

Platoon Trajectory Reconstruction with Conflict Resolution Using Semidefinite Relaxation

Junyi Ji^{*,†}, Yanbing Wang[†], William Barbour[†] and Daniel B. Work[†]

Abstract—The quality of trajectory data plays a crucial role in the investigation of microscopic traffic flow. However, the challenge of effectively and automatically eliminating false positive collisions from the measured data still remains an unsolved problem. Firstly, this paper formulates the problem of reconstructing multi-vehicle trajectories as a nonconvex quadratic constraint quadratic programming (QCQP) problem. Subsequently, it derives the equivalent semidefinite programming (SDP) problem and develops a semidefinite relaxation technique to efficiently obtain the global minimum of the relaxed problem. We then apply a randomization method to obtain a feasible near-optimal solution to the original problem. The proposed approach is validated through experiments conducted on the NGSIM I-80 camera 6 lane 5 data, and the results of these experiments demonstrate that the unreasonable vehicle dynamics are removed and the platoon inconsistency decreased from 0.622% to 0.268% in the experimental data.

I. INTRODUCTION

Vehicle trajectory data, since the release of the NGSIM dataset [1], has sparked significant growth in microscopic traffic flow research, particularly in the field of trajectory-based traffic flow analysis. Various related topics, such as car-following analysis [2], traffic oscillation modeling [3], surrogate safety measurement [4], and emission estimation [5], heavily depend on the quality of trajectory data [6]. The majority of research studies often overlook a rigorous evaluation of the quality of the released data, opting instead to directly incorporate it into the analysis. It is important to recognize that the precision of the data has a significant influence on the modeling of microscopic behavior. Inaccuracies become more noticeable when calculating speed and acceleration values during the differentiation process. Known issues include unreasonable acceleration, speed, missing data, and incompleteness of the trajectory [7], [8], [9]. It has also been highlighted the occurrence of collisions [9] in the NGSIM I-80 data, which does not align with the observed time period (see Fig. 1 and Fig. 2). The study conducted by [10] also revealed a platoon inconsistency, indicating that the inter-vehicle spacing between the leading and following vehicles was below 0. In practical observations on highways, events can be classified as either positive (a collision happened) or negative (no collision happened). However, when examining the measured data, the presence of a collision can be labeled

as either true (consistent with real-world data) or false (not consistent). This results in four distinct categories to describe the type of collision: true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TP refers to cases where collisions are correctly observed and reported. FP, which is investigated and reconstructed in this paper, indicates instances where the measured data erroneously reports *non-existent collisions*. TN represents situations where non-collision events are accurately identified as such. FN signifies cases where collisions are present in the real world but are missed or not detected in the measured data.



Fig. 1. False positive collision in the NGSIM dataset: specifically in camera 6 at frame 9690 (corresponding to 16:09 in the video file), there is a false positive collision recorded. This collision involves vehicle 3263 and vehicle 3239 in Lane 5.

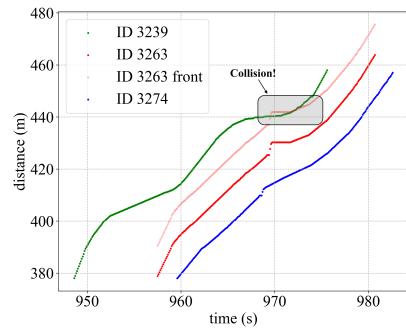


Fig. 2. The space-time diagram for the trajectory of the false positive collision sample, as depicted in Fig. 1, reveals a non-smooth trajectory pattern. Additionally, the false positive collision between vehicle 3263 and 3239 is evident in the diagram, occurring from time 968 to 975 seconds.

In this paper, conflict resolution refers to removing false positive collisions from the measured dataset. Such false positive collisions can originate from various sources, encompassing detection errors, tracking failures, and occasional smoothing errors. Motivated by the false positive collision issue, this paper focuses on dealing with longitudinal platoon trajectory reconstruction with conflict resolution to smooth

*Corresponding author, Junyi Ji, junyi.ji@vanderbilt.edu.

†Department of Civil and Environmental Engineering, Institute for Software Integrated Systems, Vanderbilt University.

the trajectory and remove false positive collisions. The structure of this paper is organized as follows: Section II presents a review of the relevant literature on trajectory reconstruction and semidefinite relaxation. Section III provides a comprehensive description of the problem statement, covering both single-vehicle and multi-vehicle reconstruction, and introduces the original non-convex QCQP problem as well as the relaxation SDP problem. Section IV presents the experimental dataset and provides details on deployment. Section V summarizes the results obtained from the experiments. Finally, Section VI concludes the paper by discussing the findings and contributions, and identifies potential directions for future research.

II. RELATED WORK

A. Trajectory reconstruction

In the context of trajectory reconstruction, Punzo et. al [7] highlight the existence of inconsistencies in the trajectory data, which have been observed since the initial release of the NGSIM data. The paper identifies two main types of inconsistencies: *internal inconsistency* and *platoon inconsistency*. These inconsistencies refer to discrepancies within individual vehicle trajectories (e.g., infeasible acceleration) and conflicts among vehicles in a platoon (e.g., collisions), respectively. Addressing and resolving these inconsistencies is crucial for accurate and reliable trajectory reconstruction. [11] designed filters for extremely high errors, which are the outliers to remove the internal inconsistency. \mathcal{L}_1 and \mathcal{L}_2 norm-regulated postprocessing algorithms were presented in [12], specifically designed for single-vehicle automatic trajectory reconstruction. This paper is extended based on the findings and techniques described in [12].

Nevertheless, there has been a limited number of papers that have effectively addressed the issue of platoon inconsistency. This problem requires multiple vehicle processing and conflict resolution, resulting in a significant increase in the problem dimension and complexity (see section III-A.2 for the proof as a non-convex QCQP problem). Conflict resolution problems are typically dealt with in the context of trajectory planning in the domain of autonomous vehicles [13], aircraft planning [14]. Semidefinite relaxation [15] is a proven technique employed in these scenarios, as it offers feasible solutions that strike a balance between optimality and computational efficiency.

B. Semidefinite relaxation

Semidefinite relaxation (SDR) is a powerful technique used in non-convex optimization, often applied to tackle non-convex QCQP optimization problems. The core idea of semidefinite relaxation is to approximate the nonconvex QCQP problem by relaxing it to a semidefinite program (SDP), which is easier to solve and provides a lower bound on the optimal solution. SDR has gained increasing attention over the past two decades and has seen applications in

TABLE I
NOTATIONS AND OPERATORS USED IN THIS PAPER

Notations	Meaning	Dimension
\mathcal{S}	vehicle set of the selected space-time area	-
\mathcal{T}	time set of the selected space-time area	-
\mathcal{V}	the set of all the vehicles	-
n_i	time interval of vehicle i alive in the selected section	\mathbb{R}
τ_i	the observed time interval of vehicle i observed in \mathcal{S}	-
$x_{i,k}$	the rear-end bumper position of vehicle i at time k	\mathbb{R}
\mathbf{x}_i	the rear-end bumper position collection of vehicle i	\mathbb{R}^{n_i}
\mathbf{v}_i	the velocity collection of vehicle i	\mathbb{R}^{n_i-1}
\mathbf{a}_i	the acceleration collection of vehicle i	\mathbb{R}^{n_i-2}
\mathbf{j}_i	the jerk collection of vehicle i	\mathbb{R}^{n_i-3}
\mathbf{w}_i	the measured rear-end bumper position collection of vehicle i	\mathbb{R}^{n_i}
$D_i^{(k)}$	the k^{th} order difference equations coefficient for vehicle i	$\mathbb{R}^{(n_i-k) \times n_i}$
N_v	the number of observed vehicles	\mathbb{R}
$\tau_i \cap \tau_j$	the intersection time interval of vehicle i and j	-
$\text{tr}(\mathbf{A})$	trace of the matrix \mathbf{A}	-
$\text{rank}(\mathbf{A})$	rank of the matrix \mathbf{A}	-
$\text{diag}()$	the diagonal matrix consists of only the matrix elements on the diagonal line.	-
T	time interval for each step	-

various fields such as signal processing and wireless communication [16]. SDR-based approaches generally offer superior performance in both theory and practice while maintaining polynomial-time worst-case complexity. Within the context of trajectory reconstruction and smoothing involving multiple vehicles, it has been established that the distance constraints manifest as nonconvex quadratic constraints [17]. This underlines the complexity and challenging nature of the problem due to the inherent non-convexity of such constraints.

III. PROBLEM STATEMENT AND METHODOLOGY

A. Problem statement

A platoon, in this context, refers to a cluster of vehicles moving collectively on a freeway along the same lane where one follows another. Trajectories of the platoon are detected and tracked by roadside cameras. The main objective of this study is to develop algorithms for reconstructing the longitudinal trajectory of vehicles. In the context of this research, the issue can be conceptualized as an optimization paradigm with objectives to smooth trajectory, modulate vehicle dynamics, and mitigate FP collision events.

1) *Single-vehicle reconstruction:* Consider the observed trajectory time interval τ_i of the vehicle i , \mathbf{z}_i is the observed position of the rear-end bumper of the vehicle i , \mathbf{x}_i is the reconstructed position of the rear-end bumper of the vehicle i , and \mathbf{w}_i is the noises, where the relationship between \mathbf{z}_i , \mathbf{x}_i , \mathbf{w}_i can be written as Eq. 1.

$$\mathbf{z}_i = \mathbf{x}_i + \mathbf{w}_i \quad \mathbf{z}_i \in \mathbb{R}^{n_i}, \mathbf{w}_i \in \mathbb{R}^{n_i}, \mathbf{x} \in \mathbb{R}^{n_i} \quad (1)$$

The speed, acceleration, and jerk of the vehicle i can be represented by the k^{th} order differentiation operator multiplied by \mathbf{x}_i , $k = 1, 2, 3$ respectively.

$$\mathbf{v}_i = D_i^{(1)} \mathbf{x}_i, \quad \mathbf{a}_i = D_i^{(2)} \mathbf{x}_i, \quad \mathbf{j}_i = D_i^{(3)} \mathbf{x}_i \quad (2)$$

where $D^{(k)}$ ($k = 1, 2, 3$) can be written as follows:

$$D_i^{(1)} = \frac{1}{T} \begin{bmatrix} -1 & 1 & & \\ & \ddots & & \\ & & -1 & 1 \end{bmatrix} \quad (3)$$

$$D_i^{(2)} = \frac{1}{T^2} \begin{bmatrix} 1 & -2 & 1 & & \\ & \ddots & & & \\ & & 1 & -2 & 1 \end{bmatrix} \quad (4)$$

$$D_i^{(3)} = \frac{1}{T^3} \begin{bmatrix} -1 & 3 & -3 & 1 & & \\ & & & & \ddots & \\ & & & & & -1 & 3 & -3 & 1 \end{bmatrix} \quad (5)$$

The problem of reconstructing a smooth trajectory for a single vehicle can be formulated as the following optimization problem (Eq. 6a - 6d). The objective function aims to minimize various factors, including the noise between the observed and reconstructed trajectory, as well as the acceleration and jerk of the reconstructed trajectory. The weight parameters, λ_1 , λ_2 , and λ_3 , need to be tuned in the optimization problem. Eq. 6b - 6d are the constraints for reasonable speed, acceleration, and jerk bounded by the maximum and minimum value $[\mathbf{v}_{\min}, \mathbf{v}_{\max}]$, $[\mathbf{a}_{\min}, \mathbf{a}_{\max}]$, $[\mathbf{j}_{\min}, \mathbf{j}_{\max}]$ respectively. This problem is a convex quadratic problem, meaning that it has a unique global optimal solution that can be efficiently solved [12].

Single-vehicle reconstruction problem

$$\min_{\mathbf{x}_i} \underbrace{\lambda_1 \|\mathbf{z}_i - \mathbf{x}_i\|_2^2}_{\mathcal{L}_2 \text{ norm of noise}} + \underbrace{\lambda_2 \|D_i^{(2)} \mathbf{x}_i\|_2^2}_{\mathcal{L}_2 \text{ norm of acceleration}} + \underbrace{\lambda_3 \|D_i^{(3)} \mathbf{x}_i\|_2^2}_{\mathcal{L}_2 \text{ norm of jerk}} \quad (6a)$$

$$\text{subject to } \mathbf{v}_{\min} \preceq D_i^{(1)} \mathbf{x}_i \preceq \mathbf{v}_{\max} \quad (6b)$$

$$\mathbf{a}_{\min} \preceq D_i^{(2)} \mathbf{x}_i \preceq \mathbf{a}_{\max} \quad (6c)$$

$$\mathbf{j}_{\min} \preceq D_i^{(3)} \mathbf{x}_i \preceq \mathbf{j}_{\max} \quad (6d)$$

follows, where \mathbf{x} represents the decision variable:

$$\mathbf{z} = \begin{bmatrix} \frac{1}{\sqrt{n_1}} \mathbf{z}_1 \\ \vdots \\ \frac{1}{\sqrt{n_k}} \mathbf{z}_k \\ \vdots \\ \frac{1}{\sqrt{n_{N_v}}} \mathbf{z}_{N_v} \end{bmatrix} \in \mathbb{R}^N, \quad \mathbf{x} = \begin{bmatrix} \frac{1}{\sqrt{n_1}} \mathbf{x}_1 \\ \vdots \\ \frac{1}{\sqrt{n_k}} \mathbf{x}_k \\ \vdots \\ \frac{1}{\sqrt{n_{N_v}}} \mathbf{x}_{N_v} \end{bmatrix} \in \mathbb{R}^N \quad (7)$$

Here, N represents the total number of trajectory points, which is calculated as the sum of individual vehicle trajectory point counts, n_i , where i ranges from 1 to N_v . The $\frac{1}{\sqrt{n_k}}$ term is to normalize the penalization for each trajectory based on the time interval length. The objective function of single-vehicle reconstruction can be extended to multiple-vehicle scenarios, as shown in Eq. 8. It is important to note that in this context, all the vehicles share the same parameter values for λ_1 , λ_2 , and λ_3 .

$$F(\mathbf{x}) = \underbrace{\lambda_1 \|\mathbf{z} - \mathbf{x}\|_2^2}_{\mathcal{L}_2 \text{ norm of noise}} + \underbrace{\lambda_2 \|D_2 \mathbf{x}\|_2^2}_{\mathcal{L}_2 \text{ norm of acceleration}} + \underbrace{\lambda_3 \|D_3 \mathbf{x}\|_2^2}_{\mathcal{L}_2 \text{ norm of jerk}} \quad (8)$$

The matrices D_2 and D_3 can be expressed as Equation 9. Here, the diag operation implies placing the difference matrix of vehicle i on the diagonal of the new matrix:

$$\mathbf{D}_k = \text{diag}(D_1^{(k)}, \dots, D_i^{(k)}, \dots, D_{N_v}^{(k)}) \quad (9)$$

In the multiple-vehicle scenarios, the focus lies on investigating the distance constraint, which can be captured by the Euclidean distance between two data points, for example, $d(\mathbf{x})_{i,j,k}$ represents the squared Euclidean distance between $x_{i,k}$ and $x_{j,k}$, can be represented by Eq. 10.

$$d(\mathbf{x})_{i,j,k} = \|x_{i,k} - x_{j,k}\|_2^2 \quad (10)$$

$$= \|\mathbf{T}_k \mathbf{S}_{i,j} \mathbf{x}\|_2^2 \quad (i \in V, j \in \mathcal{V}, i < j, k \in \tau_{i,j}) \quad (11)$$

Eq. 11 presents an alternative representation of the distance calculation using matrices and vectors. The matrix form of distance is derived using a temporal selection matrix \mathbf{T}_k and a vehicle selection matrix $\mathbf{S}_{i,j}$ applied to the data vector \mathbf{x} . The temporal selection matrix \mathbf{T}_k and the vehicle selection matrix $\mathbf{S}_{i,j}$ depend on the indices i , j , and k , with i and j belonging to a set of vertices \mathcal{V} , i being less than j , and k being within a specific time interval $\tau_{i,j}$ (see Eq. 12), which is the intersection of the time interval τ_i and τ_j .

$$\tau_{i,j} = \tau_i \cap \tau_j \quad (12)$$

If $\tau_i \cap \tau_j = \emptyset$, it will not be calculated. \mathbf{T}_k can be written as follows:

$$\mathbf{T}_k = \underbrace{\begin{bmatrix} 0 & \dots & \underbrace{1}_{\text{time } k} & \dots & 0 \end{bmatrix}}_{n_{\tau_{i,j}} \text{ time steps}} \in \mathbb{R}^{1 \times n_{\tau_{i,j}}} \quad (13)$$

The vehicle selection matrix $\mathbf{S}_{i,j}$ is constructed by concatenating the vehicle-specific selection matrices $\mathbf{S}_{i|\tau_{i,j}}$ and

$S_{j|\tau_{i,j}}$. The matrix $S_{i|\tau_{i,j}}$ is a matrix that selects the trajectory of vehicle i within the common time interval $\tau_{i,j}$ (see Eq. 15). Similarly, $S_{j|\tau_{i,j}}$ is a matrix that selects the trajectory of vehicle j within $\tau_{i,j}$ (see Eq. 16). The matrix $R_{i,j}$ is used to stack the selected trajectories of vehicles i and j together (see Eq. 17).

$$S_{i,j} = [S_{i|\tau_{i,j}} \quad S_{j|\tau_{i,j}}] R_{i,j} \in \mathbb{R}^{n_{\tau_{i,j}} \times N} \quad (14)$$

$$S_{i|\tau_{i,j}} = \left[\underbrace{\begin{matrix} & & \tau_{i,j} \\ 0 & \dots & \overbrace{\Lambda(1)}^{\tau_{i,j}} & \dots & 0 \end{matrix}}_{n_i} \right] \in \mathbb{R}^{n_{\tau_{i,j}} \times n_i} \quad (15)$$

$$S_{j|\tau_{i,j}} = \left[\underbrace{\begin{matrix} & & \tau_{i,j} \\ 0 & \dots & \overbrace{\Lambda(-1)}^{\tau_{i,j}} & \dots & 0 \end{matrix}}_{n_j} \right] \in \mathbb{R}^{n_{\tau_{i,j}} \times n_j} \quad (16)$$

$$R_{i,j} = \begin{bmatrix} \mathbf{0} & \dots & \underbrace{\Lambda(1)}_{\text{vehicle } i} & \dots & \underbrace{\Lambda(1)}_{\text{vehicle } j} & \dots & \mathbf{0} \end{bmatrix} \quad (17)$$

Eq. 18 represents the constraints placed on the squared Euclidean distance, $d(\mathbf{x})_{i,j,k}$, between two data points. It ensures that the squared Euclidean distance is greater than or equal to the minimum distance squared, d_{\min}^2 , thereby guaranteeing a minimum separation between the data points. In the context of traffic flow, this constraint corresponds to maintaining a sufficient distance or spacing between vehicles for smooth and safe traffic conditions.

$$\|d(\mathbf{x})_{i,j,k}\|_2^2 \geq d_{\min}^2 \quad (18)$$

Multiple-vehicle reconstruction problem (original)

$$\min_{\mathbf{x}} F(\mathbf{x}) \quad (\text{Eq. 8}) \quad (19a)$$

$$\text{subject to } \|\mathbf{T}_k S_{i,j} \mathbf{x}\|_2^2 \geq d_{\min}^2 \quad (19b)$$

$$\mathbf{v}_{\min} \preceq \mathbf{D}_1 \mathbf{x} \preceq \mathbf{v}_{\max} \quad (19c)$$

$$\mathbf{a}_{\min} \preceq \mathbf{D}_2 \mathbf{x} \preceq \mathbf{a}_{\max} \quad (19d)$$

$$\mathbf{j}_{\min} \preceq \mathbf{D}_3 \mathbf{x} \preceq \mathbf{j}_{\max} \quad (19e)$$

The problem can be categorized as a non-convex quadratic constraint quadratic programming (QCQP) problem since the matrix $-\mathbf{T}_k^\top \mathbf{T}_k \mathbf{T}_k S_{i,j}$ for constraint 19b is not positive semidefinite. The multi-vehicle reconstruction problem becomes challenging due to the inherent difficulty in solving non-convex QCQP problems, which are characterized by the existence of multiple local optima.

B. SDP relaxation

In order to address the original non-convex QCQP problem, this section introduces the SDP relaxation and the restoration of a viable solution to the original problem.

Consider the non-convex quadratic constraint (QC) where the non-convexity comes from as follows.

$$\mathbf{x}^\top \mathbf{Q}_{i,j,k} \mathbf{x} \geq d_{\min}^2 \quad (20)$$

where $\mathbf{Q}_{i,j,k}$ is defined as $\mathbf{S}_{i,j}^\top \mathbf{T}_k^\top \mathbf{T}_k \mathbf{S}_{i,j}$. Since the left side of the inequality is a scalar, the trace operator can be applied, and $\mathbf{x}^\top \mathbf{Q}_{i,j,k} \mathbf{x}$ can be expressed in terms of trace:

$$\mathbf{x}^\top \mathbf{Q}_{i,j,k} \mathbf{x} = \text{tr}(\mathbf{x}^\top \mathbf{Q}_{i,j,k} \mathbf{x}) = \text{tr}(\mathbf{Q}_{i,j,k} \mathbf{x} \mathbf{x}^\top) \quad (21)$$

This formulation linearizes the non-convex QC in terms of $\mathbf{x} \mathbf{x}^\top$. Let's introduce a new decision variable \mathbf{X} , defined as:

$$\mathbf{X} = \mathbf{x} \mathbf{x}^\top \quad (22)$$

The decision variable \mathbf{X} is an $N \times N$ matrix $\in \mathbb{R}^{N \times N}$, with the characteristic that $\text{rank}(\mathbf{X}) = 1$. It is important to note that the original problem is a nonhomogeneous Quadratically Constrained Quadratic Program (QCQP), as cited in [16], due to the presence of non-quadratic terms in the objective function (refer to Eq. 8). As a result, the variable \mathbf{x} must be maintained in the relaxed problem. The following equivalences are then derived (with Eq. 25 making use of the Schur complement):

$$\mathbf{X} = \mathbf{x} \mathbf{x}^\top, \text{rank}(\mathbf{X}) = 1 \quad (23)$$

$$\Leftrightarrow \mathbf{X} \succeq \mathbf{x} \mathbf{x}^\top, \text{rank}(\mathbf{X}) = 1 \quad (24)$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{X} & \mathbf{x}^\top \\ \mathbf{x} & 1 \end{bmatrix} \succeq 0, \text{rank}(\mathbf{X}) = 1 \quad (25)$$

The objective function of the original problem can then be written as the form of semidefinite programming (SDP) in Eq. 26.

$$F(\mathbf{x}, \mathbf{X}) = \text{tr}(\mathbf{X}) + \lambda_1 \text{tr}(\mathbf{D}_2^\top \mathbf{D}_2 \mathbf{X}) + \lambda_2 \text{tr}(\mathbf{D}_3^\top \mathbf{D}_3 \mathbf{X}) - 2\mathbf{z}^\top \mathbf{x} + \mathbf{z}^\top \mathbf{z} \quad (26)$$

The optimization can be written as follows (Eq. 27a - 27g):

Semidefinite relaxation problem (relaxation)

$$\min_{\mathbf{x}, \mathbf{X}} F(\mathbf{x}, \mathbf{X}) \quad (\text{Eq. 26}) \quad (27a)$$

$$\text{subject to } \text{tr}(\mathbf{Q}_{i,j,k} \mathbf{X}) \geq d_{\min}^2 \quad (27b)$$

$$\mathbf{v}_{\min} \preceq \mathbf{D}_1 \mathbf{x} \preceq \mathbf{v}_{\max} \quad (27c)$$

$$\mathbf{a}_{\min} \preceq \mathbf{D}_2 \mathbf{x} \preceq \mathbf{a}_{\max} \quad (27d)$$

$$\mathbf{j}_{\min} \preceq \mathbf{D}_3 \mathbf{x} \preceq \mathbf{j}_{\max} \quad (27e)$$

$$\begin{bmatrix} \mathbf{X} & \mathbf{x}^\top \\ \mathbf{x} & 1 \end{bmatrix} \succeq 0 \quad (27f)$$

$$\text{rank}(\mathbf{X}) = 1 \quad (\text{nonconvex}) \quad (27g)$$

The relaxation problem will be convex SDP when the rank constraint (Eq. 27g) is relaxed, which will provide a global minimal as the lower bound to the original problem. The

near-optimal solution can be recovered from the solution of SDP by randomization shown in Algorithm 1.

Algorithm 1: Randomization Feasible Solution Recovery from SDR

```

1 input: optimal solution  $\mathbf{X}^*$ ,  $\mathbf{x}^*$  to SDR problem and
   a number of randomizations  $L$ .
2 if  $\text{rank}(\mathbf{X}^*) = 1$  then
3   terminate: the optimal solution,  $\mathbf{x}^*$ , is returned
      as it is the global optimum of the original
      problem.
4 if  $\text{rank}(\mathbf{X}^*) \neq 1$  then
5   for  $l = 1, \dots, L$  do
6     generate:  $\tilde{\mathbf{x}}_l \in \mathcal{N}(\mathbf{x}^*, \mathbf{X}^* - \mathbf{x}^* \mathbf{x}^{*\top})$ 
7     if  $\tilde{\mathbf{x}}_l \in \mathcal{F}$  then
8       comment: the solution  $\tilde{\mathbf{x}}_l$  is feasible for
          the original problem ( $\mathcal{F}$  is the feasible
          set for the original problem).
9       add: the solution  $\tilde{\mathbf{x}}_l$  to the feasible set  $\mathcal{F}_c$ .
10    end
11  determine:  $l^* = \arg\min_{l \in \mathcal{F}_c} F(\tilde{\mathbf{x}}_l)$ , where
      ( $F(\mathbf{x}) = \text{Eq.8}$ )
12  terminate:  $\hat{\mathbf{x}} = \tilde{\mathbf{x}}_{l^*}$  as the approximate QCQP
      solution

```

The algorithm denoted as Algorithm 1 accepts the optimal solution \mathbf{X}^* and \mathbf{x}^* for the SDR problem, along with the specified number of randomizations L to be performed. When the rank of \mathbf{X}^* is equal to 1, indicating a unique optimal solution for the SDR problem, the algorithm terminates and outputs \mathbf{x}^* as the global optimum for the original problem. However, if the rank of \mathbf{X}^* is not 1, the algorithm proceeds to obtain a feasible solution through randomizations. The main loop of the algorithm executes from $l = 1$ to L , generating random samples $\tilde{\mathbf{x}}_l$ based on \mathbf{x}^* and the covariance matrix $\mathbf{X}^* - \mathbf{x}^* \mathbf{x}^{*\top}$. These samples are drawn from a Gaussian distribution \mathcal{N} . Each generated sample $\tilde{\mathbf{x}}_l$ is checked for feasibility in the original problem by verifying its compliance with all the constraints from Eq. 19b to Eq. 19e. If a sample is found to be feasible, it is added to the set of feasible solutions \mathcal{F}_c . Upon completing the loop, the algorithm identifies the index l^* that minimizes the objective function $F(\tilde{\mathbf{x}}_l)$ among all feasible solutions $\tilde{\mathbf{x}}_l$ in \mathcal{F}_c . The objective function is defined by Equation 8. Lastly, the algorithm terminates and outputs $\hat{\mathbf{x}}$, which is assigned as the sample $\tilde{\mathbf{x}}_{l^*}$ with the minimum objective value. This approximate solution $\hat{\mathbf{x}}$ is considered a feasible solution for the original QCQP problem.

IV. EXPERIMENTAL DEPLOYMENT

A. Experimental dataset

The paper utilized a data set derived from NGSIM data, labeled manually as the ground truth, which was shared

by [9]. The experimental dataset spans a duration of 15 minutes and includes a longitudinal range of 250 feet. The manual extraction of longitudinal vehicle trajectories was performed iteratively across all lanes, encompassing all non-motorcycle vehicles visible in camera 6 of I-80. The data schema of the NGSIM I-80 dataset includes vehicle ID, frame ID, global time, local and global coordinates, vehicle dimensions, class, velocity, acceleration, **lane ID**, preceding and following vehicle IDs, spacing, and headway.

B. Deployment details

The SDP relaxation problem is deployed on cvxpy [18] and the MOSEK [19] solver. Despite the convexity of the semidefinite program (SDP), there is still a challenge regarding the scale of the SDP [20]. In order to efficiently address the multi-vehicle reconstruction problem, a sliding window approach has been devised to decrease the dimensions in both time and space. Figure 3 illustrates a case that explains the concept of time sliding and platoon sliding windows.

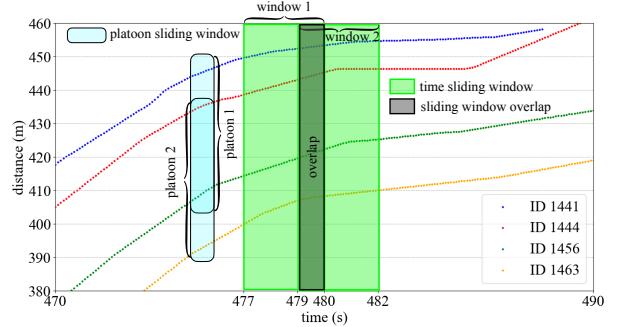


Fig. 3. This illustration depicts the sliding windows used for time and platoon analysis. The time sliding window, represented in green, generates overlapping segments (in grey). Window 1 reconstructs trajectories for all vehicles within it, with the overlap providing constraints for the next subsequent window (window 2). The platoon sliding window, shown in blue, considers three vehicles at a time. It solves the reconstruction problem for the first three vehicles, and the trajectory of the first vehicle in the next platoon (identified as ID 1444 and marked in red) becomes a constraint for the next platoon's reconstruction.

1) *Time sliding window:* Initially, a bounding box spanning N_{window} ($N_{\text{window}} = 30$ in this paper, i.e. 3 seconds) time steps is selected as a time window. This bounding box then shifts forward by 20 steps (i.e. 2 seconds). The intersection of the previous and current bounding boxes serves as a constraint to guarantee consistency and smoothness in position, speed, and acceleration for the next step.

2) *Platoon sliding window:* When the number of vehicles within the sliding window of time exceeds 3, the platoon sliding window approach is employed. The original platoon will be broken down into multiple 3-vehicle platoons. The initial platoon consists of the first three vehicles, and the position of the last vehicle in the platoon serves as a constraint for the next platoon, ensuring consistency and smoothness in their positions. By implementing the time and platoon sliding strategy, the scale of the SDP problem

will be effectively managed, remaining within the range of $3N_{\text{window}} + 1$ dimensions.

3) *Parameters setting*: The weight parameters in the objective function, namely λ_1 , λ_2 , λ_3 , are assigned values of 1, 10^{-3} , and 10^{-5} respectively. The minimum distance d_{\min} is set to the sum of the follower vehicle's length and 5 ft, representing the minimum safe distance between vehicles. To establish reasonable vehicle dynamics boundaries, the following values are set: v_{\min} (minimum velocity) is 0 mph, a_{\max} (maximum acceleration) is 10 ft/s², a_{\min} (minimum acceleration) is -10 ft/s², j_{\max} (maximum jerk) is 10 ft/s³, and j_{\min} (minimum jerk) is -10 ft/s³. These limits ensure realistic vehicle dynamics. The parameter L , which represents the number of randomization samples in the SDR feasible solution recovery process, is set to 100.

V. RESULTS

A. Computational efficiency

The code and experiment were run on a macOS 13.2.1 operating system, with a 3.22 GHz Apple M1 Pro CPU and 16GB of RAM. Solving 10 typical sliding window that spans 30 seconds (300-time steps in this case) took approximately 393 seconds of wall time. This duration is considered acceptable for offline trajectory postprocessing purposes.

B. Vehicle dynamics and smoothness

Figure 4 present the distributions of speed and acceleration for both the raw data and the reconstructed data achieved through SDR respectively. The speed and acceleration have been regulated to fall within a reasonable range.

C. Platoon consistency

Figure 5 presents a before and after comparison of the space-time diagram for Lane 5, encompassing the full space-time scale of 15 minutes and 250 feet. The comparison showcases the changes and improvements in the diagram. To magnify the space-time diagram for the specific FP collision case depicted in Figure 2, Figure 6 displays the space-time diagram reconstructed using SDR, demonstrating the successful resolution of the false positive collision between vehicle 3263 and 3239. The overall platoon inconsistency, as evidenced by the experimental data, decreased from 0.622% to 0.268% (see Fig. 4 for more details). Further investigation is required to understand the cause of the remaining 0.268% failure, which primarily occurs during lane-changing scenarios.

VI. CONCLUSIONS AND FUTURE WORK

The contributions of this paper are twofold. First, it addresses the challenging non-convex QCQP problem associated with multi-vehicle platoon trajectory reconstruction by transforming it into a convex SDP problem. Second, the paper introduces a randomization-based approach to recover a feasible near-optimal solution for the SDP problem, thereby

improving the efficiency and effectiveness of multi-vehicle trajectory reconstruction.

Future work in this field involves expanding the approach to incorporate two-dimensional (2D) trajectory smoothing techniques and collision resolution algorithms. Additionally, addressing challenges related to missing data is crucial. The robustness and scalability of the proposed method also need to be explored. The next step involves deploying the pipeline to the full-scale NGSIM data in order to assess its computational efficiency and accuracy. Another promising avenue for further exploration is the I-24 MOTION project [21], which can provide large-scale datasets and potential advancements in this domain.

ACKNOWLEDGEMENT

This study is based upon work supported by the National Science Foundation (NSF) under Grant No. 1837652 and the USDOT Dwight D. Eisenhower Fellowship program under Grant No. 693JJ322NF5201 (Wang). The authors appreciate Professor Benjamin Coifman at Ohio State University creating and sharing the manually labeled NGSIM dataset. The authors also appreciate Professor Ahmad Taha and his class CE5999 : Introduction to Optimization at Vanderbilt University [22] for his kind guidance in the field of convex optimization.

REFERENCES

- [1] V. G. Kovvali, V. Alexiadis, and L. Zhang PE, "Video-based vehicle trajectory data collection," Federal Highway Administration, Tech. Rep., 2007.
- [2] C. Chen, L. Li, J. Hu, and C. Geng, "Calibration of mitsim and idm car-following model based on ngsim trajectory datasets," in *Proceedings of 2010 IEEE International Conference on Vehicular Electronics and Safety*. IEEE, 2010, pp. 48–53.
- [3] Z. Zheng, S. Ahn, D. Chen, and J. Laval, "Freeway traffic oscillations: microscopic analysis of formations and propagations using wavelet transform," *Procedia-Social and Behavioral Sciences*, vol. 17, pp. 702–716, 2011.
- [4] Y. Li, D. Wu, J. Lee, M. Yang, and Y. Shi, "Analysis of the transition condition of rear-end collisions using time-to-collision index and vehicle trajectory data," *Accident Analysis & Prevention*, vol. 144, p. 105676, 2020.
- [5] M. Treiber, A. Kesting, and C. Thiemann, "How much does traffic congestion increase fuel consumption and emissions? applying a fuel consumption model to the ngsim trajectory data," in *87th Annual Meeting of the Transportation Research Board, Washington, DC*, vol. 71, 2008, pp. 1–18.
- [6] L. Li, R. Jiang, Z. He, X. M. Chen, and X. Zhou, "Trajectory data-based traffic flow studies: A revisit," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 225–240, 2020.
- [7] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data: accuracy and application to the next generation simulation (ngsim) program data," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1243–1262, 2011.
- [8] A. Sharma, Z. Zheng, and A. Bhaskar, "A pattern recognition algorithm for assessing trajectory completeness," *Transportation research part C: emerging technologies*, vol. 96, pp. 432–457, 2018.
- [9] B. Coifman and L. Li, "A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset," *Transportation Research Part B: Methodological*, vol. 105, pp. 362–377, 2017.
- [10] M. Montanino and V. Punzo, "Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns," *Transportation Research Part B: Methodological*, vol. 80, pp. 82–106, 2015.

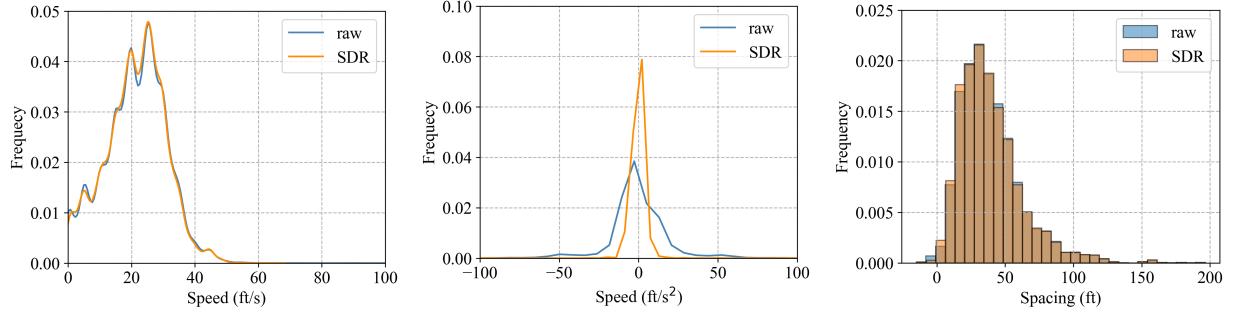


Fig. 4. Speed [left], acceleration [middle], and spacing headway [right] distribution of the experimental data before and after processing

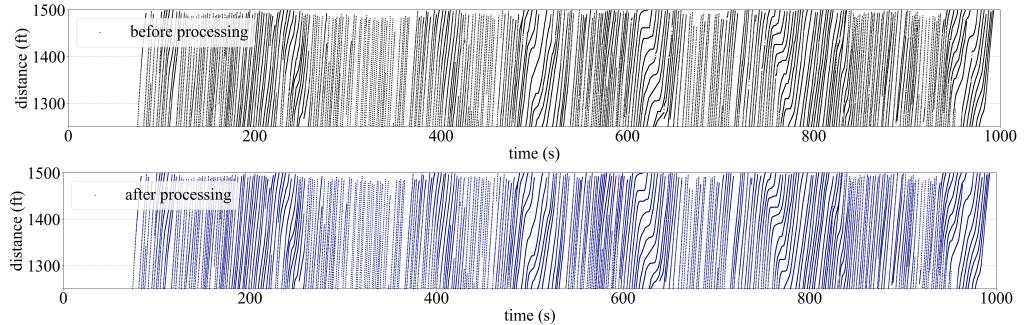


Fig. 5. The trajectory space-time diagram for the full time-space scale of the experimental data: before processing, i.e. raw data [up], after processing, i.e. reconstructed data [bottom].

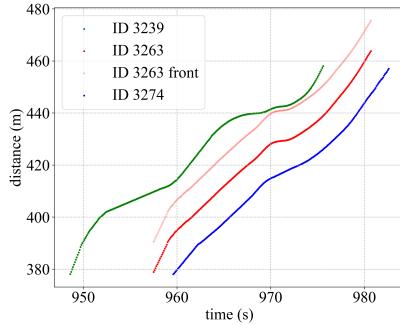


Fig. 6. The space-time diagram for the well-reconstructed false positive collision case shown in Fig. 2.

- [11] ——, “Making ngsim data usable for studies on traffic flow theory: Multistep method for vehicle trajectory reconstruction,” *Transportation Research Record*, vol. 2390, no. 1, pp. 99–111, 2013.
- [12] Y. Wang, D. Gloudemans, Z. N. Teoh, L. Liu, G. Zachár, W. Barbour, and D. Work, “Automatic vehicle trajectory data reconstruction at scale,” *arXiv preprint arXiv:2212.07907*, 2022.
- [13] D. Kang, Z. Li, and M. W. Levin, “Evasion planning for autonomous intersection control based on an optimized conflict point control formulation,” *Journal of Transportation Safety & Security*, vol. 14, no. 12, pp. 2074–2110, 2022.
- [14] T. Schouwenaars, J. How, and E. Feron, “Decentralized cooperative trajectory planning of multiple aircraft with hard safety guarantees,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 5141.
- [15] E. Frazzoli, Z.-H. Mao, J.-H. Oh, and E. Feron, “Resolution of conflicts involving many aircraft via semidefinite programming,” *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 1, pp. 79–86, 2001.

- [16] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, “Semidefinite relaxation of quadratic optimization problems,” *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.
- [17] F. Gao and S. Shen, “Quadrotor trajectory generation in dynamic environments using semi-definite relaxation on nonconvex qcqp,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6354–6361.
- [18] S. Diamond and S. Boyd, “Cvxpy: A python-embedded modeling language for convex optimization,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.
- [19] M. ApS, “Mosek optimizer api for python,” *Version*, vol. 9, no. 17, pp. 6–4, 2022.
- [20] A. Majumdar, G. Hall, and A. A. Ahmadi, “Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 331–360, 2020.
- [21] D. Gloudemans, Y. Wang, J. Ji, G. Zachár, W. Barbour, and D. B. Work, “I-24 motion: An instrument for freeway traffic science,” *arXiv preprint arXiv:2301.11198*, 2023.
- [22] A. Taha, “Lecture notes in introduction to optimization,” April 2023.