Cooperative Vehicle Motion Planning with Collision Avoidance Constraints

Hao Yi Ong Sunil Deolalikar

June 6, 2014

1 Introduction

Path planning and trajectory optimization of vehicles is an active field of research, and there is great interest in developing collision avoidance algorithms for autonomous vehicles. Previous research has focused on developing a single-vehicle collision avoidance algorithm that augments a drivers steering commands [EFG14]. In the algorithm, a graph search heuristic finds a collection of feasible trajectories avoiding the obstacles. A Model Predictive Control (MPC) scheme then determines at each time step if the driver's steering command allows for a safe vehicle trajectory, intervening only when such a trajectory does not exist. The controller is thus designed to share control with the driver in a minimally invasive manner while avoiding obstacles and preventing loss of control.

Building upon these concepts, this paper extends the single-vehicle optimal control problem to a multi-vehicle, multi-obstacle environment where the vehicles cooperatively navigate a path in an optimal fashion. A good control scheme would allow the vehicles to follow their drivers' commands while cooperatively traversing the environment without colliding with obstacles or each other. Unlike the approach in [EFG14], however, we present an MPC-based algorithm that globally solves a mixed-integer program (MIP), which encompasses dynamics and stability constraints, as well as collision avoidance integer constraints at each time step.

2 Problem Formulation

The goal is to find optimal trajectories for a set of vehicles to take along a road with obstacles, with the objective of minimizing the deviation of the lateral force input F_{yf} from that commanded by the driver $F_{yf,\text{des}}$, which is mapped from the driver's steering command. The problem is also subject to dynamics and stability constraints, as well as collision and obstacle avoidance and road boundary constraints.

Dynamics We use a linear bicycle model with five states for each vehicle v:

$$x_v = [\beta \; r \; \Delta\psi \; s \; e]^T ,$$

where β is the side slip, r the yaw rate, $\Delta\psi$ the amount of yaw, s the distance along road, and e the distance offset from the road center. The continuous vehicle dynamics are:

$$\dot{x} = A_c x + B_c F_{yf} + d_c,$$

with control input F_{yf} and the dynamics matrices A_c, B_c, and d_c defined in Appendix A.

Stability In addition to the dynamics equations used for the vehicle motion, keeping the motion of the vehicles stable involves constraints that bound the velocity state of each vehicle. These bounds represent the frictional capabilities of the tires. The stability constraints are:

$$H_{sh} x \preceq G_{sh},$$

with the stable handling matrices H_{sh} and G_{sh} defined in Appendix A.

Obstacle avoidance There are several constraints that need to be considered for inter-vehicle or obstacle collision. Following [RH05, RH02], each vehicle and obstacle is modeled as a point with a given avoidance rectangle around it. Thus if any two vehicles or any vehicle and obstacle pair lies within this same bounding rectangle a collision occurs and therefore the state is infeasible. Conversely, if a vehicle is larger than the bounding rectangle in some direction there will not be a collision in that state. Thus for any vehicle v and another vehicle or obstacle w the collision avoidance constraints can be formulated as:

$$\begin{array}{rcl} s_v - s_w & \geq & d_v - Mc_{vw1} \\ s_w - s_v & \geq & d_v - Mc_{vw2} \\ e_v - e_w & \geq & d_v - Mc_{vw3} \\ e_w - e_v & \geq & d_v - Mc_{vw4} \\ \sum_{n=1}^{4} c_{vwn} & \leq & 3 \end{array}$$

where d_v is the minimum separation distance, M some large positive number, $v \neq w$, and $c_{vwn} \in \{0, 1\}$ with 1 indicating that the corresponding binary constraint is relaxed.

Optimization problem Given the problem data and constraints above, we have:

$$\begin{array}{ll} \text{minimize} & \sum_v \|F_{yf} - F_{yf,\text{des}}\|_2 \\ \text{such that} & \dot{x} = A_c x + B_c F_{yf} + d_c \\ & H_{sh} x \preceq G_{sh} \\ & s_v - s_w \geq d_v - Mc_{vw1} \\ & s_w - s_v \geq d_v - Mc_{vw2} \\ & e_v - e_w \geq d_v - Mc_{vw3} \\ & e_w - e_v \geq d_v - Mc_{vw4} \\ & \sum_{n=1}^{4} c_{vwn} \leq 3 \\ & x \in \mathcal{X}, \quad F_{yf} \in \mathcal{F}, \end{array} \tag{1}$$

where the decision variable is F_{yf}, and $F_{yf,\text{des}}$ is the reference command mapped from the driver's steering command. The inequalities $x \in \mathcal{X}$ and $F_{yf} \in \mathcal{F}$ encode additional constraints for road boundaries and input limits owing to steering limits, respectively.

3 A Model Predictive Control Approach

In practice, the collision avoidance constraints in (1) necessitate a long enough time horizon to safely anticipate upcoming obstacles. On the other hand, the vehicle stability constraint requires the horizon to be divided into small enough time steps to capture the fast dynamics of the vehicle. A real-time implementation of (1) would thus be difficult owing to the need of a large number of small time steps.

To address this issue, we use an MPC scheme that approximately solves (1) by optimizing over a mixed-integer quadratic program at each time step with a finite look-ahead time. In our algorithm, we discretize our problem such that

$$x(k) = A_d x(k-1) + B_d F_{yf}(k) + d_d,$$

where A_d, B_d, and d_d are the zero-order hold discretizations of A_c, B_c, and d_c, respectively.

We thus solve at each time step t the discretized version of (1) with a look-ahead of τ:

$$\begin{array}{ll}
\text{minimize} & \sum\limits_{v} \sum\limits_{k=t+1}^{t+\tau-1} \|F_{yf} - F_{yf,\text{des}}\|_2 \\
\text{such that} & x(k) = A_d x(k-1) + B_d F_{yf}(k) + d_d \\
& H_{sh} x(k) \preceq G_{sh} \\
& s_v(k) - s_w(k) \geq d_v - M c_{vw1}(k) \\
& s_w(k) - s_v(k) \geq d_v - M c_{vw2}(k) \\
& e_v(k) - e_w(k) \geq d_v - M c_{vw3}(k) \\
& e_w(k) - e_v(k) \geq d_v - M c_{vw4}(k) \\
& \sum\limits_{n=1}^{4} c_{vwn} \leq 3 \\
& x(k) \in \mathcal{X}, \quad F_{yf}(k) \in \mathcal{F}, \quad k = t+1, \ldots, t+\tau,
\end{array} \tag{2}$$

with variables $x(t+1), \ldots, x(t+\tau), F_{yf}(t+1), \ldots, F_{yf}(t+\tau)$ and data $x(t), A_d, B_d, d_d, \mathcal{X}, \mathcal{F}$. We then take the control input $F_{yf}(t+1)$ computed at each time step, which gives us a complicated state feedback control $u(t) = \phi_{\text{mpc}}(x(t))$ that's computed at each time step:

Algorithm 1: *MPC algorithm for cooperative vehicle motion planning*

given initial condition x_0 and reference command $F_{yf,\text{des}}$.
for each time t

1. Solve (2) and record $F_{yf}^{(t)}(1)$ and $x^{(t)}(1)$.
2. Output $F_{yf}^{(t)}(1)$ and set $x_0^{(t+1)} := x^{(t)}(1)$.

---

4 Example: Paired Navigation

To test and demonstrate our method, we applied it to paired vehicles navigating a shared road in two different obstacle configurations.

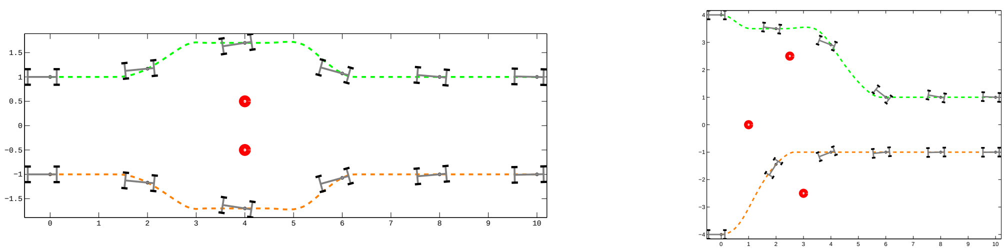

Figure 1: Two vehicles navigating between two sets of obstacles. Note: not to scale.

In the left inset of Figure 1, we see that there was not enough space for either of the two vehicles to pass between the obstacles. Thus the optimal path requires them both to go around the two obstacles and then to return to the desired paths. In the right inset there was not enough space for both vehicles to pass through the center of the three obstacles, thus one vehicle went between obstacles while the other one went around the obstacle and then returned to its desired path. Both of the simulations were run for 100 iterations with a look ahead of 2 s and time step of 0.1 s (*i.e.*, a total simulation time of 10 s).

Runtime Details The algorithm was implemented on a 2.4GHz Intel i5 quad-core processor using CVX with the bundled Gurobi solver on Matlab, which uses the branch and bound method with heuristics to arrive at a globally optimal solution.

Test Case	CPU time(s)	Variables	Integer Variables	Constraints
Left	35.87	1574	20	900
Right	49.39	1784	28	1100

While the computational times were reasonable (~0.4 s runtime for 0.1 s time steps), improvements in efficiency will be needed for this algorithm to run in real-time. We also note that the computational cost increases just by adding one more obstacle.

5 Conclusion and Future Directions

The MPC approach here presents a promising method for cooperative vehicle collision avoidance. For practical use, however, computational times need to be drastically reduced. Since the branch and bound method works by partitioning the state space into convex regions and finding optimality bounds in those regions, we can parallelize the process and run the algorithm faster with more cores. Another direction would be to model uncertainty in obstacles for situations where obstacles are not fully observed.

Acknowledgments

We thank Professor Marco Pavone and Steve Erlien for motivating this project and providing advice. We also extend our gratitude to Ed Schmerling for his help in reviewing our proposal.

References

[EFG14] S. Erlien, S. Fujita, and J. C. Gerdes. Shared steering control using safe envelopes for obstacle avoidance and vehicle stability. 2014.

[RH02] A. Richards and J. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. *American Control Conference*, 2002.

[RH05] A. Richards and J. How. Mixed-integer programming for control. *American Control Conference*, pages 2676–2683, 2005.

A Problem data and stability and dynamics matrix coefficients

The problem data are defined as follows:

$$
\begin{aligned}
\overline{\alpha}_r &\rightarrow \text{Rear tire slip angle} \\
\overline{\alpha}_f &\rightarrow \text{Front tire slip angle} \\
\overline{C}_{\bar{\alpha}_r} &\rightarrow \text{Rear tire coefficient of stiffness} \\
\overline{C}_{\bar{\alpha}_f} &\rightarrow \text{Front tire coefficient of stiffness} \\
a &\rightarrow \text{Distance from center of mass to front axle} \\
b &\rightarrow \text{Distance from center of mass to back axle} \\
U_x &\rightarrow \text{Vehicle speed (car frame)} \\
I_{zz} &\rightarrow \text{Moment of inertia} \\
m &\rightarrow \text{Vehicle mass} \\
\overline{F}_{yf} &\rightarrow \text{Lateral force on vehicle (front)} \\
\mu_p &\rightarrow \text{Peak coefficient of friction} \\
\mu_s &\rightarrow \text{Sliding coefficient of friction} \\
\alpha_{r,sat} &\rightarrow \text{Rear tire slip angle (saturation)} \\
r_{ss,max} &\rightarrow \text{Maximum steady-state yaw rate.}
\end{aligned}
$$

The dynamics matrices are defined as follows:

$$
\begin{aligned}
A_c(\bar{\alpha}_r) &= \begin{bmatrix} -\frac{C_{\alpha_r}}{mU_x} & \frac{b\bar{C}_{\bar{\alpha}_r}}{mU_x^2} - 1 & 0 & 0 & 0 \\ \frac{b\bar{C}_{\bar{\alpha}_r}}{I_{zz}} & -\frac{b^2\bar{C}_{\bar{\alpha}_r}}{I_{zz}U_x} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ U_x & 0 & U_x & 0 & 0 \end{bmatrix} \\
B_c &= \begin{bmatrix} \frac{1}{mU_x} & \frac{a}{I_{zz}} & 0 & 0 & 0 \end{bmatrix}^T \\
d_c(\bar{\alpha}_r) &= \begin{bmatrix} \frac{\bar{F}_{yf} - \bar{\alpha}_r\bar{C}_{\bar{\alpha}_r}}{mU_x} & -\frac{b\bar{F}_{yf} - \bar{\alpha}_r}{I_{zz}} & 0 & U_x & 0 \end{bmatrix}^T.
\end{aligned}
$$

The stability matrix coefficients are defined as follows:

$$
\begin{aligned}
H_{sh} &= \begin{bmatrix} 1 & -\frac{b}{U_x} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & \frac{b}{U_x} & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \end{bmatrix} \\
G_{sh} &= \begin{bmatrix} \alpha_{r,sat} & r_{ss,max} & \alpha_{r,sat} & r_{ss,max} \end{bmatrix}.
\end{aligned}
$$