

Model Predictive Control for Autonomous Driving considering Actuator Dynamics

Mithun Babu¹, Raghu Ram Theerthala¹, Arun Kumar Singh², Baladhurgesh B.P.¹,
Bharath Gopalakrishnan¹, K. Madhava Krishna¹, Shanti Medasani³

Abstract—In this paper, we propose a new model predictive control (MPC) formulation for autonomous driving. The novelty of our MPC stems from the following results. Firstly, we adopt an alternating minimization approach wherein linear velocities and angular accelerations are alternately optimized. We show that in contrast to the joint optimization, the alternating minimization exploits the structure of the problem better, which in turn translates to reduction in computation time. Secondly, our MPC explicitly incorporates the time dependent non-linear actuator dynamics that captures the transient response of the vehicle for a given commanded velocity. This added complexity improves the predictive component of MPC resulting in improved margin of inter-vehicle distance during maneuvers like overtaking, lane-change, etc. Although, past works have also incorporated actuator dynamics within MPC, there has been very few attempts towards coupling actuator dynamics to collision avoidance constraints through the non-holonomic motion model of the vehicle and analyzing the resulting behavior. We use a high fidelity simulator to benchmark our actuator dynamics augmented MPC with other related approaches in terms of metrics like inter-vehicle distance, trajectory smoothness, and velocity overshoot.

I. INTRODUCTION

Autonomous driving represents an interesting research problem which is at the intersection of many different fields including robotics and control. Motion planning is a core component of any autonomous driving set-up. Off late, there has been an upsurge in applying model predictive control (MPC) based formulation for navigation of autonomous vehicles [1], [2], [3]. The motivation behind this is clear. MPC provides a unified optimization based approach which can handle arbitrary constraints on state and control while minimizing a user-defined cost function. In spite of the recent success of MPC on autonomous driving, some key bottlenecks still remain. Among these include improving the computational efficiency of the underlying optimization and improving the predictive component of MPC to better reflect the actual behavior of the vehicle. In this paper, we make contributions towards both these problems.

On the optimization front, we make a case for adopting the alternating minimization (AM) approach which alternately searches in the space of angular accelerations and linear velocities. More precisely, the optimization operates in two separate layers wherein at the first layer, the angular

accelerations are optimized while fixing the linear velocities. Subsequently, at the second layer, the linear velocities are optimized while the angular accelerations are fixed to the values obtained at the first layer and so on. We show that AM approach has a very distinct advantage. For a given angular acceleration profile, the non-holonomic motion model reduces to an affine form with respect to linear velocity. This in turn leads to a difference of convex structure in the velocity optimization layer which allows us to solve it efficiently using the *convex-concave procedure* [4], [5]. Although AM has been extensively used in robotics and machine learning applications, we are not aware of any existing works which applies it to simplify trajectory optimization with non-holonomic motion model.

Our second contribution lies in incorporating actuator dynamics in the underlying optimization of our MPC. To be precise, we incorporate the time dependent non-linear mapping between the commanded velocity and the actual body velocity attained by the vehicle. The idea of MPC with actuator dynamics is not new. However, in the context of autonomous vehicles, existing works around this idea have been mostly restricted to either only path tracking control [6] or have considered collision avoidance only along the longitudinal direction of motion [7]. In contrast, the novelty of our approach stems from how we relate the actuator dynamics to the collision avoidance constraints through the non-holonomic motion model.

On the implementation side, we show the usefulness of our AM approach by showing reduction in computation time over the joint formulation that simultaneously optimizes linear velocities and angular accelerations. Furthermore, we use a high fidelity simulator to compare our actuator dynamics augmented MPC with a more conventional formulation with no-actuator dynamics and piece-wise constant acceleration input. The specific metrics used for comparison include margin of inter-vehicle distance and velocity oscillation during standard maneuvers like lane-change, overtaking and lane following. The rest of the paper is organized as follows: Section II gives an overview of the current work in this field, Section III describes the actuator dynamics in detail along with the formulation of cost function and constraints. Section IV, describes the alternating minimization routine and section V gives a detail evaluation of our AM approach.

II. RELATED WORKS

MPC for autonomous driving is an active area of research and there exists a lot of literature that justifies its potential.

¹ Robotics Research Center, International Institute of Information Technology, Hyderabad, India. ² TUIT, University of Tartu, Estonia. ³ MathWorks, Inc.

This work was partly supported by the MathWorks, Inc. Arun Kumar Singh was supported by Estonian Information Technology Foundation for Education, IT Academy program.

See [3], [8], [9] for some excellent surveys. For better comparison with our proposed work, we classify the existing MPC formulations into two categories. Those in the first category like [1], [10] formulate the underlying optimization of the MPC as a rigorous non-linear programming problem and then use iterative techniques like sequential convex programming [11] for the solution. The advantage of these approaches is that they consider the exact motion model of the vehicle and thus, are guaranteed to produce a kinematically feasible trajectory. The so called warm-start or using trajectories obtained at the previous iteration to initialize the optimization at the current iteration has been shown to speed up computation in practice. The formulations in the second category directly works with an affine approximation of the vehicle motion model [12], [13]. Consequently, the optimization becomes simpler although at the expense of obtaining trajectories that may not be kinematically feasible or even collision free. Our AM based approach falls into the first category. But in contrast to existing works, we try to exploit the inherent structure in the optimization leading to tangible computational improvements.

All the above cited works employ a sophisticated trajectory optimization but do not consider the actuator dynamics within the optimization. Most of the current MPC formulations with actuator dynamics either have not considered collision avoidance or have worked with simplified collision benchmarks. For example, [6] claims improved tracking performance by incorporating actuator dynamics within the MPC. Authors in [7] formulate cooperative adaptive cruise control for a fleet of vehicles and have considered collision avoidance but only along the longitudinal direction of motion. Along the same line, [14] builds a MPC with actuator dynamics but does not consider collision avoidance.

III. PROBLEM FORMULATION

In this section, we formulate the underlying optimization associated with our MPC formulation. We begin by introducing the motion model and actuator dynamics followed by systematic description of cost function and constraints of our optimization problem.

A. Motion model

The discrete time model of a non-holonomic autonomous vehicle is given by the following set of equations.

$$\begin{aligned} x(t_{i+1}) &= x(t_i) + v(t_i) \cos \theta(t_i) \Delta t. \\ y(t_{i+1}) &= y(t_i) + v(t_i) \sin \theta(t_i) \Delta t. \\ \theta(t_{i+1}) &= \theta(t_i) + \dot{\theta}(t_i) \Delta t + \frac{1}{2} \ddot{\theta}(t_i) \Delta t^2 \\ \dot{\theta}(t_{i+1}) &= \dot{\theta}(t_i) + \ddot{\theta}(t_i) \Delta t \\ \dot{\theta}(t_i) &= u_1(t_i) \\ v(t_i) &= f_{act}(t, v_c(t_i), v(t_{i-1})) \\ v_c(t_i) &= u_2(t_i) \end{aligned} \quad (1)$$

Where, $x(t_i)$, $y(t_i)$ and $\theta(t_i)$ are respectively the position and heading of the vehicle. The term $v(t_i)$ represents the linear velocity of the vehicle while $\dot{\theta}(t_i)$ corresponds to the angular velocity at time t_i . As shown the motion model is controlled by angular accelerations, $\ddot{\theta}(t_i)$ and velocity commands, $v_c(t_i)$. Furthermore, we assume that commanded

velocity gets translated to actual body velocity $v(t_i)$ of the vehicle through a non-linear time dependent function, $f_{act}(\cdot)$.

B. Actuator Dynamics

In the context of the proposed work, the role of actuator dynamics, $f_{act}(\cdot)$ is to model how actual body velocity varies with respect to the commanded velocity over time. As a simple example, consider a $f_{act}(\cdot)$ which relates $v(t)$ and $v_c(t_i)$ linearly over time. Thus, the relationship between $v(t)$ and $v_c(t_i)$ takes the following form:

$$v(t) = v(t_{i-1}) + \overbrace{\frac{v_c(t_i) - v(t_{i-1})}{t_i - t_{i-1}}}^{a(t_{i-1})} (t - t_{i-1}), \forall t \in (t_{i-1}, t_i] \quad (2)$$

It is clear that (2) is equivalent to non-holonomic motion model with piece-wise constant acceleration $a(t_{i-1})$ as the control input. In our work, we adopt a more expressive $f_{act}(\cdot)$ which accounts for the inherent relationship between $v(t_i)$ and $v_c(t_i)$.

First order system: We conducted several experiments with our autonomous vehicle hardware [15] and a high-fidelity simulator to analyze the response to a step velocity input. Some sample results are presented in Fig. 1(a)-1(c). Predominantly, our autonomous vehicle shows actuator dynamics similar to a first order system (1(a)). However, during decelerations (Fig. 1(b)), we observed occasional velocity overshoots indicating that the actual dynamics could have multiple modes. The response from our simulator (1(c)) is more typical of a second order system.

Motivated by the results from our autonomous vehicle hardware, we model our actuator dynamics as a first order system (3), where $v(t_{i-1})$ is the velocity acquired by the system at t_{i-1} instant and $v(t)$ is time varying response for a given step input. Model parameter τ is called time constant of the system. Finding τ analytically is difficult as many components of our vehicle actuation mechanism are difficult to model. So, we adopted a data-driven approach and estimated τ following a linear regression based on the response to different velocity step inputs. We note that our first order actuator dynamics is a simplified model and it leads to a simpler optimization problem (see equation (4)). Moreover, our extensive simulations show that even with a simulator which predominantly has second order actuator dynamics, our first order model could ensure collision avoidance with high fidelity. Finally, a first order model has only one parameter and thus require simpler system identification as compared to a second order actuator dynamics.

$$v(t) = v_c(t_i) + \left(v(t_{i-1}) - v_c(t_i) \right) \exp \frac{-(t-t_{i-1})}{\tau} \quad t \in (t_{i-1}, t_i] \quad , \quad (3)$$

Expression for vehicle velocity: If the system is subjected to a set of n commanded velocity inputs, then vehicle velocity $v(t_n)$ after n time-steps each of Δt duration, can be represented as

$$v(t_n) = (\sum_{i=1}^n v_c(t_i)(1 - m_i) \prod_{l=i+1}^n m_l) + v_0 \prod_{l=0}^n m_l. \quad (4)$$

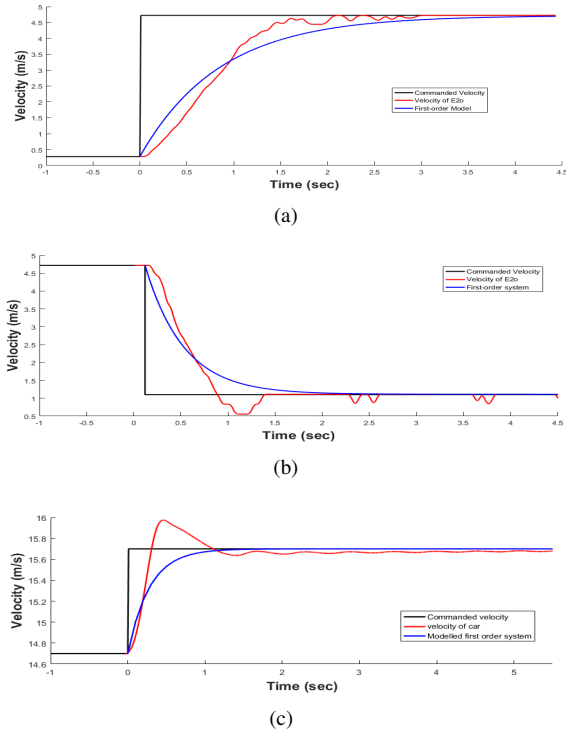


Fig. 1. Figures show response to a step velocity input. The commanded velocity is shown in black, the response in red and the fitted first order model in blue. (a) and (b) show results from our autonomous vehicle hardware a Mahindra E20 [15]. Predominantly, the observed response was similar to a first order system. However, occasionally during acceleration, the vehicle did show velocity overshoots. The response from our simulator shown in (c) is more typical of a second order system.

$$m_i = \exp^{-\Delta t/\tau} \quad (5)$$

The usefulness of (4) stems from the fact that it provides time varying vehicle velocity response purely as a function of the commanded velocity inputs $v_c(t_i)$. Importantly, (4) is affine with respect to $v_c(t_i)$.

C. Optimization

The underlying optimization of our MPC can be described in the following manner:

$$\arg \min_{\ddot{\theta}(t_i), v_c(t_i)} J = J_{smooth} + J_{goal} \quad (6a)$$

$$\mathbf{x}(t_{i+1}) = \mathbf{f}(\mathbf{x}(t_i), \mathbf{u}(t_i)). \quad (6b)$$

$$\dot{\theta}_{min} \leq \dot{\theta}(t_i) \leq \dot{\theta}_{max} \quad (6c)$$

$$\ddot{\theta}_{min} \leq \ddot{\theta}(t_i) \leq \ddot{\theta}_{max} \quad (6d)$$

$$v_c(t_i) \leq v_{max} \quad (6e)$$

$$a_{min} \leq \frac{v_c(t_i) - v(t_{i-1})}{\Delta t} \leq a_{max} \quad (6f)$$

$$-\kappa_{max} v(t_i) \leq \dot{\theta}(t_i) \leq \kappa_{max} v(t_i) \quad (6g)$$

$$c_{obst}(x(t_i), y(t_i), x_i(t_i), y_i(t_i), R_i) \leq 0 \quad (6h)$$

$$J_{smooth} = \underbrace{\sum_{i=1}^N \ddot{\theta}(t_i)^2}_{J_{\theta}} + \underbrace{\sum_{i=1}^N \left(\frac{v_c(t_{i-1}) - 2v_c(t_i) + v_c(t_{i+1}))}{\Delta t^2} \right)^2}_{J_v} \quad (7)$$

$$J_{goal} = (x(t_N) - x_f)^2 + (y(t_N) - y_f)^2 + (\theta(t_N) - \theta_f)^2 \quad (8)$$

Where, $\mathbf{x}(t_i) = [x(t_i), y(t_i), \theta(t_i), \dot{\theta}(t_i)]$ and represents the state of the system. The vector valued function $\mathbf{f}(\cdot)$ is just a compact representation of (1), where we have used $\mathbf{u}(t_i) = [u_1(t_i), u_2(t_i)]$ as mentioned in 1. The cost function (6a) is a summation of smoothness and goal reaching costs (7)-(8). As can be seen from (7), smoothness cost penalizes high value of angular accelerations and jerk modeled as a second order finite difference of linear velocity. The terminal cost ensures that the obtained trajectory terminates as close as possible to the goal position (x_f, y_f) . The equality (6b) constrains the control variables and states to be compatible with the motion model of the robot. The inequalities (6c)-(6d) represent the bounds on angular velocities and accelerations respectively. Inequality (6e) ensures that the commanded velocity is less than the physical limit of the vehicle. In (6f) we have modeled acceleration as a finite difference between the vehicle velocity at t_{i-1} and the next commanded velocity. Inequalities, (6g) represent the curvature bounds for the vehicles. Note how these have been written with respect to the actual body velocity and not the commanded velocity. Inequalities (6h) models the collision avoidance constraints and has the following algebraic form:

$$c_{obst}(\cdot) \leq 0 : -(x(t_i) - x_i(t_i))^2 - (y(t_i) - y_i(t_i))^2 + R_i^2 \leq 0. \quad (9)$$

Where, $x_i(t_i)$ and $y_i(t_i)$ describe the position of i^{th} obstacle. Following [1], [16], we model our ego-vehicle and the neighboring obstacles as an overlap of circles. Consequently, R_i represents the combined radius of each circle of the ego-vehicle and i^{th} obstacle. For static obstacles, the position would be independent of t_i . The form of (9) assumes that the vehicle and the obstacles are both modeled as circles. It is convenient to obtain an affine approximation for (9) by linearizing it around a guess trajectory $(\hat{x}(t_i), \hat{y}(t_i))$. In (10), ∇_x, ∇_y stands for partial derivative of $c_{obst}(\cdot)$ with respect to $x(t_i)$ and $y(t_i)$ respectively. Further, $\hat{c}(\cdot)$ is obtained by evaluating right hand side of (9) at $(\hat{x}(t_i), \hat{y}(t_i))$.

$$affine_{c_{obst}}(\cdot) = \hat{c}_{obst} + \nabla_x(x(t_i) - \hat{x}(t_i)) + \nabla_y(y(t_i) - \hat{y}(t_i)) \quad (10)$$

The core complexity of optimization (6a)-(6h) stems from the non-linear motion model (6b). This is because, for an affine motion model, (10) turns out to be globally valid convex approximation of the original collision avoidance constraints (9). [5], [4]. In other words, satisfaction of (10) ensures satisfaction of (9) as well. However, the non-linearity of non-holonomic motion model destroys this structure and makes the optimization more difficult.

In the next section, we propose an optimization routine which tries to explore as much as possible the inherent structure of the problem. The core idea hinges on the observation that if the heading trajectory of the vehicle is

fixed, the motion model becomes affine with respect to linear velocities.

IV. ALTERNATING OPTIMIZATION

Algorithm 1 summarizes the proposed alternating optimization routine. It starts with (line 1) choosing an initial guess for commanded velocity $\hat{v}_c^k(t)$, angular acceleration $\hat{\theta}^k(t)$ and a counter k along with two positive weights w_θ, w_v . The function *InitialTraj(.)* (line 3) computes an initial guess for position and heading trajectory, $(\hat{x}^k(t), \hat{y}^k(t), \hat{\theta}^k(t))$ using $\hat{v}_c^{k-1}(t)$ and $\hat{\theta}^{k-1}(t)$ in the motion model (1). Lines 6 and 12 represent the angular acceleration and linear velocity optimization layer respectively. Both the layers continue till the change in the cost functions between subsequent iteration is greater than the threshold ϵ and the collision avoidance constraints are not satisfied. The optimal solution obtained after each layer is used to update the initial guesses of angular accelerations and commanded velocities (lines 11 and 17).

A. Angular Acceleration Layer

The angular acceleration layer is obtained by extracting the $\theta(t)$ dependent terms from the optimization (6a)-(6h). The following points are worth pointing out.

- First, note the motion model $f^\theta(.)$ which is obtained by first order Taylor series expansion of the first two equations in (1) around $\hat{\theta}^k(t)$. Consequently, we obtain a motion model which is affine with respect to heading angle $\theta(t)$.
- The affine approximation holds only in the vicinity of $\hat{\theta}^k(t)$. Thus, a trust region needs to be incorporated to ensure that $\hat{\theta}^k(t)$ and $\hat{\theta}^{k+1}(t)$ are sufficiently close to each other. The last inequality in line 6 of Algorithm 1 which puts a box constraints on $\theta(t)$ serves this purpose. The trust region is modified and guess point is updated at each iteration as discussed in [5].
- The collision avoidance constraints have been augmented with a non-negative slack variable $s_\theta(t)$. This is to ensure that the Algorithm 1 continues to make progress towards the optimal solution even if the initial guess trajectory $(\hat{x}^k(t), \hat{y}^k(t))$ renders *affine_c(.)* infeasible (or in other words is not collision free). Consequently, we also incorporate a penalty on the slack variables in the cost function. The weights w_θ of the penalty is sequentially increased using a positive factor δ till $c_{obst}(.) > 0$.

B. Linear Velocity Layer

This layer has only such terms from the cost and constraint functions of optimization, (6a)-(6h) which explicitly depends on the $v_c(t_i)$. The following key points should be noted.

- Note, the motion model, $\mathbf{f}^v(.)$ which has been obtained from (1) by using $\hat{\theta}^k(t), \hat{\theta}^k(t), \hat{\theta}^k(t)$ obtained by the previous angular acceleration layer.
- The collision avoidance constraints are augmented with non-negative slacks $s_v(t)$ similar to angular acceleration

layer. The penalty on the slacks also follows the same reasoning.

- Note that this layer does not have any trust region constraints. This is because in this layer we do not make any linearization which in turn is due to the motion model being already affine in terms of forward velocity. The physical implication of this is that the velocity optimization layer can take as large as possible step towards the optimal solution [5]. In contrast, the progress of the angular acceleration layer is limited by the size of the trust region.

C. MPC

Algorithm 1 is solved in a receding horizon manner to serve as our MPC. At the start of the MPC, the Algorithm 1 is initialized with the output of a high-level planner such as [17]. In the subsequent iteration of the MPC, we adopt the warm-start approach where we initialize with the trajectory and controls obtained in the previous iteration.

V. RESULTS AND DISCUSSION

A. Benchmarking Alternating Minimization (AM)

Here we compare our proposed AM with the more conventional formulation where angular accelerations and linear velocities are simultaneously obtained. We prototyped both the approaches in MATLAB using CVX [18]. These simulations are ran in a Intel core i5-3230M @ 2.6 GHz CPU with 6GB RAM. The results are summarized in Fig.2(a)-2(b) and 3(a)-3(b).

Fig.2(a)-2(b) presents the comparison of the computational aspects in terms of run time and number of iterations. As shown, on an average, our proposed AM approach takes around 17% less time and 32% less iterations. As shown in Fig.3(a)-3(b), both our AM and joint formulation results in low smoothness cost. However, smoothness cost obtained with joint formulation is significantly lower than that obtained with our AM. It can be thus concluded that our proposed AM approach sacrifices a bit of optimality in the pursuit of computing reasonably smooth, kinematically feasible collision avoiding trajectories in quick time. We believe such a feature can be useful for autonomous driving.

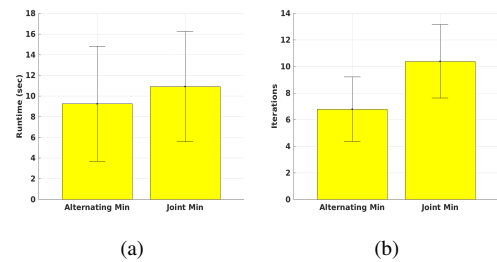


Fig. 2. (a) shows a comparison of runtime of both approaches. Proposed approach shows 1.8sec improvement in runtime. (b) shows comparison on number of iterations taken by both approaches in-order to converge to a similar solution -compared in 3(a)-3(b)

Algorithm 1 Alternating Optimization

```

1: Initialization: Initial guess for  $\hat{v}_c^k(t)$ ,  $\hat{\theta}^k(t)$ , iteration
   counter,  $k = 0$ ,  $w_\theta, w_v$ 
2:
3:  $(\hat{x}^k(t), \hat{y}^k(t)) = InitialTraj(\hat{v}_c^k(t), \hat{\theta}^k(t))$ 
4:
5: while  $|J_\theta^{k+1} - J_\theta^k| \geq \epsilon$  and  $|J_v^{k+1} - J_v^k| \geq \epsilon$  do
6:
   
$$\ddot{\theta}^k(t) = \arg \min J_\theta + \sum w_\theta s_\theta.$$

   
$$\mathbf{x}(t_{i+1}) = \mathbf{f}^\theta(\mathbf{x}(t_i), \mathbf{u}(t_i)).$$

   
$$\dot{\theta}_{min} \leq \dot{\theta}(t_i) \leq \dot{\theta}_{max}.$$

   
$$\ddot{\theta}_{min} \leq \ddot{\theta}(t_i) \leq \ddot{\theta}_{max}$$

   
$$-\kappa_{max} \hat{v}_c(t_i) \leq \dot{\theta}(t_i) \leq \kappa_{max} \hat{v}_c(t_i).$$

   
$$affine_{C_{obst}}(\ddot{\theta}(t_i)) - s_\theta(t_i) \leq 0.$$

   
$$s_\theta(t_i) \geq 0.$$

   
$$\theta^{k-1}(t_i) - \theta^{trust}(t_i) \leq \theta^k(t_i) \leq \theta^{k-1}(t_i) + \theta^{trust}(t_i).$$

7:   if  $C_{obst}(\cdot) > 0$  then
8:      $w_\theta \leftarrow w_\theta * \delta$ 
9:   end if
10:
11:    $\hat{\theta}^k(t) \leftarrow \ddot{\theta}^k(t)$ 
12:
   
$$v_c^k(t) = \arg \min J_v + \sum w_v s_v.$$

   
$$\mathbf{x}(t_{i+1}) = \mathbf{f}^v(\mathbf{x}(t_i), \mathbf{u}(t_i)).$$

   
$$v_c(t_i) \leq v_{max}.$$

   
$$a_{min} \leq \frac{v_c(t_{i+1}) - v_c(t_i)}{\Delta t} \leq a_{max}$$

   
$$-\kappa_{max} v(t_i) \leq \dot{v}(t_i) \leq \kappa_{max} v(t_i).$$

   
$$affine_{C_{obst}}(v_c(t_i)) - s_v(t_i) \leq 0.$$

   
$$s_v(t_i) \geq 0.$$

13:   if  $C_{obst}(\cdot) > 0$  then
14:      $w_v \leftarrow w_v * \delta$ 
15:   end if
16:
17:    $\hat{v}_c^k(t) \leftarrow v_c^k(t)$ 
18:    $k \leftarrow k + 1$ 
19: end while

```

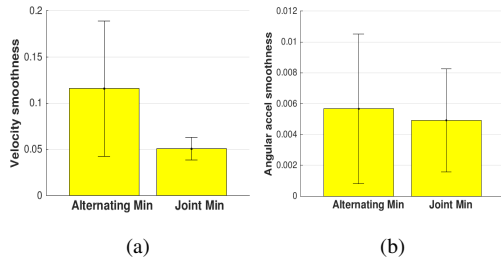


Fig. 3. (a) and (b) shows comparison of velocity smoothness and angular acceleration smoothness respectively. We observe reduction in smoothness of control profile due to alternating minimization

B. Effect of Actuator Dynamics

To analyze the effect of actuator dynamics, we integrated our MPC with a state of the art vehicle simulator. The simulator provides state feedback for the ego-vehicle along with the information about the state of the obstacle through a virtual LIDAR with a sensing range of 70m. For this implementation, we prototyped Algorithm 1 in C++ using Gurobi solver [19] obtaining significant speed up. We were able to iterate our MPC at 10Hz with a planning horizon of 50 steps each of duration 0.1s. To ensure that the ego vehicle respects the road geometry, the boundaries of the road are modeled as imaginary obstacles and included into Algorithm 1. The bounds on velocity (v_{max}), acceleration (a_{max}), and deceleration (a_{min}) were kept at 25m/s, 4m/s², and -6m/s² respectively to make simulations closer to real-world scenarios.

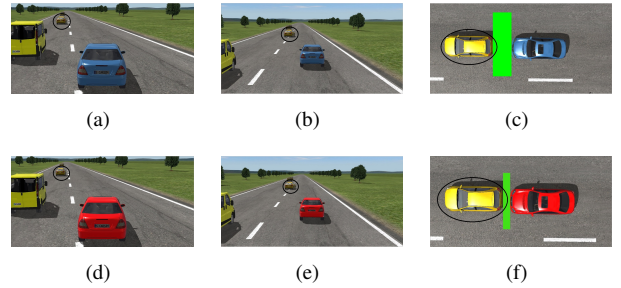


Fig. 4. (a)-(c) and (d)-(f) shows a scenario where in an occluded vehicle (marked with black circle) makes a sudden overtaking maneuver and comes within collision range of an ego vehicle. Vehicle in blue is modelled with actuator dynamics as in eqn. (3) and Vehicle in red is modelled with actuator dynamics as in eqn. (2). By comparing (d) and (h) we observe a significant improvement in inter-vehicle distance (marked in green)

1) Benchmark Scenarios: Occlusion during overtaking:

In this benchmark, the ego-vehicle traveling at 15m/s initiates an overtaking maneuver. Midway during the maneuver, it notices another vehicle (which was previously occluded from its field of view) also performing a similar maneuver. In such a situation, the ego-vehicle should decelerate quickly to 12m/s to avoid collision. Fig. 4(a)-4(c) shows the occlusion scenario, wherein the ego-vehicle is shown in blue. The vehicle it is overtaking is the yellow van and the occluded vehicle is marked with a black circle. Fig.4(d)-4(f) repeats the same benchmark with an ego-vehicle (shown in red) that uses MPC with actuator dynamics (2) to compute its motion. Fig.5(a),5(b) shows the plot of inter-vehicle distance and vehicle velocity for a specific portion of time. It can be seen that the ego-vehicle that incorporates actuator dynamics (3) within MPC can pro-actively anticipate the transient velocity response and thus de-accelerate faster resulting in improved inter-vehicle distance. The complete velocity profile is shown in Fig.5(c) and as shown it exhibits less overshoot and oscillation with the incorporation of the actuator dynamics.

Lane change : In this benchmark, the ego-vehicle moving at a speed of 10m/s performs a lane change maneuver to an adjacent high speed lane (15m/s). In this situation, a delay in achieving the increased velocity would bring the ego-vehicle

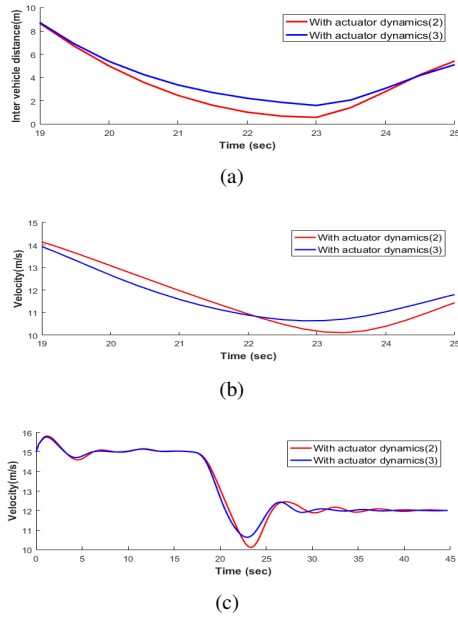


Fig. 5. (a) shows inter vehicle distance between ego vehicle and the leading vehicle during critical situation using both actuator models. (b) and (c) show the velocity plot during the maneuver. We can notice advantage of our approach in terms of decreased velocity overshoot and improved damping.

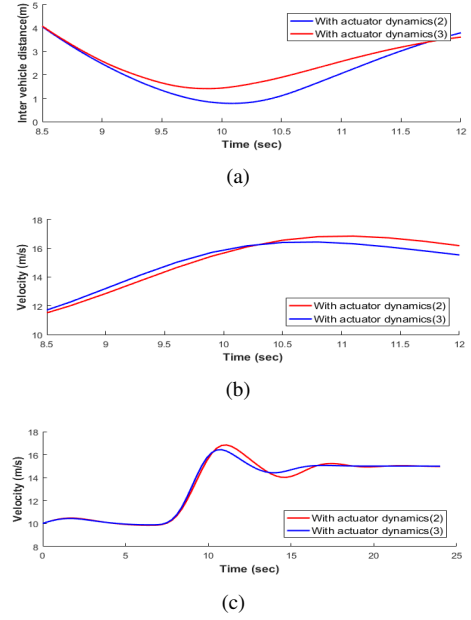


Fig. 7. (a) plots inter vehicle distances between the ego vehicle and obstacle in rear using both actuator models. The blue curve maintains better inter vehicle distances compared to the red curve, thus validating our approach. (c) signifies the advantage of our approach in terms of decreased velocity overshoot and improved damping.

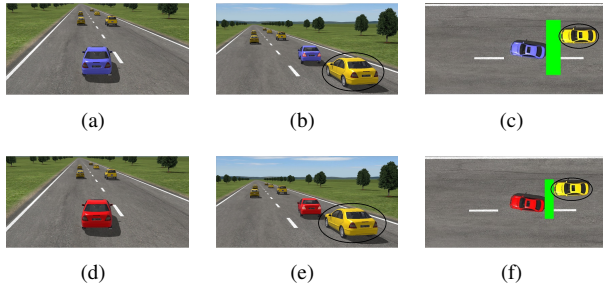


Fig. 6. (a)-(c) shows a lane change scenario with an ego vehicle (blue) with actuator dynamics modelled as in eqn.(3). While the ego vehicle is executing the lane change, a vehicle from the rear (marked with black circle) suddenly comes into the collision range of the ego vehicle. The shaded region in green is an indicator of the inter-vehicle distance. (d)-(f) repeats the same experiment with an ego vehicle (red) with actuator dynamics modelled as in eqn.(2). The improvement in the inter-vehicle distance can be seen in (c).

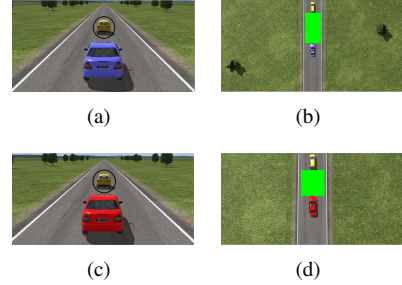


Fig. 8. (a)-(b) show a scenario where an ego vehicle (blue) tries to follow the yellow vehicle (marked in black circle) ahead. Later, this vehicle brake abruptly and consequently comes into the collision range. In (c)-(d), we show the performance of the ego vehicle (red) with actuator dynamics modelled as eqn.(2). The improvement in the inter-vehicle distance is shown by comparing (b) with (d).

in collision course with the vehicle approaching from the rear. Fig.6(a)-6(c) show the lane change scenario. The ego-vehicle is shown in blue while the rear vehicle is marked with a black circle. Similar to the previous benchmark, in Fig.6(d)-6(f), we repeat the simulation with a different ego-vehicle (shown in red) that has actuator dynamics modeled as eqn.(2). It can be seen from Fig.7(b) that use of actuator dynamics (3) allows the vehicle to accelerate faster to the required velocity and thus maintain higher clearance from the vehicle approaching from behind (Fig.7(a)). The complete vehicle velocity shown in Fig.7(c) shows improved transient response with reduced overshoot and oscillation when using actuator dynamics (3) within the MPC.

Sudden Braking: This benchmark is shown in Fig.8(a)-8(b). Here, the ego-vehicle (shown in blue) is following a

yellow vehicle (marked by a black circle) in the front at a speed of $15m/s$. Suddenly, the front vehicle de-accelerates bringing our ego-vehicle in collision course. The ego-vehicle responds by reducing its speed. Fig.8(c)-8(d) repeats the same simulation with an ego-vehicle (shown in red) with actuator dynamics (2). As shown in Fig.9(b), bringing actuator dynamics explicitly within the MPC allowed the ego-vehicle to achieve significantly faster de-acceleration. This in turn improves the inter-vehicle distance (Fig.9(a)). Similar to previous benchmarks, in Fig.9(c), we observe velocity profile with improved transient response with actuator dynamics (3) incorporated within the MPC.

Quantitative Comparison: The bar-graph shown in Fig. 10 summarizes the results on inter vehicle distance observed during previous benchmarks. Here, we also present an additional comparison that is based on actuator dynamics (eqn.3)

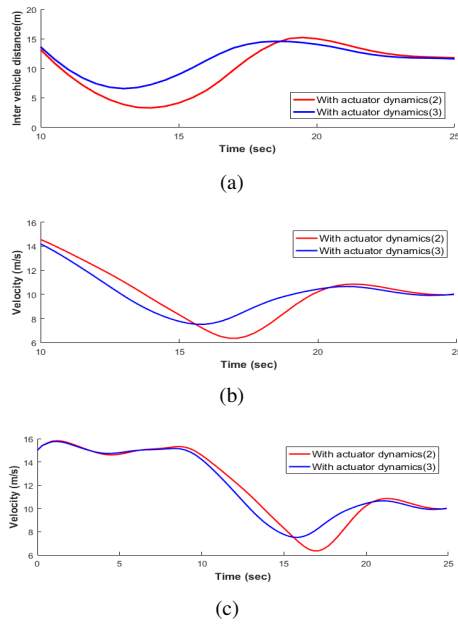


Fig. 9. (a) plots inter vehicle distance between the ego vehicle and leading vehicle using two different models of actuator dynamics. (c) further shows advantages of our approach in terms of decreased velocity overshoot and improved damping.

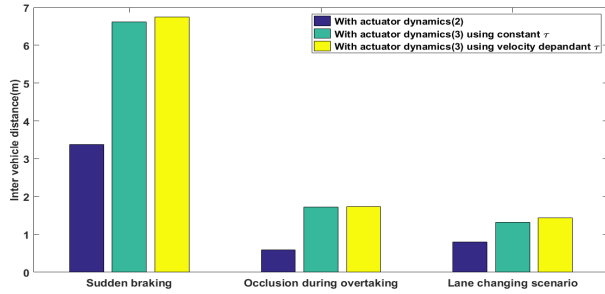


Fig. 10. Figure summarizes results on inter-vehicle distance obtained in previous benchmarks. It is clear that the our MPC based on AM and actuator dynamics (3) has a clear advantage while maneuvering in typical urban scenarios

but with a τ which varies with vehicle velocity $v(t)$. We used a function approximator based on radial basis function to learn the $\tau-v(t)$ mapping. As shown in Fig.10, we noticed only marginal benefit of using a complicated $v(t)$ dependent τ in our first order actuator dynamics (3).

VI. CONCLUSIONS AND FUTURE WORK

In this work, we built an MPC formulation, on a novel alternating minimization based optimization routine coupled with a non-holonomic motion model with a first order actuator dynamics. We performed extensive simulations to show how incorporation of actuator dynamics within the MPC improves autonomous driving. In particular, we showed improved inter-vehicle distance during different maneuvers like overtaking, lane-change and vehicle-following. We also benchmarked our alternating minimization and showed that it can compute feasible collision avoiding trajectories faster while still achieving quite low smoothness cost.

There are several directions where our current formulation can be extended to remove some of the existing limitations. For example, one of our primary future focus is on incorporating steering actuator dynamics within our MPC and analyzing the resulting benefits in the context of autonomous driving. We are also working on implementing the proposed formulation on our autonomous car prototype.

REFERENCES

- [1] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1928–1935.
- [2] D. Kim, J. Kang, and K. Yi, "Control strategy for high-speed autonomous driving in structured road," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 186–191.
- [3] C. Katrakazas, M. Quddus, W.-H. Chen, and L. Deka, "Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 60, pp. 416–442, 2015.
- [4] X. Shen, S. Diamond, Y. Gu, and S. Boyd, "Disciplined convex-concave programming," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 1009–1014.
- [5] S. Boyd, "Sequential convex programming," *Lecture Notes, Stanford University*, 2008.
- [6] E. Kim, J. Kim, and M. Sunwoo, "Model predictive control strategy for smooth path tracking of autonomous vehicles with steering actuator dynamics," *International Journal of Automotive Technology*, vol. 15, no. 7, pp. 1155–1164, 2014.
- [7] N. Murgovski, B. Egardt, and M. Nilsson, "Cooperative energy management of automated vehicles," *Control Engineering Practice*, vol. 57, pp. 84–98, 2016.
- [8] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [9] A. Carvalho, S. Lefèvre, G. Schilbach, J. Kong, and F. Borrelli, "Automated driving: The role of forecasts and uncertainty control perspective," *European Journal of Control*, vol. 24, pp. 14–32, 2015.
- [10] A. Carvalho, Y. Gao, A. Gray, H. E. Tseng, and F. Borrelli, "Predictive control of an autonomous ground vehicle using an iterative linearization approach," in *Intelligent Transportation Systems (ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 2335–2340.
- [11] S. Boyd, "Sequential convex programming," *Lecture Notes, Stanford University*, 2008.
- [12] J. Nilsson, P. Falcone, M. Ali, and J. Sjöberg, "Receding horizon maneuver generation for automated highway driving," *Control Engineering Practice*, vol. 41, pp. 124–133, 2015.
- [13] J. Nilsson, M. Brännström, J. Fredriksson, and E. Coelingh, "Longitudinal and lateral control for automated yielding maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1404–1414, 2016.
- [14] Y. Luo, A. Serrani, S. Yurkovich, D. B. Doman, and M. W. Oppenheimer, "Model predictive dynamic control allocation with actuator dynamics," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2. IEEE, 2004, pp. 1695–1700.
- [15] A. Modh, S. Singh, A. Kumar, K. M. Krishna *et al.*, "Gradient aware-shrinking domain based control design for reactive planning frameworks used in autonomous vehicles," *arXiv preprint arXiv:1804.08679*, 2018.
- [16] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–574, 1998.
- [17] S. Sharma and M. E. Taylor, "Autonomous waypoint generation strategy for on-line navigation in unknown environments," *environment*, vol. 2, p. 3D, 2012.
- [18] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.
- [19] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: <http://www.gurobi.com>