Improved Tracking of Articulated Vehicles in Self-Driving Applications Using Phantom Observations

Cuichun Xu, Peter Kingston, Nemanja Djuric

***Abstract*— One of the critical challenges in a self-driving system addressed by the perception module is the tracking of actors in the vicinity of an autonomous vehicle. This is a difficult and complex task, that entails inferring higher-order states of the actors (such as velocity, acceleration, heading, and so on) as well as their tracking through time. The task is particularly challenging in the case of vehicles that have a non-rigid shape, such as articulated vehicles where two or more rigid parts are connected by hitch points. In addition, if we consider a highway case where a large number of such vehicles can be present, the task becomes even more demanding as all surrounding actors need to be tracked very efficiently in real-time. While there exist accurate vehicle kinematic models that account for articulation, applying them directly to tracking of a large number of articulated vehicles can be difficult due to the complexity and high latency cost of such models when used in the update step of the tracker. We focus on this problem and propose an approach that allows for the accurate tracking of articulated vehicles in a very efficient manner. In particular, we achieve this by tracking each rigid part independently and applying tracker updates using both actual observations as well as articulated phantom observations. These phantom observations are computed by considering the articulated kinematic model and the state of the neighboring connected part, which allows for efficient and kinematically consistent update steps. We evaluated the proposed approach on large-scale real-world data collected on highways in Texas, and compared it to a number of state-of-the-art baselines. The results strongly indicate the benefits of using phantom updates for accurate and efficient tracking of articulated vehicles.**

I. INTRODUCTION

The development of autonomous driving technology has a transformative potential that can impact and advance our environment and everyday lives [1]. The potential of the positive impact of autonomous vehicles (AVs) holds particularly true when considering the highways and the amount of truck traffic observed there. In the US alone, in 2011 there were more than 164 thousand miles of highways that were used to transport 16.1 billion tons of freight, with more than 70% being carried by trucks [2]. This clearly exemplifies the significant importance of the highways and in particular trucks to the economy. However, trucks were also involved in a significant number of highway crashes that occurred in 2020, with nearly 47% of fatal and 43% injury-related accidents involving trucks [3]. Improving modeling of the movement of such actors within an autonomous system could have an effect on these statistics by better anticipating the behavior of trucks and other articulated vehicles, thus impacting the efficiency of highway traffic.

Authors are with Aurora Innovation, Pittsburgh, PA 15222, USA. {cuichun.xu, pkingston, ndjuric}@aurora.tech

An autonomous system typically consists of several sequential modules and tasks, commonly including the detection of nearby actors, their tracking through time, and forecasting of their behavior, which is then used within a motion planning system to compute AV movement. In this work we consider the tracking aspect of such a system, focusing on the problem of tracking of articulated vehicles through time that are commonly found on highways. Tracking is not an easy task in general, and becomes significantly more involved for non-rigid objects where it becomes critical to model the relationship between the connected parts. Kinematic and dynamic models that can be used for this purpose exist [4], [5], [6], however equations are complex and expensive to calculate, resulting in significant latency that may not be acceptable in a real-time system. Moreover, given that on highways a large number of articulated actors is usually observed, the latency problem becomes further exacerbated.

We look into this important problem and describe a method to improve the tracking of articulated vehicles by speeding up the use of complex kinematic models, while maintaining good accuracy of the tracker on such actors. Our main contributions are summarized below:

- We propose an efficient method for tracking articulated vehicles through phantom observations used in the update step of the tracker;
- We evaluate the method on a large-scale, real-world highway data set, showing the benefits of the proposed approach when compared to the existing state-of-the-art.

The rest of the paper is organized as follows. In Section II we give an overview of the relevant literature, summarizing the prior work on the modeling of articulated vehicles as well as prior work on their tracking. In Section III we describe the problem setup and the considered vehicle kinematic models used for modeling of articulated vehicles, followed by the description of the proposed approach in Section IV. In Section V we describe the experimental setup, as well as the evaluation on a large-scale data set collected on highways in Texas, followed by Section VI that concludes the work.

II. RELATED WORK

In this section we give a brief overview of related prior work that motivated the proposed method.

A. Articulated vehicle models

Vehicle models are used to encode actor kinematics and compute their next state, given the current state of a traffic actor. A commonly used kinematic model for rigid vehicles (e.g., passenger cars) is a bicycle model [7], however that model is not applicable in the case of non-rigid objects considered in this work. Such models have been expanded through the development of kinematic models for articulated vehicles, which model their articulation by considering the articulation joints between various rigid parts, most commonly between a tractor and a trailer [4], [8]. These models are mainly used in control and planning applications [5], [8], [9], where parameters such as tractor and trailer extents, hitch position, and tractor/trailer weights are available. A large portion of detection and tracking work for articulated vehicles is related to tracking the trailer angle, which is the yaw difference between the trailer and tractor, in order to support control and planning tasks [10], [11]. These approaches all focus on ego-tracking and require specific sensing on the trailer, such as a camera or ultrasonic sensors, while in our work we focus on tracking through the detection of articulated vehicles in the vicinity of an AV.

B. *Tracking of articulated vehicles*

For general tracking, where the autonomous vehicle needs to track other articulated vehicles, to the best of our knowledge there exists a very limited number of publications. This is likely due to the fact that some of the parameters required for accurate tracking, such as hitch position and tractor/trailer weight, are usually hard to measure. Nevertheless, the authors of [12] described an effort to track articulated vehicles using radar measurements. The resolution of automotive radar has increased a lot in recent years which could benefit such an approach, however as discussed in [12], the task of clustering, segmentation, and bounding box estimation are still facing many corner case challenges. The authors of [13] proposed a novel approach to describe an articulated object's state based on deformed super-ellipses, which is shown to have a closed-form equation. On the other hand, we consider tracking-by-detection [14] and use common vehicle models for articulated vehicles, proposing an efficient way to apply them to tracking of such traffic actors.

III. Problem setup

Without the loss of generality, in the remainder of the paper we consider tracking of articulated vehicles that contain a single point of articulation (i.e., consist of exactly one tractor and one trailer). Moreover, we assume that the AV is equipped with a sensor suite allowing it to sense its surroundings, such as cameras and lidars. In addition, we adopt a paradigm of tracking-by-detection where a reasonably-performing 3D actor detector is assumed, whose detections of tractor and trailer are fed to a Kalman filter-based tracker that outputs 3D tracks (discussed in Section III-A). This is a reasonable assumption as a number of deep learning models for object detection have been published recently, relying on lidar and/or camera inputs [15], [16].

When an articulated model is not used, the tractor and trailer will be tracked either as one large entity covering both or as two independent entities, with no interconnected or consistent kinematics. In such cases, the tractor and trailer tracks can start to have inconsistent tracks, since the articulation is not taken into account. This is particularly problematic when the detector performance is reduced, such as in the case of occlusions. For example, when the actor is in front of the AV the trailer can partially occlude the view of the tractor leading to degraded tractor detections, or vice versa for actors in the rear of the AV. Another example is when a fast articulated vehicle is overtaking the AV, in that case the sensor viewing angle on the articulated vehicle could change considerably in a short period, leading to an increase in detection error.

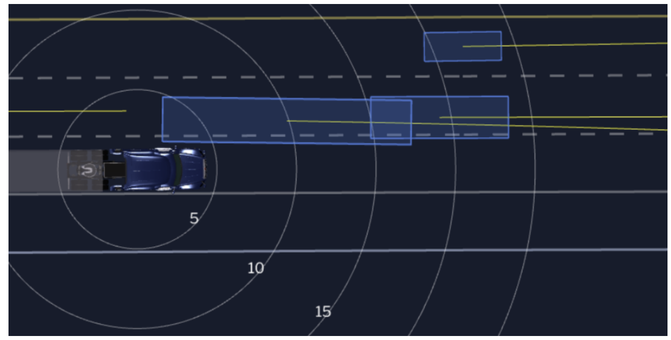

Fig. 1. Inconsistent tracking without articulated vehicle kinematics

This situation is illustrated in Figure 1, showing a real-world example captured in our data. In particular, the AV is a semi-truck located at the center of the range rings. The adjacent actor, which is an articulated vehicle, is shown as two independently-tracked overlapping bounding boxes that are overtaking the AV. The yellow lines starting from the centers of the track boxes indicate predicted ballistic trajectories computed using the track states such as speed, acceleration, and yaw rate. Upon closer inspection of Figure 1 we can see that the predicted track trajectory of the trailer is inconsistent with the tractor state, although they should be near-identical as in reality the two tracks are connected.

A. *Brief explanation of the tracker*

A typical tracker has three highly related components: (1) associator, (2) lifetime manager, and (3) updater. The associator component associates current tracks to new detections, thus establishing temporal consistency. The lifetime manager component proposes new tracks on detections not associated with any of the current tracks, and also retires tracks that have no associated detection for a certain period. Finally, the updater component updates the track state with the associated detections. The updater also smooths out the detection noise and can be used to compute higher-order states that may not be directly observable, such as yaw rate or acceleration.

Depending on the detector output and system requirements, different states could be used to describe a track. Without loss of generality, in this work we assume that a track state for a rigid vehicle is a 12-D vector given as

$$\mathbf{z} = [x, y, z, \theta_x, \theta_y, \theta_z, v, a, c, l, w, h]^\top, \tag{1}$$

where x, y and z represent track center position in the world coordinate, θ_x, θ_y, and θ_z are the vehicle orientation at the

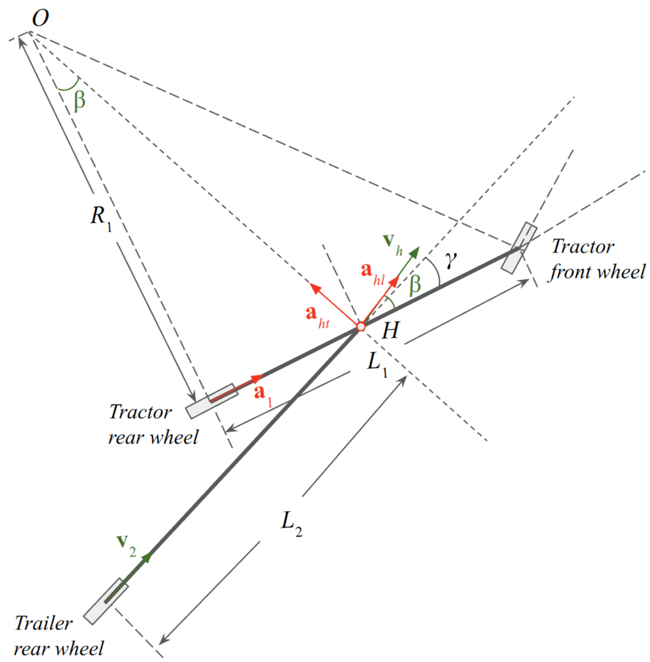

Fig. 2. Simplified articulated vehicle model

3 dimensions, v is the signed speed at a reference point on the vehicle (e.g., at the middle of the two rear wheels), a is the signed acceleration along the centerline of the vehicle at the reference point, c is the curvature of the reference point (computed as $1/R$, where R is the reference point's turning radius), and finally l, w, and h are the length, width, and height of the vehicle bounding box, respectively. We note that it can be shown that the vehicle yaw rate at the reference point is cv. With the state vector defined as in (1), we can see that the state covariance would be a 12×12 matrix.

The updater component is most commonly chosen to be an iterative Bayesian filter [17], which has two steps in one update cycle. First, in the prediction step, the prior track state is predicted to the observation time, and as a result track state covariance increases. Then, in the update step, information from the associated detection is incorporated into the track state, and as a result track state covariance decreases. The most popular iterative Bayesian filter is the Kalman filter [18], which we also employ in our experiments.

To optimally track an articulated vehicle, one needs to combine both the tractor state vector and the trailer state vector into a single state vector of length 24, and reason about the joint kinematics in the prediction and update steps. The hitch position can be determined by some heuristics (e.g., at the rear end of the tractor). Alternatively, we can add hitch position to the state vector, in which case the total state dimension is equal to 25.

B. Tracking articulated vehicles using artic. vehicle model

Based on the simplification of a popular bicycle model [19], [20], a simplified articulated vehicle model considered in our work is illustrated in Figure 2. Note that the actual length of both tractor and trailer can be larger than shown in the figure, as the vehicle extents can go beyond the wheelbase. Two front wheels of the tractor are combined into one wheel, and similar holds for two rear tractor wheels and two rear trailer wheels. The reference point of the tractor is defined as the center of the rear wheel, which the tractor is assumed to be rotating around. Similarly, the reference point of the trailer is defined to be the center of the trailer's rear wheel. In Figure 2, H marks the hitch position where the trailer is attached to the tractor. Hitch displacement L_h is defined as a signed distance from the tractor reference point along the tractor axis. If the hitch is in front of the tractor reference point (as in Figure 2) the hitch displacement will be positive. Other variables in Figure 2 are defined in the top half of Table I, while the bottom half defines variables not shown in Figure 2 to avoid clutter, but that are still used in kinematic equations in the later parts of this section. In Table I vectors are shown in bold and scalars in italics.

TABLE I
VARIABLES IN THE ARTICULATED VEHICLE MODEL

O	instantaneous center of rotation of tractor
R_1	tractor rear wheel turning radius
H	hitch position
L_1	tractor wheelbase
L_2	trailer wheelbase
β	slip angle at the hitch
γ	trailer angle
$\mathbf{a}_1$	tractor longitudinal acceleration at rear wheel
$\mathbf{a}_{ht}$	tangential hitch acceleration
$\mathbf{a}_{hl}$	longitudinal hitch acceleration
$\mathbf{v}_2$	trailer rear wheel velocity vector
$\mathbf{v}_h$	hitch velocity vector
$\mathbf{v}_1$	tractor rear wheel velocity vector
L_h	hitch displacement
c_1	tractor curvature, or $1/R_1$
c_2	trailer curvature, or $1/R_2$
θ_1	tractor angle in world coordinate
θ_2	trailer angle in world coordinate
ω_1	tractor yaw rate
ω_2	trailer yaw rate
$\dot{\omega}_1$	tractor yaw acceleration
$\dot{\omega}_2$	trailer yaw acceleration

While in the updater we can directly use the full joint state vector and the provided articulated vehicle model (as discussed in Section III-A), there are certain concerns associated with such an approach. Firstly, increased state dimensionality leads to larger complexity of the tracker, making it more difficult to tune and deploy. Secondly, and importantly for real-world applications, using the combined observation will significantly increase the tracker latency due to the use of matrix inverses and other matrix operations in the update steps [18], which in the case of inverses at best amounts to sub-cubic complexity in the observation dimensionality [21].

IV. PROPOSED APPROACH

To address the problems discussed in the previous section, we introduce phantom state updates that will help make tractor and trailer states more kinematically consistent, while

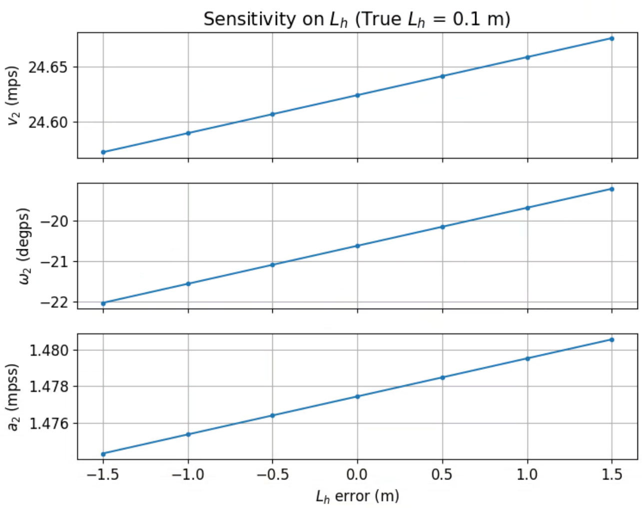

Fig. 3. Phantom observation of the trailer state using noisy L_h

at the same time avoiding a large increase in tracker complexity and latency. In particular, we start from the common approach where two separate tracks for trailer and tractor are used, which are predicted and updated independently in the updater. However, after these independent updates of the two components, we propose to introduce a second phase of the update step and create a phantom observation of one of the components that is consistent with the articulated vehicle model. This phantom update is thus kinematically consistent with its connected component, and we use it to additionally update the state of the original component. If there are kinematic inconsistencies between the two components as exemplified in Figure 1, updating with the phantom observation would bring their kinematics into a better alignment. This keeps the updates efficient as the states of both trailer and tractor are still 12-D vectors (as opposed to a 24-D vector for a joint tractor-trailer state), while ensuring that the updates do encode kinematic consistency.

The form or dimension of the phantom observation can be very flexible. Depending on what we would like to improve for the two independently tracked parts, the phantom observation could be as simple as single-dimensional (e.g., only curvature), or alternatively it can be a full higher-order vector as in (1), depending on the modeler's choice. In this work we focus on computing phantom observation $\tilde{\mathbf{z}} = [v, a, c]^\top$, which we derive in the following section.

A. Computing phantom observations

We assume that state estimates are known for both tractor and trailer ($\mathbf{z}_1$ and $\mathbf{z}_2$, resp.), as well as an estimate of the hitch location L_h (e.g., can be approximated as the mid-point between the bounding boxes for the two independent parts).

Let us first compute phantom observation for the trailer $\tilde{\mathbf{z}}_2$ given the tractor state $\mathbf{z}_1$, and we will consider the reversed roles later. From the basic bicycle vehicle model [20], the speed at the hitch point can be computed as

$$v_h = \sqrt{v_1^2 + (\omega_1 L_h)^2}, \tag{2}$$

while the hitch slip angle is

$$\beta = \arctan(\omega_1 L_h / v_1). \tag{3}$$

Since $\omega_1 / v_1 = c_1$, equation (3) is equivalent to

$$\beta = \arctan(c_1 L_h). \tag{4}$$

The trailer reference point speed v_2 can be obtained by projecting $\mathbf{v}_h$ to the trailer center line. Considering Figure 2 and denoting a magnitude of $\mathbf{v}_h$ as v_h, v_2 is given by

$$v_2 = v_h \cos\alpha, \tag{5}$$

where $\alpha = \beta - \gamma$ is often called the "fifth-wheel" angle or the trailer's virtual front wheel angle. Using the basic bicycle model, the trailer yaw rate can be derived from the trailer's fifth-wheel velocity and the fifth-wheel angle as

$$\omega_2 = v_h \sin\alpha / L_2. \tag{6}$$

Using (5) we can derive $\omega_2 = c_2 v_2 = c_2 v_h \cos\alpha$, and equating the last expression to the right-hand side of (6) we can compute the curvature at trailer reference point as

$$c_2 = \tan\alpha / L_2. \tag{7}$$

Using point O as the reference point and assuming the tractor is a rigid body, the acceleration vector at the hitch point is given by

$$\mathbf{a}_h = \dot{\boldsymbol{\omega}}_1 \times \overrightarrow{OH} + \boldsymbol{\omega}_1 \times (\boldsymbol{\omega}_1 \times \overrightarrow{OH}), \tag{8}$$

where $\dot{\boldsymbol{\omega}}_1$ is the yaw acceleration vector, $\times$ denotes cross product, and $\overrightarrow{OH}$ is a vector pointing from O to H. From (8) we can see that $\mathbf{a}_h$ has two orthogonal components. The two components are denoted in Figure 2 as $\mathbf{a}_{hl}$ and $\mathbf{a}_{ht}$, respectively, both plotted in red. Since $\omega_1 = v_1 c_1$, by rigid body assumption we can compute the norm of $\dot{\boldsymbol{\omega}}_1$ as

$$\dot{\omega}_1 = a_1 c_1. \tag{9}$$

From Figure 2 it can be seen that the signed magnitude of $\overrightarrow{OH}$ is $\overline{OH} = R_1 / \cos(\beta) = \frac{1}{c_1 \cos(\beta)}$, and using (4) and trigonometry we can derive

$$\overline{OH} = \frac{1}{c_1}\sqrt{c_1^2 L_h^2 + 1}. \tag{10}$$

Using (9) and (10) the $\mathbf{a}_{hl}$ magnitude can be computed as

$$a_{hl} = a_1 \sqrt{c_1^2 L_h^2 + 1}. \tag{11}$$

With the same derivation but applied to the trailer, we have

$$a_{hl} = a_2 \sqrt{c_2^2 L_2^2 + 1}. \tag{12}$$

Equating the right-hand sides of (11) and (12), the trailer longitudinal acceleration at the reference point is

$$a_2 = a_1 \frac{\sqrt{c_1^2 L_h^2 + 1}}{\sqrt{c_2^2 L_2^2 + 1}}. \tag{13}$$

Thus, we have found that using the tractor state, hitch position, and trailer angle we can get phantom observations on trailer speed, curvature, and acceleration, shown in equations (5), (7), and (13), respectively. We can see that the

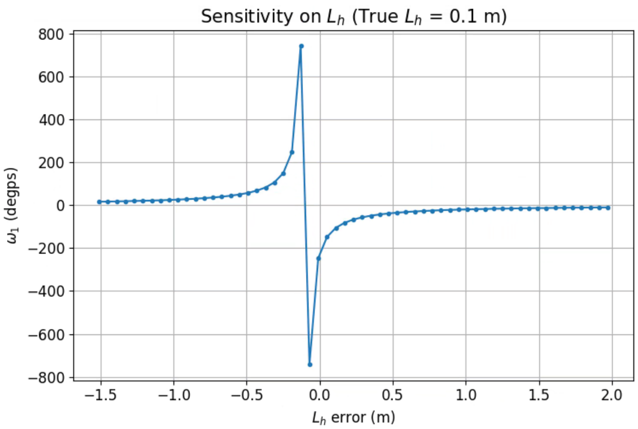

Fig. 4. Phantom observation of the tractor state using noisy L_h

phantom observations of the trailer do not depend on L_1, and empirically we found that the system is also not overly sensitive to noise in L_h and L_2.

We investigate the sensitivity by considering typical state values for an articulated vehicle. In particular, in Figure 3 we assume $L_h = 0.1m$, $L_2 = 12m$, $\omega_1 = 0.2$ rad$/s$, $v_1 = 25m/s$, $a_1 = 1.5m/s^2$, and $\gamma = 10°$, with the three subplots showing v_2, ω_2, and a_2 computed with different L_h errors. It can be seen that phantom trailer observations are quite robust, with limited sensitivity to the noise in L_h.

Let us now consider the case where we want to compute phantom observation of tractor $\tilde{\mathbf{z}}_1$ given trailer state $\mathbf{z}_2$. Through similar derivations as above it can be shown that

$$v_1 = v_h \cos(\alpha + \gamma), \tag{14}$$

where α is the trailer's fifth-wheel angle, given by

$$\alpha = \arctan(c_2 L_2). \tag{15}$$

The tractor reference point curvature can then be derived as

$$c_1 = \tan(\alpha + \gamma)/L_h, \tag{16}$$

and finally

$$a_1 = a_2 \frac{\sqrt{c_2^2 L_2^2 + 1}}{\sqrt{c_1^2 L_h^2 + 1}}. \tag{17}$$

The above equations show that using the trailer state and hitch position we can derive phantom observations of the tractor's speed, curvature, and acceleration. Note that the phantom speed observation is expected to be accurate, as both v_h and α can usually be accurately estimated. However, from (16) we see that c_1 and equivalently ω_1 are quite sensitive to L_h, and thus very susceptible to noise.

We investigate this sensitivity in Figure 4. Here we also assume typical state values for an articulated vehicle, namely $L_h = 0.1m$, $L_2 = 12m$, $\omega_2 = -0.4$ rad$/s$, $v_2 = 25m/s$, $a_2 = 1.5m/s^2$, and $\gamma = 10°$, and compute ω_2 with different errors in L_h. It can be seen that small errors in L_h could result in large changes in the tractor yaw rate. For this reason we do not use the derived yaw rate in the phantom tractor update, and only focus on phantom speed and acceleration.

V. Experiments

As discussed in Section IV, the choice of which phantom update to use is very flexible. In this section we present an empirical evaluation of the phantom update approach, applied to two important cases where an AV commonly interacts with an articulated vehicle. In particular, we consider a case of an articulated actor overtaking the AV, where a phantom observation of $\tilde{\mathbf{z}}_2 = [v_2, a_2, c_2]^\top$ is used to update the trailer track state. In addition, we consider a case where an AV is following a lead articulated vehicle with a heavily occluded tractor, where a phantom observation $\tilde{\mathbf{z}}_1 = [v_1, a_1]^\top$ is used to update the tractor track state.

A. Experimental setup

The track state vector is given by (1) and the reference point (i.e., the location of the rear wheel) is assumed to be 30% of the actual track length behind the track midpoint (x, y, z). The hitch position is assumed to be at the tractor reference point if the tractor and trailer bounding boxes overlap, otherwise it is assumed to be at the end of the tractor's bounding box. In the prediction step of the Kalman filter the independent tracks are predicted with a bicycle model. When applying the phantom observations in the update step, we increase the observation noise covariance matrix by a factor of 4 to ensure the state variances are not overconfident due to two state updates.

We compare the proposed approach (referred to as "phantom update") with the following relevant baselines:

- treat the tractor and the trailer as one rigid track whose state is represented with a 12-D vector, referred to as "single update", where the single bounding box is obtained by computing the minimal enclosing bounding box over the tractor and trailer detections;
- treat the tractor and the trailer as independent vehicles and update them separately, referred to as "independent update" (note that this is a default setting when no articulation is assumed);
- treat the tractor and the trailer as one articulated track whose state is represented with a full 24-D vector, referred to as "full update".

The competing approaches were evaluated on labeled data of large vehicles collected on Texas highways between 2021 to 2023, comprising around $100{,}000$ actor samples. We compare the results on three separate situations involving articulated vehicles that are encountered during normal highway operations of autonomous vehicles. In particular, we consider the following cases: (1) an articulated actor is passing the AV, (2) an articulated actor is turning in front of the AV (encountered on surface roads before and after the highway portion of the AV's route), and (3) the AV is following a lead articulated vehicle. In each of the cases we apply the proposed approach only to tracks that are less visible and would thus benefit most from additional phantom observations, resulting in phantom updates to trailer, trailer, and tractor, respectively, where we also report the metrics only on these particular tracks for the considered cases.

TABLE II

COMPARISON OF THE COMPETING METHODS SHOWING PERFORMANCE RELATIVE TO "INDEPENDENT UPDATE" (BEST RESULTS BOLDED)

	Passing trailer				**Turning trailer**				**Lead tractor**			
Method	loc.	orien.	velo.	accel.	loc.	orien.	velo.	accel.	loc.	orien.	velo.	accel.
Single update	0.96	1.01	0.72	0.90	3.74	6.62	0.82	**0.61**	1.31	1.07	1.05	1.04
Independent update	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	1.00	1.00
Full update	0.93	1.04	1.12	1.53	**0.78**	**0.93**	0.81	0.75	**0.93**	1.14	1.22	1.30
Phantom update	**0.83**	**0.98**	**0.71**	**0.82**	0.96	1.23	**0.72**	0.63	**0.93**	1.16	**0.81**	**0.97**

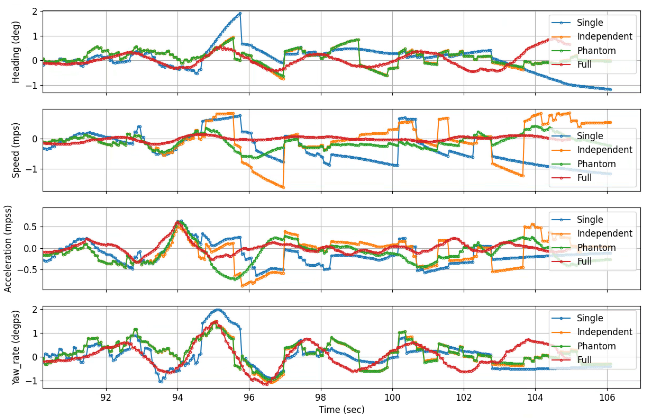

Fig. 5. State error on the occluded tractor of a lead articulated vehicle

B. Results

In Table II we present results of the competing methods on the three considered cases reported relative to the results of the default "independent update" method. We report errors on the relevant track states, including error of the nearest track corner to the AV (shown as "loc."), as well as errors in orientation ("orien."), velocity ("velo."), and acceleration ("accel."), where in all cases lower values are better.

We can see that "single update" had quite competitive performance, especially when it comes to the results on passing trailer and lead tractor, where the rigid parts of an articulated vehicle are expected to be aligned well. However, in the turning case this method expectedly broke down, as a single large track is no longer sufficient to model both tractor and trailer. When it comes to the full update, we can see that the location error was quite low, while the higher-order states had a larger error relative to other baseline methods. This is somewhat unexpected, as one would expect that the full update is the best choice to track articulated vehicles as it best fits the underlying kinematics of such actors. However, due to the complexity of the model and the number of parameters that need to be calibrated, this method is difficult to tune and is much less robust than the competing approaches. During the empirical analysis we attempted to tune this method to ensure good performance, however large dimensionality of the problem proved the effort both cumbersome and costly. Moreover, the latency of the "full update" approach is another important downside, discussed in more detail at the end of this section. Lastly, we can see that the proposed "phantom update" method had the most consistent performance compared to the other baselines, reaching the best performance in many reported metrics and across different conditions.

In Figure 5 we provide detailed tractor state errors of a case of a lead articulated vehicle. During this event the tractor was either partially or fully occluded, leading to suboptimal detections that impacted the quality of the tracker output. We can see that the proposed phantom approach had less extreme and in general smaller state errors than either the single or independent update schemes, due to kinematic-constrained updates that the phantom observations provide. We can also see that the phantom approach had a performance closely matching the one by the full update scheme. For this particular case the full update scheme had the best performance with the lowest state errors, however in the following paragraph we discuss a major downside that needs to be considered when using the full update.

Lastly, we compared the latency of running a single iteration of the Kalman filter when a full joint state is used, versus when running phantom updates on separate tractor and trailer states. On our mid-tier workstation the full update took $3.05\times$ more time than running a phantom update on a single track. Even assuming that we apply phantom observations on both trailer and tractor which would take two times longer, the full update is still around 50% slower than running phantom updates on individual tractor and trailer tracks. This additional latency becomes particularly impactful when a large number of articulated vehicles is encountered in a traffic scene, which is not uncommon in everyday highway driving.

VI. CONCLUSION

We considered the task of improving the perception system used for highway driving of autonomous vehicles. We focused on the problem of tracking of articulated vehicles, which are commonly observed on highways and pose significant challenges to the tracker due to the complexity of modeling articulation between the rigid parts of such traffic actors. To ensure accurate and efficient tracking, we proposed to use phantom observations in the tracker's update step, computed by considering the current state of the connected part and using existing articulated kinematic models. This results in a significant speedup of the update step for articulated vehicles, that avoids expensive and complex computations. The proposed approach was evaluated on large-scale data consisting of thousands of driven miles on Texas highways, showing that the method allows for accurate and efficient tracking of large numbers of articulated vehicles.

References

[1] T. J. Kim *et al.*, "Automated autonomous vehicles: prospects and impacts on society," *Journal of Transportation Technologies*, vol. 8, no. 03, p. 137, 2018.

[2] FHWA, "Our nation's highways: 2011," US Department of Transportation, Tech. Rep. FHWA-PL-10-023, May 2010.

[3] NHTSA, "Traffic safety facts annual report," National Highway Traffic Safety Administration, Tech. Rep. DOT HS 812 826, 2020.

[4] C. Chen and M. Tomizuka, "Dynamic modeling of articulated vehicles for automated highway systems," in *Proceedings of 1995 American Control Conference-ACC'95*, vol. 1. IEEE, 1995, pp. 653–657.

[5] T. Nayl, "Modeling, control and path planning for an articulated vehicle," Ph.D. dissertation, Luleå tekniska universitet, 2013.

[6] Q. Meng, L. Gao, H. Xie, and F. Dou, "Analysis of the dynamic modeling method of articulated vehicles." *Journal of Engineering Science & Technology Review*, vol. 10, no. 3, 2017.

[7] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, 2017, pp. 812–818.

[8] J. Yang, S. Kim, and K. Huh, "Development of a unified lane-keeping and collision avoidance system for semi-trailer truck," *IEEE Access*, vol. 8, pp. 149 751–149 763, 2020.

[9] M. Bellone and J. Qutteineh, "Extension of trajectory planning in parameterized spaces to articulated vehicles," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–6.

[10] M. Bahramgiri, S. Nooshabadi, K. T. Olutomilayo, and D. R. Fuhrmann, "Hitch angle estimation for trailer backup system—an object detection and tracking approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, 2022.

[11] A. Habibnejad Korayem, A. Pazooki, L. Durali, A. Khajepour, B. Fidan, A. V. Ponnuswami, and S. P. Khaligh, "Hitch angle estimation of a towing vehicle with arbitrary configuration," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7535–7546, 2022.

[12] F. Gustavsson and L. Hazard, "Tracking of articulated vehicles using radar measurements," 2020.

[13] L. A. Giefer and J. Clemens, "State estimation of articulated vehicles using deformed superellipses," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–8.

[14] K. R. Reddy, K. H. Priya, and N. Neelima, "Object detection and tracking–a survey," in *2015 International conference on computational intelligence and communication networks (CICN)*. IEEE, 2015, pp. 418–421.

[15] Y. Zhou, P. Sun, Y. Zhang, D. Anguelov, J. Gao, T. Ouyang, J. Guo, J. Ngiam, and V. Vasudevan, "End-to-end multi-view fusion for 3d object detection in lidar point clouds," in *Conference on Robot Learning*. PMLR, 2020, pp. 923–932.

[16] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4604–4612.

[17] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[18] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[19] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *2015 IEEE intelligent vehicles symposium (IV)*. IEEE, 2015, pp. 1094–1099.

[20] M. Guiggiani, "The science of vehicle dynamics," *Pisa, Italy: Springer Netherlands*, vol. 15, 2014.

[21] R. Duan, H. Wu, and R. Zhou, "Faster matrix multiplication via asymmetric hashing," *arXiv preprint arXiv:2210.10173*, 2022.