Neural Mixed Platoon Controller Design

Ailing Xie
1. School of Transportation Science and Engineering, Beihang
2. Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems&Safety Control
Beijing 100191, China
xieailing@buaa.edu.cn

Jianshan Zhou*
1. School of Transportation Science and Engineering, Beihang
2. Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems&Safety Control
Beijing 100191, China
jianshanzhou@foxmail.com

Daxin Tian
1. School of Transportation Science and Engineering, Beihang
2. Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems&Safety Control
Beijing 100191, China
dtian@buaa.edu.cn

Xuting Duan
1. School of Transportation Science and Engineering, Beihang
2. Beijing Key Laboratory for Cooperative Vehicle Infrastructure Systems&Safety Control
Beijing 100191, China
duanxuting@buaa.edu.cn

Zhengguo Sheng
Department of Engineering and Design
University of Sussex
Richmond, UK
z.sheng@sussex.ac.uk

Dezong Zhao
James Watt School of Engineering
University of Glasgow
Glasgow, UK
Dezong.Zhao@glasgow.ac.uk

***Abstract*—Vehicle platooning can be formulated as an optimal control problem and many solving paradigms, such as Pontryagin's maximum principle-based and dynamical programming methods, have been recently developed. However, these methods usually rely on solving a group of necessary conditions or Hamilton-Jacobi-Bellman (HJB) partial differential equations, which is hard to calculate. Besides, due to the heterogeneous dynamics of different vehicles in a mixed and complex platoon which comprises of not only connected autonomous vehicles (CAVs), but also human-driven vehicles (HDVs), it is also challenging to coordinate the behaviors of different vehicles in an unified control framework. Here we provide a Neural Mixed Platoon Control (NMPC) framework, a novel control design for mixed vehicle platooning based on a neural ordinary differential equation (NODE). We first formulate an optimal control model that incorporates the heterogeneous dynamics of a leading CAV and several following HDVs. We use a neural network to parameterize a state-feedback controller and join the neural controller and the mixed platooning dynamics into the NODE solver to create a closed-loop and learnable controlled system. The resulting system can learn optimal control inputs driving the mixed platoon to evolve from a given beginning condition to the target state within a finite duration in an unsupervised manner. Finally, simulation results validate our suggested method's usefulness in terms of space headway and velocity tracking.**

***Index Terms*—connected vehicles, platoon control, neural network, deep learning**

∗Corresponding Author
This research was supported in part by the National Postdoctoral Program for Innovative Talents under Grant No. BX2021027, the China Postdoctoral Science Foundation under Grant No. 2020M680299, the Opening Project of the Ministry of Transport Key Laboratory of Technology on Intelligent Transportation Systems under Grant No. F20211746, the National Natural Science Foundation of China under Grant No. 52202391 and No. U20A20155, and the Beijing Municipal Natural Science Foundation under Grant No. L191001.

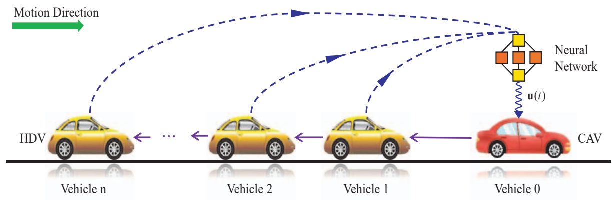

Fig. 1. Mixed platooning scenario.

I. Introduction

Connected and autonomous vehicles (CAVs) can acquire information from other vehicles through vehicle-to-vehicle (V2V) communications to help formation control, which can improve safety, alleviate traffic disturbance propagation, and increase road capacity. CAVs will co-drive with the human-driven vehicles (HDVs) on roadways throughout the conversion to a CAV-only environment, producing a mixed traffic environment [1]. Linear quadratic regulators, model predictive control (MPC), optimal control, robust control, adaptive dynamic programming (ADP), etc., have been designed for controlling a CAV to moderate the mixed traffic flow [2]–[4]. However, all the above control designs are based on a large number of parameters as preconditions. We know that the control problems of complex systems often rely on their dynamic models as well as optimization theory [5]. A dynamic system is controllable, indicating that it can reach the target state z^* in period t_e. Typically the control strategy depends on minimizing the cost function which usually includes the strength, length, and frequency of control signals [6]. This article selects "control energy" as the cost function and the solution of optimum control issues is usually based on Pontryagin's maximum principle, that is, solving the BVP in a Hamiltonian architecture, and performing state changes and system evolution, which is called POC.

In recent years, more and more attention is paid to artificial neural networks (ANNs) that use physical model formulations based on Lagrangian and Hamiltonian functions as a prior for a variety of tasks [7]–[9]. As for the Euler-Lagrangian differential equation of motion, the difference between kinetic energy and potential energy can distinguish between state variables, derivatives of state variables, and Lagrangian functions [7]. Neural networks may simulate the energy variables independently [10] or simultaneously [11]. Reference [12] builds a neural network using the Hamiltonian, which is based on the principle of minimizing the difference between the gradients and the time derivatives to update the controller parameters. Reference [13] uses a differentiable leapfrog integrator and to make the control process independent of the time derivatives, this method uses an ODE solver to propagate the measured state discrepancy backward. Reference [14] provides a neural network for solving ordinary-differential-equations to learn dynamics, profiling the benefits of not relying on model parameters while ensuring stability.

For the physics problem of vehicle platooning, we need a model that obtains physically plausible representations. The integration of deep learning and physics is theoretically possible, and the complex architecture of deep networks necessarily makes it possible to compute derivatives efficiently with extreme precision, which creates the conditions for the encoding and solution of physical differential processes. Therefore, for the high complexity and dimensional dynamical system of mixed platooning, we propose a Neural Mixed Platoon Controller (NMPC), an artificial neural network method that makes an extension from the particular to the general [15] and does not rely on the evaluation of energy cost function. The primary contributions of our study are as follows:

- We present a network topology called NMPC encoding the equation originating from Hamiltonian Mechanics. This topology will keep physical plausibility and train optimization techniques.
- We evaluate the proposed approach by controlling a mixed vehicle platoon and demonstrate NMPC's control ability better than POC.

The remainder of the paper is written in the following manner. We first present the dynamical model of the mixed vehicle platooning scenario in Section II. Then we demonstrate the principle of the POC and NMPC in Section III. We analyze the ability of NMPC to learn control signals of CAV in terms of deviation from the target state in Section IV. Finally, Section V brings us to a conclusion.

II. Dynamic Model

Connected and autonomous vehicles (CAVs) are a type of vehicle equipped with advanced onboard sensors, controllers, and actuators, which can integrate modern communication and network technologies to realize the information exchange

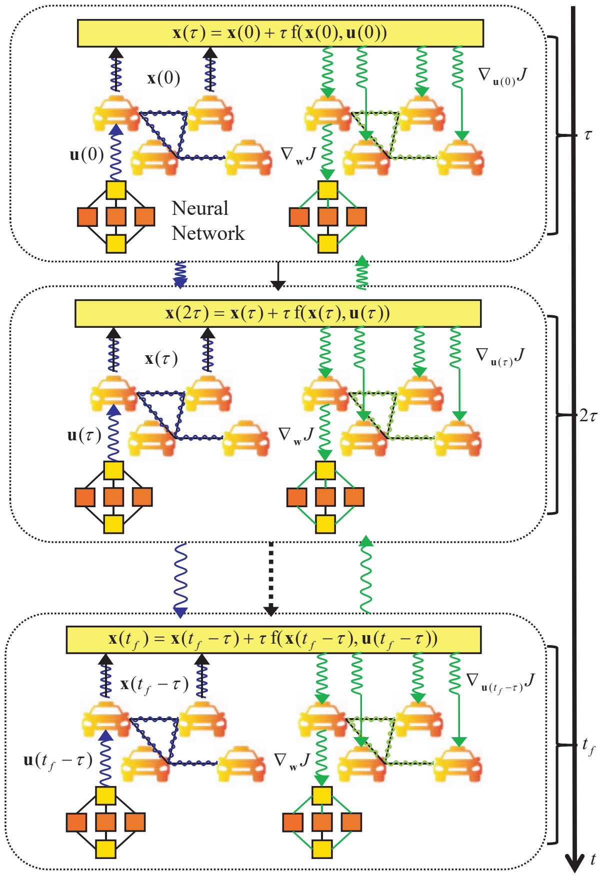

Fig. 2. Neural-ODE based controller.

and sharing between vehicles and X (human, vehicle, road, cloud, and other systems). Based on linearized car-following dynamics, we focus on a scenario: the CAV is driving freely as a leading agent, which is used as a mobile actuator for traffic control, and can obtain motion information from its different numbers of following vehicles under human control (HDVs) as well as generate a learning control signal to attenuate disturbances throughout the mixed traffic flow (see Fig. 1).

The model is based on the longitudinal dynamics of the head car in a mixed platoon. Consider a single-line setup, where the CAV at the top position of the queue is numbered as vehicle 0. Then we use $\mathcal{F} = \{1, 2, \ldots, a\}$ to denote the consecutive vehicles following behind. The position of vehicle j $(j \in \{0\} \cup \mathcal{F})$ is denoted as x_j, velocity is defined as v_j, and acceleration is $\dot{d}_j$. The distance between vehicle j from its adjacent vehicle $j-1$, is set to $d_j(t) = x_{j-1}(t) - x_j(t)$. The vehicle length is ignored.

The linear continuous-time vehicle-following dynamics of HDVs can be described by the following formula $(j \in \mathcal{F})$

$$\frac{\mathrm{d}v_j(t)}{\mathrm{d}t} = G(d_j(t), \frac{\mathrm{d}d_j(t)}{\mathrm{d}t}, v_j(t)), \tag{1}$$

where $\frac{\mathrm{d}d_j(t)}{\mathrm{d}t} = v_{j-1}(t) - v_j(t)$. For the function $G(\cdot)$, the acceleration of vehicle j is determined by the relative distance, relative velocity to its adjacent preceding vehicle, and its own velocity. The explicit expression of (1) can be presented as

$$\frac{\mathrm{d}v_j(t)}{\mathrm{d}t} = S_1(d_j(t) - Tv_j(t)) + S_2\frac{\mathrm{d}d_j(t)}{\mathrm{d}t}, \tag{2}$$

where S_2 is affected by the speed difference between its own car and the car ahead, and S_1 is the reactivity to the discrepancy between the expected and the actual vehicle spacing. The target distance of each HDV (to its adjacent preceding vehicle) is proportional to its speed, with the correlation coefficient persists at T (the time headway). We presume that all the HDVs have identical tracking intervals at different time periods. Define the state of the error between the actual and balanced condition of car j as $\tilde{d}_j(t) = d_j(t) - Tv_j(t)$, and then we can transform (2) into

$$\frac{\mathrm{d}v_j(t)}{\mathrm{d}t} = S_1\tilde{d}_j(t) + S_2(v_{j-1}(t) - v_j(t)). \tag{3}$$

As for the leading agent, the input u_t is obtained from the specified control method and is utilized as its acceleration. The longitudinal dynamics of vehicle 0 can be represented as

$$\begin{cases} \frac{\mathrm{d}x_0(t)}{\mathrm{d}t} = v_0(t), \\ \frac{\mathrm{d}v_0(t)}{\mathrm{d}t} = u(t), \end{cases}$$

which is a second-order model.

The state of the whole system is defined as

$$\begin{aligned} \boldsymbol{z}(t) =& [x_0(t), v_0(t), \tilde{d}_1(t), v_0(t) - v_1(t), \\ & \ldots, \tilde{d}_a(t), v_{a-1}(t) - v_a(t)]^\mathsf{T}, \end{aligned} \tag{4}$$

therefore, the state-space formulation for the whole system can be summarized as

$$\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = \boldsymbol{M}\boldsymbol{z}(t) + \boldsymbol{N}u(t). \tag{5}$$

The coefficient matrix $\boldsymbol{M} \in \mathbb{R}^{(2a+2)\times(2a+2)}$, $\boldsymbol{N} \in \mathbb{R}^{(2a+2)\times 1}$, which are respectively defined as

$$\boldsymbol{M} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & s_1 & s_2 & 0 & 0 & \ldots \\ 0 & 0 & -S_1 & -S_2 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & s_1 & s_2 & \ldots \\ 0 & 0 & S_1 & S_2 & -S_1 & -S_2 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \tag{6}$$

$$\boldsymbol{N} = [0, 1, 0, 1, 0, 0, \ldots, 0]^\mathsf{T}, \tag{7}$$

where $s_1 = -S_1T$, $s_2 = 1 - S_2T$.

III. Controlling Dynamical Systems with NMPC

As for the control problem mentioned above, the function in (1) considers both the interplay between points $1, 2, \ldots, 2a+2$ and the influence of the control input u_t upon the system. At first, points are in condition $\boldsymbol{z}_0$ and transited towards a desired state $\boldsymbol{z}^*$ at time t_e through suitable control input that minimizes the cost function

$$C = \int_0^{t_e} D(\boldsymbol{z}(t), \boldsymbol{u}(t))dt + \phi(\boldsymbol{z}(t_e)). \tag{8}$$

TABLE I
Following Vehicles

Vehicle Number	Initial States	
	Position	***Velocity***
Vehicle 1	475 m	15 m/s
Vehicle 2	430 m	20 m/s
Vehicle 3	400 m	15 m/s
Vehicle 4	350 m	20 m/s
Vehicle 5	330 m	10 m/s

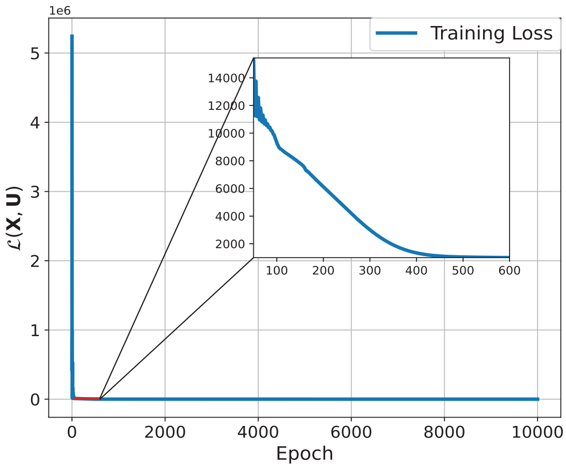

Fig. 3. Training loss after 10000 epochs.

The first term in (8) represents the cost from time 0 to t_e, and the control energy

$$D = \|\boldsymbol{u}(t)\|_2^2, \tag{9}$$

an explicit form, can be adopted. To minimize the control energy $\int_0^{t_e} \|\boldsymbol{u}(t)\|_2^2 dt$, there exist analytical POC inputs

$$\boldsymbol{u}^*(t) = \boldsymbol{N}^\mathsf{T} e^{\boldsymbol{M}^\mathsf{T}(t_e - t)}\boldsymbol{Q}(t_e)^{-1}\boldsymbol{p}(t_e). \tag{10}$$

Equation (10) is obtained by solving the Hamiltonian $H = \|\boldsymbol{u}(t)\|_2^2 + \boldsymbol{\lambda}(t)^\mathsf{T}[\boldsymbol{M}\boldsymbol{z}(t) + \boldsymbol{N}\boldsymbol{u}(t)]$ through the use of Pontryagin's maximum principle. The vector $\boldsymbol{p}(t_e) = \boldsymbol{z}(t_e) - e^{\boldsymbol{M}t_e}\boldsymbol{z}_0$ represents the discrepancy between the objective state $\boldsymbol{z}(t_e)$ and the preliminary state $\boldsymbol{z}_0$. The matrix

$$\boldsymbol{Q}(t_e) = \int_0^{t_e} e^{\boldsymbol{M}t}\boldsymbol{N}\boldsymbol{N}^\mathsf{T}e^{\boldsymbol{M}^\mathsf{T}t}dt \tag{11}$$

is the controllability Gramian.

NMPC provides an idea of using neural ODEs for optimal control problems, which offers an alternative method to achieve the aim state $\boldsymbol{z}^*$ in period t_e. We demonstrate a schematic of NMPC and it is combined with the dynamical system of mixed vehicle platooning (see Fig. 2). First, we input the control $\boldsymbol{u}(t)$ with weight vector $\boldsymbol{g}$ in the artificial neural network so that the evolution of the state vector $\boldsymbol{z}(t)$ (5) turns to a function of $\hat{\boldsymbol{u}}(t; \boldsymbol{g})$. Second, we use an appropriate

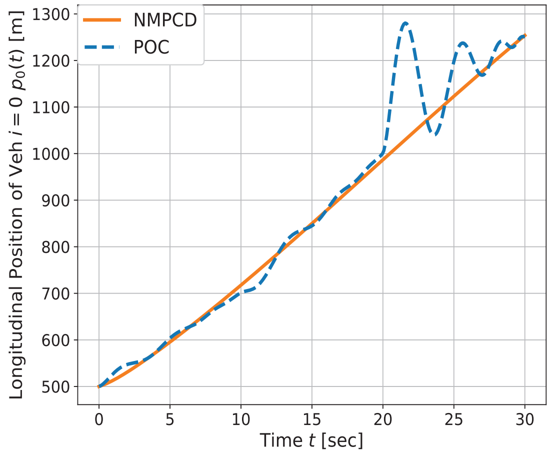

Fig. 4. Comparison of the leading vehicle position under the two controls.

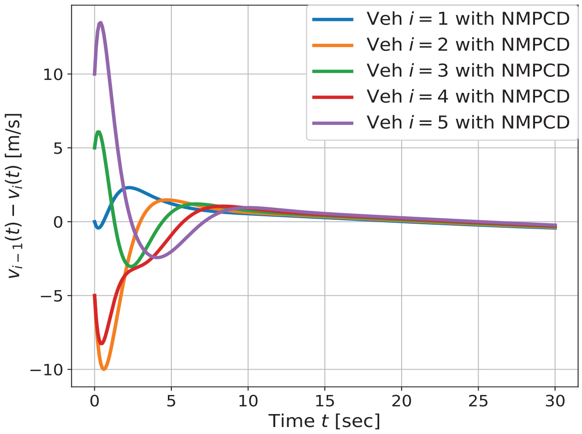

Fig. 6. Speed difference between the two adjacent vehicles under NMPC.

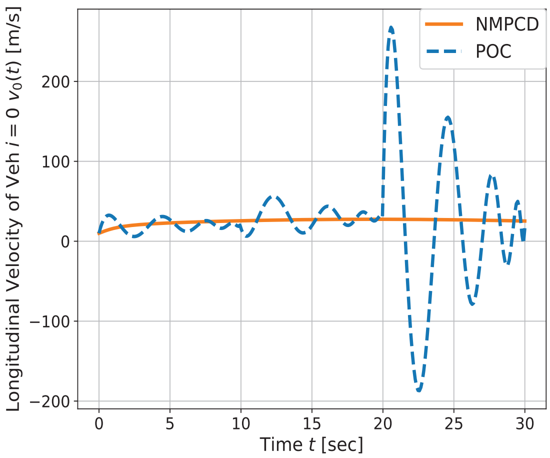

Fig. 5. Comparison of the leading vehicle velocity under the two controls.

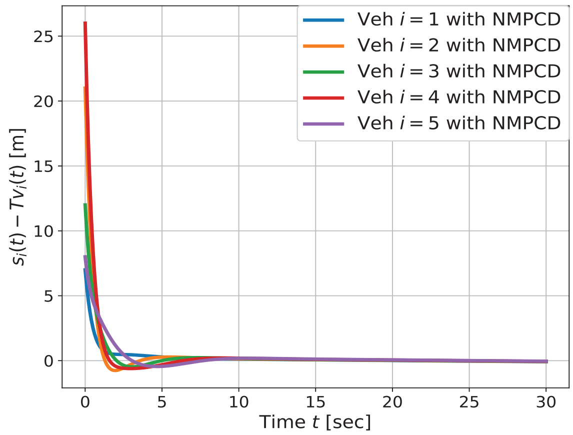

Fig. 7. The discrepancy between the actual distance and the expected distance between the two adjacent vehicles under NMPC.

loss function with mean-squared error and a gradient-descent method

$$J(\boldsymbol{z}(t_e), \boldsymbol{z}^*) = \frac{1}{2a+2}\|\boldsymbol{z}(t_e) - \boldsymbol{z}^*\|_2^2, \tag{12}$$

to evaluate $\boldsymbol{g}$ on the basis of

$$\boldsymbol{g}^{(i+1)} = \boldsymbol{g}^{(i)} - l\nabla_{\boldsymbol{g}^{(i)}} J(\boldsymbol{z}(t_e), \boldsymbol{z}^*), \tag{13}$$

in which l is the learning rate, and the superscript indicates the order of gradient correction currently used. Third, it is known from the literature [15] that the decreasing gradient in $\boldsymbol{g}$ will lead to the same effect in the control input $\hat{\boldsymbol{u}}(t; \boldsymbol{g})$. The expansion

$$\hat{\boldsymbol{u}}(t; \boldsymbol{g}^{(i+1)}) = \hat{\boldsymbol{u}}(t; \boldsymbol{g}^{(i)}) + \mathcal{J}_{\hat{\boldsymbol{u}}} \Delta \boldsymbol{g}^{(i)}, \tag{14}$$

where $\Delta \boldsymbol{g}^{(i)} = -l\nabla_{\boldsymbol{g}^{(i)}} J$, and $\mathcal{J}_{\hat{\boldsymbol{u}}}$ is the Jacobian of $\hat{\boldsymbol{u}}$, which can be described as $(\mathcal{J}_{\hat{\boldsymbol{u}}})_{xy} = \partial \hat{\boldsymbol{u}}_x / \partial \boldsymbol{g}_y$. A small learning rate l, will make $\Delta \boldsymbol{g}^{(i)}$ arbitrarily small, which will keep t constant. We can rearrange (14) as

$$\hat{\boldsymbol{u}}(t; \boldsymbol{g}^{(i+1)}) = \hat{\boldsymbol{u}}(t; \boldsymbol{g}^{(i)}) - l\mathcal{J}_{\hat{\boldsymbol{u}}}\mathcal{J}_{\hat{\boldsymbol{u}}}^{\mathsf{T}} \nabla_{\hat{\boldsymbol{u}}} J, \tag{15}$$

as a result of $\nabla_{\boldsymbol{g}^{(i)}} J = \mathcal{J}_{\hat{\boldsymbol{u}}}^{\mathsf{T}} \nabla_{\hat{\boldsymbol{u}}} J$, where the square matrix $\mathcal{J}_{\hat{\boldsymbol{u}}}\mathcal{J}_{\hat{\boldsymbol{u}}}^{\mathsf{T}}$ plays a role as a linear transformation on $\nabla_{\hat{\boldsymbol{u}}} J$.

NMPC usually begins with a modest control input $\hat{\boldsymbol{u}}^{(0)}(t; \boldsymbol{g}^{(0)})$, combining with the dynamical system, and executes a gradient descent in $\boldsymbol{g}$ (13). The smaller the loss, the less $\boldsymbol{g}$ changes, and the less the control input $\hat{\boldsymbol{u}}$ changes as the end state $\boldsymbol{z}(t_e)$ approaches the expected state $\boldsymbol{z}^*$, due to (15). The trajectory generated by the control behavior will track the system's vector field to its limit if the initial control input and learning rate are sufficiently small. The resultant control is similar to POC which reduces control energy due to the induced gradient descent. Therefore, instead of relying on the Hamiltonian and solving the adjoint system to minimize the control energy obviously, NMPC is a way of solving control problems that include implicit energy regularization.

IV. Numerical Simulation of NMPC

We now provide the effect of NMPC using simulations in the scenario, in which all six vehicles are driving in a single lane in the same direction. The leading autonomous vehicle (vehicle 0), with an initial position of $500\,\mathrm{m}$ and an initial speed of $10\,\mathrm{m/s}$, will obtain the control input trained by NMPC in the next 30 seconds to adjust its acceleration, speed, and position, and simultaneously update the state space vector to control the following five manual-driving vehicles to steer to the desired state. In Table. I, we give the assumed initial speed and position of the following vehicles.

For the setting of the required parameters in (6), we assume that $S_1 = 1.65$, $S_2 = 0.5$, $T = 1.2\,\mathrm{s}$, which will ensure stable performance for the whole system. These were acquired utilizing the handbook that was accessible, e.g., in [16]. To verify the ability of NMPC to control the acceleration and deceleration of the vehicle as required, we assume that $30\,\mathrm{s}$ is divided into three stages, each stage being $10\,\mathrm{s}$. The expected leading position of the three stages is $700\,\mathrm{m}$, $1000\,\mathrm{m}$, and $1250\,\mathrm{m}$ respectively. The expected velocity of all six vehicles in three stages are 20, 30, $15\,\mathrm{m/s}$ respectively.

Neural Network Controller is very sensitive to learning rate and often requires several adaptations to converge to desirable performance. More epochs on lower learning rates may further improve the performance. After optimizing the hyperparameters of the ANN that is used to form a mixed platooning, we consider it to be composed of 2 hidden layers with 15 exponential linear units (ELUs) as hidden layer neurons each, training epochs 10000, learning rate 0.01, and sample time is defined as $0.1\,\mathrm{s}$. To avoid the vanishing or exploding gradients in the back-propagation, we establish the weights $\boldsymbol{g}$ with the Kaiming uniform initialization algorithm and then let the optimizer (Adam) update the network parameters. The control output $\hat{\boldsymbol{u}}(t;\boldsymbol{g})$ is time-unfolded. We include no energy regularization term in the loss, and the loss weight of each variable in the loss function is set to 1. Dormand-Prince's approach with step-size control and dense output can better integrate the studied dynamical systems during solving ODE problems, so we choose to use it.

We show in Fig. 3 that NMPC after 10000 iterations makes the training loss tend to almost zero. When the number of iterations is small, the loss can reach the 6^{th} power of 10. As shown in the enlarged area in the figure, as the number of iterations increases, especially between the 50^{th} and 600^{th} iterations, the loss shows a clear downward trend. Further analysis shows that after tens of thousands of iterations, the loss is almost negligible, and the purpose of minimizing the loss function is achieved, which proves the effectiveness of NMPC control.

We analyze the control behavior of NMPC, which solves (5) utilizing neural ODEs, with that of POC on leading CAV. Fig. 4 and Fig. 5 show the evolution of the longitudinal position and velocity of the CAV (vehicle 0). We observe that under POC (dashed blue line): there exists obvious oscillation, and POC does not play an actual control role in this task. In the contrast, smooth curves illustrate that NMPC (solid orange line) can control the leading vehicle to approach the desired state based on neural ODE, and can freely accelerate or decelerate within the specified time as desired.

For the following HDVs, we see that although they have different initial positions and speeds, NMPC can make the two adjacent vehicles converge to have roughly the same speed in a short period (about $8-9\,\mathrm{s}$, possibly due to the first control target being set to the tenth-second state) (see Fig. 6), and the difference between the actual distance and the ideal distance based on ideal headway T of the two adjacent vehicles converges to 0 (see Fig. 7), achieving an ideal state.

In summary, NMPC does not need to solve an adjoint system compared to the traditional POC based on Pontryagin's maximum principle. NMPC only needs the information of the initial and the desired terminal state of a dynamical system to solve an optimal control solution. Our simulation results also validate the control performance as expected.

V. Discussion

We have investigated the control of a mixed vehicle platoon composed of a CAV leader and multiple HDV followers in this article. We formulate the heterogeneous dynamics of the mixed platoon and present a neural mixed platoon control framework that connects a state-feed neural network controller with the mixed platooning dynamics by using a NODE solver. This design enables us to learn optimal control inputs that can drive the platooning vehicles to reach their desired states within a given period in an unsupervised manner. We have also conducted simulations and show the effectiveness of the neural mixed platoon controller. In particular, numerical results indicate that our design has better scalability and control performance in terms of platooning errors and stability when compared to an energy-based optimal control approach. The results reveal the significant potential of combining deep learning techniques and optimal control formulation in vehicle platooning, which may open a promising direction for solving high-dimensional numerically complex control problems.

References

[1] Y. Liu, A. Zhou, Y. Wang, and S. Peeta, "Proactive longitudinal control to manage disruptive lane changes of human-driven vehicles in mixed-flow traffic," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 321–326, 2021.

[2] J. Lan, D. Zhao, and D. Tian, "Data-driven robust predictive control for mixed vehicle platoons using noisy measurement," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[3] J. Zhou, D. Tian, Z. Sheng, X. Duan, G. Qu, D. Cao, and X. Shen, "Decentralized robust control for vehicle platooning subject to uncertain disturbances via super-twisting second-order sliding-mode observer technique," *IEEE Transactions on Vehicular Technology*, 2022.

[4] J. Zhou, D. Tian, Z. Sheng, X. Duan, G. Qu, D. Zhao, D. Cao, and X. Shen, "Robust min-max model predictive vehicle platooning with causal disturbance feedback," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[5] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035006, 2016.

[6] G. Yan, J. Ren, Y.-C. Lai, C.-H. Lai, and B. Li, "Controlling complex networks: How much energy is needed?" *Physical review letters*, vol. 108, no. 21, p. 218703, 2012.

[7] M. A. Roehrl, T. A. Runkler, V. Brandtstetter, M. Tokic, and S. Obermayer, "Modeling system dynamics with physics-informed neural networks based on lagrangian mechanics," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9195–9200, 2020.
[8] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," *arXiv preprint arXiv:1907.04490*, 2019.
[9] Y. D. Zhong, B. Dey, and A. Chakraborty, "Symplectic ode-net: Learning hamiltonian dynamics with control," *arXiv preprint arXiv:1909.12077*, 2019.
[10] M. Lutter, K. Listmann, and J. Peters, "Deep lagrangian networks for end-to-end learning of energy-based control for under-actuated systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7718–7725.
[11] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," *arXiv preprint arXiv:2003.04630*, 2020.
[12] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[13] Z. Chen, J. Zhang, M. Arjovsky, and L. Bottou, "Symplectic recurrent neural networks," *arXiv preprint arXiv:1909.13334*, 2019.
[14] T. Duong and N. Atanasov, "Hamiltonian-based neural ode networks on the se (3) manifold for dynamics learning and control," *arXiv preprint arXiv:2106.12782*, 2021.
[15] L. Böttcher, N. Antulov-Fantulin, and T. Asikis, "Ai pontryagin or how artificial neural networks learn to control dynamical systems," *Nature Communications*, vol. 13, no. 1, pp. 1–9, 2022.
[16] D. T. McRuer and E. S. Krendel, "Mathematical models of human pilot behavior," ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT NEUILLY-SUR-SEINE (FRANCE), Tech. Rep., 1974.