2020 16th IEEE International Conference on Automation Science and Engineering (CASE)
August 20-21, 2020, Online Zoom Meeting

Optimization based path planning for a two-body articulated vehicle

Deyuan Chen, Zhiqiang Yang, Lars Svensson and Lei Feng

***Abstract*— An articulated vehicle is a two-body design capable of precise maneuvering around obstacles, while carrying heavy loads over rough terrain. In the context of path planning for automated articulated vehicles, it is desirable to fully utilize the maneuverability of the vehicle to enable autonomous operation in confined areas. In this paper we study the impact of model accuracy in an optimization based path planner for an articulated vehicle. For this purpose, we compare the traditional kinematic bicycle model with a two-body articulated model. We evaluate performance in terms of path length, path quality, success rate and computation time through performing test queries in artificial environments and through experiments on a full scale articulated hauler. Results show that for simple, unidirectional maneuvers, performance differences are small, but for more difficult bidirectional maneuvers, the articulated model produces shorter and higher quality paths at a higher success rate. However, the articulated model has 2.75 times longer computation time on average.**

I. Introduction

The articulated hauler illustrated in Fig. 1 is a two-body vehicle designed for a variety of heavy-duty transportation tasks, with an articulated steering mechanism that enables agile maneuvering with heavy loads over rough terrain. Automation of hauler vehicles in mining and construction environments is motivated by improving productivity, fuel efficiency and safety of operators. For any automated vehicle, path planning is a key functionality that computes a sequence of states between the initial and the goal states of the vehicle. The sequence should avoid collision with obstacles and be feasible with respect to the dynamics of the vehicle [1]. Performance of a path planning algorithm is determined by the length and quality of the produced paths, as well as the computation time. For example in a production context such as mining or construction, a planner must produce a path that is short, accurate and feasible for complex maneuvers in tight spaces at an acceptable computational cost. Overly restrictive planning, leading to inability to find a feasible collision free solution may significantly reduce productivity in a tightly spaced production environment.

At the core of any path planning algorithm is a vehicle representation that ensures dynamic feasibility of the obtained path. The overwhelming majority of academic works in path planning for non-holonomic vehicles use variations of the kinematic bicycle model [2, 3] for this purpose. For most path planning applications, the kinematic bicycle model represents a suitable trade-off between accuracy and complexity. However, since the kinematic bicycle model does not account for the two-body motion of an articulated machine, it is less accurate in predicting the motions of such a machine at low speeds [4]. Implications of modelling error in the path planning context are twofold.

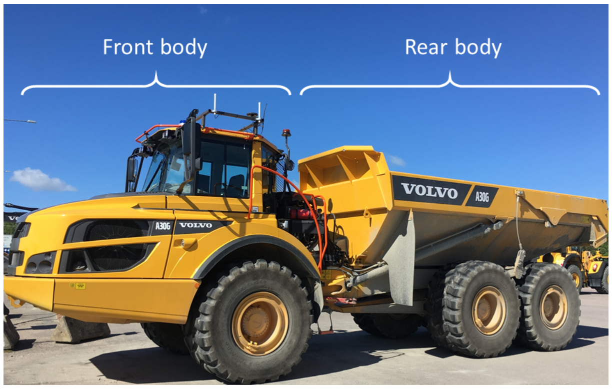

Fig. 1: The articulated hauler used for experiments. The articulated steering mechanism steers the vehicle by manipulating the angle between the front and rear bodies. Image courtesy of Volvo CE

i) The planner produces low quality paths, which the real vehicle is unable to follow accurately, possibly leading to collision with obstacles.
ii) The planner is unable to find a solution in an environment that the vehicle is actually capable of traversing.

On the other hand, a more complex model with a larger state space, and multiple bodies to keep out of collision will increase the computational cost [1].

In this paper, we evaluate the performance implications of using the single-body kinematic bicycle model or a two-body articulated kinematic model [5] in path planning for an articulated hauler. The comparison is done by evaluating performance metrics such as path length and quality (capability of the real vehicle to track the path), as well as average success rate and computation time in our implementation of a state of the art optimization based path planning framework [6]. Results are obtained through performing test queries in artificial environments and through full scale experiments. In summary, the contributions of the paper are as follows:

- We extend an optimization based path planning framework [6], to accommodate a two-body articulated vehicle model [5].
- We evaluate path planning performance with the new model, using the single-body kinematic bicycle model as a benchmark.

The remaining sections of the paper are structured as follows. In Section II, we introduce related academic works in path planning algorithms and modelling of articulated ve-

The authors are with the Department of Machine Design, KTH Royal Institute of Technology, Stockholm, Sweden. {deyuan,zyan,larsvens,lfeng}@kth.se

978-1-7281-6904-0/20/$31.00 ©2020 IEEE

hicles. In Sections III and IV, we introduce the two kinematic models used in the evaluation, and the optimization based path planning framework. Sections V and VI present the experiments and results, and finally, conclusions are stated in Section VII.

II. Related Work

Path planning algorithms are commonly classified into random sampling based, graph search based, interpolating curve based and optimization based methods [3]. For this work we focus on optimization based path planning in which the motion planning problem is represented as an optimization problem, in which the cost function and constraints of the problem encode the desired behavior. We select this method for three specific reasons. First, with an optimization based approach, we can explicitly express the vehicle model differential equations as constraints [7]. Second, the method does not rely on discretization and search of the state space, and therefore allows more precise maneuvering in confined spaces with obstacles [6]. Third, computational cost of numerical optimization based path planners scales comparatively well with the number of states in the model [8], thus potentially allowing the usage of the more complex articulated model.

In optimization based path planning, collision avoidance is handled by adding constraints to the state space, such that the vehicle body is not allowed to intersect with obstacles. This is straightforward if the vehicle is represented as a point mass model by limiting the minimum distance between the controlled point and the obstacle [9, 10, 11, 12]. This method generates relatively conservative paths and may fail to find a path in confined areas where the geometric shape of the vehicle needs to be considered, especially for the articulated vehicle which has multiple bodies. A more accurate method was recently introduced by Zhang et al. [6]. By utilizing strong duality of convex optimization, the full-dimensional obstacle avoidance constraints can be formulated into an optimization problem in which the obstacles and the bodies of the vehicle are regarded as different convex sets without penetration with each other. This convex optimization problem is then integrated into the motion planning problem, providing collision avoidance with respect to a precise representation of the vehicle shape.

The path planning problem for articulated steering vehicles have been previously studied in the fields of robotics and vehicle automation. Choi and Huhtala [12] proposes a three stepped approach. An initial guess from a lattice planner is parameterized by a sequence of Bezier splines. Parameters of the splines are are optimized by solving an NLP and finally a gradient based path smoothing procedure is applied to inhibit unnecessary swerving. The method uses collision checking along paths in a discretized distance map to determine proximity to obstacles, which is less precise then for example [6].

Furthermore, there are several previous academic works proposing accurate kinematic models of articulated vehicles for the purpose of path tracking control [4, 5]. An accurate two-body kinematic model in combination with a precise representation of the vehicle shape in relation to obstacles have the potential to fully utilize the maneuvering potential of an articulated vehicle in a tightly spaced production environment.

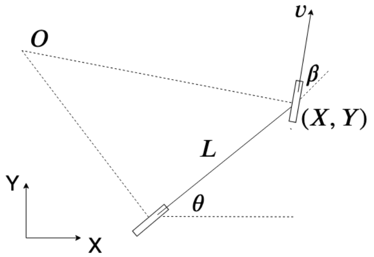

Fig. 2: Geometric relations of the kinematic bicycle model. Differential equations in (1).

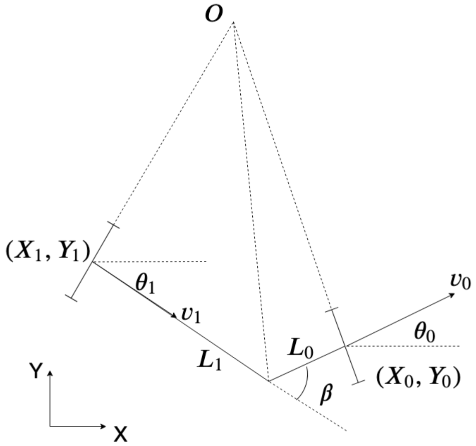

Fig. 3: Geometric relations of the kinematic articulated vehicle model [5]. Differential equations in (2).

III. Vehicle models

When operating in unstructured areas, the speed of an articulated hauler is typically low. Therefore, we do not take the dynamic effects caused by inertia into consideration, nor the effects of tire slip. This section introduces two kinematic models to represent the vehicle in the path planning algorithm.

The geometric relations of the bicycle model are shown in Fig. 2, with the state propagation described in Equation (1).

$$\begin{aligned}\dot{X} &= v\cos(\theta+\beta)\\ \dot{Y} &= v\sin(\theta+\beta)\\ \dot{\theta} &= \frac{v\sin(\beta)}{L}\end{aligned} \tag{1}$$

In the model, X and Y represent the coordinates of the center of front axle in an inertial frame. θ corresponds to the heading angle of the vehicle. β is the steering angle, v is the speed of the vehicle and L is the distance between the center of the front axle and the center of the rear axle.

The corresponding geometric relations of the kinematic articulated model [5] are shown in Fig 3. The differential equations describing the model are given in (2).

$$\begin{aligned}
\dot{X}_0 &= v_0 \cos\theta_0 \\
\dot{Y}_0 &= v_0 \sin\theta_0 \\
\dot{\theta}_0 &= \frac{v_0 \sin\beta + L_1\dot{\beta}}{L_1 + L_0\cos\beta} \\
\dot{X}_1 &= v_1 \cos\theta_1 \\
\dot{Y}_1 &= v_1 \sin\theta_1 \\
\dot{\theta}_1 &= \frac{v_1 \sin\beta - L_0\dot{\beta}}{L_0 + L_1\cos\beta}
\end{aligned} \tag{2}$$

In the kinematic articulated vehicle model, (X_i, Y_i), $i = 0, 1$ are the coordinates of the center of front axle and the center of rear axle respectively. θ_i corresponds to the heading angle of the chassis at the centers of each axle of the vehicle. β is the steering angle, v_i is the speed at the front and rear axles respectively, L_i is the distance between the center of each axle and the articulation point. See [5] for the derivation of the model.

For the bicycle model, the state vector is $x := [X, Y, \theta, v, \beta]$ and for the articulated model, $x := [X_0, Y_0, \theta_0, v_0, \beta]$. We discretize (1) and (2) using forward Euler discretization and compactly write the state propagation as $x_{k+1} = f(x_k, u_k)$.

IV. Path Planning Algorithm

In this paper, a collision-free path is computed in a two step process, which has been proven successful in several previous works [6, 13, 14, 15]. First, a coarse path is generated by an A* search algorithm [16] with motion primitives [17]. Second, the generated path is used as initial guess of a nonlinear optimization scheme to compute the final path, where the two kinematic models introduced in Section III are used.

A. Generating Initial Guess

A set of motion primitives is generated as straight lines and arcs with limited curvature, to represent the limited steering angle of the vehicle. The heading angle is discretized into 24 directions. For each direction, the search space consists of 6 available motion primitives: $\{forward,\ forward\,\&\,left,\ forward\,\&\,right,\ reverse,\ reverse\,\&\,left,\ reverse\,\&\,right\}$. Examples of motion primitives are shown in Fig 4.

The cost of each motion primitive is proportional to its path length. We also penalize the direction switch to limit the back and forth motion in the obtained path.

The initial guess is generated as a sequence of piecewise constant curvature arcs, but when used as an initial guess for the optimization problem, the path is interpreted as the reference trajectory of the control point of each of the models in Section III by combining the The trapezoidal velocity profile.

B. Obstacle Avoidance

For collision avoidance of the full-dimension body of two models, we adopt the convex optimization based method proposed in [6]. Vehicle bodies are represented by rectangles. One rectangle is used for the single-body bicycle model and two rectangles for the two-body articulated machine model. We take the bicycle model as an example: the space occupied by the vehicle at step k is represented by $\mathcal{E}(x_k)$, where x_k is the state of the vehicle at step k. Obstacles are represented by polygons $\mathcal{O}^{(m)}$, $m = 1, \dots M$, where M is the total number of obstacles. When both vehicle body and obstacles are all defined as convex sets, convex optimization can be applied to determine the minimum distance between the vehicle and the obstacles. We define the optimization problem

$$\min_{p,\,q} \quad dist \tag{3a}$$

$$\text{s.t.} \quad p \in \mathcal{E}(x_k), \tag{3b}$$

$$q \in \mathcal{O}^{(m)},\ m = 1, \dots M, \tag{3c}$$

$$dist = ||p - q|| \tag{3d}$$

where

$$\begin{aligned}
\mathcal{E}(x_k) =& R(x_k)\mathcal{B} + t(x_k),\ \mathcal{B} := \{\, p : Gp \preceq g \,\}, \\
\mathcal{O}^{(m)} =& \{\, q \in \mathcal{R}^2 : A^{(m)}q \preceq b^{(m)} \,\}
\end{aligned} \tag{4}$$

$\mathcal{B}$ is the convex set which describes the initial position of the vehicle. The matrix G and g determine the edges of the convex set. $\mathcal{E}(x_k)$ can be seen as the rotation and translation of the initial convex set $\mathcal{B}$. R is the rotation matrix and t is the translation matrix. Matrix A can be regarded as the normal vector of the obstacle's edge. p and q are a pair of points which are in vehicle's convex set and in obstacle's convex set respectively. In the optimization problem (3). $dist$ equals to 0 if the vehicle model and obstacle intersect. The bicycle model is collision free at step k when the $dist$ is positive for all obstacles.

To enable application to the articulated model, we extend the method by defining, two convex sets $\mathcal{E}_0, \mathcal{E}_1$ that are used instead of $\mathcal{E}$ to describe front body and rear body separately.

C. Optimizing The Path

By incorporating the vehicle kinematic constraints and collision avoidance constraints. The final path is computed by solving the nonlinear optimization problem

$$\min_{\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}} \quad \sum_{k=0}^{N} {u^T}_k H u_k \tag{5a}$$

s.t.

$$x_0 = x_S, x_{N+1} = x_F, \tag{5b}$$

$$x_{k+1} = f(x_k, u_k), \tag{5c}$$

$$h(x_k, u_k) \leq 0, \tag{5d}$$

$$-g^\top \mu_k^{(m)} + (A^{(m)}t(x_k) - b^{(m)})^\top \lambda_k^{(m)} > d_{min}, \tag{5e}$$

$$G^\top \mu_k^{(m)} + R(x_k)^\top {A^{(m)}}^\top \lambda_k^{(m)} = 0, \tag{5f}$$

$$||{A^{(m)}}^\top \lambda_k^{(m)}|| \leq 1, \lambda_k^{(m)} \geq 0, \mu_k^{(m)} \geq 0, \tag{5g}$$

$$k = 0, \dots, N, \quad m = 1, \dots, M, \tag{5h}$$

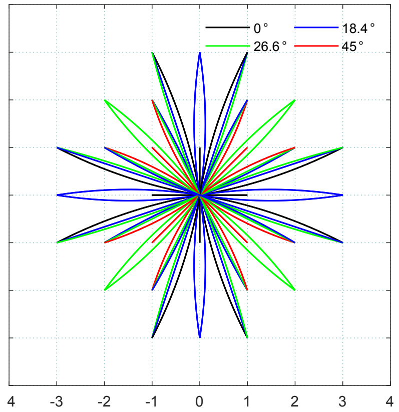

Fig. 4: The set of all motion primitives with the vehicle located at the origin. We discretize a full rotation into 24 steps.

where $V = \sum_{k=0}^{N-1} u^T{}_k H u_k$ is the objective function consisting of the sum of the weighted L2-norms of two control input arrays. H is a positive semi-definite matrix which defines the weight of each control variable. N represents the horizon of the problem and M represents the number of obstacles. The variables consist of three parts: state variables of the vehicle $\boldsymbol{x}$, control inputs $\boldsymbol{u}$ and Lagrange multipliers $\boldsymbol{\lambda}, \boldsymbol{\mu}$. They are the collections of all state vectors, control inputs and Lagrange multipliers over all steps. $\boldsymbol{x} := [x_0, x_1, \ldots, x_{N+1}]$, $\boldsymbol{u} := [u_0, u_1, \ldots, u_N]$ $\boldsymbol{\lambda} := [\lambda_0, \lambda_1, \ldots, \lambda_N]$, $\boldsymbol{\mu} := [\mu_0, \mu_1, \ldots, \mu_N]$. An extra x_{N+1} is used to define the final state constraint. Initial values of variables $\boldsymbol{x}, \boldsymbol{u}$ are obtained by the A* search algorithm described in Section IV-A.

Equations (5b) - (5h) are constraint functions. f represents the state propagation of the kinematic models introduced in Section III. Vehicle and environmental constraints are represented by h. The vehicle constraints include the allowed value ranges of the steering angle β, the speed of the vehicle v and the control input. The environmental constraints define the boundary of the map. The control input of the system is $u_k := [\omega_k, a_k]$, where ω is the angular velocity of the steering, and a is the longitudinal acceleration. Constraints (5e) - (5g) represent the collision avoidance part of the problem and are reformulated from the optimization problem (3) using strong duality of convex optimization. d_{min} is the safety distance between vehicle and obstacles. Details on the reformulation are found in [6].

The total number of steps $N+1$ is positively related to the length of the initial guess. The trapezoidal velocity profile is used to estimate the total traveling time of the path of the initial guess. The nonlinear problem is subsequently solved using Ipopt [18], warmstarted at the initial guess. Ipopt is an open source software package for solving nonlinear problems through interior point methods.

V. Experiment Setup

A. Modelling Parameters

Both models have the same width, and the total length of the articulated model is equal to the length of the bicycle model, i.e., $L = L_0 + L_1$. For the state variables, the steering angles are limited by the interval $\beta \in [-0.6, 0.6]\,rad$ and the velocity by $v \in [-2, 2]\,m/s$. For the control inputs, the acceleration α and the angular velocity ω of the steering angle are limited as $\alpha \in [-0.5, 0.5]\,m/s^2$ and $\omega \in [-0.6, 0.6]\,rad/s$. The sampling time $T = 1s$ is selected. The map and the lateral error are scaled linearly due to the confidential requirement of Volvo CE.

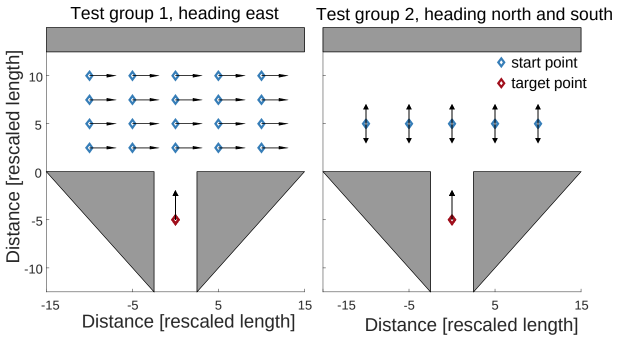
Test group 1, heading east
Test group 2, heading north and south
start point
target point
Distance [rescaled length]
10
5
0
-5
-10
-15
-5
5
15
Distance [rescaled length]
-15
-5
5
15
Distance [rescaled length]

Fig. 5: Test queries in parking scenario.

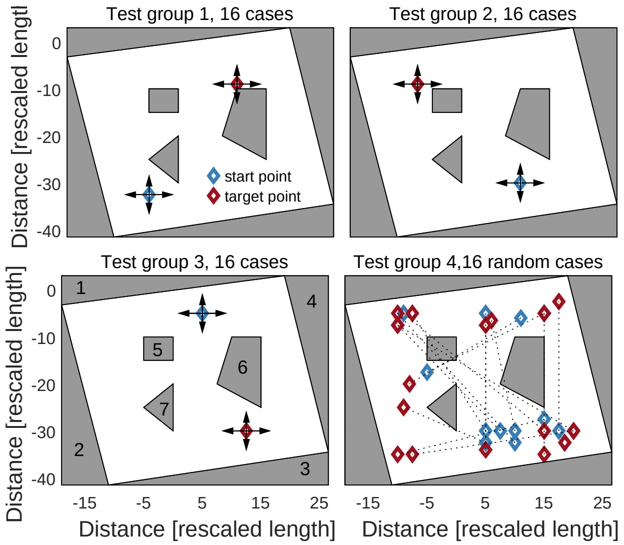
Test group 1, 16 cases
Test group 2, 16 cases
Test group 3, 16 cases
Test group 4,16 random cases
start point
target point
Distance [rescaled length
Distance [rescaled length]
0
-10
-20
-30
-40
1
2
3
4
5
6
7
-15
-5
5
15
25
Distance [rescaled length]
Distance [rescaled length]

Fig. 6: Test queries in transportation scenario.

B. Test Queries

We evaluate the planners over a set of test queries in two scenarios that exemplify typical operations of an articulated hauler: reverse parking and transportation. Initial and final velocities are always 0 m/s.

i) *Reverse parking scenario*
This scenario aims to test the ability of path planner to find the path in a confined environment. Test queries are divided into two groups, in the first group, 20 start points are selected evenly in the map with the same heading toward east. In the second group, 5 start points are evenly selected on a line parallel to the X axis and all start points in this group have two initial headings: north and south. Test queries are shown in Fig. 5.

ii) *Transportation scenario*

The transportation scenario map is created according to a real test track. In this map, Obstacle 1 to Obstacle 4 are used to define the boundaries of the test track area. Obstacle 5 to Obstacle 7 are virtual obstacles that the vehicle shall avoid.

Test queries, Fig. 6, are divided into four groups. In the first three groups, three pairs of the start point and target point are specified along with four different heading directions represented by the arrows in the figure. The paths are computed from all combinations of start directions and target heading directions, that is, 16 cases for each pair of the start point and target point. The fourth group contains several random start points and target points with random heading directions.

The path planning algorithm is implemented in Matlab 2018a and executed on a computer with an AMD 2700x CPU at 3700 MHz and 16GB of RAM.

C. Experiments

The quality of the computed path is evaluated by experiments on a full scale prototype articulated hauler, depicted in Fig. 1. Data are collected from the Global Navigation Satellite System (GNSS). The hauler is equipped with an automation system including an accurate navigation system and a path following controller. The system has been extensively tested by Volvo CE. To reduce the impact of environmental variations, each case is repeated 5 times.

VI. Results and Discussion

We wish to compare the length and quality of paths produced using the two models in Section III, as well as the reliability of the algorithm in the two cases. Path length is defined as the distance travelled by the center of the front axle. Lateral tracking error and heading error are recorded as a measure of the path's quality. In terms of reliability, we run multiple path planning queries in a known map, and evaluate average computation time and success rate over the set of queries.

A. Results From Test Queries

We start by studying single queries in the parking and transportation scenarios. Path plots are shown in Fig. 7. In the parking scenario, path lengths are 28.27 m for the bicycle model and 26.61 m for the articulated model. The models have the same steering angle constraint, but the maximum curvature of the bicycle model's path is less than that of the articulated machine model's path due to the difference in steering structure. The larger curvature range enables the articulated machine model to find a shorter feasible path in confined areas. The corresponding computation times are 4.54 s for the bicycle model and 21.93 s for the articulated machine model. The larger number of optimization variables associated with the more complex model has considerable impact on computation time in this case. We notice a similar trend in the transportation scenario. For the bicycle model, the path length is 40.55m, with a computation time of 49.69 s. For the articulated machine model, the path length is 38.90m with a computation time of 240.26 s.

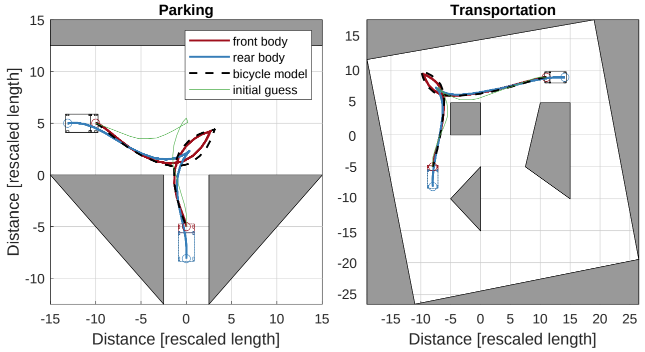

Fig. 7: Path planner outputs from single queries in the parking scenario (left) and transportation scenario (right).

TABLE I: Simulation result summary

	Success rate	Mean length (normalized value)	Mean computation time
Parking			
bicycle	63%	24.25	14.06
articulated	73%	23.05	30.17
Transport			
bicycle	92%	36.71	54.61
articulated	100%	35.86	166.08

To investigate the generality of our results, we evaluate the performance metrics over a set of queries, described in Section V. In total, we run 30 queries in the parking scenario and 64 queries in the transportation scenario. Results are summarized in Table I. In 85% of the queries, the articulated model generates a shorter path than the bicycle model, among these cases, the distance difference is mostly less than 10%. On the other hand, when the articulated machine model generates a longer path length, the length difference is within the range of 5%.

As start and goal points are generated programmatically, queries vary in difficulty and some do not have a solution. We measure success rate by computing the ratio of successful queries over the set. A query is regarded as successful if

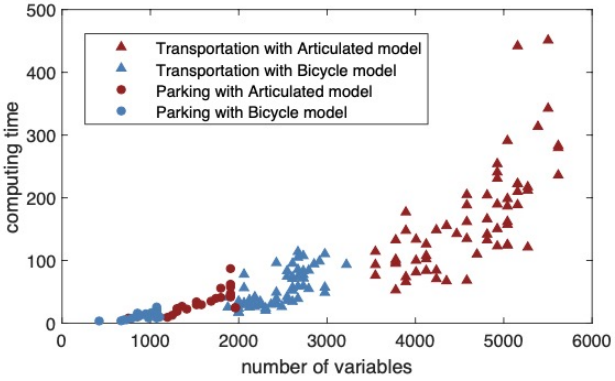

Fig. 8: The Computation time for all converged queries.

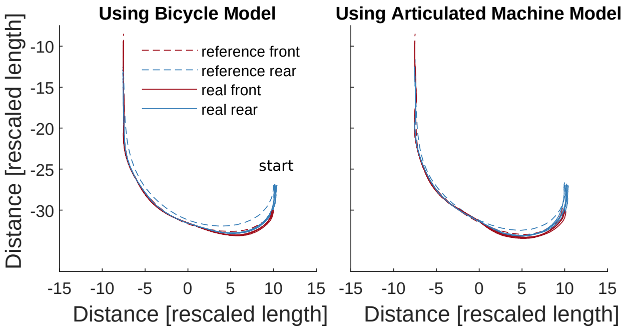

Fig. 9: Trajectories in real test, query 1. Repeated 5 times.

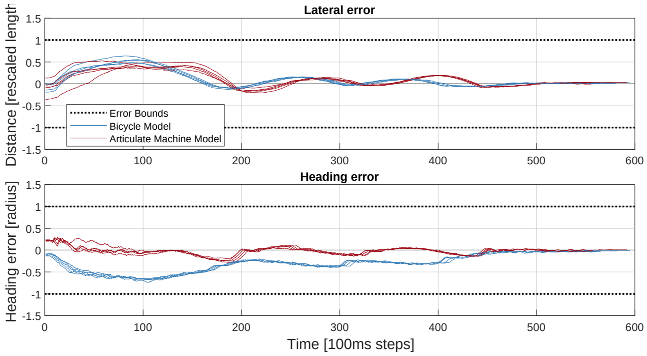

Fig. 10: Errors in real test, query 1. Repeated 5 times.

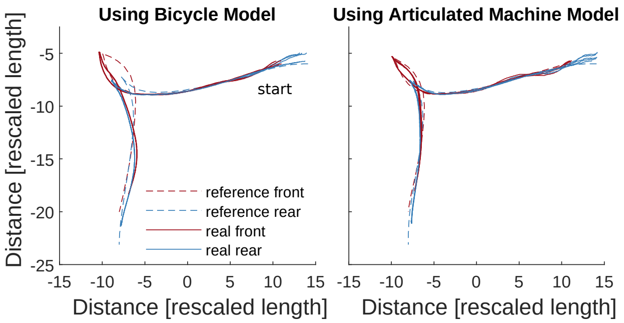

Fig. 11: Trajectories in real test, query 2. Repeated 5 times.

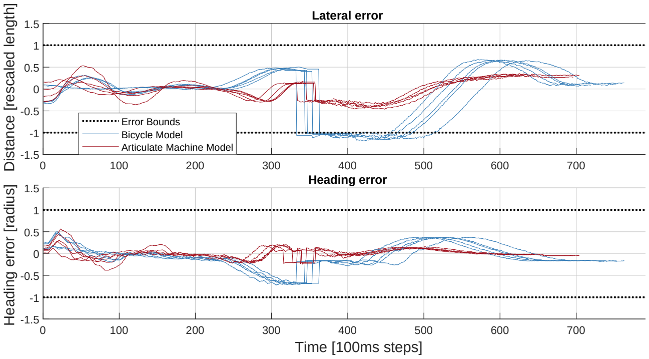

Fig. 12: Errors in real test, query 2. Repeated 5 times.

the solution is feasible in simulation. We note that provided with the same warm start algorithm, the articulated model has higher success rate for both scenarios. The underlying reason is that the articulated model better captures the kinematics that gives the articulated vehicle its maneuverability. Therefore the planner is able to find solutions for more challenging queries.

Computation times for all converged queries are shown in Fig. 8. We notice that for both scenarios, the bicycle model tends to have less computation time than the articulated model. On average, the articulated model has 2 times longer the computation time in the parking map, and 3 times longer in the transportation map.

The difference in computation time is mainly due to model complexity. The articulated model has more states than the bicycle model, and more constraints to describe the collision avoidance of the second body. Therefore its associated optimization problem has more variables than its bicycle model counterpart. By comparing the computation time of two models in different maps, the results show that the computation time of the articulated model increases faster than that of the bicycle model when the map gets complicated or when the horizon expands. In general nonlinear programming, the relationship between problem size and computation time is highly problem-dependent. However, in this case, computation time clearly increases with the number of variables. In addition, the variance of the calculation time also increases with the number of variables, indicating that computation time becomes more difficult to estimate for large problems.

B. Experiment Result

To evaluate the quality of the path, we select two different queries from the transportation scenario and track the obtained paths with the prototype hauler, while recording the lateral and heading errors. The experiment is repeated five times for each query.

In query 1, the generated paths are all unidirectional. As shown in Fig. 10, the two models' lateral error and heading error are both within the allowable range without much difference. In Fig. 9, we see that the vehicle tracks the path reasonably well with both models.

In query 2, the generated paths are instead bidirectional. From Fig. 12, we notice that the path computed using the bicycle model can not be properly followed, as the lateral error exceeds the allowable range of the controller after the hauler switches direction. The reason behind this unsatisfactory behavior is discrepancy between the bicycle model in the planner and the real articulated vehicle. The vehicle cannot accurately track the planned motion at the direction change, which is evident from Fig. 11. Furthermore, we note that when the vehicle tracks the path associated with the articulated model, both lateral and heading errors stay well within the allowable range during the direction change.

VII. Conclusions

We present a novel optimization based path planner for articulated vehicles that extends an existing state of the art framework [6] to accommodate a two-body articulated vehicle model. We evaluate the performance of the planner in terms of path length, quality and computation time, by comparing it to a benchmark planner, based on the single-body kinematic bicycle model. Results show that in general, paths generated by the more accurate two-body articulated model are shorter and of higher quality, i.e, the vehicle is able to track them more accurately. Also, the path planning algorithm with the articulated model has higher success rate for difficult missions.

The improvement in terms of maneuverability utilization for the two-body model is most significant at low speeds and at direction changes in close proximity to obstacles, i.e., for the more challenging motion planning queries. This in combination with the result that the mean computation time for the two-body model is 2.75 times longer over our set of queries suggests that this design may be most suitable as a fallback alternative for extra challenging queries, where the nominal path planner fails to find a solution. For nominal conditions, i.e., less challenging queries, the added precision is not required, and shorter compute times to enable faster re-planning rate to react to dynamic changes to the environment is to be prioritized. Hence, for nominal conditions, a simpler model, e.g., the bicycle model, or a spline based planning framework [12] can be more suitable. Adding such a fallback functionality to a nominal planning framework would enable full utilization of the vehicle's maneuverability when needed, avoiding costly production stops.

References

[1] Steven M LaValle. *Planning algorithms*. Cambridge university press, 2006.

[2] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[3] D. González, J. Pérez, V. Milanés, and F. Nashashibi. "A Review of Motion Planning Techniques for Automated Vehicles". *IEEE Transactions on Intelligent Transportation Systems* 17.4 (Apr. 2016), 1135–1145.

[4] P. I. Corke and P. Ridley. "Steering kinematics for a center-articulated mobile robot". *IEEE Transactions on Robotics and Automation* 17.2 (Apr. 2001), 215–218.

[5] Claudio Altafini. "A Path-Tracking Criterion for an LHD Articulated Vehicle". *The International Journal of Robotics Research* 18.5 (1999), 435–441.

[6] Xiaojing Zhang, Alexander Liniger, and Francesco Borrelli. "Optimization-based collision avoidance". *arXiv preprint arXiv:1711.03449* (2017).

[7] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[8] Alexander Liniger, Alexander Domahidi, and Manfred Morari. "Optimization-based autonomous racing of 1: 43 scale RC cars". *Optimal Control Applications and Methods* 36.5 (2015), 628–647.

[9] T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist. "Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles". *IEEE Transactions on Automation Science and Engineering* 7.1 (Jan. 2010), 167–172.

[10] Thaker Nayl, Mohammed Q Mohammed, and Saif Q Muhamed. "Obstacles avoidance for an articulated robot using modified smooth path planning". In: *2017 international conference on computer and applications (ICCA)*. IEEE. 2017, 185–189.

[11] Amr Mohamed, Jing Ren, Haoxiang Lang, and Moustafa El-Gindy. "Optimal collision free path planning for an autonomous articulated vehicle with two trailers". In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. Toronto, ON, Canada, 2017, 860–865.

[12] J. Choi and K. Huhtala. "Constrained Global Path Optimization for Articulated Steering Vehicles". *IEEE Transactions on Vehicular Technology* 65.4 (Apr. 2016), 1868–1879.

[13] Dmitri Dolgov, Sebastian Thrun, Michael Montemerlo, and James Diebel. "Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments". *The International Journal of Robotics Research* 29.5 (2010), 485–501.

[14] R. Oliveira, M. Cirillo, J. M. artensson, and B. Wahlberg. "Combining Lattice-Based Planning and Path Optimization in Autonomous Heavy Duty Vehicle Applications". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. Changshu, Suzhou, China, June 2018, 2090–2097.

[15] Kristoffer Bergman, Oskar Ljungqvist, and Daniel Axehill. "Improved Path Planning by Tightly Combining Lattice-based Path Planning and Numerical Optimal Control". *arXiv preprint arXiv:1903.07900* (2019).

[16] P. E. Hart, N. J. Nilsson, and B. Raphael. "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". *IEEE Transactions on Systems Science and Cybernetics* 4.2 (July 1968), 100–107.

[17] Mihail Pivtoraiko and Alonzo Kelly. "Efficient Constrained Path Planning via Search in State Lattices". In: *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*. Munich, Germany, 2005.

[18] Andreas Wächter and Lorenz T Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Mathematical programming* 106.1 (2006), 25–57.