Motion Planning and Control of an Autonomous Car for Pedestrian Avoidance

Nikhil H N, Shantanu S, Varun K, Weiming Chen
5049628, 5032814, 4980611, 5015006
Robot Motion Planning and Control
ME47035 - Final Project 2020

Abstract— **Path planning for autonomous vehicles in various environments is an important problem, due to several constraints from vehicle dynamics and existence of surrounding vehicles. There are different methods for path planning such as RRT method, A* method and so on. In this paper, we perform path planning and obstacle avoidance of a car using Model Predictive Control (MPC). A car represented by a classical bicycle model is made to travel along the center lane of a road and maintain the trajectory. Two moving pedestrians are modelled as obstacles which the car avoids. After avoiding the obstacles, the car returns back to its original trajectory quickly with zero error.**

I. INTRODUCTION

In robot motion planning, a sequence of valid configurations of the robot is found that moves the robot from initial position to final position. An example of a robot can be an autonomous car avoiding obstacles present in its path. Several methods can be used for path planning and obstacle avoidance and each has its own specific purpose. To this end, MPC has been employed partly due to its ability to handle constraints which are necessary in the modeling of all physical system. The main idea of MPC is to solve a finite horizon optimal control problem at certain time steps and compute an optimal control sequence of which only the first control action is implemented. The procedure is then repeated at the next sampling step as new information about the state has been obtained from a model.

In this paper, motion and path planning of an autonomous car is performed for a moving obstacle such as a pedestrian using MPC. Due to its capability of systematically handling nonlinear time-varying models and constraints as well as operating close to the limits of admissible states and inputs, MPC has been popularly used. Here, a bicycle model is considered for its simplicity and a simple MPC problem is formulated to avoid pedestrians. The controller will be developed and tested in simulations perform in MATLAB using CasADi nonlinear solver [1]. Current state of the art involves a nonlinear MPC problem which considers the rate of change of lane in order to have a comfortable and smooth lane change during path planning [2][3].

There also exist methods where the proposed MPC planner automatically selects the type of maneuvers under a unified optimization framework to be performed during lane change in an actual highway scenario for obstacle avoidance. Car with polygon shapes are also considered to avoid collision. Different scenarios such as highway driving, lane keeping and lane merging can be conducted [4]. Extended bicycle models which include tire longitudinal and lateral forces are also currently used as compared to the classical bicycle model used here [3]. Therefore, in this paper, we stick to a simple framework as discussed earlier.

The rest of the paper is defined as follows: In Section II, the vehicle model and set of equations of motions are described along with its workspace and configuration space. In Section III, the cost function of the MPC problem is defined along with the constraints on the model. The planned scenario to be performed is also defined and includes the pre-planned obstacle trajectory. In Section IV, the results are shown which show the path taken by the autonomous car to avoid the pedestrian. Finally in Section V, the limitations, conclusions and possible future work are discussed.

II. ROBOT MODEL

A kinematic model representing the lateral motion of a vehicle can under some specific assumptions described below is developed. In contrast to a dynamic model where the forces acting on the system are taken into account, the kinematic model is derived from mathematical relationships governing the system of the vehicle's motion. Furthermore, compared to higher fidelity vehicle models, the system identification on the kinematic model is easier because of the low number of system parameters needed to describe the system. In addition, studies have shown that the kinematic model is under some conditions able to actually perform similarly well compared to a more complex and computationally heavy dynamic model. For this application where the velocity of the vehicle as well as the lateral acceleration are low the kinematic model performs well [6]. This motivates the design of the controller based on a kinematic model.

The robot is an autonomous car driving along a road. A car is a non-holonomic system and can be described by different models. The simplest model would be a quarter car model but it does not have sufficient data for path planning. Hence, a non-linear bicycle model is selected for path planning problem. To define the non-linear bicycle model for a car, certain variables have to be considered as explained in the following subsection.

A. Kinematic Model

In a bicycle model, the body frame of the car is place at the center of the rear axle and the x-axis is along the center main axis of the car. Let v denote the speed of the car, ϕ the steering angle and wheelbase as L as shown in the figure below [5].

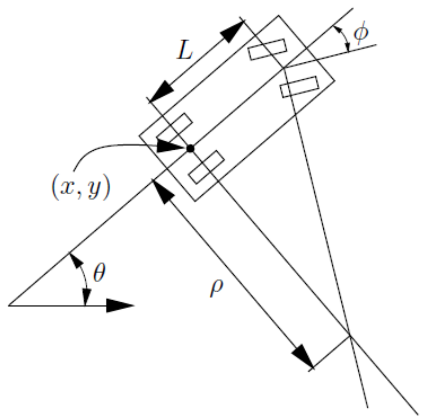

Fig. 1. Bicycle Model - Configuration Space

The model has four states:

- x - Global X position of the car center
- y - Global Y position of the car center
- θ - Heading angle of the car (counterclockwise positive)
- v - Speed of the car (positive)

There are two manipulated variables:

- ϕ - Steering angle (counterclockwise positive)
- δ - Throttle input to the car (positive when accelerating)

The nonlinear continuous-time equations that describe this model in an inertial frame are:

$$\begin{aligned} \dot{x} &= v \cdot \cos(\theta) \\ \dot{y} &= v \cdot \sin(\theta) \\ \dot{\theta} &= \frac{v}{L} \cdot \tan(\phi) \\ \dot{v} &= 0.5 \cdot \delta \end{aligned} \tag{1}$$

In the kinematic bicycle model it is assumed that the slip angles at both wheels are zero. Consequently the lateral component of the velocity is zero. This is reasonable assumption to make for low velocities.

Also note that in this model, it is assumed that both left and right front wheel have the same steering angle. In reality the inner wheel turns with greater angle due to the fact that it covers a shorter distance than that of the outer wheel. However, this difference is in this case small enough to be neglected.

The analytical Jacobian of nonlinear model are used to build the linear prediction model at a nominal operating point. They are given as:

$$\begin{aligned} \dot{x} &= v \cdot \sin(\theta) \cdot \theta + \cos(\theta) \cdot v \\ \dot{y} &= v \cdot \cos(\theta) \cdot \theta + \sin(\theta) \cdot v \\ \dot{\theta} &= \frac{v}{L} \cdot \tan(\phi) + \frac{v}{L} \cdot (\tan(\phi)^2 + 1) \cdot \phi \\ \dot{v} &= 0.5 \cdot \delta \end{aligned} \tag{2}$$

The equation of motion above relates system inputs to the change in configuration and this would be the update equation for the defined problem.

B. Workspace and Configuration Space

The workspace for a car which is a non-holonomic system as the constraints are expressed as a function of the car velocity as well as seen in equation (2). The workspace here is defined as:

$$W = \mathbb{R}^2 \tag{3}$$

For the configuration space, we define a space where the car can move in x and y directions and also rotate in a restricted manner. The configuration space is described by the Fig (1). It is defined as:

$$C = \mathbb{R}^2 \times S^1 \tag{4}$$

The configuration space defined above can be seen as the manifold of vehicle transformations in the absence of any collision constraints. In motion planning, the configurations that either cause the robot to collide with obstacles or cause some specified links of the robot to collide with each other should be removed. The removed part of the configuration space is referred to as the obstacle region. The leftover space is precisely what a solution path must traverse. A motion planning algorithm must find a path in the leftover space from an initial configuration to a goal configuration.

III. MOTION PLANNING

In this section, the path planner chosen is described. As explained earlier, an MPC approach is taken for path planning. MPC is an optimisation problem wherein it solves a sequence of finite-time trajectory optimisation problems in a receding horizon. It takes into account the update of the environment states during its path planning course. The principal advantages of MPC are:

- The constraints are taken into account in the design itself.
- Let solve problems with linear and nonlinear systems or variable and multi-variable systems without changing the controller formulation.
- The path planning or optimization is done online as the environment is dynamic with the pedestrian moving across.

Some challenges with MPC would be that it takes time to compute the solution online. However, in this case, a bicycle model is chosen which should make computations easier. The structure of the path planning (similar for kinematic model) based on MPC is considered [2]. The outputs which generate the driving path are derived through the MPC.

In this paper, the value of the environmental sensors is assumed that accurately obtained and there is no error or disturbance. The outputs which are derived from the cost function generate a path and the outputs are returned to the MPC.

A. Problem Description

A car is travelling along the center lane of a road and maintains a desired velocity. Two moving pedestrians are modelled as obstacles which the car avoids. A predicted horizon of 10 steps is considered. Constraints on the steering angle ϕ and throttle δ are given. The car is given a definite rectangular boundary. After avoiding the obstacles, the car should be able to return back to its original trajectory quickly with zero error.

B. Cost Function and Constraints

Constraints are necessary to define the areas to avoid and to provide restrictions on certain parameters such as the inputs so that they are feasible. The cost function to be minimized for the path planning problem and obstacle avoidance is given as,

$$\begin{aligned} arg \min_{x,y,\theta,\delta,\phi} \Big(&\sum_{k=1}^{N} \frac{1}{2} x_k^2 + \sum_{k=1}^{N} \frac{1}{2} \theta_k^2 \\ & - 50 \cdot log \sum_{k=1}^{N/2} ((x_k - v_k) + (y_k - 50))^2 \\ & - 50 \cdot log \sum_{k=1}^{N/2} ((x_k - v_k) + (y_k - 20))^2 \\ & + \sum_{k=1}^{N} (\delta_k - \delta_{k+1})^2 + \sum_{k=1}^{N} (\phi_k - \phi_{k+1})^2 \Big) \end{aligned} \tag{5}$$

where,

- 1^{st} and 2^{nd} term are for deviation from lane and straight position.
- 3^{rd} and 4^{th} term are weighted error between car position and pedestrian position.
- 5^{th} term is the throttle error.
- 6^{th} term is the steering error.

The cost function subjects to certain constraints. The following linear constraints are implemented for MPC path planner:

- Steering angle constrained between -0.5 and 0.5 rad.
- The throttle δ (acceleration or deceleration) is restricted between 1 and -1 (Full throttle/Full braking).

Therefore, the overall constraints are written as,

$$\begin{aligned} x(k+1) = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \\ -1 \leq \delta \leq 1 \\ -0.5 \leq \phi \leq 0.5 \end{aligned} \tag{6}$$

It should also be noted here that since we do not travel at a speed higher than 50 km/h, we do not consider the effect of steering angle on the speed.

C. Predefined Obstacle Trajectory

The pedestrian is crossing the road at a constant speed of 2 km/h and is shown as a blue dot in Fig (4). A boundary is defined for the pedestrian so the the car does not intersect with the pedestrian boundary. In total, there are 2 pedestrians crossing the road.

IV. RESULTS

A. Simulation Setup

The MPC formulation in Section III was solved on MATLAB using the non-linear CasADi solver package. The problem was solved on an i7 processor laptop with 16GB of RAM. The total runtime is approximately 26 seconds. The static figures below do not easily convey the evolution in time of the vehicle trajectories.

As seen in Fig (2), the car represented by the green rectangle initially follows the center lane as desired. Upon encountering a moving pedestrian represented in a blue circle with a defined circular boundary, the car slows down and turns away from the pedestrian and gets back on to the center lane as soon as possible. The same process is followed for the second pedestrian as well. After avoiding the pedestrians, the car comes back to the center lane and follows the same until full simulation time. This successfully shows the implementation of a basic path planning for a car using MPC with obstacle avoidance. The car ensures that its desired trajectory is followed even after encountering an obstacle.

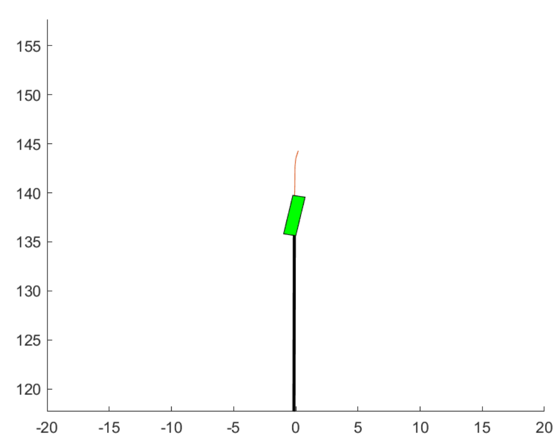

Fig. 2. Car in Motion

The profile of the inputs to the car, namely the steering angle ϕ and the throttle δ are shown in the figure below.

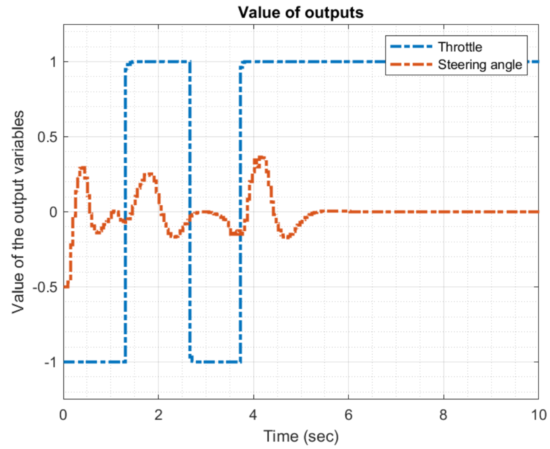

Fig. 3. Throttle and Steering Angle Input

The overall trajectory of the car is shown in the figure below.

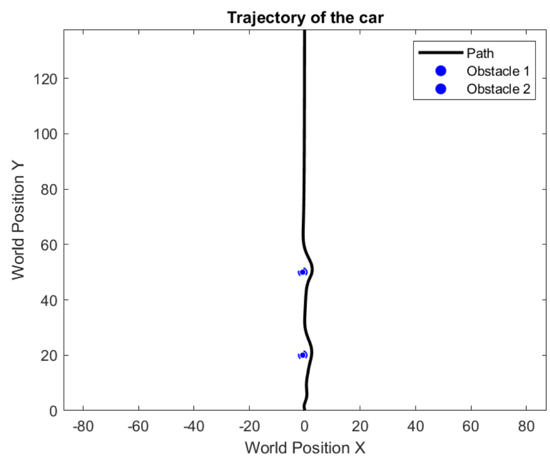

Fig. 4. Path Trajectory of the Car

Thus, the car is able to achieve the required goal wherein it maintains its trajectory and avoids the pedestrians.

V. CONCLUSION

A. Implementation

In this paper, an algorithm to evaluate two main scenarios, which are, to avoid an obstacle and follow a trajectory using MPC is proposed. MPC provides a locally optimal solution for the given cost function and constraints only if the problem is convex. In this case, the local solution is also a global solution since the cost function is convex. However, the accuracy of the path depends on the kinematic model of the system chosen. A more accurate/complex kinematic model will tend to give a better solution. The solution here is however complete as the car is able to reach its goal of maintaining its trajectory. The car is defined as a classical bicycle model with non-linear kinematics. In order to utilize MPC, the model can be linearized for simplicity or a non-linear solver can be used. Using a non-linear solver involves high computation which can take time if implemented online in an actual vehicle. However, since the model is simple, with the usage of high end processors, a non-linear solver CasADi is used here.

In this case, the pedestrians were avoided fairly successfully and the car was able to maintain its desired trajectory after that although not in an accurate manner.

B. Limitation

The car follows a defined trajectory to maintain its position in the lane center and avoids obstacles, however, it can be seen that the car orientation does not accurately define its heading. This is because the model is a classical bicycle model with just a steer wheel and a driven wheel. With a planar model of the car, it would be a more accurate representation and would provide precise motion. However, it would be computationally intense, making it an unsuitable choice for path planning in the case of autonomous cars in actuality.

C. Improvement

The problem can be further improved upon by including the vehicle dynamics such as tire, suspensions and steering into account for modelling the kinematics of the car. This would give better accuracy of the car model and improve motion planning. More constraints such as rate of change of steering in order to avoid jerk can also be included. Also, it can be adapted to more complex situations such as parking or lane merging on highways as in [4]. The most optimal way of implementing MPC would be for close range planning such as obstacle avoidance while utilizing other path planning algorithms such as RRT or RRT* for trajectory planning. This is because, defining a very large horizon in case of path planning would again involve huge processing of data and since MPC is typically local, it would not yield a global solution for every scenario.

REFERENCES

[1] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi – A software framework for nonlinear optimization and optimal control, Mathematical Programming Computation, In Press, 2018 https://web.casadi.org/

[2] J. C. Kim, D. S. Pae and M. T. Lim, Obstacle Avoidance Path Planning Algorithm Based on Model Predictive Control, 2018 18th International Conference on Control, Automation and Systems (ICCAS 2018), Oct. 17 20, 2018

[3] V. Turri, A. Carvalho, H. E. Tseng K. H. Johansson, F. Borrelli, Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013

[4] C. Liu, S. Lee, S. Varnhagen, H. E. Tseng, Path Planning for Autonomous Vehicles using Model Predictive Control, 2017 IEEE Intelligent Vehicles Symposium (IV), June 11-14, 2017

[5] S. M. LaValle, Planning Algorithms, Cambridge University Press, 2006

[6] P. Polack, F. Altché, B. d'Andréa-Novel and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, 2017, pp. 812-818.