CLEMSON UNIVERSITY INTERNATIONAL CENTER FOR AUTOMOTIVE RESEARCH

CPSC 8810: Motion Planning Final Project Report

Motion Planning of a Vehicle in a Roundabout Scenario

Professor: Dr. Ioannis Karamouzas

Students: Sanket Bachuwar, Ardashir Bulsara, Huzefa Dossaji

Date: April 28th, 2019

Contents

Introduction:

Roundabouts are a recent development in urban environments and thus prove to be a challenge to drivers of all types and ages. Their objective is to improve the traffic flow in traditional intersections with traffic lights. The purpose of this project is to design a program that can successfully navigate multiple agents through a roundabout scenario. The approach taken is to use a forced based local navigation method where the behavior of each agent is modelled as a collection of forces with respect to each other to essentially plan a collision free route to its goal. The simulation incorporates a finite state machine with 3 states relating to the roundabout scenario: entry, roundabout, and exit to closely mimic the driver behavior in real life. The entry and exit approach strategy will rely on Bezier Curves to allow for smooth transition of the agent. The agents are circular in shape and have predetermined entry and exit positions. The environment is a 4-way anti-clockwise roundabout scenario consisting of entry and exit lanes while also having double lanes in the roundabout itself [1].

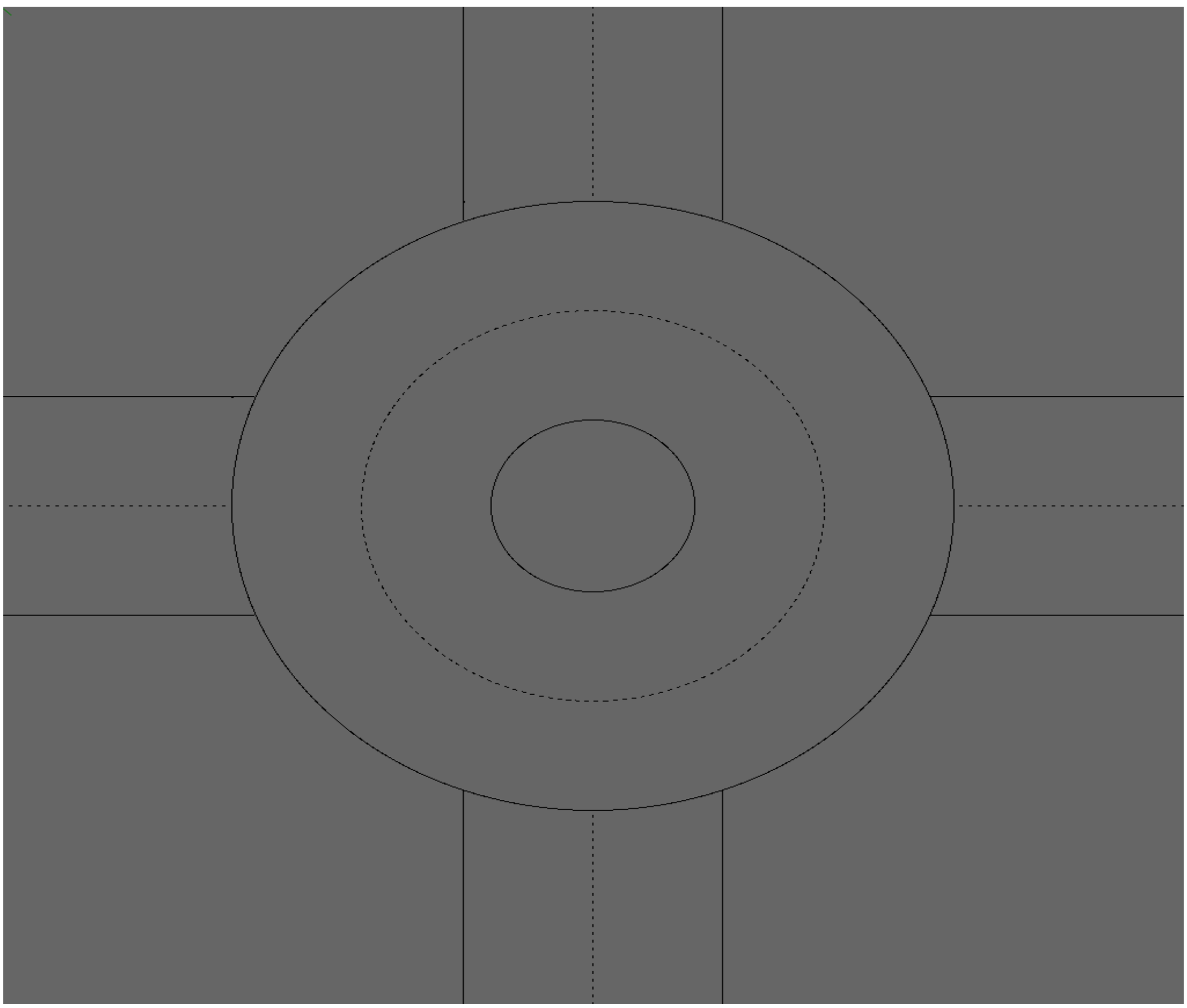

Figure 1: Roundabout Scenario

Definitions

Roundabout*:*

- Entrance: This is where the entry curve is created and is the spawn of our agents which will traverse through the roundabout. The entrance curve will exist from the extreme side of the canvas to a little bit after the roundabout entrance or the start of the large circle.
- Circulatory Roadway: This roadway is where the roundabout occurs. The roadway can consist of one or two lanes and in our initial approach, we used two lanes. A state machine would be necessary to change between lanes after transitioning from an entrance or into an exit.
- Exit: The exit curve is created from a little bit outside of the large outer circle to the extreme edge of the frame towards the goal position. The entrance and exit curves are same in generation but are followed in opposite sequence.
- Central Island: The size of the central island is crucial for path generation during all three stages of the roundabout. It determines the radius to follow during the circulatory roadway and the angle as to which the entrance and exit curves begin and end.
- The size of the lanes inside the circulatory roadway and the outside for entry and exit lanes are 112 pixels wide.

Agents*:*

- Agents in roundabout are all part of the same group excluding the central island
- Agents all have a preferred speed of 1.3 and a max speed of 2 based on force applied from other agents
- Agents all have a radius of 25.

Bezier Curves:

The main idea of this path planner revolves around setting certain rules of the road for the driver to follow. One of the most important rules is to maintain a certain predefined lane while driving, especially when transitioning from a straight road to an intersection or a roundabout. This is where the idea of parametric path generation comes into play. A Bezier curve is an extension of the spline curve in Bernsteinian coordinates where the curve can be continuous, and its implementation is simple enough to carry out for multiple sets of control points.

The equations for the curves are as follows [1]:

$$P_0 = R_i + L_0 \frac{R_{i-1} - R_i}{||R_{i-1} - R_i||}$$

$$P_1 = R_i + L_1 \frac{R_{i-1} - R_i}{||R_{i-1} - R_i||}$$

$$P_2 = I_1 + L_2 \frac{I_1 - I_2}{||I_1 - I_2||}$$

$$P_3 = P_4 + L_3 \overrightarrow{T}$$

$$P_{4_{(x,y)}} = \begin{pmatrix} r\cos(\arctan(\frac{R_{i-1y} - R_{iy}}{R_{1-1x} - R_{ix}}) + \varphi) + R_{ix} \\ r\sin(\arctan(\frac{R_{i-1y} - R_{iy}}{R_{i-1x} - R_{ix}}) + \varphi) + R_{iy} \end{pmatrix}$$

The control points are based on set places within the roundabout environment. They are dependent on the position of [1]:

- Start point P0
- End point P4 (dependent on the radius of roundabout)
- P2 and P3 are constraints which must be generated such that the vehicle envelope stays within the bounds of the lane.
- P1 which decides the smooth entry point of the curve into the circulatory path.

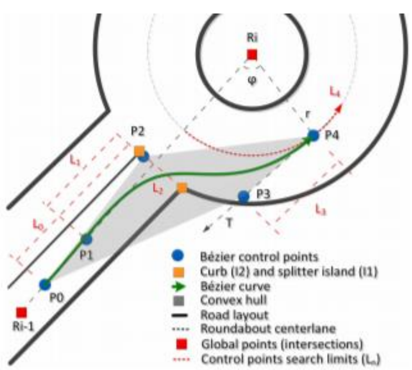

Figure 1: Bezier curve control points [1]

Scenario:

In the current simulated scenario, agents must enter, exit and negotiate a roundabout with four-way entry – exit and a central island. The agents should not collide with each other during this entire maneuver. The agents have different entry and exit points. No agents have the same entry point but share the same exit points.

Implementation:

Base Code:

The code from homework 2 is being used as the base, while the simulation environment and other modules such as the state machine are built by the team.

Modifications to the Base Code:

Following are the modifications done on the base code:

- A simulation environment was created using python's Tkinter library.
- Implementation of the finite state machine.
- Module to generate the Bezier curve and waypoints on the Bezier curve for the agents to follow.
- Module for agents to navigate the scenario using the local force-based navigation approach.
- Import file to read newly defined parameters for the agents.

Finite State Machine:

A finite state machine with three states is used in this simulation for the agents to successfully navigate the scenario. The three states used are:

- State 1: Entry to the roundabout, Bezier curve-based trajectory
- State 2: Traversing the roundabout, local forced based navigation
- State 3: Exiting the roundabout, Bezier curve-based trajectory

The agents during entry to the roundabout switch to state 1 and follow the Bezier curve trajectory with a specified velocity, as soon as the agent enters the roundabout it switches to state 2 and complies with force-based navigation till it reaches the desired exit point, in this state the agent performs collision avoidance and therefore multiple agents can be on the circulatory road at the same time. At the desired exit point on the roundabout, the agent switches to state 3 to follow the exit Bezier curve trajectory with a specified velocity.

The finite state machine is designed such that the agents can switch from any one state to any another based on the initial position and goal position. As agents using the first exit at the roundabout switch from state 1 to state 3 directly, thereby following only the Bezier curves.

In state 1 and state 3, a function generates certain number of points (defined by the user) that lie on the Bezier curve. Every point acts as a way point or local goal for the agent on this trajectory. The agent then follows every local goal to negotiate the curve with a predetermined velocity.

In state 2, the agent switches to local force-based navigation, where it senses the agents around and exerts force based on the other agent’s velocity and distance between the agents. The central island is also an agent therefore acting as bounds for agents to move around it.

Every curve has been assigned a number; every entry curve has positive number while exit curve has odd number. This is used to as a state switching parameter for entry. It also doubles up to add to the robustness of the code, as entering an exit curve number at entry will not allow the agent to traverse on the wrong side.

Another state switching parameter to switch to state 2 or force-based navigation is assigned in the input file, this parameter can attain only two values (1 – to switch to state 2 and 0 – to skip state 2).

To switch to state 3 an additional condition is applied at sate 2 itself, if the x or y co - ordinate of agent is within a specified range of its goal position the agent switches to state 3, exits the roundabout and follows the exit curve to its goal.

Tuning and Optimizing:

The tuning and optimizations of the parameters is done in order to achieve a smooth motion plan execution for the agents and an optimal performance for the simulation.

For Bezier curve following, to make the agent navigate through the curves smoothly, there are two tuning parameters for every time interval of the simulation: the number of waypoints generated per curve and the velocity of the agent. If the number of waypoints generated is increased with high velocity, it leads to a jerky motion for the agent. While if the number of waypoints is smaller with low agent velocity, the agent is unable to reach the local goal and thereby loses track of its path. Therefore, an optimal set of values are being used for a particular time interval of simulation [2][3].

For local force-based navigation, agent’s sensing radius and time horizon are the tuning parameters [2][3]. A smaller sensing radius will cause collision as it is unable to sense other agent in time. The time horizon parameter decides on at what time force is to be exerted and the magnitude of force that is to be exerted onto the other agent. A small value of this parameter may not be able to generate enough force to avoid collision. While a higher value will exert high force displacing agents out of simulation environment bounds.

The above-mentioned parameters also affect the computational performance of the simulation. A small time step or large values of sensing radius and number of waypoints require large amount of computations, thereby slowing down the simulation. Therefore, it is necessary to choose an optimal set of values so that it does not affect the simulation results but also uses minimal amounts of computation.

Results / Outcome / Problems:

Several different scenarios and environments of roundabouts were made before we could finally settle on one that gave good results. Positioning and visualization of Bezier curves significantly improved along the iterations allowing the agent to improve its traversal through the roundabout. We learned quickly that having a square canvas frame (implying that the length and width of the visualization were the same) made creating Bezier curves much simpler and effective. All Bezier curves from the entrances and exits are symmetric and with respect to the origin (512,512) of the canvas, while designing a frame of 1024x1024 pixels. During our first iteration, designing Bezier curves that were similar in scale and symmetrical proved much more difficult than our second iteration.

In our second iteration (Figure 3), one of the biggest problems was that the Bezier curves overlapped each other which caused conflicts when switching state machines from entry, to circular roadway to exit. If the exit was directly to the right of the entrance the agent would end the entry curve farther than the beginning of the exit curve which would force the agent to traverse back the circulatory roadway to the start of the exit curve.

In the final iteration, for simplification we decided to widen the central island so we could deal with one lane on the roundabout and shortened the Bezier curves to 4 points instead of 5 so the agent would stay longer in state 2, circulatory roadway. Transitions were much smoother and realistic while the forced based approach to stay a certain distance away from the center island also worked better. Since the large center landmark is also considered an agent and the scaling of the frame was considerably increased, it was necessary to also greatly increase parameters such as the sensing radius, time horizon, maximum force applicable to the agent, simulation time step max iterations, and number of points when generating the Bezier curve to get optimal behavior patterns from the simulation. Figure 5 shows the final motion of the agents when spawning at different entrance curves and having random exit goal lanes.

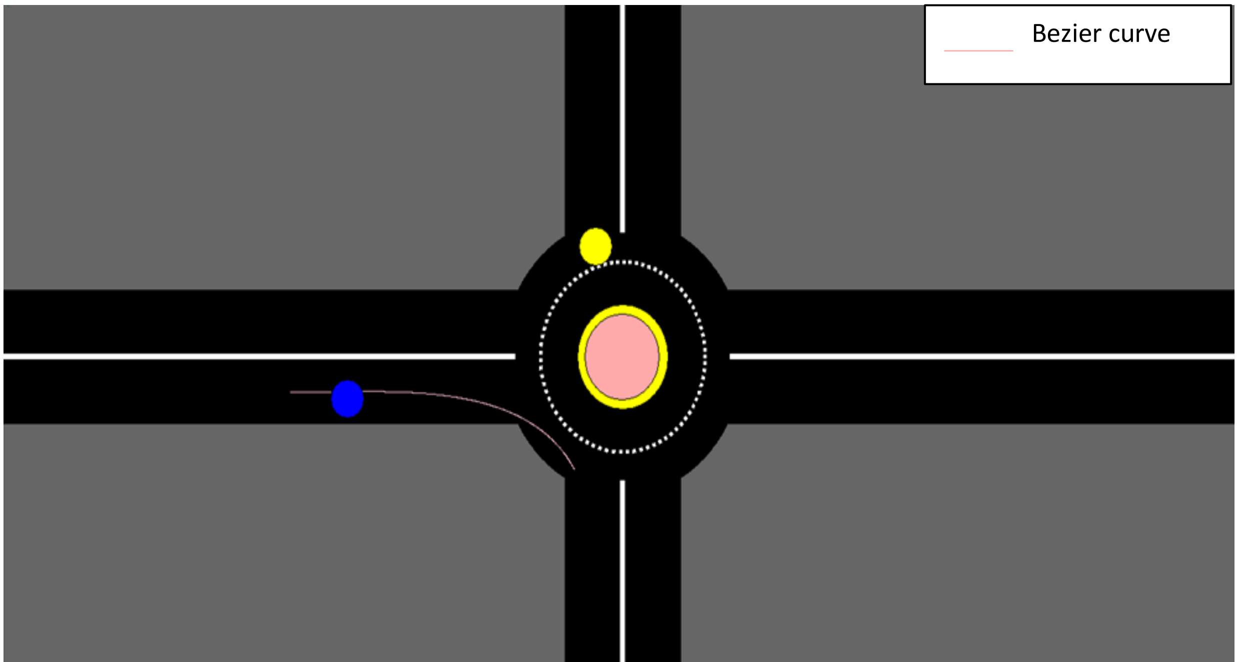

Figure 2: First Iteration

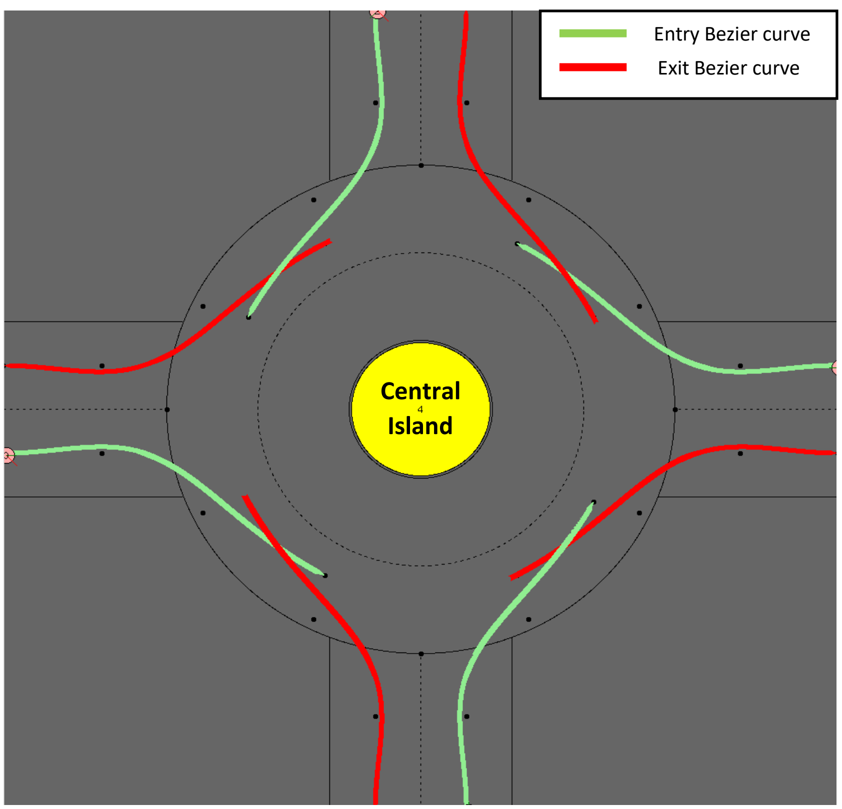

Figure 3: Second Iteration

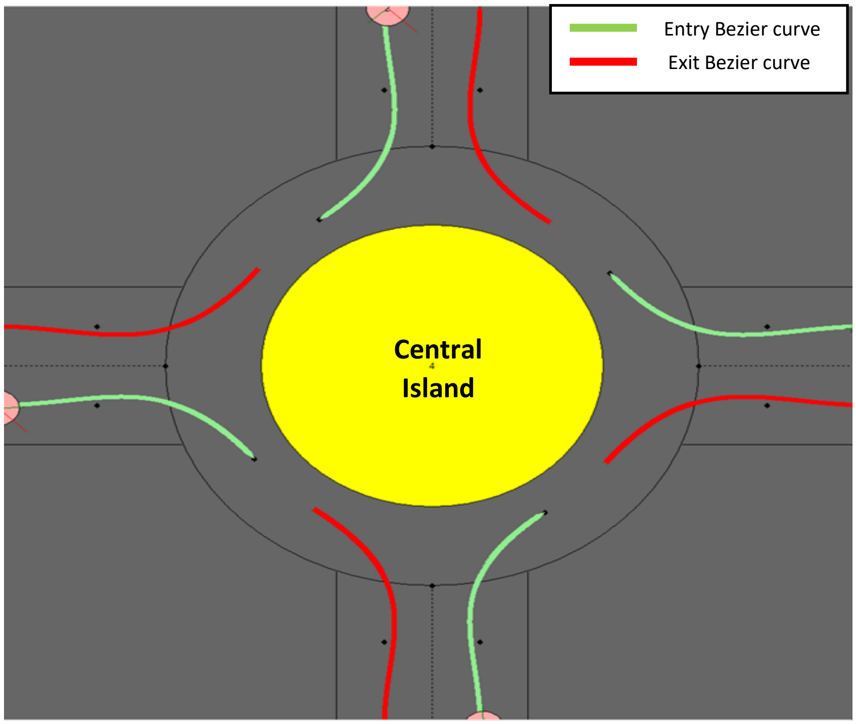

Figure 4: Third and Final Iteration

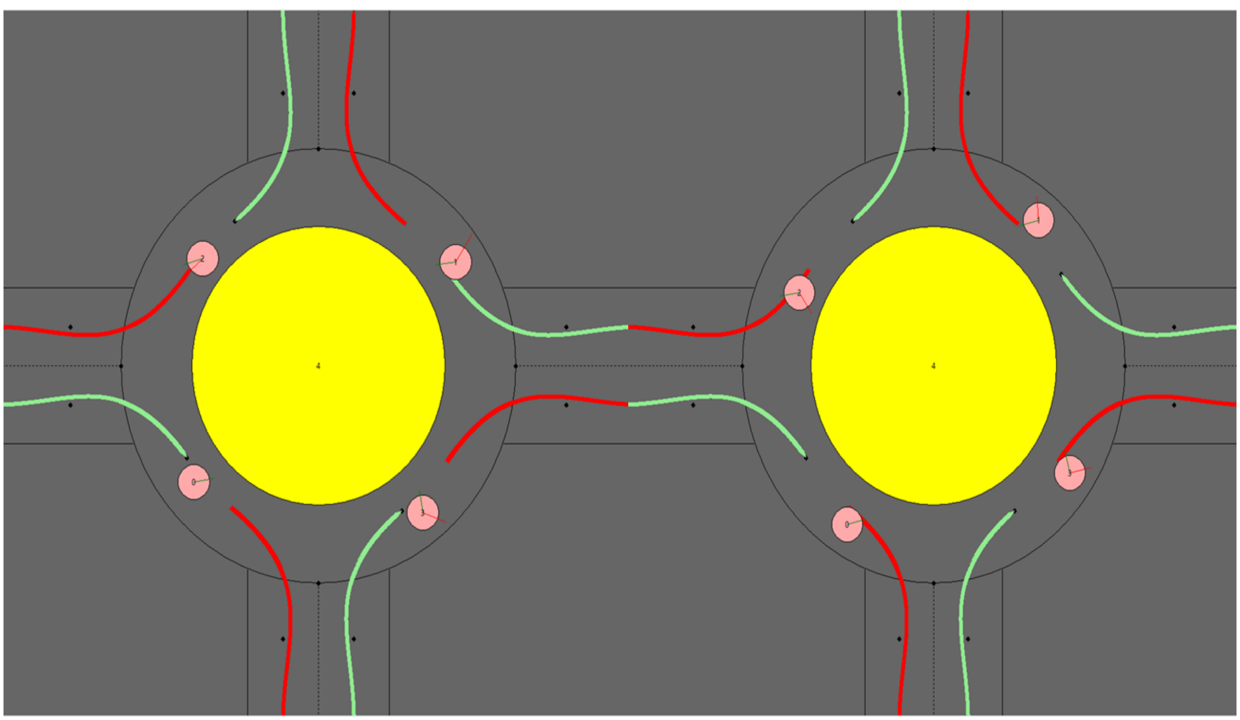

Figure 5: Agent Motion inside roundabout with local force-based navigation

Future Plans:

The plan is to implement an additional state to the roundabout scenario where we consider two lanes in the circulatory roadway. The agent will have to switch to the inside lane after traversing through the entrance Bezier curve when the exit goal of the agent is not directly to the right.

In the current implementation, use of Bezier curves based on a fixed number of control points having a predefined location relative to the splitter islands and the central island is being done. To implement this in real world scenarios, the agent must generate its own path based on the traffic pattern of vehicles in the roundabout.

Another possibility would be to implement a state where it senses traffic within the roundabout as it is about to enter, and yield such that it has enough space to merge onto the roundabout safely.

In this implementation of path planning, the behavior of drivers within the roundabout isn't completely accurate, therefore there is a scope to implement a behavioral state machine.

References:

1. Gonzalez, David, Joshué Pérez, and Vicente Milanés. "Parametric-based path generation for automated vehicles at roundabouts." *Expert Systems with Applications* 71 (2017): 332-341.
2. Local Navigation with Predicted Forces (HW-2 CPSC 8810 - Motion Planning) Author: Ioannis Karamouzas (ioannis@g.clemson.edu)
3. Forootaninia, Zahra, Ioannis Karamouzas, and Rahul Narain. "Uncertainty Models for TTC-Based Collision-Avoidance." *Robotics: Science and Systems*. 2017.