Optimized Path Following in Truck Platoons Using Non-Linear Model Predictive Control with Geo-spatial Data

Akhil Bandamidapalli
Dept. of Mechanical Engineering
Indian Institute of Technology Madras
Chennai, 600036, India

K. B. Devika
Faculty of Environment, Science & Economy
University of Exeter
Exeter, EX4 4QF, United Kingdom

Shankar C. Subramanian
Dept. of Engineering Design
Indian Institute of Technology Madras
Chennai, 600036, India

***Abstract*—A truck platoon is a truck formation that moves at high speeds along a specific route with relatively small inter-vehicular distances. A fundamental problem in such formations is that the platoon must follow the prescribed route while avoiding collisions and keeping the platoon stable. We must also ensure that the energy consumed in the process is minimal. To solve this problem, a non-linear model predictive controller has been proposed while also considering the two-dimensional dynamics of the truck operation. To solve the problem of route following, we implement a technique that reads from the previously obtained Geo-spatial data and generates waypoints as markers for the platoon to follow. The tracking of these waypoints are formulated as a Non-Linear optimization problem while integrating velocity minimization among the vehicles. This problem is then solved using the CasADi optimal control framework. The controller's capability to establish a stable, working platoon for different routes has been studied.**

***Index Terms*—Geo-spatial data, CasADi, Model Predictive Control, Non-Linear Optimization, Truck Platoon**

I. Introduction

An autonomous truck platoon consists of several trucks that follow a prescribed route while avoiding collisions. One crucial fact to note is that, due to the frontal area of these vehicles, a significant amount of energy is lost by aerodynamic drag force. Hence, by having a closed formation, truck platoons allow for transportation and logistics of large-scale items over long distances while also having better energy and fuel efficiency [1], [2]. Hence, truck platoons are an active area of research, and constant developments are being made in the field to improve the efficiency of long-distance transportation [3].

—

— Geo-spatial route planning is a method of planning a route from GPS data, deciding parameters, and controlling actions to ensure the route is followed. It is used widely in many areas to plan paths while also enforcing certain constraints [4], [5], [6].

—

— Model Predictive Control (MPC) is a control algorithm that uses the system model to optimize control inputs to achieve a target while obeying certain constraints. Naturally, the method gained traction in the autonomous vehicles industry to execute paths efficiently [7], [8], [9]. An added challenge is constructing such a controller for a platoon, where inter-vehicular collisions must be avoided. A traditional approach in this direction can be seen in [10], but as suggested by the author, a noticeable advancement to this could be the addition of a genetic algorithm [11].

—

— Another approach in this direction has been worked out, where a non-linear MPC is used to control a platoon on a linear path while also ensuring the stability of the platoon [12]. However, the major drawback of this approach is ensuring its feasibility in real-world scenarios, which forces us to include lateral dynamics as well. In this regard, a working implementation of a Non-linear MPC for route following and platoon stability has been developed, which may work in any route after some processing. It is imperative to note that a simplified vehicle model has been considered while considering the lateral dynamics. This resulting non-linear control problem has no closed-form solution, making computation challenging. To overcome this, the optimization software CasADi has been used to numerically solve the Non-Linear Programming (NLP) problem [13], [14]. A schematic representation of the proposed controller is shown in Fig 1.

II. Truck Platoon Model

A. Dynamics model

A 3-Degree of Freedom vehicle model is used in this section as the controller's prediction model, as shown in Fig.2 and as prescribed in [15], [16]. This model is used to progress the system at every step. The model is simplified for calculation; we enforce the following assumptions:

- We ignore the influence of vertical motion, including the suspensions
- The vehicle motion is described using the bicycle model, hence neglecting left and right load transfers
- We ignore lateral aerodynamics as they are minute and do not affect the system significantly

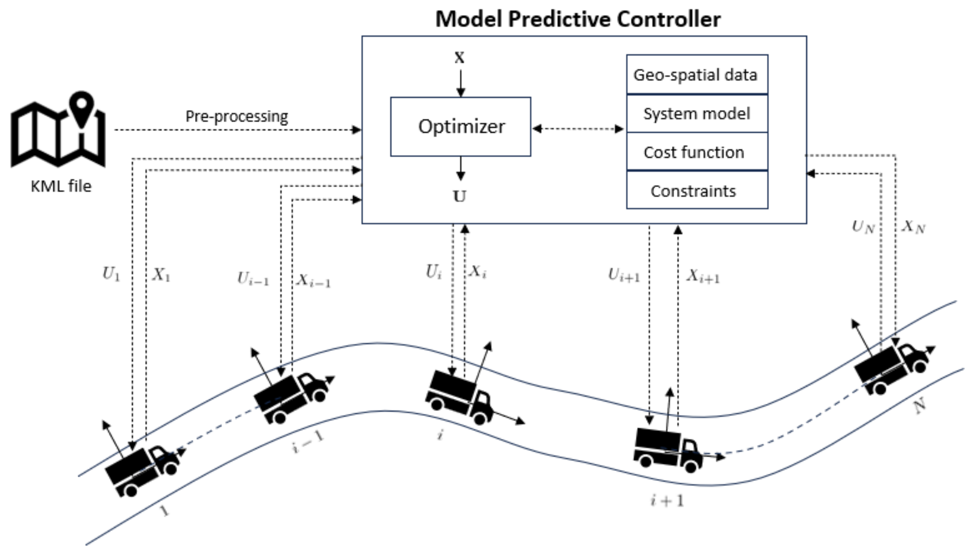

Fig. 1. Overview of the controller

The equations of motion for the i^{th} truck in the platoon are,

$$m_i\ddot{x}_i = m_i\dot{y}_i\dot{\psi}_i + 2F_{i,x,f}\cos(\delta_i) - 2F_{i,y,f}\sin(\delta_i) - \frac{1}{2}C_d\rho A_i{\dot{x}_i}^2 \tag{1}$$

$$m_i\ddot{y}_i = -m_i\dot{x}_i\dot{\psi}_i + 2F_{i,x,f}\sin(\delta_i) + 2F_{i,y,f}\cos(\delta_i) + 2F_{i,y,r} \tag{2}$$

$$J_i\ddot{\psi}_i = a_i\left(F_{i,y,f}\cos(\delta_i) + F_{i,x,f}\sin(\delta_i)\right) - b_iF_{i,y,r} \tag{3}$$

where,

- m_i is the mass of the i^{th} vehicle
- $\dot{x}_i$ is the velocity in the local x- direction of the i^{th} vehicle
- $\dot{y}_i$ is the velocity in the local y- direction of the i^{th} vehicle
- $\ddot{x}_i$ is the acceleration in the local x- direction of the i^{th} vehicle
- $\ddot{y}_i$ is the acceleration in the local y- direction of the i^{th} vehicle
- $\dot{\psi}_i$ is the yaw rate of the i^{th} vehicle
- $\ddot{\psi}_i$ is the yaw acceleration of the i^{th} vehicle
- $F_{i,x,f}$ is the longitudinal force in the front tyre of the i^{th} vehicle
- $F_{i,y,f}$ is the lateral force in the front tyre of the i^{th} vehicle
- $F_{i,x,r}$ is the longitudinal force in the rear tyre of the i^{th} vehicle
- $F_{i,y,r}$ is the lateral force in the rear tyre of the i^{th} vehicle
- δ_i is the steering angle of the i^{th} vehicle
- C_d is the drag coefficient
- ρ is the density of air
- A_i is the frontal area of the i^{th} vehicle
- a_i,b_i are the dimensions of the i^{th} vehicle
- J_i is the moment of inertia of the i^{th} vehicle

where $i = 1...N$,N is the number of vehicles in the platoon.

1) Aerodynamic Drag Force: The aerodynamic drag force of the i^{th} vehicle in the longitudinal direction is modelled as ,

$$F_{i,aero} = \frac{C_d\rho A_i{\dot{x}_i}^2}{2} \tag{4}$$

We assume a constant drag force acting along the longitudinal direction for each truck. The drag in the lateral direction is neglected as it is negligible.

2) Local to Global frame conversion: The velocities of the i^{th} vehicle in the global frame are given by,

$$\begin{bmatrix} \dot{X}_i \\ \dot{Y}_i \end{bmatrix} = \begin{bmatrix} \cos\psi_i & -\sin\psi_i \\ \sin\psi_i & \cos\psi_i \end{bmatrix} \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \tag{5}$$

3) Non-Linear State Space Model: The derived non-linear equations describing the dynamics of the vehicle are written in the form of a state space vector as below,

$$\dot{X}_i(t) = f_{pred}\left(X_i(t), U_i(t)\right) \tag{6}$$

where, $\dot{X}_i(t)$ represents the states of the system with time, $U_i(t)$ represents the inputs applied to the system at time t, where i goes from 1 to N, describing the vehicle number.

$$X_i(t) = \left[X_i(t), Y_i(t), \dot{x}_i(t), \dot{y}_i(t), \psi_i(t), \dot{\psi}_i(t)\right]^T$$

$$U_i(t) = [F_{i,x,f}(t), F_{i,y,f}(t), F_{i,x,r}(t), F_{i,y,r}(t), \delta_i(t)]^T \tag{7}$$

B. Platoon Model

The above state space model is now used to describe the entire platoon. We also use a non-linear approximation to progress the system with time.

$$\dot{\mathbf{X}}(t) = \mathbf{f_{pred}}\left(\mathbf{X}(t), \mathbf{U}(t)\right) \tag{8}$$

where $\mathbf{f_{pred}}$ is a non-linear function, invariant with time, $\mathbf{X}(t)$ is the state vector for the platoon and $\mathbf{U}(t)$ is the control vector for the platoon.

$$\mathbf{X}(t) = \left[X_1(t), Y_1(t), \dot{x}_1(t), \dot{y}_1(t), \psi_1(t), \dot{\psi}_1(t), X_2(t)...\dot{\psi}_N(t)\right]^T$$
$$\mathbf{U}(t) = [F_{1,x,f}(t), F_{1,y,f}(t), F_{1,x,r}(t), F_{1,y,r}(t), \delta_1(t)...\delta_N(t)]^T \tag{9}$$

To progress the system, we discretize the system using a first-order Euler approximation, hence.

$$\mathbf{X}(t + \Delta T) = \mathbf{X}(t) + \dot{\mathbf{X}}(t)\Delta T \tag{10}$$

where, ΔT is the sampling time.

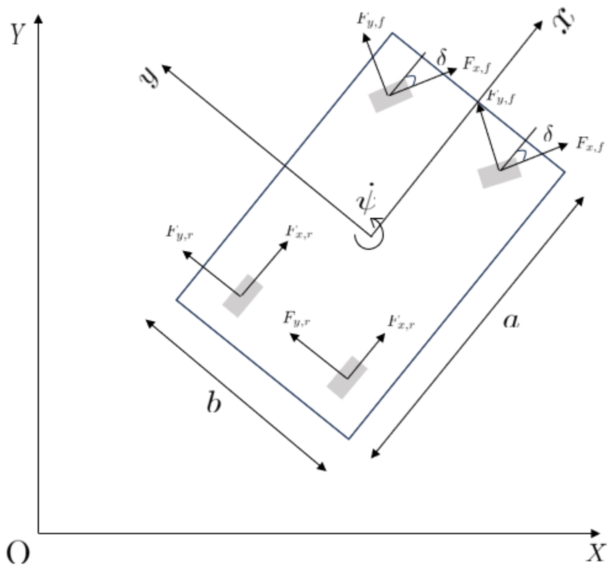

Fig. 2. Schematic diagram of the vehicle model

III. Geo-spatial Route Model

Our approach necessitates us to be able to find out the waypoints from a Keyhole Markup Language (KML) file. This data can be obtained from web mapping platforms like Google Maps®. We use the following procedure.

- Once we input a KML file, we extract latitude and longitude values from it.
- We then project these coordinates onto the cartesian plane. We also account for the curvature of the earth by using the following formula,

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 1,11,321m \\ 1,11,000m \end{bmatrix}^T \begin{bmatrix} \Delta Longitude \\ \Delta Latitude \end{bmatrix} \tag{11}$$

  The above transformation projects the distance between two points on a 3D sphere to the distance between them on the cartesian plane as prescribed in [17].
- With the obtained points on the cartesian plane, we linearly interpolate between them while ensuring that no two consecutive coordinates exceed 50m. This is to ensure that the route generated is smooth and also resembles real-world scenarios.

This post-processed data is stored in the form of $X_{i,ref}$,$Y_{i,ref}$ and $\psi_{i,ref}$ for $i = 1....N$.

IV. Control Problem

The main aim of the proposed controller is to make sure that the vehicles in the platoon follow the given route and also maintain stability in the long run. We consider a platoon to be stable when the velocities between two consecutive vehicles are minimal, given that the route is non-linear. To achieve these objectives, we express them as mathematical expressions that result in an optimal control problem using an MPC framework.

A. *Model Predictive Control Framework*

In our work, we have implemented a receding control framework that follows an MPC (Model Predictive Control) framework. At every discrete moment in time, a set of inputs is computed over a control horizon N_c, while the states are calculated over a prediction horizon N_p. During this process, the cost function is minimized. The MPC framework and the optimal control problem are presented below,

1) Cost function: The cost function has five components, represented by five weights, w_t, w_a, w_v, w_u, and w_s.

$$J_d[k] = \omega_t \sum_{i=1}^{N} [(X_i[k] - X_{i,ref}[k])^2 + (Y_i[k] - Y_{i,ref}[k])^2] \tag{12}$$

$$J_a[k] = \omega_a \sum_{i=1}^{N} (\psi_i[k] - \psi_{i,ref}[k])^2 \tag{13}$$

$$J_v[k] = \omega_v \sum_{i=2}^{N} [(\dot{x}_{i-1}[k] - \dot{x}_i[k])^2 + (\dot{y}_{i-1}[k] - \dot{y}_i[k])^2] \tag{14}$$

$$J_{u,f}[k] = \omega_u \sum_{i=1}^{N} [F_{i,x,f}[k]^2 + F_{i,x,r}[k]^2 + F_{i,y,f}[k]^2 + F_{i,y,r}[k]^2] \tag{15}$$

$$J_{u,s}[k] = \omega_s \sum_{i=1}^{N} \delta_i[k]^2 \tag{16}$$

We combine the two input costs, (15) and (16), as one to get,

$$J_u[k] = J_{u,f}[k] + J_{u,s}[k] \tag{17}$$

Hence, the combined cost function is,

$$J = \sum_{k=1}^{N_p} [J_d[k] + J_a[k] + J_v[k] + J_u[k]] \tag{18}$$

Here, J_d is the cost associated with route deviations. The values, $X_{i,ref}$,$Y_{i,ref}$ are obtained from the processed Geo-spatial data and constantly update as time progresses, depending on the individual vehicle's location. With this data, we can track if a vehicle deviates too much from its actual path and appropriately penalize it. J_a also performs a similar task but penalizes the steering angle of the vehicle with the slope of the route at that instant. The slope of the road, $\psi_{i,ref}$ is also obtained from the Geo-spatial data. As the platoon tracks the given route profile, there may be certain sections of the route where the vehicles may abruptly accelerate/decelerate. To avoid this, we introduce the J_v term, which minimizes the velocity difference between consecutive vehicles. Hence, in the long run, this term gets minimized, ensuring that all the vehicles in the platoon are of similar velocities, which ensures platoon stability. Finally, we have the terms $J_{u,f}$ and $J_{u,s}$, which minimize the tractive forces exerted by the wheels and the amount of steer needed, respectively. This also ensures that minimum energy is expended by the engines and minimum steering is needed.

2) System Model: To progress the model forward, we use the system dynamics defined by the equations (8) and (10). Hence, we obtain a non-linear model as,

$$\mathbf{X}(t+\Delta T) = \mathbf{X}(t) + \mathbf{f_{pred}}(\mathbf{X}(t), \mathbf{U}(t)\Delta T \tag{19}$$

3) Constraints: Naturally, the prescribed model would need constraints to ensure that the controller computes solutions within the feasible limits of the control inputs. Here, we assume that the trucks in the platoon are rear-wheel driven. We also have to ensure that the velocities of the truck are within a prescribed range.

$$F_{\min} \le F_{if}[k] \le 0 \tag{20}$$

$$F_{\min} \le F_{ir}[k] \le F_{\max} \tag{21}$$

$$\delta_{\min} \le \delta_i[k] \le \delta_{\max} \tag{22}$$

$$\dot{x}_{\min} \le \dot{x}_i[k] \le \dot{x}_{\max} \tag{23}$$

$$\dot{y}_{\min} \le \dot{y}_i[k] \le \dot{y}_{\max} \tag{24}$$

for all $i = 1....N$ and $k = 1....N_p$.

We also have additional constraints to ensure that inter-vehicular distance between the trucks does not go lower than a certain value to prevent collisions.

$$X_i[k] - X_{i-1}[k] \ge \epsilon_{min} \tag{25}$$

$$Y_i[k] - Y_{i-1}[k] \ge \delta_{min} \tag{26}$$

for all $i = 2....N$ and $k = 1....N_p$.

Also, for all $i = 1....N$ and $k = N_c....N_p - 1$, we ensure that,

$$\mathbf{U}[k] = \mathbf{U}[N_c - 1] \tag{27}$$

B. Non-Linear Optimization Problem

The MPC can now be formulated as an optimization problem by expressing the functions in a matrix form. The resultant cost function is a quadratic function of the control inputs and can be brought into a quadratic form by using augmented matrices,

$$\min J(\mathbf{X}, \mathbf{U}) = \sum_{k=1}^{N_p} \tilde{\mathbf{X}}[k]^\top \mathbf{Q}_k \tilde{\mathbf{X}}[k] + \mathbf{U}[k]^\top \mathbf{R}_k \mathbf{U}[k] \tag{28}$$

where,

$$\tilde{\mathbf{X}}[k] = \left[\tilde{X}_1[k], \tilde{X}_2[k], \dots, \tilde{X}_N[k]\right]$$

$$\tilde{\mathbf{X}}_i[k] = [X_i[k], X_{i,ref}[k], Y_{i,ref}[k], \psi_{i,ref}[k]]$$

$$\mathbf{U}[k] = [U_1[k], U_2[k], \dots, U_N[k]] \tag{29}$$

$\mathbf{Q}_k$ and $\mathbf{R}_k$ are the cost matrices. These matrices control the weights of the cost function. Hence, we can simplify the expression as a non-linear optimization problem,

$$\min J(\mathbf{X}, \mathbf{U})$$

$$\mathbf{X}[k+1] = \mathbf{X}[k] + f_p(\mathbf{X}[k], \mathbf{U}[k])\Delta T, \forall k = \{0 \dots N_p - 1\}$$

subject to,

$$\mathbf{F}_{\min} \le \mathbf{U}[k] \le \mathbf{F}_{\max}, \forall k = \{0 \dots N_p - 1\}$$

$$\mathbf{M}\mathbf{X}[k] \ge \mathbf{C}, \forall k = \{1 \dots N_p\} \tag{30}$$

where. the constraints mentioned in (20-27) are written in the form of matrices, $\mathbf{F}_{min}$,$\mathbf{F}_{max}$,$\mathbf{M}$ and $\mathbf{C}$.

C. Numerical Solution

The given study has now been posed as a numerical optimization problem. To solve the NLP, we use the open-source framework CasADi, which enables non-linear optimization. CasADi enables us to formulate the problem efficiently and choose a suitable solver [18]. Our study uses the Interior Point OPTimizer (IPOPT) solver. The IPOPT solver uses interior point methods to find the optimum values [19].

V. Simulation

A system simulation is carried out, with the results in the later section. For testing, we use the Bangalore to Chennai route and the Hyderabad to Chennai route. We first import the corresponding KML file from Google Maps® [20], [21]. This data is then processed and converted to a route on the cartesian plane by using (11). Once this data is generated, we run the controller. The controller sets up the MPC formulation at every step, solves for the optimum values, and then takes the next step. We also warm start the solver at every step by taking a single step shift from the previous optimal value. This process is run till the end of the route data, which results in the end of the simulation.

VI. Results and discussions

The proposed approach has been tested on with a platoon of 4 vehicles. The parameters of the non-linear MPC and the physical characteristics of the vehicles have been mentioned in Table 1. We consider all the vehicles in the platoon to be similar, although one may alter them as well. We test our controller on two different routes, the Bangalore to Chennai route and the Hyderabad to Chennai route.

We study the controller's ability to primarily follow the prescribed routes while avoiding collisions and keeping the velocities between the vehicles at a minimum. The values of the weights were individually tuned based on the requirements. Compared to w_u, w_s, higher values for w_t,w_a, and w_v are chosen to minimize any deviations from the route and to penalize any high difference in velocities between the trucks. Since the geo-spatial route data is pre-compiled, the entire controller's time efficiency is solely dependent on the optimization capabilities of the CasADi solver.

Parameters	Values
N	4
N_p, N_c	5,3
ΔT	1 sec
w_t, w_a, w_v	10,1,5
w_u, w_s	$10^{-10}, 10^{-30}$
ϵ, δ	50m,50m
$\dot{x}_{\max}, \dot{y}_{\max}$	50m/s,50m/s
$\dot{x}_{\min}, \dot{y}_{\min}$	-50m/s,-50m/s

TABLE I
PARAMETERS OF THE SIMULATION

A. Bangalore to Chennai Route

1) Path Following: In Fig.4(i), the path following is almost perfect, with only slight deviations during turns. The presence of steep turns on the route is critical to test it on actual road conditions. However, the controller faces a challenge while turning as it has to account for changes in the value of $\psi_{i,ref}$ (where i ranges from 1 to 4) provided by the processed KML file. Due to the rapid changes in the value of $\psi_{i,ref}$ during turns, the controller must optimize them at every iteration, which creates an additional load on the controller. This may lead to slight deviations in the executed path compared to the reference path.

It is worth noting that the controller's accuracy is particularly promising in linear regions. This behavior can be attributed to the fact that the controller needs to optimize fewer parameters, as the $\psi_{i,ref}$ values do not change significantly.

2) Velocity minimization: In Fig.4(ii) and Fig.4(iii), it can be observed that the controller ensures that the local velocity differences between the trucks are minimized during execution. This is important to avoid collisions and maintain a stable platoon in the long run. For example, in segment Y of the route, which is between $0m$ and $200m$, the controller sets $\dot{y}_i$ to 0 and ψ_i close to $\pi/2$, as the route has negligible deviation in the X direction. Therefore, the controller tries its best to optimize the route direction. However, since a higher weight is assigned to path following, as shown in Table I, following the path accurately takes precedence.

B. Hyderabad to Chennai Route

1) Path following: Based on the results shown in Fig.6(i), the path-following is accurate, with only minor deviations in the initial stages. These deviations can be attributed to the vehicles being in their starting state and requiring some time to reach the desired states. However, it is essential to note that the vehicles tend to deviate slightly from the reference path during the road turns. The reasoning is similar to the behavior seen in the previous route.

2) Velocity Minimization: Similar to the previous route, the controller attempts to maintain the local velocity components to near zero based on the direction of the route. For instance, in the region of X between $600m$ and $800m$, the route has minimal deviation in the Y direction, as depicted in Fig.6(i). As a result, the controller sets $\dot{y}_i$ to 0. [1]

[1]The simulations were run on a 1.6 GHz Intel Core i5 Processor, with 8 GB of installed RAM

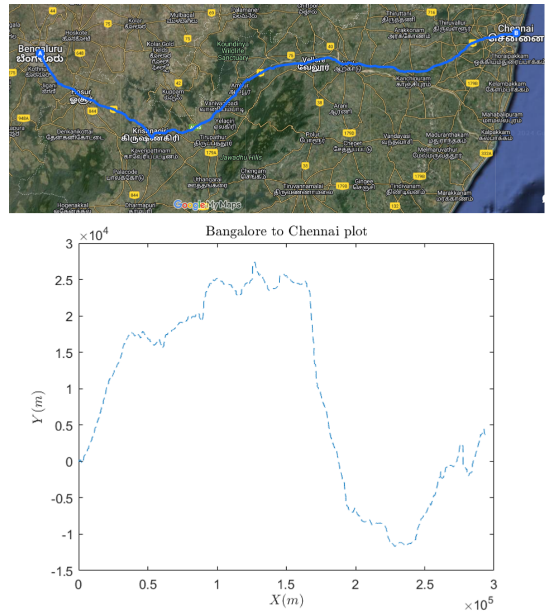

Fig. 3. Conversion of the Bangalore to Chennai KML file from Google Maps® to usable data represented as a plot

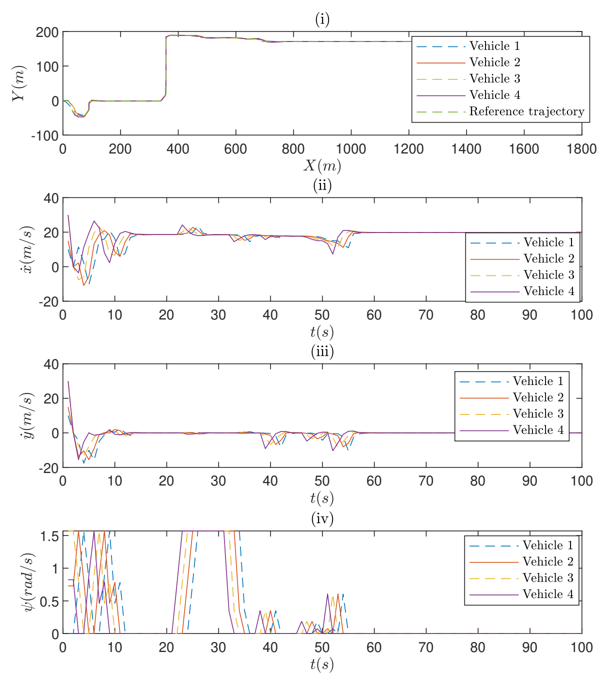

Fig. 4. (i) Plot of the route trajectory (ii) Local velocity in the x - direction (iii) Local velocity in the y - direction (iv) Yaw angle for the individual vehicles for the Bangalore to Chennai route

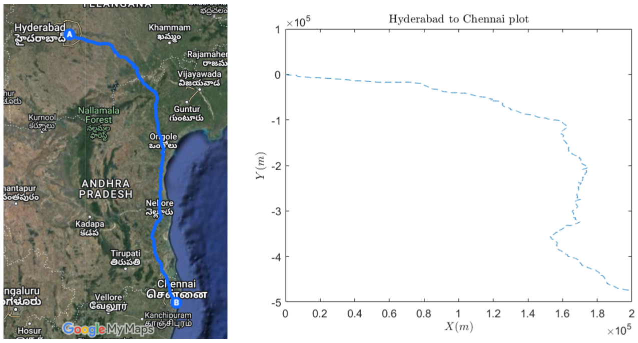

Fig. 5. Conversion of the Hyderabad to Chennai KML file from Google Maps® to usable data represented as a plot

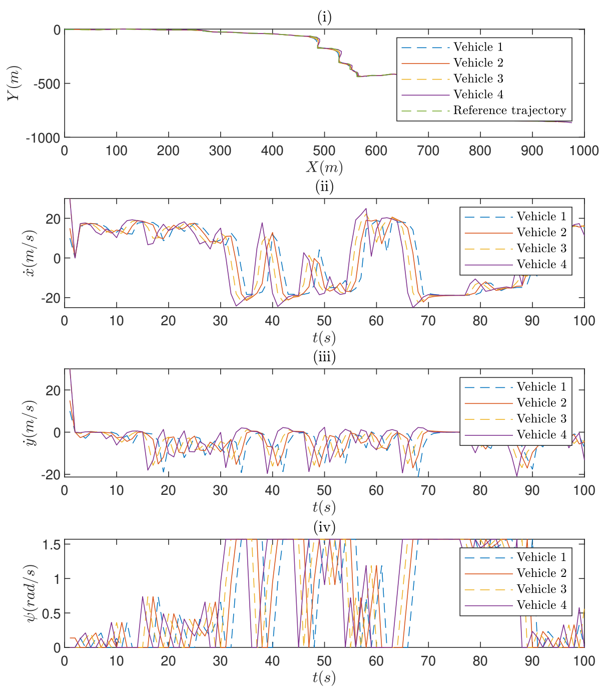

Fig. 6. (i) Plot of the route trajectory (ii) Local velocity in the x - direction (iii) Local velocity in the y - direction (iv) Yaw angle for the individual vehicles for the Hyderabad to Chennai route

VII. Conclusions

Hence, using geospatial data from traditional sources for path-following and collision avoidance while ensuring string stability using a model predictive controller-based approach has been shown here. We also illustrate the use of the optimization framework CasADi. A three-degree of freedom model further strengthens the capabilities of the above controller to be used in various route scenarios. Further research may extend the idea to a real-time environment with dynamically changing routes.

References

[1] S. Tsugawa, S. Jeschke, and S. E. Shladover, "A review of truck platooning projects for energy savings," IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 68–77, 2016.

[2] G. Rohith, K. B. Devika, P. P. Menon, and S. C. Subramanian, "An adaptive time-headway policy for lower energy consumption in autonomous vehicle platoons," in 2022 European Control Conference (ECC). IEEE, 2022, pp. 1734–1739.

[3] Muratori, M., Holden, J., Lammert, M., Duran, A. et al., "Potentials for Platooning in U.S. Highway Freight Transport," SAE Int. J. Commer. Veh. 10(1):45-49, 2017, https://doi.org/10.4271/2017-01-0086.

[4] Keenan, Peter B. "Modelling Vehicle Routing in GIS." Operational Research (2008): n. pag. Print.

[5] Jeon, Soo Jung et al. "GPS waypoint fitting and tracking using model predictive control." 2015 IEEE Intelligent Vehicles Symposium (IV) (2015): 298-303.

[6] A. Al Arabi, H. Ul Sakib, P. Sarkar, T. P. Proma, J. Anowar and M. A. Amin, "Autonomous Rover Navigation Using GPS Based Path Planning," 2017 Asia Modelling Symposium (AMS), Kota Kinabalu, Malaysia, 2017, pp. 89-94, doi: 10.1109/AMS.2017.22.

[7] Chen, Shu-ping, et al. "MPC-based path tracking with PID speed control for high-speed autonomous vehicles considering time-optimal travel." Journal of Central South University 27.12 (2020): 3702-3720.

[8] Kim, Minsung, et al. "Model predictive control method for autonomous vehicles using time-varying and non-uniformly spaced horizon." IEEE Access 9 (2021): 86475-86487.

[9] Vu, Trieu Minh, et al. "Model Predictive Control for Autonomous Driving Vehicles." Electronics, vol. 10, no. 21, 24 Oct. 2021, p. 2593, https://doi.org/10.3390/electronics10212593.

[10] S. Graffione, C. Bersani, R. Sacile and E. Zero, "Model predictive control of a vehicle platoon," 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2020, pp. 513-518, doi: 10.1109/SoSE50414.2020.9130541.

[11] Meng Wang, Juexuan Chen, Hanwen Yang, Xin Wu, Longjiao Ye, "Path Tracking Method Based on Model Predictive Control and Genetic Algorithm for Autonomous Vehicle", Mathematical Problems in Engineering, vol. 2022, Article ID 4661401, 16 pages, 2022. https://doi.org/10.1155/2022/4661401

[12] S. Tadeparti, K. B. Devika and S. C. Subramanian, "Computationally Efficient Non-linear Model Predictive Control for Truck Platoons," 2023 European Control Conference (ECC), Bucharest, Romania, 2023, pp. 1-6, doi: 10.23919/ECC57647.2023.10178412.

[13] Andersson, Joel A. E. et al. "Dynamic optimization with CasADi." 2012 IEEE 51st IEEE Conference on Decision and Control (CDC) (2012): 681-686.

[14] Elhesasy, M.; Dief, T.N.; Atallah, M.; Okasha, M.; Kamra, M.M.; Yoshida, S.; Rushdi, M.A. Non-Linear Model Predictive Control Using CasADi Package for Trajectory Tracking of Quadrotor. Energies 2023, 16, 2143. https://doi.org/10.3390/en16052143

[15] Liu, Yang, et al. "Lateral control system for vehicle platoon considering vehicle dynamic characteristics". IET Intelligent Transport Systems, vol. 13, no. 9, 2019, p. 1356-1364. https://doi.org/10.1049/iet-its.2018.5504.

[16] Guo, Jinghua et al. "Adaptive non-linear coordinated optimal dynamic platoon control of connected autonomous distributed electric vehicles on curved roads." IET Intelligent Transport Systems (2020): n. pag.

[17] Ashkenazi, V., 1986. Coordinate systems: how to get your position very precise and completely wrong. The Journal of Navigation 39 (2), 269–278

[18] Andersson, Joel A. E., et al. "CasADi: A Software Framework for Nonlinear Optimization and Optimal Control." Mathematical Programming Computation, vol. 11, no. 1, 11 July 2018, pp. 1–36, https://doi.org/10.1007/s12532-018-0139-4.

[19] A. Forsgren, P. Gill, and M. Wright, "Interior methods for nonlinear optimization," Society for Industrial and Applied Mathematics, vol. 44, pp. 525–597, 12 2002.

[20] Google. (n.d.). [Google Maps directions to drive from Bangalore, Karnataka, to Chennai, Tamil Nadu]. Retrieved February 13, 2024, from https://maps.app.goo.gl/KDbsAuXZajn4UkBHA.

[21] Google. (n.d.). [Google Maps directions to drive from Hyderabad, Telangana to Chennai, Tamil Nadu]. Retrieved February 13, 2024, from https://maps.app.goo.gl/wat1Yk6s1RmqsfkW7.