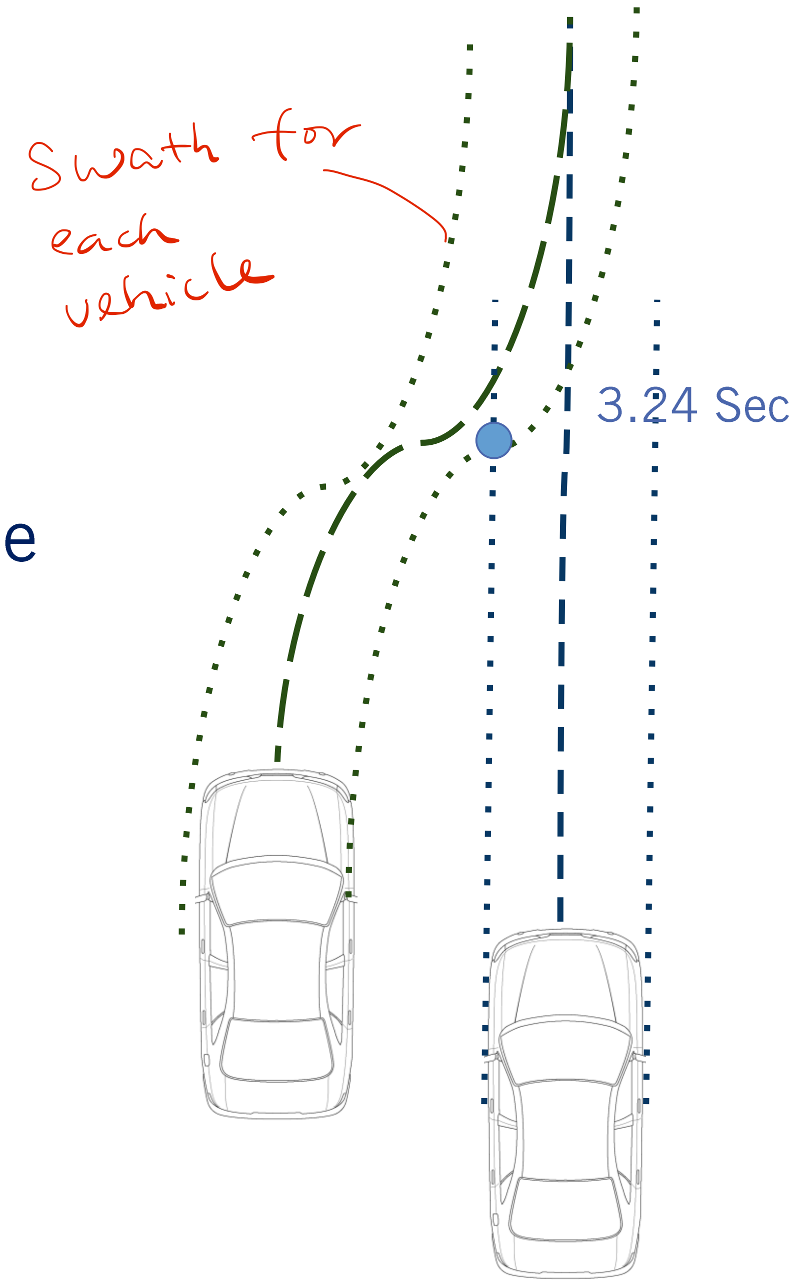
Simulation approach

- Simulate the movement of each vehicle as time passes
- Taking account of the vehicle model over time
- Checking if any part of the two dynamic object has collided



Estimation approach

- Geometries of the vehicles are approximated over duration of the predicted path
- Collision point is estimated based the cars predictions
- Many assumptions are usually made by this method usually to estimate time to collision



Relative Strengths and Weaknesses

Simulation Approach

- Computationally expensive

  step by step evaluations

  & object geometry intersection computations

- Higher accuracy if simulated with high fidelity
- Offline Applications (Dataset evaluation or Simulations)

Estimation Approach

- Computationally inexpensive
  - Memory footprint
  - Computational time
- Less accurate due to approximations and estimations
- Real Time Applications (In Car Prediction)

  online

Simulation approach Pseudocode

Inputs:

- D – list of all dynamic objects + planned path of ego vehicle
  - Predicted paths
- dt – time between simulation steps → can be different from the predicted path time steps.
- N_c – number of circles for collision approximation

Outputs:

- P_{coll} – list of all collision points
- TTC – list of all times to collision points

Time to collision

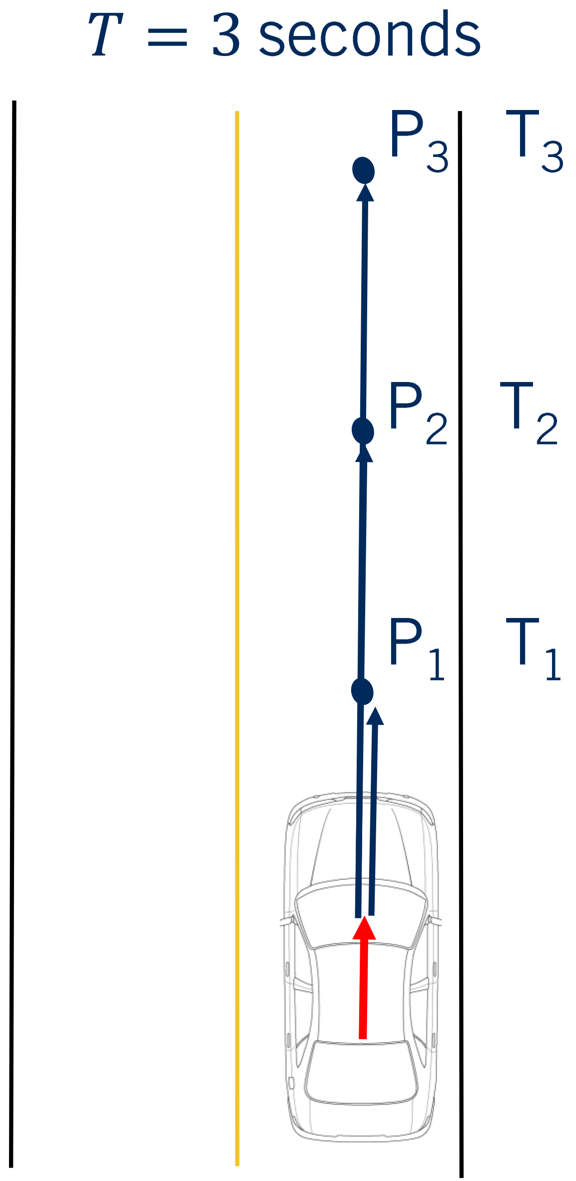
Simulation approach Pseudocode

Algorithm Constant Velocity TTC(D, T, dt, N_c)

$t \leftarrow 0$
$x_0 = x_{obj}$
while $t < T$ **do**
  $t = t + dt$
  for i $\in \{1, \dots |D|\}$ **do**
    $d_i.x_t \leftarrow$ PositionEstimation(d_i, t)
    for $\in \{i, \dots |D|\}$ **do**
      $d_j.x_t \leftarrow$ PositionEstimation(d_j, t)
      $P_{coll,ij} \leftarrow$ CollisionEstimation($d_i.x_t$, $d_j.x_t$, N_c)
      if $P_{coll,ij}$ **then**
        $TTC_{ij} \leftarrow t$
      end
    end
  end
end while
return P_c, TTC

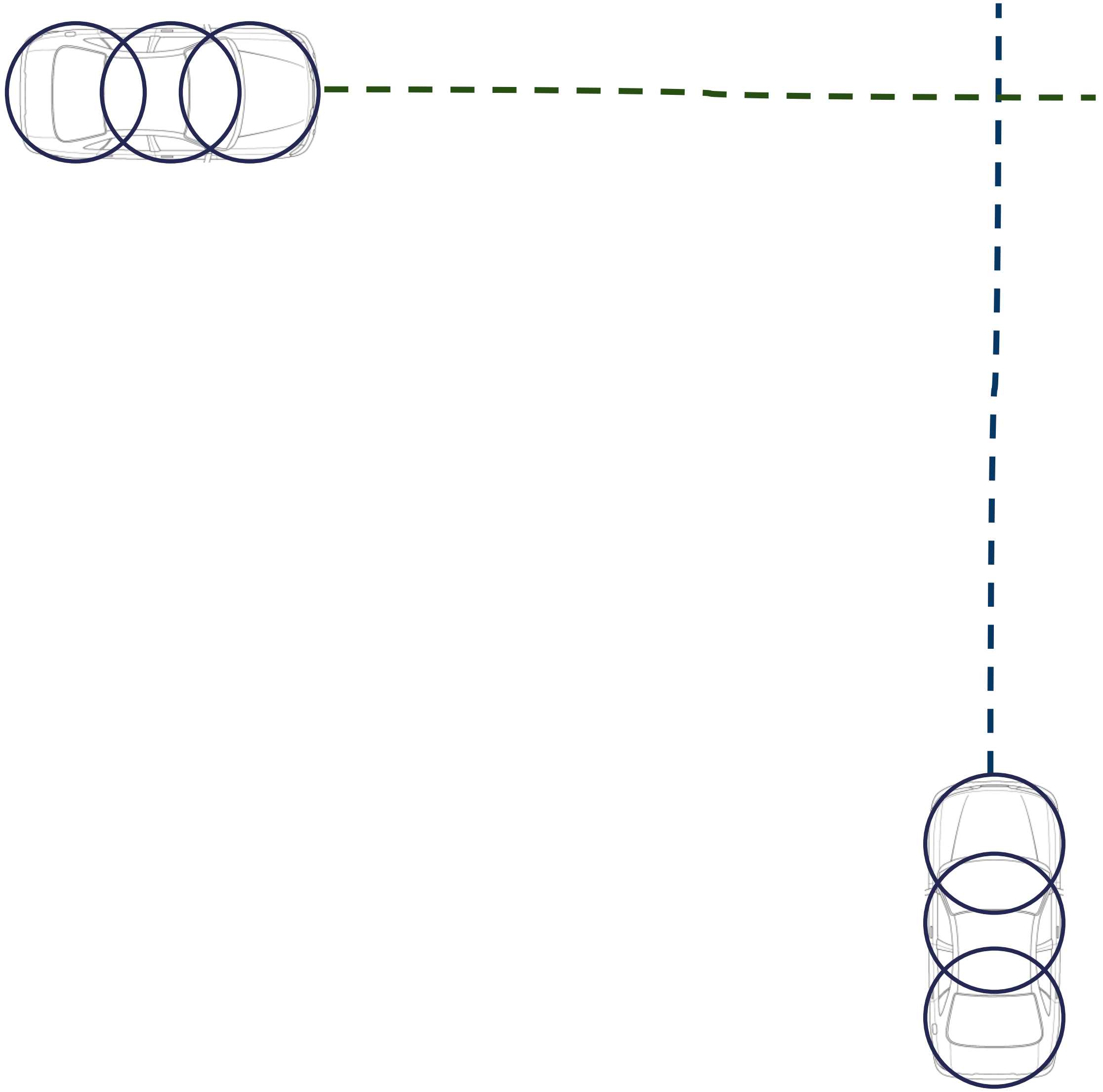
Estimation of Dynamic Object State

- Each predicted vehicle state has a predicted time at each location
- Find the closest vehicle state along the predicted path to the current simulation time



Efficient Collision Detection Method

1. Represent each car as a set of circles



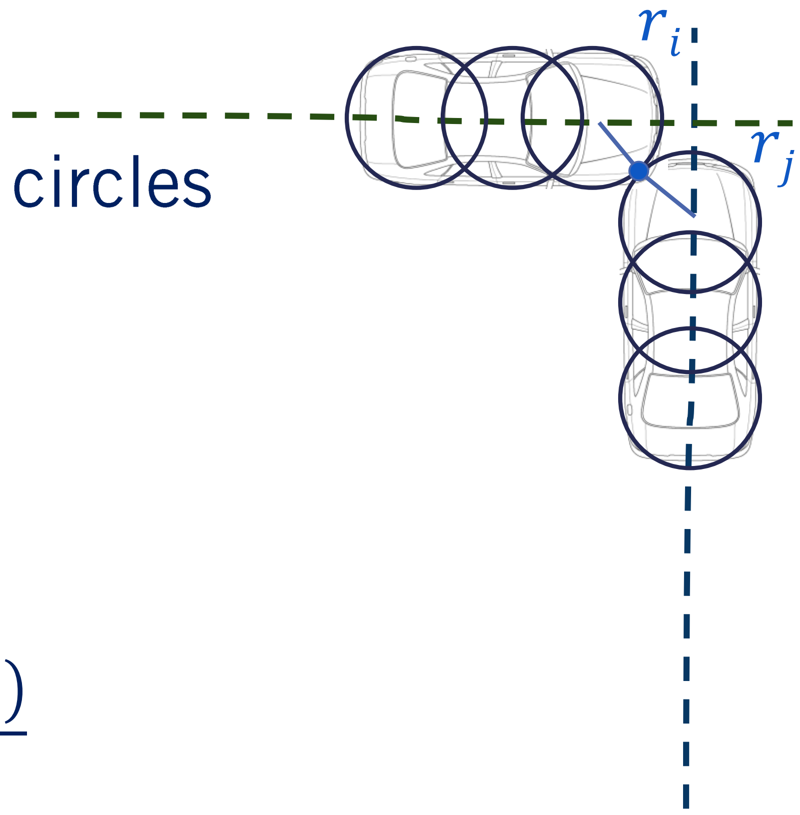
Efficient Collision Detection Method

1. Represent each car as a set of circles
2. Check if a collision will occur between two circles

$$d_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \qquad r_i + r_j \geq d_{i,j}$$

3. Calculate collision point

$$C_X = \frac{(x_i * r_j) + (x_j * r_i)}{(r_i + r_j)}, \qquad C_Y = \frac{(y_i * r_j) + (y_j * r_i)}{(r_i + r_j)}$$



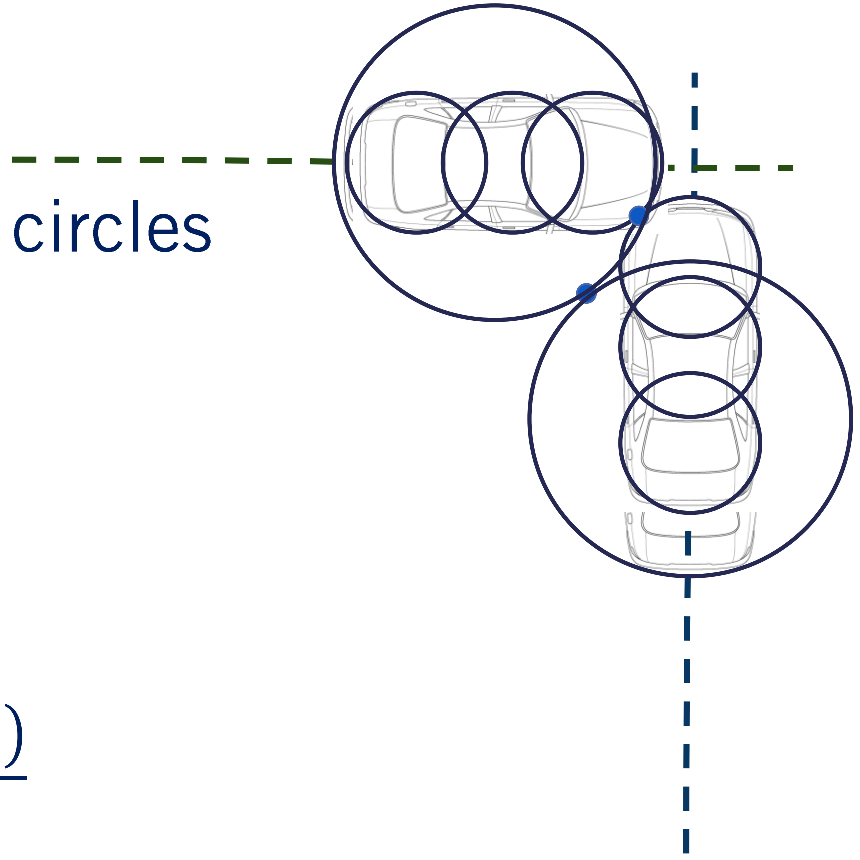
Efficient Collision Detection Method

1. Represent each car as a set of circles
2. Check if a collision will occur between two circles

$$d_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, \qquad r_i + r_j \geq d_{i,j}$$

3. Calculate collision point

$$C_X = \frac{(x_i * r_j) + (x_j * r_i)}{(r_i + r_j)}, \qquad C_Y = \frac{(y_i * r_j) + (y_j * r_i)}{(r_i + r_j)}$$



- Tradeoff between:
  - Accuracy
  - Number of computations

Summary

- Define time to collision and all its components
- Identify two approaches to calculate time to collisions and discuss their strengths and weaknesses
  - Simulation approach
  - Estimation approach
- Outline a simulation based algorithm to calculate time to collision

- **Next**: Behavior Planning and its Principles