This is an Author's Accepted Manuscript version of the paper:
H. Febbo, J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "Moving Obstacle Avoidance for Large, High-Speed Autonomous Ground Vehicles," American Control Conference, 2017.

Moving Obstacle Avoidance for Large, High-Speed Autonomous Ground Vehicles

Huckleberry Febbo, Jiechao Liu, Paramsothy Jayakumar, Jeffrey L. Stein, and Tulga Ersal*

***Abstract*—This work introduces the use of hard constraints to avoid moving obstacles for navigating a large, high-speed autonomous ground vehicle in an unstructured environment using nonlinear model predictive control in a single-level architecture, where path planning and tracking are combined into a single optimization problem. Additionally, the hard constraints approach is compared to the traditional approach in this context which implements obstacle avoidance by augmenting the obstacle avoidance requirements into the cost function as soft constraints. In both approaches, the control signals, which are steering angle command and reference longitudinal speed, are optimized using a nonlinear vehicle dynamics model, where the objective is to minimize the time-to-goal. Results indicate that the hard constraints approach outperforms the soft constraints approach both in terms of obstacle avoidance performance and optimization time.**

I. Introduction

Avoiding collisions with obstacles is an important problem for mobile robots, autonomous vehicles (AVs) and unmanned ground vehicles (UGVs). For computational efficiency, control of these vehicles is often carried out using a hierarchical scheme wherein a high-level path planner quickly generates a reference trajectory and then a vehicle-level controller is employed to track the reference trajectory. In the literature this hierarchical approach is refereed to as a two-level architecture [1], [2]. However, when it becomes necessary to push the vehicle to its dynamical limits by either minimizing time-to-goal or maximizing progress-on-track, as in racing situations [3], [4] or with military applications, the high-level path planner may create dynamically infeasible trajectories, because it often only considers simple vehicle dynamics. Additionally, collisions with obstacles may occur if the vehicle deviates from the reference trajectory, because vehicle-level controllers do not generally constrain the vehicle to avoid obstacles.

To mitigate these issues, several researchers optimize the control commands for the vehicle using a single-level architecture [1], [2]. In a single-level architecture, there is no reference trajectory available to the vehicle-level controller. Instead, path planning and vehicle-level control are carried out simultaneously. Thus, for a properly constrained system with an appropriate vehicle model, the trajectories generated for all feasible solutions will be both dynamically feasible and collision free. Additionally, when the goal of the optimization is to minimize time-to-goal or progress-on-track and the problem is setup using a single-level architecture, the entire state-space can be explored and the control signals that push the vehicle to its dynamic limits can be identified.

AVs and UGVs are often controlled using model predictive control (MPC) [2], [5]–[10]. MPC is capable of controlling complex nonlinear systems bound by nonlinear constraints and it works by optimizing the control signals using a model of the system over a given prediction horizon and subsequently executing a portion of these optimized signals.

Using a two-level architecture, where the higher level is a path planner and at the vehicle level MPC is used for path tracking, researchers controlled a sports vehicle to drive autonomously at high speeds along a mountain road [4]. Additionally, MPC has also been used to develop active steering algorithms to assist drivers in avoiding obstacles [11], to limit the driver input inside a safe handling envelope [5], and to stabilize a vehicle using an AV steering system [12]. In [11], authors compare the performance of an MPC controller, where the vehicle model is linearized at the beginning of the prediction horizon, with one where the vehicle model is linearized about a reference trajectory over the entire prediction horizon, an approach often referred to as linear time-varying MPC. In [5], authors develop an active steering algorithm that optimizes the front lateral tire forces in lieu of steering angle to formulate a convex optimization problem that can be solved quickly. In [12] authors introduced the notion of using nonlinear MPC (NLMPC) to control the steering angle of a vehicle along a reference trajectory. However, all of these optimizations utilize a two-level architecture and time-to-goal minimization or progress-on-track maximization are not explicitly considered.

Compared to the two-level architecture approach, there has been much less work that focuses on controlling AVs and UGVs using a single-level architecture, but favorable results have been reported. In particular, researchers control $1:43$ scale race cars in highly nonlinear operating regimes while maximizing progress-on-track [2]. However, [2] focused on small radio-controlled race cars, whereas this work focuses on large vehicles with a high center of gravity such as a High Mobility Multipurpose Wheeled Vehicle (HMMWV),

Manuscript submitted September, 2016. The authors wish to acknowledge the financial support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-14-2-0001 U.S. Army Tank Automotive Research, Development and Engineering Center (TARDEC) Warren, MI.

H. Febbo, J. Liu, J. L. Stein, and T. Ersal are with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: febbo@umich.edu; ljch@umich.edu; stein@umich.edu; tersal@umich.edu).

P. Jayakumar is with the U.S. Army RDECOM-TARDEC, Warren, MI 48397 (email: paramsothy.jayakumar.civ@mail.mil)

*Corresponding author (tersal@umich.edu)

where rollover is a major concern and must be accounted for in the constraints. In [8], an NLMPC algorithm that uses a single-level architecture is introduced that operates a large AV in unstructured environments (without lanes or traffic rules) to optimize the steering angle in order to minimize time-to-goal while avoiding static obstacles. This work is extended to include the optimization of reference longitudinal speed in addition to the steering angle in [9]. The obstacles, however, were still considered to be static.

Using a single-level architecture, researchers utilized soft constraints for moving obstacle avoidance while considering the vehicle's dynamical limits using NLMPC, but this approach does not guarantee obstacle avoidance for a feasible solution and the focus was on a low speed vehicle with a short prediction horizon [10]. Obstacle avoidance can only be guaranteed for a feasible solution if it is implemented using hard constraints.

In summary, a hard constraints approach to handling moving obstacles using a single-level architecture for large vehicles with significant dynamics has not yet been investigated.

This work aims to fill this gap by developing an optimal control formulation that uses hard constraints to avoid moving obstacles using a single-level architecture based on the prior effort in [8], [9], [13], [14]. It is assumed that an obstacle detection and tracking algorithm such as the one developed in [15] is utilized, so that both the shapes and time-varying positions of all obstacles are known. Additionally, the soft constraints approach is also implemented into the algorithm and compared to the hard constraints approach. It is shown that the hard constraints method both yields a better obstacle avoidance performance and reduces optimization time when compared to the soft constraints approach.

Therefore, the novel and salient contributions of this work are:

1) Using hard constraints in a single-level optimal control architecture to avoid collisions with moving obstacles for a large, high-speed UGV in an unstructured environment,
2) Comparing soft constraints to hard constraints in current context.

The remainder of this paper is organized as follows. Section II presents the overall problem formulations for both hard constraints and soft constraints methods. In Section III two examples are investigated that compare the hard and soft constraints methods. Finally, in Section IV the work is summarized and conclusions are given.

II. Problem Formulation

This work leverages the single-level optimal control problem (OCP) formulation in [9] and modifies it to accommodate moving obstacles using the hard constraints approach. As a benchmark, the soft constraints approach is also implemented. For simplicity and space limitations, these two approaches are compared by comparing the OCP solutions without closing the loop.

The controller executes the solution from the following general OCP:

$$\underset{\boldsymbol{\xi},\,\boldsymbol{\zeta},\,T_p}{\text{minimize}} \quad J = \mathcal{T}\left(T_p\right) + \left\{\int_0^{T_p} \mathcal{I}\left[\boldsymbol{\xi}\left(t\right),\boldsymbol{\zeta}\left(t\right)\right]\mathrm{d}t\right\} \tag{1}$$

$$\text{subject to} \quad \dot{\boldsymbol{\xi}}\left(t\right) = \boldsymbol{\mathcal{V}}\left[\boldsymbol{\xi}\left(t\right),\boldsymbol{\zeta}\left(t\right)\right] \tag{2}$$

$$\boldsymbol{\xi}_{\min}\left(t\right) \leq \boldsymbol{\xi}\left(t\right) \leq \boldsymbol{\xi}_{\max}\left(t\right) \tag{3}$$

$$\boldsymbol{\zeta}_{\min}\left(t\right) \leq \boldsymbol{\zeta}\left(t\right) \leq \boldsymbol{\zeta}_{\max}\left(t\right) \tag{4}$$

$$\boldsymbol{\mathcal{F}}\left[\boldsymbol{\xi}\left(T_p\right),\boldsymbol{\xi}\left(0\right)\right] \leq 0 \tag{5}$$

$$\boldsymbol{\mathcal{R}}\left[\boldsymbol{\xi}\left(t\right)\right] \leq 0 \tag{6}$$

$$\boldsymbol{\mathcal{S}}\left[\boldsymbol{\xi}\left(t\right)\right] \leq 0 \tag{7}$$

$$T_p \leq T_{p,\max} \tag{8}$$

When Eq. (1) is minimized subject to the constraints in Eq. (2) - Eq. (8), the optimal control vectors $\boldsymbol{\zeta}$, state vectors $\boldsymbol{\xi}$, and prediction time T_p can be calculated. In the following sections, these equations are expanded and described starting with the constraints and finishing with the cost function.

There are several sets of constraints that are identical in both the hard constraints and soft constraints approaches; the vehicle dynamics must be satisfied (Eq. (2)), both the state and control trajectories must lie within their respective bounds (Eq. (3) and Eq. (4)), the vehicle must reach the goal (Eq. (5)), the vehicle must avoid collisions with moving obstacles (Eq. (6)), the maneuver must be dynamically safe (Eq. (7)), and the final prediction time must be less than $T_{p,\max}$ (Eq. (8)). The difference between the two approaches is the way that obstacle avoidance is implemented and is described in detail below.

Eq. (2)*: Vehicle Dynamics*

The level of model fidelity necessary in the model predictive controller for a large UGV represented by a 14 Degree of Freedom (DoF) vehicle model is well captured by a 3 DoF vehicle model with nonlinear bounds on acceleration/deceleration, a nonlinear tire model, and longitudinal load transfer [13], [14]. The NLMPC vehicle model, governed by the state space equation shown in Eq. (9), is leveraged within the OCP to identify to control inputs over the prediction horizon.

$$\dot{\boldsymbol{\xi}} = \boldsymbol{\mathcal{A}}\left(\boldsymbol{\xi}\right) + \boldsymbol{\mathcal{B}}\boldsymbol{\zeta} \tag{9}$$

where

$$\boldsymbol{\mathcal{A}}(\boldsymbol{\xi}) = \begin{bmatrix} U\cos\psi - (V + L_f\omega_z)\sin\psi \\ U\sin\psi + (V + L_f\omega_z)\cos\psi \\ \omega_z \\ a_x \\ (F_{y,f} + F_{y,r})/M_t - U\omega_z \\ (F_{y,f}L_f - F_{y,r}L_r)/I_{zz} \\ 0 \\ 0 \end{bmatrix} \text{ and } \boldsymbol{\xi} = \begin{bmatrix} x \\ y \\ \psi \\ U \\ V \\ \omega_z \\ \delta_f \\ a_x \end{bmatrix}$$

$$\boldsymbol{\mathcal{B}^T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{\zeta} = \begin{bmatrix} \gamma_f \\ J_x \end{bmatrix}$$

where the state and control vectors are $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$, respectively, a_x and J_x represent the longitudinal acceleration and jerk,

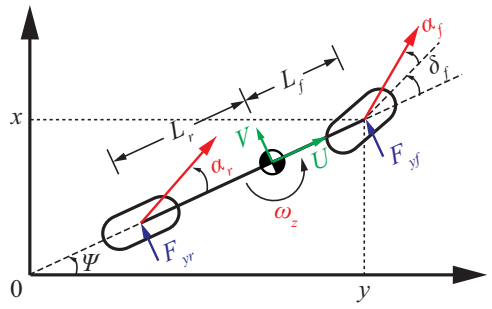

Fig. 1. The 3 DoF vehicle model used in the optimal control problem formulation [9]

respectively, γ_f is the steering rate, M_t is the total vehicle mass, and I_{zz} is the moment of inertia about the center of mass. As shown in Fig. 1, ψ represents the heading angle, L_f and L_r represent the distances between the front and rear axles to the center of mass, respectively, U and V represent the longitudinal and lateral speeds, respectively, x and y describe the global position of the center of the front axle, δ_f is the front steering angle, ω_z is the yaw rate, $F_{y,f}$ and $F_{y,r}$ represent the front and rear lateral tire forces, respectively, and finally, α_f and α_r represent the front and rear tire slip angles, respectively.

Eq. (3)-*Eq.* (4)*: State and Control Bounds*

γ_f and J_x are chosen as control variables to achieve smoother responses for δ_f and a_x, respectively. Additionally, γ_f and J_x are added to the optimization so that they can be bounded at each instant in time (t) based on the physical limits of the vehicle:

$$\gamma_{f,\min} \leq \gamma_f(t) \leq \gamma_{f,\max} \tag{10}$$
$$J_{x,\min} \leq J_x(t) \leq J_{x,\max} \tag{11}$$

Nonlinear acceleration/deceleration bounds, determined by studying the acceleration/deceleration limits of the 14 DoF plant model in the previous work [9], are incorporated, which are functions of longitudinal vehicle speed:

$$a_{x,\min}[U(t)] \leq a_x(t) \leq a_{x,\max}[U(t)] \tag{12}$$

Additional bounds based on both the vehicle limits and the desired vehicle behavior are placed on x, y, U, ψ, and δ_f:

$$x_{\min} \leq x(t) \leq x_{\max} \tag{13}$$
$$y_{\min} \leq y(t) \leq y_{\max} \tag{14}$$
$$U_{\min} \leq U(t) \leq U_{\max} \tag{15}$$
$$\psi_{\min} \leq \psi(t) \leq \psi_{\max} \tag{16}$$
$$\delta_{f,\min} \leq \delta_f(t) \leq \delta_{f,\max} \tag{17}$$

There are no explicit restrictions on lateral speed or yaw rate.

Eq. (5)*: Final State Constraints*

Constraints are also placed on the vehicle's position (x, y) to be within a small distance σ from the goal position (x_g, y_g) at T_p. Mathematically, these constraints are expressed as:

$$x_g - \sigma \leq x(T_p) \leq x_g + \sigma \tag{18}$$
$$y_g - \sigma \leq y(T_p) \leq y_g + \sigma \tag{19}$$

Eq. (6)*: Moving Obstacle Avoidance using Hard Constraints*

In the hard constraints approach, obstacle avoidance is guaranteed for the NLMPC vehicle model for all feasible solutions, because hard constraints are added to insure that the trajectories of the vehicle and the obstacles do not intersect over T_p. Obstacles are represented as super-ellipses; thus, the following constraint is enforced on the trajectory for each obstacle:

$$\left|\frac{x(t) - x^i{}_{\text{obs}}(t)}{e+m}\right|^P + \left|\frac{y(t) - y^i{}_{\text{obs}}(t)}{f+m}\right|^P > 1$$

where $x^i{}_{\text{obs}}(t)$ and $y^i{}_{\text{obs}}(t)$ describe the global position of the center of the ith obstacle at t, e and f describe the semi-major and semi-minor axes, respectively, P defines the shape of the super-ellipse, and m is a safety margin that is added around each obstacle to account for the size of the vehicle.

Eq. (7)*: Dynamical Safety Constraints*

For the type of vehicles this work is concerned with (i.e., large, high-speed vehicles with high center of gravity), tire lift-off is the main dynamical safety concern. Identical to [8], [9], tire-lift-off is prevented by constraining the vertical tire force on the rear tires to be above a particular threshold, $F_{z,\text{threshold}}$. Constraints are applied to only the rear tires due to the observation that, for the specific vehicle considered, rear tires experience lift-off before the front tires [9].

Eq. (1)*: Cost Function*

The cost function consists of four terms that are linearly combined and multiplied by their respective weighing parameters w_t, w_{ψ_f}, w_{F_z}, and w_{cf}.

$$\begin{aligned} J = {} & w_t T_p + \\ & w_{\psi_f} \int_0^{T_p} \left[\sin(\psi_g)(x - x_g) - \cos(\psi_g)(y - y_g)\right]^2 \mathrm{d}t + \\ & w_{F_z} \int_0^{T_p} \left[\tanh\left(-\frac{F_{z,rl} - a}{b}\right) + \tanh\left(-\frac{F_{z,rr} - a}{b}\right)\right] \mathrm{d}t + \\ & w_{\text{cf}} \int_0^{T_p} \left[w_\delta \delta_f^2 + w_\gamma \gamma_f^2 + w_J J_x^2\right] \mathrm{d}t \end{aligned} \tag{20}$$

The first term term minimizes T_p. For a feasible solution, T_p is also the time that it takes for the vehicle to reach the goal. By minimizing this time, the vehicle is effectively pushed towards its dynamical limits. The next term helps ensure that the vehicle passes the goal point (x_g,y_g) through a desired direction (ψ_g). The third term in the cost function is a soft constraint on the vertical tire load that dissuades the vehicle from operating too close to the threshold on the vertical tire load; a and b are parameters. More information on the third term can be found in [9]. The final term in the cost function

penalizes the control effort of the vehicle over the entire prediction horizon. w_δ, w_γ, and w_J are additional weighing terms on the steering angle, steering rate and longitudinal jerk, respectively.

Moving Obstacle Avoidance using Soft Constraints: In this approach, the hard constraints on obstacle avoidance (Eqn. (6)) are removed and an additional term SC_i is added to the cost function (Eq. (20)) for each obstacle to promote obstacle avoidance.

$$SC_i = w_{\text{obs}} \int_0^{T_p} \frac{\beta(t,i)}{\left(d(t,i)+\varepsilon\right)^2} dt$$

with

$$d(t,i) = \sqrt{\left(x(t) - {x^i}_{\text{obs}}(t)\right)^2 + \left(y(t) - {y^i}_{\text{obs}}(t)\right)^2}$$

where $\beta(t,i)$ is a boolean that is set to unity if the vehicle is within a certain distance of the obstacle. For circular obstacles (e = f and P = 2) $\beta(t,i)$ is set to unity if $d(t,i) \leq e + m$. If the obstacle is not a circle, there is no analytical expression to determine if the vehicle is outside of the obstacle and m, so $d(t,i)$ is calculated numerically as described below. If the vehicle is at least a distance of m from the nearest point on the obstacle, then $\beta(t,i)$ is set to zero. Finally, w_{obs} is a weighing term that promotes obstacle avoidance, and ε is a small number used to avoid singularities.

To calculate $d(t,i)$ for an elliptical obstacle, the initial approach, referred to as SC_1, was to find the nearest point to the vehicle on the edge of a rectangular grid of query points enlarged around the obstacle by $1.2 \times m$ in both the x and y directions similar to [10]. While this works for simple cases, this approach results in a lack of information in the gradient function when the optimization is evaluating solutions where the vehicle trajectory is within the grid of query points. This can result in convergence on a solution that crashes into an obstacle even when there is another feasible obstacle-free path. Therefore, in this work another approach, referred to as SC_2, was adopted wherein $d(t,i)$ is calculated to the center of the obstacle, so that there is gradient information available to the optimization; it is more costly to drive through the middle of the obstacles than through the sides. In the cases tested, SC_2 was found superior to SC_1 in terms of obstacle avoidance performance and convergence speed, so the Results and Discussion section focuses on comparing hard constraints to SC_2 and only a small section is included that demonstrates how SC_1 can fail to highlight that the designer must be careful when building SC_i.

Solving the OCP

The aforementioned continuous time OCP is transcribed into a nonlinear programming problem using Euler's Backward Difference method and solved using the Interior Point Method implemented in IPOPT [16].

III. Results and Discussion

Demonstration of the hard constraints formulation developed in this work is provided using two scenarios that involve multiple moving obstacles. Comparisons to the soft constraints approach are also included.

Case 1

In the first example, there are three circular obstacles; one is large and static, another is medium size and moving left in front of the vehicle between the start and goal points, and the last one is roughly the size of a HMMWV (called the small obstacle for this case) and is moving in the direction from the start point to the goal point. Both hard and soft constraints formulations identify feasible solutions, albeit they are quite different as shown in Fig. 2. Using the soft constraints method the vehicle overtakes the small obstacle to the right and maneuvers just to the left of the large obstacle and to the right of the medium obstacle, whereas the hard constraints method identifies a solution that overtakes the small obstacle to the left and is able to maneuver to the left of both the medium and large obstacles. Thus the hard constraints method is able to reduce time-to-goal by 30% compared to the soft constraints approach. In Fig. 3, the vertical tire force, longitudinal speed and steering angle traces are shown for both methods. Towards the end of the trajectories, both vehicles are operated very close to their limit for minimum tire vertical load.

Another important consideration when comparing these optimizations is the optimization time. The optimization time was 0.74 and 1.83 min for the hard and soft constraints methods, respectively, on a 2.9 GHz CPU. While neither one of these times are considered to be fast enough for real-time experimentation, it is worth noting that the optimizations were implemented in MATLAB for fast development purposes, and an implementation in a compiled language can be expected to be significantly more efficient. More importantly, the hard constraints method was able to reduce the optimization time by 60% compared to the soft constraints approach.

Finally, it is noted that the soft constraints formulation converged on a local minimum, as the solution obtained by the hard constraints method yields a smaller objective function value also for the objective function used for soft constraints.

Case 2

In the second example, both algorithms are tested in a scenario with 17 moving obstacles of various shapes (to represent cars, HMMWVs, and tanks) and speeds that are moving horizontally between the vehicle starting and the goal points. Both the hard and soft constraints methods avoid all of the obstacles and successfully attain the goal position as shown in Fig. 4. To complete the mission, both vehicles follow very similar trajectories; this is likely due to the limited number of feasible paths for this particular scenario. As a result, time-to-goal is the same for both approaches and is 10 s. In Fig. 5, the vertical tire load, longitudinal speed and steering angle are shown for both methods. Between about 2.5 s and 4 s, both vehicles are operating at the minimum vertical load (set to 1000 N for these tests). This corresponds to the extreme maneuvers that the vehicles make between Frame 2 and Frame 3 in the top and bottom traces in Fig. 4 to avoid colliding with obstacles.

It is also noted that for both the hard and soft constraints cases the solution identified is not feasible at the last step, because at the last time step the vehicle accelerates above

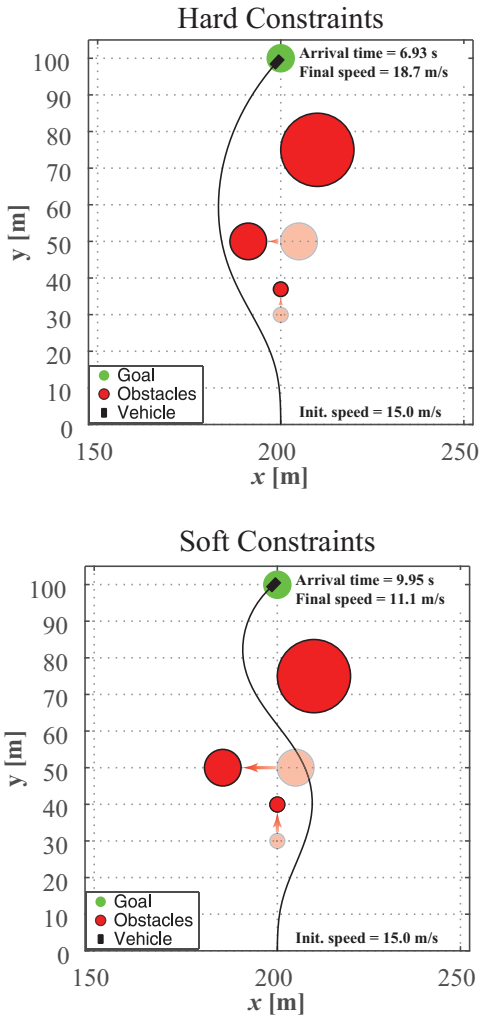

Fig. 2. In Case 1, compared to the soft constraints approach, the hard constraints approach reduces the time-to-goal by 30%, number of iterations by 11% , and optimization time by 60%.

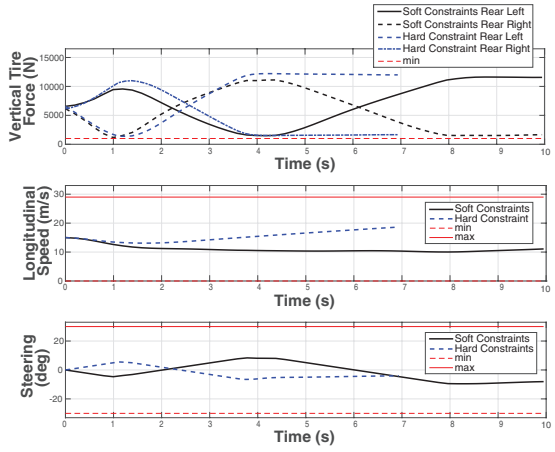

Fig. 3. This figure shows three traces for Case 1; vertical tire force, optimized steering and optimized longitudinal velocity.

the maximum allowable acceleration in order to satisfy the constraint that the vehicle is within a particular distance (σ) in both the x and the y directions of the goal (Eq. (5)). This occurs only at the last time step and is an artifact of a discretization with a fixed time step as part of the solution strategy. Our preliminary investigations indicate that a variable time step implementation resolves this issue.

One of the major drawbacks of the soft constraints method is the amount of time that it takes to complete the optimization, which is 364.5 min and two orders of magnitude longer compared to the 3.21 min for the hard constraints formulation. This is likely due to the fact that when solving this problem using the soft constraints method for elliptical obstacles, there is no analytical expression for the distance from the vehicle to the nearest point on the ellipse and a numerical solution for this distance is needed. Without analytic expressions for the objective function, its gradient, the constraints and their Jacobian, IPOPT typically takes much longer to converge.

Additionally, collision avoidance is not guaranteed for a feasible solution using soft constraints. To demonstrate this, the formulation SC_1 is used to find a solution in Case 2. The optimization converges on a solution illustrated in Fig. 6 that drives the vehicle into an obstacle at $t = 1.76$ s.

IV. Conclusion

In this work, a hard constraints formulation is developed for moving obstacle avoidance in a large, high-speed autonomous ground vehicle in an unstructured environment. To this end, a 3 DoF vehicle model is utilized in a single-level nonlinear optimization framework to find the optimal control signals (steering rate and longitudinal jerk) subject to constraints on both the dynamical limits of the vehicle and obstacle avoidance. This formulation is benchmarked against a soft constraints approach that also utilizes the same models, but relaxes the hard constraints on obstacle avoidance and augments the cost function with a term that promotes obstacle avoidance. Two comparative simulation case studies are given. It is found that both algorithms successfully avoid colliding with obstacles, however the proposed hard constraints formulation is found superior due to both a faster convergence time in optimization as well as better obstacle avoidance performance in terms of time-to-goal. Future work includes improvement of optimization times, identification of obstacles based on sensor data, prediction of the future paths of the obstacles, performing closed-loop evaluations with a vehicle model that has higher fidelity than the 3 DoF model used in the NLMPC formulation, evaluating the robustness of the algorithm and validating it experimentally.

References

[1] J. V. Frasch, A. Gray, M. Zanon, H. J. Ferreau, S. Sager, F. Borrelli, and M. Diehl, "An auto-generated nonlinear MPC algorithm for real-time obstacle avoidance of ground vehicles," in *European Control Conference*, 2013, pp. 4136–41.

[2] A. Liniger, A. Domahidi, and M. Morari, "Optimization-based autonomous racing of 1:43 scale RC cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–47, 2015.

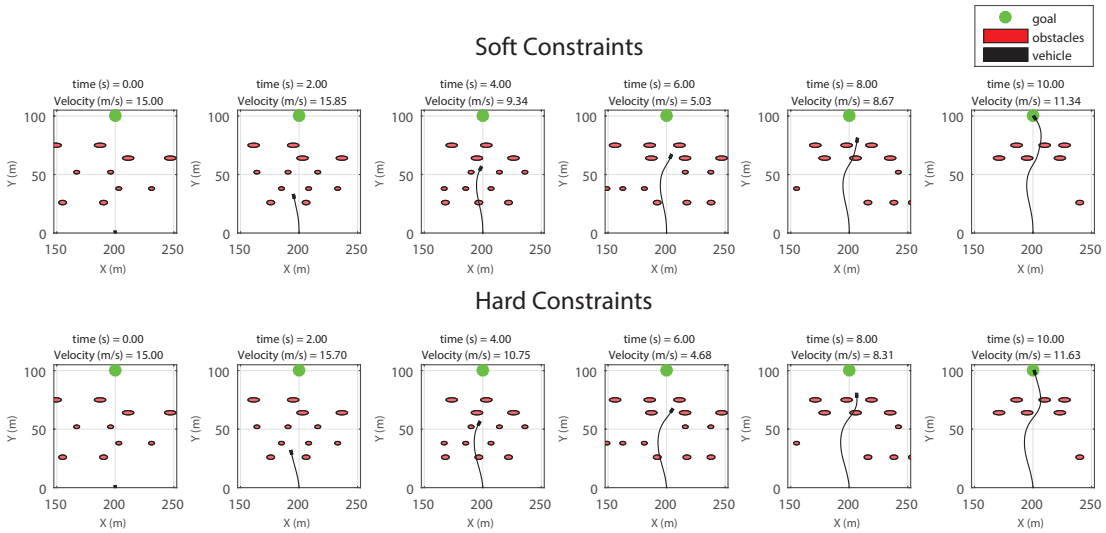

Fig. 4. In Case 2, the hard and soft constraints approaches yield almost the same trajectory and time-to-goal. However, compared to the soft constraints approach, using hard constraints reduces optimization time by 99.1%.

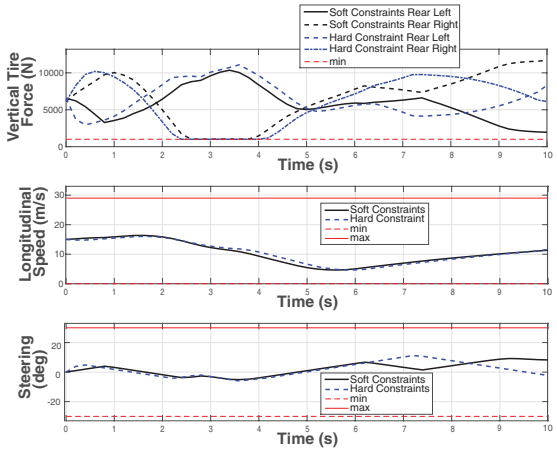

Fig. 5. This figure shows three traces for Case 2; vertical tire force, optimized steering and optimized longitudinal velocity.

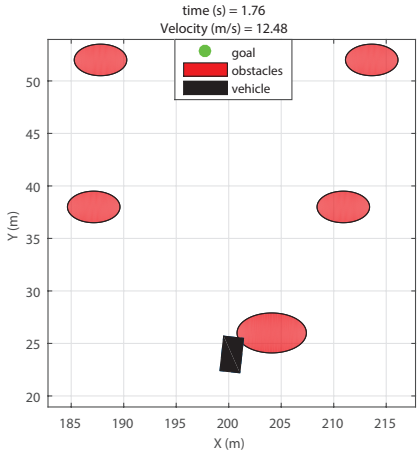

Fig. 6. This figure demonstrates that if the soft constraints are not designed properly, the vehicle may crash into an obstacle . This particular example uses the SC_1 formulation and illustrates the solution at $t = 1.76$ s.

[3] E. Velenis, P. Tsiotras, and L. Jianbo, "Modeling aggressive maneuvers on loose surfaces: The cases of trail-braking and pendulum-turn," in *European Control Conference*, 2007, pp. 1233–40.

[4] K. Kritayakirana and J. C. Gerdes, "Using the centre of percussion to design a steering controller for an autonomous race car," *Vehicle System Dynamics*, vol. 50, no. sup1, pp. 33–51, 2012.

[5] C. E. Beal and J. C. Gerdes, "Model predictive control for vehicle stabilization at the limits of handling," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1258–1269, 2013.

[6] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566–80, 2007.

[7] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, "Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads," in *Dynamic Systems and Control Conference*, 2010, pp. 265–272.

[8] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A multi-stage optimization formulation for MPC-based obstacle avoidance in autonomous vehicles using a LIDAR sensor," in *Dynamic Systems and Control Conference*, 2014.

[9] ——, "An MPC algorithm with combined speed and steering control for obstacle avoidance in autonomous ground vehicles," in *Dynamic Systems and Control Conference*, 2015.

[10] Y. Yoon, J. Shin, H. J. Kim, Y. Park, and S. Sastry, "Model-predictive active steering and obstacle avoidance for autonomous ground vehicles," *Control Engineering Practice*, vol. 17, no. 7, pp. 741–750, 2009.

[11] A. Katriniok and D. Abel, "LTV-MPC approach for lateral vehicle guidance by front steering at the limits of vehicle dynamics," in *IEEE Conference on Decision and Control and European Control Conference*, pp. 6828–33.

[12] F. Borrelli, P. Falcone, T. Keviczky, J. Asgari, and D. Hrovat, "MPC-based approach to active steering for autonomous vehicle systems," *International Journal of Vehicle Autonomous Systems*, vol. 3, no. 2-4, pp. 265–291, 2005.

[13] J. Liu, P. Jayakumar, J. L. Overholt, J. L. Stein, and T. Ersal, "The role of model fidelity in model predictive control based hazard avoidance in unmanned ground vehicles using LIDAR sensors," in *Dynamic Systems and Control Conference*, 2013.

[14] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "A study on model fidelity for model predictive control based obstacle avoidance in high speed autonomous ground vehicles," *Vehicle System Dynamics*, vol. 54, no. 11, pp. 1629–1650, 2016.

[15] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2-3, pp. 123–139, 2009.

[16] A. Wachter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.