Model Predictive Control for Coordination of Vehicle Platoons

Christopher Au
ECE 1505 Convex Optimization
University of Toronto - Systems Control
Email: christopher.au@mail.utoronto.ca

***Abstract*—This paper describes the application of model predictive controllers for decentralized control and coordination of autonomous vehicle platoons. Information about the road trajectory and surrounding vehicles are used to solve a constrained nonlinear optimization problem to plan the system behavior over a finite horizon. System actuation restrictions are taken into consideration in the controller design as optimization constraints.**

Keywords: Nonlinear Model Predictive Control, Receding Horizon Control, Autonomous Vehicle Platoons

I. Introduction

Increasing traffic demand on existing road infrastructure poses a serious problem in many urban areas. In addition to increasing road capacity, the concept of grouping vehicles into platoons is very desirable because it promises greater fuel economy due to reduced air resistance, shorter commute times during peak hours and fewer traffic collisions.

This paper describes the use of model predictive controllers (MPCs) for a vehicle platoon to optimally track a trajectory subject to the vehicles' motion dynamics and actuation limits. Although optimization solvers are computationally more intensive compared to conventional controllers, current convex solvers are fairly sophisticated and the speed with which optimization problems can be solved on modern systems makes it a very attractive approach for real-time applications.

The vehicles are modeled with an equivalent bicycle model with steering angle, steering rate, velocity and acceleration actuation limits, which provides the constraints for the optimization problem. The goal is to implement a decentralized MPC formation controller based on [1] and [2] to enable a multi-agent nonlinear system to follow a trajectory defined by the road. Its overall performance is also analyzed through simulations conducted in Matlab.

This paper is organized as follows: in Section II, a mathematical background on MPCs and their stability conditions are discussed. An overview of the MPC theoretical approach, specific to vehicle platoons, is provided in Section III. The simulation setup is detailed in Section IV and its results are presented in Section V. Section VI discusses future improvements to the system and the paper concludes with Section VII.

II. Problem Formulation

A. MPC Mathematical Formulation

Optimizing a system's input can be considered a convex optimization problem. By considering all future costs over a finite horizon, a set of sub-optimal control actions based on the current state can be calculated. This problem can be mathematically represented by:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{t=0}^{T-1} l(x(t), u(t)) \\ \text{subject to} \quad & u(t) \in U, x(t) \in X \\ & x(t+1) = Ax(t) + Bu(t) \\ & x(0) = z,\ x(T) = 0 \end{aligned} \tag{1}$$

Where $l(x(t), u(t))$, a positive definite function, is the stage cost of being in state x and applying input u; U and X are the input and state constraints; $Ax(t) + Bu(t)$ is the discretized system model constraints, z is the measured state at the current time step, and $x(T)$ is the final state or terminal constraint that is driven to 0. The goal is to minimize the stage cost over a time horizon defined by T.

Although this problem formulation offers a sub-optimal solution, in practice, it can be easily implemented in real-time on the physical system. Furthermore, the MPC solution can provide very near optimal results with significantly less computation time when compared to solving over an infinite horizon [3].

The process of calculating the optimal input is as follows:

1) Measure/estimate the current system state z
2) Solve the optimization problem for the optimal control action plan (set U_t^*) based on the horizon defined by T
3) Execute the first optimal control action u_t^* in the action plan
4) Repeat

Note that while the control actions for T time steps in the future are solved, only the first control action is used before the optimization problem is solved again. This ensures that the problem remains stable and feasible after executing the first control action.

B. MPC Stability and Robustness

Some important observations about the feasibility and stability of MPCs can be found in [4] and [5]. In general, the solutions calculated by the MPCs are not guaranteed to be feasible and stable. This is especially true for an uncertain system:

$$x_t = Ax_{t-1} + Bu_{t-1} + d \tag{2}$$

Where d is a bounded disturbance. It is possible that the disturbance can cause a loss of feasibility and recovering from infeasibility may not be guaranteed in real-time [6].

As a result, in a closed-loop system, it is necessary to choose proper terminal costs and constraints to explicitly ensure feasibility (robustness) and stability. The simplest approach, which is used in this paper, is to choose the terminal constraint, $x(T)$, to be 0. The Lyapunov based proof can be found in [7]. Other methods include modifying the MPC scheme by making more restrictive assumptions or introducing additional degrees-of-freedom [1]. However, because each vehicle in the platoon is controlled by independent controllers, there is no guarantee that the platoon will operate collision free at all times outside of the simulation. It is therefore necessary to include an additional controller that actuates the vehicle's brakes if its forward distance with the next vehicle reaches a certain danger threshold. This is a very mature technology in the industry and will be outside the scope of this paper.

III. Proposed Solution

A. Vehicle Dynamics

Due to the complexity in modelling an Ackermann-steering vehicle, a more simple discrete bicycle kinematics model is used instead to model the vehicle dynamics.

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t cos\theta_{t-1} dt \\ y_{t-1} + v_t sin\theta_{t-1} dt \\ \theta_{t-1} + \frac{v_t tan\delta_t}{L} dt \end{bmatrix} \tag{3}$$

Where $[\ x \quad y \quad \theta\ ]^T \in R^3$ represent the Cartesian position coordinates of the rear wheel in meters and the absolute heading angle in radians. The rear wheel speed, $v \in R$, and steering angle, $\delta \in R$, are the two system inputs. It is assumed that the vehicle does not operate at its performance limits. Thus, the vehicle does not experience tire slip and it is sufficient to model the relationship between the vehicle's inputs and states with a bicycle model.

The acceleration and steering rate of the vehicle can also be calculated using Eq. 4 and 5 respectively.

$$a_t = \frac{v_t - v_{t-1}}{dt} \tag{4}$$

$$\dot{\delta}_t = \frac{\delta_t - \delta_{t-1}}{dt} \tag{5}$$

B. MPC Constraints for Trajectory Tracking

For the car leading the platoon, the stage cost is chosen, by design, to be the sum of the 2-norm errors between the desired trajectory location, x_d, and vehicle's estimated current position, x_{cur}, in the time horizon:

$$\sum_{t=0}^{T-1} l(x(t), u(t)) = \sum_{t=0}^{T-1} ||x_d(t) - x_{cur}(t)||_2 \tag{6}$$

The state and input constraints are interpreted as lower and upper bounds on the system state:

$$lb \leq x \leq ub \tag{7}$$

Also, the initial state constraint is equal to the measured state and the final state is driven to zero:

$$x(0) = z \tag{8}$$

$$x(T) = 0 \tag{9}$$

Actuator limitations is defined by A_{eq} which is a matrix with the first 3×3 entry equal to the identity matrix. B_{eq} is set in the beginning of each optimization iteration to be the measured system state.

$$A_{eq} x = B_{eq} \tag{10}$$

Unfortunately, the bicycle model defined in Eq. 3 is a nonlinear system and not affine. This necessitates an additional nonlinear constraint to substitute the non-affine function. Eq. 11 is used to calculate the equality constraints based on all the discrete states of the system defined in Eq. 3, 4 and 5 in the time horizon.

$$c_{eq} = \begin{bmatrix} x_t - (x_{t-1} + v_t cos\theta_{t-1} dt) \\ y_t - (y_{t-1} + v_t sin\theta_{t-1} dt) \\ \theta_t - (\theta_{t-1} + \frac{v_t tan\delta_t}{L} dt) \\ -v_t + v_{t-1} + a_t dt \\ -\delta_t + \delta_{t-1} + \dot{\delta}_t dt \end{bmatrix} = 0, \quad t = 1...T \tag{11}$$

Combining all the constraints into standard form:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{t=0}^{T-1} l(x(t), u(t)) \\ \text{subject to} \quad & c_{eq}(x) = 0 \\ & A_{eq} x = B_{eq} \\ & lb \leq x \leq ub \\ & x(0) = z, x(T) = 0 \end{aligned} \tag{12}$$

C. MPC Constraints for Platoon Formation

Vehicle formation behavior is a challenging problem because it involves coordinating the actions of multiple agents with nonlinear dynamics which is a non-convex problem. [1] briefly describes the multiple ways the problem has been approached, including linear matrix inequalities, control Lyapunov functions and vision based solutions.

It is assumed that the vehicles are dynamically decoupled and independently actuated based on information available through the vehicle's own sensors. The vehicles inside the platoon will then plan trajectory paths using a similar procedure described in part B. Essentially, each vehicle will run its own MPC and will use information about its surroundings to plan collision-free trajectories that maintains the formation with safe distances between the vehicles.

Optimal control will be used to minimize the error between the relative distances between the vehicles with a desired offset. The new objective function for the following vehicles is:

$$f_0(x) = ||x_{target} - x_{pos}(T_{follow})||_2 \tag{13}$$

Where x_{target} is the Cartesian coordinate of the vehicle being followed which is compared to x_{pos}, the estimated position of the actual vehicle in T_{follow} time steps. The time horizon specified by T_{follow} will be smaller than the horizon used by the leader vehicle. This is because the leader has more visibility of the road trajectory in front of the vehicle compared to the other vehicles in the platoon. T_{follow} directly affects the distance that the vehicles follow each other and is chosen to produce a desired safe following distance. In general, the larger the value of T_{follow}, the larger the distance between vehicles in the platoon. This also produces better tracking results since the horizon is longer for the MPC planning. The designer should choose a horizon that offers a good balance between good tracking and safe following distances.

IV. Simulation Setup

This section describes how the simulation for the vehicle platoon was implemented in Matlab. A vehicle platoon size of 5 is arbitrarily chosen with T = 10 for the MPC horizon for the leading vehicle and $T_{follow} = 4$ for the MPC horizon for the other vehicles in the platoon. A simulation step size of $dt = 0.1s$ is used.

The nonlinear equality constraints is calculated using Eq. 11 for each time step in the horizon length to satisfy the upper and lower bound constraints detailed in Table 1. After solving the optimization problem at each time step, the first control action is executed and B_{eq} is updated with the new measured system state.

Table 1: System parameters

Parameter	Symbol	Value	Unit
Distance between rear and front axle	L	2	m
Maximum steering angle	δ_{max}	$\pm 50 \times \pi/180$	rad
Maximum steering rate	$\dot{\delta}_{max}$	$\pm 60 \times \pi/180$	rad/s
Maximum speed	v_{max}	27	m/s
Minimum speed	v_{min}	0	m/s
Maximum acceleration	a_{max}	2.8	m/s^2
Maximum deceleration	a_{min}	-4	m/s^2

In Matlab, the 'fmincon' function is used to solve the constrained nonlinear optimization problem using the Sequential Quadratic Programming (SQP) method. Essentially, the constrained problem is transformed to its dual / unconstrained problem by using a penalty function for constraints that are near or beyond the constraint boundary. [8] indicates that this algorithm will outperform other methods in terms of efficiency, accuracy, and percentage of successful solutions.

V. Simulation Results

The performance of the MPC for vehicle platoons is illustrated in this section. The first simulation, shown in Figure 1, is implemented on a semi-realistic road trajectory described by a continuous function. The initial vehicle positions are chosen such that they are close to the leader vehicle but off the actual road to show the convergence of the vehicle trajectories to the desired trajectory without colliding.

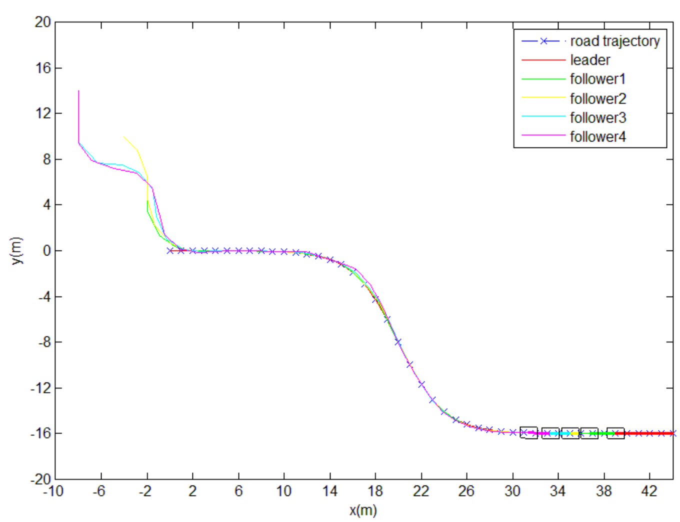

Fig. 1. Curved trajectory

The next simulation, in Figure 2, illustrates the tracking performance of the vehicle platoon for a square road trajectory. Due to the abrupt changes in the desired trajectory and the length of the planning horizon, the system is not expected to track the road perfectly. However, it can be seen that the results are quite good.

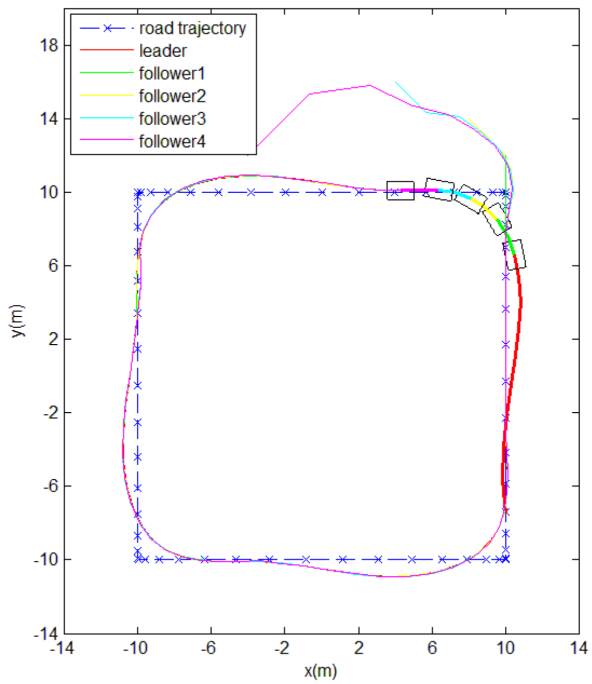

Fig. 2. Square trajectory

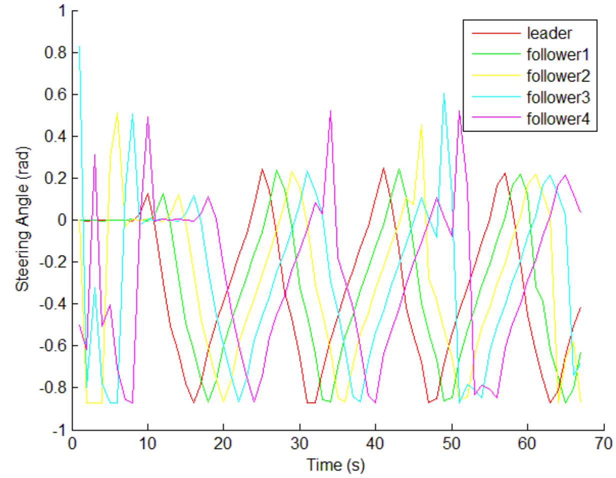

Fig. 4. Steering angle inputs

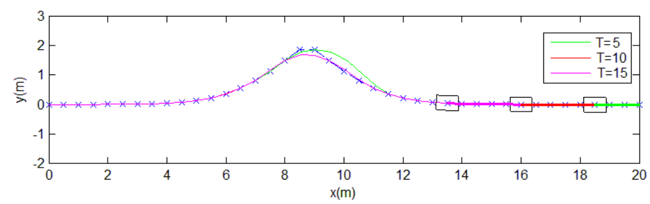

Fig. 5. Double lane change trajectory

From Figure 3 and 4, it can be seen that the vehicles' velocity and steering inputs respect the acceleration and steering rate constraints. Such constraints are necessary for the MPC to create realizable input commands that can be implemented on an actual system.

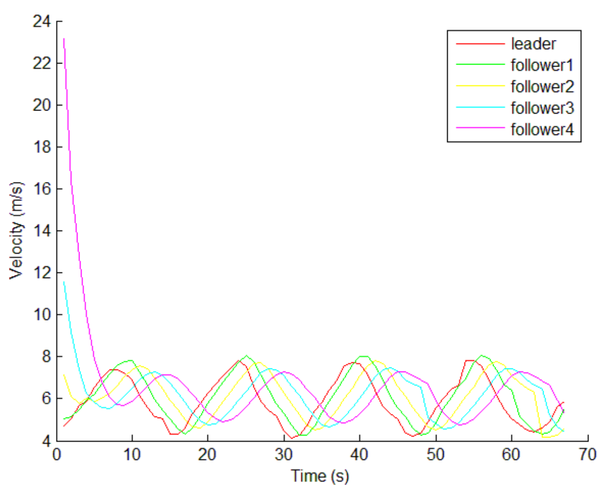

Fig. 3. Velocity inputs

The plot in Figure 5 shows the effect of changing the time horizon on the tracking performance of the controller. Clearly, increasing the horizon length results in better tracking.

For low computational power systems, a trade off must be made between computation times and tracking accuracy through the proper choice of the time horizon constant: T. Table 2 shows the computation time for various horizon lengths for a double lane change trajectory. Note that for $T = 10$ and 15, there is no noticeable difference in vehicle's trajectory. Clearly, tracking accuracy will not continue to increase past $T = 10$ for a double lane change maneuver.

Table 2: Computation time for various time horizons

Time Horizon	**Computation Time (s)**
5	18
10	102
15	257

VI. Future Improvements

There are some limitations in the MPC for platoon formation. Most notably, the tracking effectiveness decreases with decreasing time horizon. As such, the vehicles that are following other vehicles are constrained in this respect since T is restricted by the following distance. Realistically, these vehicles inside the platoon have less information to work with because their sensors can only perceive what is in front of them. In order to improve the trajectory tracking while also maintaining tight platoon groupings, it may be necessary to incorporate additional sensor information from the car's immediate surroundings for the MPC to use.

In addition, the MPC algorithm presented in this report can be improved by solving the planning problem approximately. [3] describes a method called "Fast MPC" which fixes the barrier parameter and uses warm-start to limit the total number of Newton steps. It will then be possible to run the MPC algorithm on a real-time system in the kilohertz range.

The MPC implementation in this report also assumes that the vehicles can only use information from its own sensors. However, if additional information is communicated by the leader vehicle about the subsequent road trajectory that will be traversed by the reset of the platoon, then this type of controller can be improved. This will require car-to-car communication technology that is also a very popular research area.

Furthermore, for this system to be realizable, good forward distance measurements between the vehicles are required. This is especially important when there are bends in the road where a directionally fixed forward looking radar may be at a disadvantage. Vision information from cameras fused with 3D mapping information from LIDAR sensors may be more suitable.

VII. Conclusion

This paper outlines a method of designing a vehicle platoon formation controller using MPC. The simulation results show that the method can provide good tracking accuracy while operating within the actuator limitations and vehicle holonomic constraints. Feasibility and stability is also ensured for both short and long prediction horizons, guaranteeing that the vehicles reach the road trajectory. Also, the choice of a short or long prediction horizon is a trade-off between shorter computational times versus better tracking errors.

References

[1] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G. Balas. *Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations*, IEEE Transactions on Control Systems Technology 16.1: 19-33, 2008.

[2] W. Dunbar and R. Murray. *Model predictive control of coordinated multi-vehicle formations*, IEEE Transactions on Decision and Control: 4631-4636 vol.4, 2002.

[3] S. Boyd. (n.d.). *EE364b Course Notes: Model Predictive Control*. Stanford, CA.

[4] T. Keviczky, F. Borrelli and G. Balas. *A study on decentralized receding horizon control for decoupled system*, Proc. Amer. Control Conf., pp. 49214926, 2004.

[5] T. Keviczky, F. Borrelli, and G. Balas. *Decentralized receding horizon control for large scale dynamically decoupled systems*, Automatica, vol. 42, no. 12, pp. 21052115, 2006.

[6] C. Jones, M. Zeilinger, S. Richter (2011). *Real-time online Model Predictive Control*, cole Polytechnique, Fdrale De Lausanne.

[7] C. Jones, F. Borrelli, M. Morari. (2014). *Course Notes - Model Predictive Control*, Institut fur Automatik, ETH Zurich.

[8] Mathworks (2015). *Constrained Nonlinear Optimization Algorithms*.