The project in question aims to develop a model by the means of either deep reinforcement learning1 or the use of genetic algorithms2 capable of navigating a remote-controlled car from A to B in a non-static environment. Imagine you would want a remote-controlled car to move from the elevator to the coffee room in an office building, manoeuvring around static objects such as pillars, desks, chairs, etc. as well as non-static objects like people stochastically walking around the office. Provided you can develop a model with good performance, we can make an attempt at doing just that, using a real remote-controlled car we have at the Deloitte office. The remote-controlled car built according to the design specifications of the MIT RACECAR3, the framework around which is a state-of-the-art research platform for testing autonomous vehicle algorithms. The inputs available to the model will be the relative coordinates of the car to the desired endpoint and a 270-degree LiDAR sensor mounted on the car giving distances to the nearest objects. The model should aim to predict a speed and an angle for the car. The angle being between the forward wheels and the car frame, a measure of the car’s direction. It should also be noted, that since we intend on trying the algorithm out on a real car, it is necessary to add physical restrictions to the model to help emulate more accurately the real world. An example being that the angle of the front wheels and the speed of the car have to be altered in a continuous fashion, i.e. the angle cannot go from –20 degrees to 40 degrees in one-time step. A very simplistic physics engine has been developed to allow for low-memory consuming simulations of environments to be run for training purposes. It is also possible to emulate the car in a more realistic engine using Gazebo, such that the true performance can be evaluated before using the algorithm with ROS (Robot Operating System) on the actual car. Inspirational thoughts/questions regarding neural architecture:

• Will the network need memory?

• Is it necessary to create a hybrid between a rule-based model, and a neural network to account for

the complexities?

• Will fabricated heuristics help the model (ex. Distance to goal, angle to goal, number of objects in

sight) perform better?