IET Intelligent Transport Systems

WILEY

IET The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

LSTM-Based Centralized/Decentralized Controller Design for Vehicular Platooning

Ryota Nakai | Kazumune Hashimoto | Xun Shen | Shigemasa Takai

Graduate School of Engineering, Osaka University, Suita, Japan

Correspondence: Kazumune Hashimoto (hashimoto@eei.eng.osaka-u.ac.jp)

Received: 25 February 2025 | **Revised:** 22 December 2025 | **Accepted:** 5 January 2026

ABSTRACT

Cooperative adaptive cruise control, also known as vehicular platooning, has gained significant interest for its ability to enhance fuel efficiency and comfort in vehicle operations. This study proposes novel control strategies for vehicular platooning based on long short-term memory (LSTM) neural networks. By learning temporal dependencies in vehicle behaviour, the proposed LSTM-based controllers improve string stability within the platoon, particularly under varying velocity patterns of the lead vehicle. Two distinct frameworks are investigated: centralized and decentralized control models. The centralized model makes use of the states of all vehicles within the platoon, whereas the decentralized model focuses on the states of only a limited number of preceding vehicles. Simulation experiments demonstrate that both the centralized and decentralized LSTM controllers significantly outperform traditional, non-LSTM-based controllers in minimizing cumulative inter-vehicle error. This study contributes a novel controller training methodology that integrates LSTM-based architectures with optimal control principles, offering improved adaptability and flexibility in real-time platoon management.

1 | Introduction

Recent investigations into urban vehicle dynamics underscore considerable advancements in automation and safety, achieved through the sophisticated integration of information and communication technology. These advancements are particularly evident in the implementation of intelligent transportation systems and cooperative adaptive cruise control, which are pivotal for enhancing vehicular efficiency and safety. Various methodologies for vehicular communication have been explored, serving as a cornerstone for the efficacious deployment of automated systems within metropolitan infrastructures [1, 2]. Furthermore, the realm of autonomous vehicular navigation and assistance systems within urban environments is experiencing significant advancements with the incorporation of radar and LIDAR technologies. This integration markedly improves vehicular environmental perception, a critical aspect for navigating the complex and dynamic urban terrains (see e.g., [3]).

Traffic congestion remains a paramount challenge within the transportation sector, manifesting not only in significant safety hazards but also in considerable time inefficiencies. The concept of *platoon control*, which minimizes the inter-vehicle spacing, emerges as a viable solution offering multiple benefits. These include the mitigation of traffic congestion, alleviation of driver workload, and enhancement of fuel economy (see, e.g., [4–7]). The determination of optimal inter-vehicle spacing within a platoon is critical for ensuring both stability and safety. To establish this optimal spacing, three predominant methodologies are employed: constant spacing (CS), constant time headway (CTH), and variable time headway (VTH). The CS strategy ensures safety within the platoon's coordination; however, it is associated with reduced road utilization and demonstrates limited adaptability in complex scenarios, as indicated by [8]. In contrast, the CTH strategy effectively manages platoons, improves road utilization, and proves efficacious for control objectives, as evidenced in research conducted by [9, 10]. Nonetheless, its efficacy may

diminish at reduced velocities. Of late, the VTH approach has gained an increased attention [11], as it excels in handling complex situations and has proven benefits in fuel efficiency [12, 13].

String instability is one of the crucial phenomena in platooning coordination, characterized by the amplification and propagation of errors resulting from abrupt manoeuvres by the leading vehicle within a platoon [14]. Previous research has primarily focused on using transfer functions in the frequency domain for platoon control, with an emphasis on linear systems [15, 16]. Additionally, studies have explored time-domain control methodologies for platoon control [17] and examined the impact of time delays in these systems [18]. In addition, some optimal control or model predictive control-based approaches have been investigated to explicitly consider hard constraints [19–22]. For a comprehensive overview of string stability, including its advantages and disadvantages, refer to the insightful review presented in [23].

While various control strategies have been explored as above, effectively integrating the temporal behaviours of a human-operated leading vehicle into platoon control designs remains a significant challenge. The behaviour of the leading vehicle can vary dramatically under different circumstances, such as maintaining constant acceleration/deceleration or transitioning between these states. Consequently, there is a crucial need for a more sophisticated platoon control system architecture that can accommodate these temporal variations, an aspect not sufficiently addressed in the existing literature. To address the challenges posed by the variable and temporal nature of the leading vehicle, this paper proposes the integration of long short-term memory (LSTM) networks into our control design. LSTMs, a variant of recurrent neural networks, incorporate a cell, input gate, output gate and forget gate to effectively manage the vanishing gradient problem [24]. This methodology enables the controller to assess not only the current state of the system but also to account for temporal dependencies by incorporating information from previous states. The primary goal of this research is to develop a controller that achieves string stability within the platoon, thereby significantly enhancing its adaptability to the fluctuating behaviour patterns of the leading vehicle. Further, we explore the design of two control models within our framework: a centralized controller and a decentralized controller. The centralized model processes the states of all vehicles within the platoon, generating appropriate control inputs for each. Conversely, the decentralized model considers the states of immediately preceding vehicles, along with its own state, to generate specific control inputs. Both models leverage optimal control principles, where the objective involves minimizing a loss function to achieve string stability in terms of the error defined from the VTH. Particularly, decentralized controllers are designed by initializing their parameters with those optimized for the immediately preceding vehicle. This approach utilizes principles of transfer learning [25], leveraging pre-trained models to expedite training for new tasks. The effectiveness of our proposed methods is validated through numerical simulations, which compare scenarios with and without the integration of LSTMs. These simulations demonstrate the superior performance of the LSTM-based controllers, confirming their enhanced capability in managing the dynamic behaviours of the leading vehicle.

1.1 | Aim and Research Problem

This study aims to develop a control strategy for vehicle platooning that enhances string stability under varying lead-vehicle behaviours. Specifically, given the longitudinal platoon dynamics and a set of time-varying lead-vehicle velocity profiles, we seek a feedback controller that maps available state histories to control inputs so as to minimize the cumulative variable-time-headway (VTH) tracking error while fulfilling string stability (e.g., in the L_2 sense) across all profiles. In particular, the main contributions of this paper are threefold:

- LSTM-based controller design: We develop an LSTM-based controller for vehicle platooning. By exploiting temporal dependencies in the historical states, the LSTM enables adaptive and robust responses to changes in the lead vehicle's behaviour.
- Centralized and decentralized architectures: We investigate both centralized and decentralized configurations within the same framework. The centralized controller utilizes the states of all vehicles, whereas the decentralized controller relies only on the states of preceding vehicles, improving scalability and reducing computational burden.
- Validation via simulation: We demonstrate the effectiveness of the proposed methods through numerical simulations that compare scenarios with and without LSTM integration, showing improved string stability and adaptability under variable lead-vehicle behaviours.

1.2 | Related Works

This section reviews relevant studies in platoon control architectures, string stability metrics, and recent applications of neural networks in control.

1.3 | Modelling Car-Following Dynamics With LSTM

Our approach is related to the neural network-based modelling of car-following dynamics [26–28]. A primary focus in this research is to reproduce the driving behaviour based on the actual driving data, aiming to embed the car-following model in the simulation. Among these approaches, application of the LSTM has been recognized as particularly powerful and one of the state-of-the-art models [26, 27]. Note that the problem setup as well as the proposed approach of this paper are significantly different from those of the prior studies in the following sense. Instead of learning the car-following models from training data, we here investigate to design an *LSTM-based, feedback controller* as the solution to the optimal control problem. In particular, we propose the development of algorithms to craft both *centralized and decentralized controllers* represented by the LSTM, such that the loss function for achieving string stability is minimized. This introduces a novel training procedure characterized by an iterative process for evaluating a loss of the current controller propagated through the dynamics (forward computation), and updating parameters for controller through back propagation.

Through the integration of LSTM, the resulting controller can handle various motion patterns of the human-operated vehicle.

1.3.1 | Urban Transport Planning

Sustainable and smart mobility initiatives increasingly shape modern transportation systems. Ref. [29] emphasizes that expanding electric vehicle (EV) infrastructure supports renewable energy integration and accelerates sustainable mobility. Institutional and governance factors also play a crucial role, as highlighted in [30], while transit-oriented development approaches [31] demonstrate how land use and public transportation planning can reduce private car reliance. Urban traffic congestion remains a critical challenge, with studies such as [32] analysing congestion patterns and mitigation strategies. Complementary research on Riyadh's public transportation system [33] and smart transportation planning [34] underscores the need for operational solutions that align with infrastructure and policy goals. Our work addresses this gap by proposing an LSTM-based platoon control framework focused on micro-level operations. While prior studies mainly examine planning and governance, our approach enhances string stability and energy efficiency, providing a practical operational layer that complements broader smart transportation strategies.

In summary, prior research has made significant progress in improving platoon stability and responsiveness; however, most approaches either rely on simplified dynamics or lack the ability to capture long-term temporal dependencies inherent in real driving scenarios. These limitations motivate the present study, which integrates LSTM-based architectures with optimal control principles to enhance adaptability and robustness in real-time platoon management.

2 | Problem Formulation

2.1 | Dynamics

In this study, we consider the longitudinal control of a system composed of a lead vehicle driven by a human and $N \in \mathbb{N}$ autonomous followers. We denote the state of the followers as $x_i = [v_{i-1}, v_i, d_i]^\top$, where v_{i-1}, v_i, and d_i represent the velocity of the preceding vehicle, the velocity of the following vehicle, and the inter-vehicle distance, respectively. The control input of the followers is the force applied to the vehicle and is denoted as $u_i \in \mathbb{R}$ for all $i = 1, \ldots, N$.

The dynamics of the i-th following vehicle is characterized by the discrete-time nonlinear system as follows:

$$x_i(k+1) = f_i(x_i(k), u_i(k)), \tag{1}$$

where $x_i(k)$ and $u_i(k)$ represent the state of x_i and the control input u_i at time step k, respectively, and $f_i(\cdot)$ is a given continuously differentiable function. Regarding the leading (human-operated) vehicle, we assume access to its various velocity patterns, represented by $v_0^{(m)}(0), \ldots, v_0^{(m)}(T)$ for all $m = 1, \ldots, M$, where T denotes a predetermined horizon, and M denotes the total number of velocity patterns. The notation $v_0^{(m)}(k)$ specifies the velocity of the leading vehicle at time k for the m-th pattern. These patterns encapsulate potential behavioural patterns of the leading vehicle, such as maintaining constant acceleration or deceleration, and transitioning from acceleration to deceleration. Figure 1 provides an example of visualizing these velocity patterns. Throughout this study, we focus on highway platooning under a flat-terrain assumption, where road grades and signalized intersections are not modelled. Accordingly, gravitational grade forces are omitted from the longitudinal dynamics, and the lead-vehicle velocity v_0 is treated as an exogenous highway profile. Scenario-specific elements such as downhill grades, and traffic-light starts are left for targeted validation in future work.

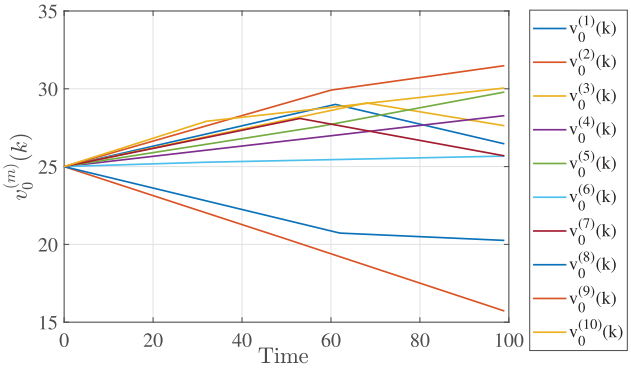


FIGURE 1 | An example of the set of the velocity patterns of the leading vehicle $v_0^{(m)}(0), \ldots, v_0^{(m)}(T)$, $m = 1, \ldots M$. It can be seen from the figure that some of the patterns represent the constant deceleration (e.g., $v_0^{(9)}(k)$), constant acceleration (e.g., $v_0^{(4)}(k)$), and changing from acceleration to deceleration at a certain time instant (e.g., $v_0^{(1)}(k)$).

2.2 | String Stability

In this section, we introduce the performance index so as to characterize spacing policy. Deciding on the right distance between the vehicles is crucial for controlling the platoon effectively and efficiently, and it can serve as a measure to see how well the controller is handling the platoon. Each vehicle aims to keep a distance from the one in front of it that matches this desired distance. This helps regulate how the vehicles in the platoon follow each other.

In this paper, we introduce the concept of VTH [11] for characterizing the spacing policy. This concept is employed to achieve the desired distance between i-th and $(i-1)$-th vehicle at each time according to the following equation:

$$d_{\mathrm{des},i}(k) = w_i(k) v_i(k) + d_{\min} \tag{2}$$

$$w_i(k) = w_0 - c_0(v_{i-1}(k) - v_i(k)), \tag{3}$$

for all $i = 1, \ldots, N$, where $w_i(k)$ is the (time-varying) headway of the i-th vehicle at time k, $d_{\min} > 0$ denotes a given prescribed minimum distance between vehicles to enhance safety guarantees, $v_{i-1}(k), v_i(k)$ are $(i-1)$-th and i-th speed at time k, and $c_0 > 0$ and $w_0 > 0$ are the constant parameters, respectively. The concept of VTH has originally been inspired from the CTH, wherein the headway is assumed to be constant for all times [35]. VTH

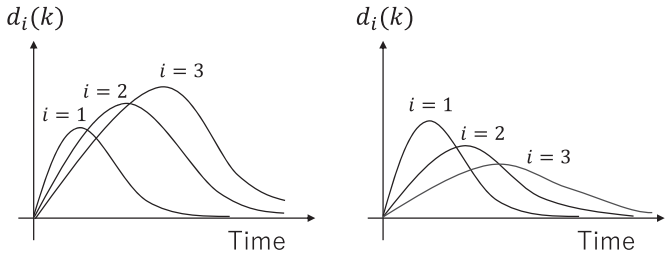


FIGURE 2 | Illustration of the string instability. The left figure shows the case where string instability occurs. The right figure shows the case where string instability can be avoided, which means that string stability is satisfied.

introduces the time-varying characteristics of the headway, and is determined by the deviation of the velocities from the preceding vehicle. It is known that this time-varying adjustment under VTH enhances traffic flow capabilities, surpassing the constant time headway approach employed in CTH [8].

When the leading vehicle in a platoon experiences a marked speed change, such as sudden deceleration or acceleration, the error caused by this speed variation can be transmitted to, or even magnified in, the following vehicles. This issue is widely known as *string instability*, a phenomenon that must be mitigated due to its potential to disrupt platoon coordination (see Figure 2 for this illustration). In this paper, we define the string stability based on the error in terms of the VTH as follows. First, let $e_i(k)$ be the error given by

$$e_i(k) = d_i(k) - d_{\mathrm{des},i}(k). \tag{4}$$

Then, string stability is considered achieved if the following condition is met for all vehicles $i = 1, \ldots, N$:

$$\sum_{k=1}^{T} e_{i-1}^2(k) > \sum_{k=1}^{T} e_i^2(k). \tag{5}$$

The left-hand side denotes the cumulative error for the $(i-1)$-th vehicle, while the right-hand side represents the cumulative error for the i-th vehicle over a given horizon T. Consequently, this condition signifies that the total error diminishes as one moves towards the end of the platoon. In other words, when the above conditions are met, the error arising from the sudden change in speed of the leading vehicle does not amplify or propagate towards the subsequent vehicles.

Remark 1. The definition of the string stability in this paper is indeed corresponding to $\mathcal{L}_2$ *string stability* (see, e.g., [23]). □

2.3 | Control Architectures

As previously mentioned, our primary objective is to enable the following vehicles to maintain a desired distance despite the effect of the leading vehicle for ensuring string stability (5). In pursuit of this goal, we aim at building control architectures that account for the time-varying nature of the leading vehicle's behaviour characterized by its velocity patterns $v_0^{(m)}(0), \ldots, v_0^{(m)}(T)$, $m = 1, \ldots M$. This means that the controller must generate control

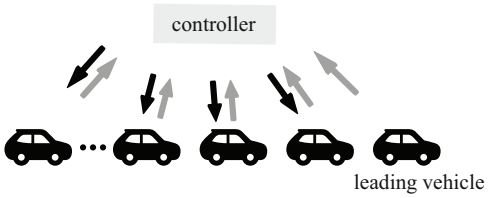


FIGURE 3 | Architecture of the centralized controller.

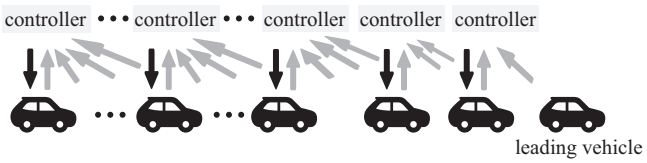


FIGURE 4 | Architecture of the decentralized controller.

inputs based not only on the information of the current states, but also on the past ones. As we will see later, we address such temporal aspects by building controllers through the LSTM. In addition, we propose to design two control architectures: the centralized and the decentralized controller. The specifics of these architectures will be detailed in the next subsections.

2.3.1 | Centralized Controller

The architecture of the centralized controller is illustrated in Figure 3. As shown in the figure, the centralized controller is designed to receive the states from all vehicles and then generate suitable control inputs for all vehicles accordingly. More specifically, we aim to design a controller characterized as

$$u(k) = \pi_G\left(\{x(\ell)\}_{\ell=0}^{k}; W_G\right), \tag{6}$$

where $\pi_G(\cdot; W_G)$ denotes a policy of the centralized controller with W_G being a parameter to characterize π_G, and $x(k)$ (respectively, $u(k)$) is a vector that stacks all the state information (respectively, control inputs), that is, $x(k) = [x_1(k), x_2(k), \ldots, x_N(k)]^\top$ (respectively, $u(k) = [u_1(k), u_2(k), \ldots, u_N(k)]^\top$). Note that, as previously mentioned, the controller depends not only on the current state $x(k)$ but also on the past ones $x(0), \ldots, x(k-1)$, which is due to the time-varying and various patterns of the leading vehicle's behaviour.

2.3.2 | Decentralized Controller

The architecture of the decentralized controller is illustrated in Figure 4. The decentralized controller is designed to receive the set of states only from a limited number of the preceding vehicles and its own state, and to generate the corresponding control input for itself. Hence, each vehicle in the platoon is equipped with its individual decentralized model-based controller, contributing collectively to the platoon's coordinated control.

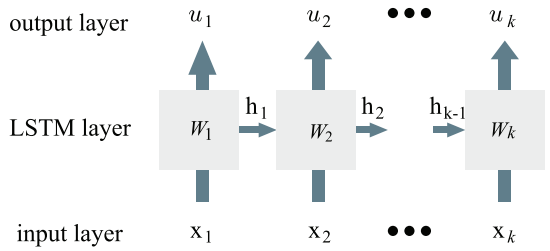


FIGURE 5 | Architecture of the LSTM-based controller.

Specifically, we design a controller for the decentralized case as follows: for all $i = 1, \ldots, N$,

$$u_i(k) = \pi_L(\{x_{L_i}(\ell)\}_{\ell=0}^{k}; W_L) \tag{7}$$

where, for a given $n_L < N$, $x_{L_i} = [x_{i-n_L}(k), x_{i-(n_L-1)}(k), \ldots, x_i(k)]^\top$ if $i \geq n_L$ and $x_{L_i} = [x_0(k), x_1(k), \ldots, x_i(k)]^\top$ if $i < n_L$. In other words, n_L expresses the number of the preceding vehicles that the decentralized controller for each following vehicle utilizes the state information.

2.4 | Discussion on the Choice of Control Architectures

There are both advantages and disadvantages for choosing between centralized and decentralized control architectures. The primary advantage of implementing a centralized controller is the simplicity of the training process, as it involves developing only one controller. However, this approach has scalability limitations; the computational complexity involved in designing the controller can significantly increase with the number of vehicles. Additionally, a centralized controller offers limited flexibility, struggling to adapt to changes in the platoon configuration, such as vehicles merging or catching up. In contrast, decentralized controllers provide superior flexibility by allowing precise management of individual vehicles within the platoon. This capability facilitates adaptation to changes, such as the integration of new vehicles from behind, achieved through the introduction of additional controllers for each new vehicle. However, control performance may degrade compared to the centralized approach because decentralized controllers generate control inputs based only on a limited number of immediately preceding vehicles. Consequently, selecting the better control architecture requires consideration of specific driving scenarios, including the number of vehicles involved and whether vehicle merging is anticipated, as well as the consideration of allowable computational time and desired control performance.

3 | Training LSTM-Based Controller for Platoon

This section presents detailed methodologies for the development of the controllers introduced in the previous section. To address the various temporal behaviours of the leading vehicle as indicated by its velocity profile, we propose the training of both centralized and decentralized controllers through the usage of LSTM networks. Figure 5 illustrates the control architecture that incorporates LSTM. This architecture enables the capture of temporal correlations through hidden states h within the LSTM's intermediate layer, facilitating the formulation of control inputs

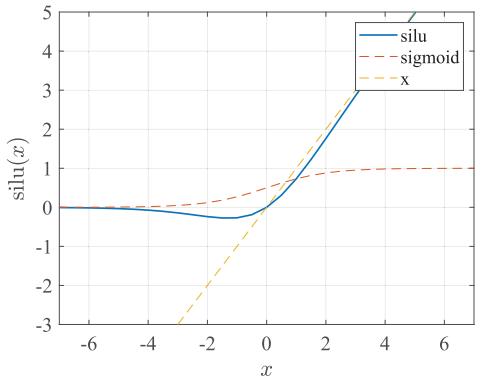


FIGURE 6 | Illustration of the SiLU function.

that consider both present and past states. In a platoon context, employing an LSTM-based controller allows for the recognition of the lead vehicle's temporal behaviours, including instances of abrupt acceleration or deceleration. Such functionality significantly improves the coordination of multiple vehicles, especially in scenarios where the lead vehicle is operated by a human. Moreover, it offers potential advancements in control performance in terms of string stability, surpassing the capabilities of traditional deep neural networks devoid of LSTM components.

The following subsections are dedicated to defining a suitable loss function intended for minimization during the controllers' training phase. Subsequently, algorithms for the training of both centralized and decentralized controllers are presented, aiming to minimize the identified loss function.

3.1 | Loss Function and Optimization Problem

We define an appropriate loss function so as to maintain a target speed and to achieve the string stability under various velocity patterns of the leading vehicle. First, given the initial state $x(0)$, m-th velocity profile of the leading vehicle $v_0^{(m)}(0), \ldots, v_0^{(m)}(T)$ and a controller (either a centralized one π_G or a decentralized one π_L), we can produce the state trajectory denoted as $x^{(m)}(0), \ldots, x^{(m)}(T)$ with $x^{(m)}(k) = [x_1^{(m)}(k), \ldots, x_N^{(m)}(k)]^\top$ and $x_i^{(m)}(k+1) = f_i(x_i^{(m)}(k), \pi_G(\{x(\ell)\}_{\ell=0}^{k}; W_G))$ (or $x_i(k+1) = f_i\left(x_i(k), \pi_L(\{x_{L_i}(\ell)\}_{\ell=0}^{k}; W_L)\right)$ for the decentralized controller case) for all $k = 0, \ldots, T$ and $i = 1, \ldots, N$. Based on this notation, let us define the loss function as follows:

$$\text{Loss} = \sum_{m=1}^{M} \sum_{k=1}^{T} \left\{ \text{Loss}_E(x^{(m)}(k)) + \alpha \text{Loss}_{\text{ST}}(x^{(m)}(k)) \right\} \tag{8}$$

where $\alpha > 0$ is a weight parameter, and

$$\text{Loss}_E(x^{(m)}(k)) = \sum_{i=1}^{N} (e_i^{(m)}(k))^2 \tag{9}$$

$$\text{Loss}_{\text{ST}}(x^{(m)}(k)) = \sum_{i=2}^{N} \text{SiLU}((e_i^{(m)}(k))^2 - (e_{i-1}^{(m)}(k))^2) \tag{10}$$

where $e_i^{(m)}(k)$ denotes the error with respect to the VTH computed from $x^{(m)}(k)$. The function $\text{SiLU}(\cdot)$ is given by $\text{SiLU}(x) = x \cdot \text{sigmoid}(x)$ (for the illustration, see Figure 6). The main advantage

of using the SiLU function instead of the ReLU function is that it is differentiable at $x = 0$.

The first term of the loss function Loss_E is designed to reduce errors with respect to the VTH over time, while the second term Loss_{ST} aims to ensure string stability within the platoon, that is, the following vehicle must diminish its error compared to the vehicle ahead in the platoon under all velocity patterns of the leading vehicle.

Now, recall that our goal is to design both the centralized and the decentralized controllers, wherein the former one utilizes the state information of all vehicles (6), while the latter one utilizes the state information from only a limited number of the preceding vehicles (7). The optimization problem for training the centralized controller is summarized as follows:

Problem 1 (Centralized controller). Given the initial state $x(0)$, the system (1) and the set of the velocity patterns of the leading vehicle $\{v_0^{(m)}(k)\}_{k=1}^{T}, m = 1, 2, \ldots, M$, find the parameter W_G by solving the following optimization problem.

$$\begin{aligned} &\min_{W_G} \sum_{m=1}^{M}\sum_{k=0}^{T} \left\{\text{Loss}_E(x^{(m)}(k)) + \alpha \text{Loss}_{\text{ST}}(x^{(m)}(k))\right\} \\ &\text{s.t. } x_i^{(m)}(k+1) = f_i\left(x_i^{(m)}(k), \pi_G\left(\{x(\ell)\}_{\ell=0}^{k}; W_G\right)\right), \\ &\quad \forall k = 0, 2, \ldots, T-1,\ i = 1, 2, \ldots, N. \end{aligned} \tag{11}$$

□

In Problem 1, $\pi_G(\cdot, ; W_G)$ represents a controller as the LSTM network, where W_G is the corresponding weight parameter to be optimized. The optimization problem for training the decentralized controller is summarized as follows:

Problem 2 (Decentralized controller). Given the initial state $x(0)$, the system (1) and the set of the velocity patterns of the leading vehicle $\{v_0^{(m)}(k)\}_{k=1}^{T}, m = 1, 2, \ldots, M$, find the parameters $W_{L,n}$ for all $n = 1, \ldots, N$ by solving the following optimization problem.

$$\begin{aligned} &\min_{W_{L,1}\ldots W_{L,N}} \sum_{m=1}^{M}\sum_{k=0}^{T} \left\{\text{Loss}_E(x^{(m)}(k)) + \alpha \text{Loss}_{\text{ST}}(x^{(m)}(k))\right\} \\ &\text{s.t. } x_i^{(m)}(k+1) = f_i\left(x_i^{(m)}(k), \pi_L(\{x_{L_i}(\ell)\}_{\ell=0}^{k}; W_{L,n})\right), \\ &\quad \forall k = 0, 2, \ldots, T-1,\ i = 1, 2, \ldots, N. \end{aligned} \tag{12}$$

□

In Problem 2, $\pi_L(\cdot, ; W_{L,n}), n \in \{1, \ldots, N\}$ represents a controller for n-th vehicle as the LSTM network, where $W_{L,n}$ is the corresponding weight parameter to be optimized.

3.2 | Proposed Approach to Training Controllers

In this section, we provide detailed approaches to solving Problems 1 and 2.

ALGORITHM 1 | Algorithm to learn the centralized controller.

Input : $x(0)$ (initial state); $v_0^{(m)}(0), \ldots, v_0^{(m)}(T), m = 1, \ldots, M$ (the set of velocity patterns of the leading vehicle); W_G (initial weight parameter); N_{ite} (number of iterations); N_{batch} (mini-batch size)

Output: Optimized weight parameter for the centralized controller W_G^*

1 **for** $l = 1 : N_{\text{ite}}$ **do**
2 $\text{Loss}_{\text{total}} \leftarrow 0$;
3 **for** $o = 1 : N_{\text{batch}}$ **do**
4 $m_o \leftarrow$ randomly select from $\{1, \ldots, M\}$;
5 Given $x(0), v_0^{(m_o)}(0), \ldots, v_0^{(m_o)}(T)$ and W_G,compute the state trajectory $x_i^{(m_o)}(0), \ldots, x_i^{(m_o)}(T)$ for all $i \in \{1, \ldots, N\}$ according to (11);
6 Compute the corresponding loss as $\text{Loss}_o = \sum_{k=0}^{T}\left\{\text{Loss}_E(x^{(m_o)}(k)) + \alpha \text{Loss}_{ST}(x^{(m_o)}(k))\right\}$
7 $\text{Loss}_{total} \leftarrow \text{Loss}_{total} + \text{Loss}_o$;
8 **end**
9 Using Loss_{total}, compute $\text{Loss}/\partial W_G$;
10 $W_G \leftarrow W_G - \beta \partial \text{Loss}/\partial W_G$;
11 **end**
12 $W_G^* \leftarrow W_G$;

3.2.1 | Training Centralized Controller

As depicted in Figure 3 and Equation (6), for each time step k, the centralized controller receives the current states from all vehicles within the platoon and stack them to the historical data, that is, $\{x(\ell)\}_{\ell=0}^{k}$, and then control inputs are generated accordingly for all the following vehicles. To address the controller's dependency on the historical data $\{x(\ell)\}_{\ell=0}^{k}$, we develop a controller as an LSTM network trained by minimizing the loss (8), so that the resulting controller can handle various temporal behaviours of the leading vehicle.

The learning process involves iterations of both the forward computation and the back-propagation. The detailed algorithm is given in Algorithm 1. To enhance efficiency, we employ a mini-batch strategy with a size of N_{batch} within the iteration count until N_{ite}. In the forward computation (lines 3–8), for every trial $o \in \{1, \ldots N_{\text{batch}}\}$, one of the velocity patterns is randomly selected, which is denoted as $v_0^{(m_o)}(0), \ldots, v_0^{(m_o)}(T)$, $m_o \in \{1, \ldots, M\}$, and then compute the state trajectory $x_i^{(m_o)}(0), \ldots, x_i^{(m_o)}(T)$ for all $i \in \{1, \ldots, N\}$ according to the dynamics and the corresponding loss using the current parameter W_G. Then, the parameter W_G is updated by computing the gradient of the loss with respect to W_G from the back-propagation through time (line 9). This computation can be easily implemented by combining the auto-differentiation tools designed for NNs, such as PyTorch. The above process is iterated until the prescribed maximum number of iteration N_{ite} is reached.

3.2.2 | Training Decentralized Controller

As depicted in Figure 4, the decentralized controller equipped for each following vehicle $\pi_{L,i}$ utilizes the information from only n_L preceding vehicles along with its own state, producing the control inputs accordingly through the LSTM as elaborated in the previous section.

The overview of the learning process is shown in Figure 7, where, in contrast to the centralized case, the decentralized controller

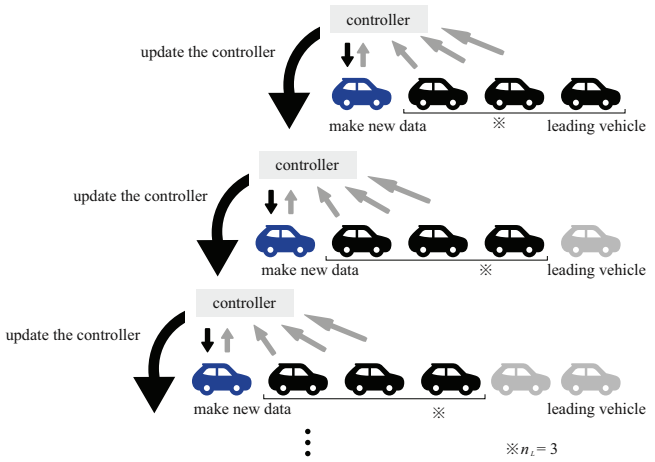


FIGURE 7 | Overview of the training procedure for decentralized controller (for the case $n_L = 3$).

is repeatedly constructed towards the rear vehicles. First, we assume that the centralized controller for the first $n_L - 1$ vehicles has been constructed in advance (i.e., Algorithm 1 is implemented with N being replaced by n_L). It is argued that this construction is computationally light in contrast to the original centralized case, as it is only necessary to construct a controller for $n_L < N$ vehicles. We denote by $x_n^{(m)}(0), \ldots, x_n^{(m)}(T)$ for all $n = 1, \ldots, n_L$ and $m = 1, \ldots, M$ the corresponding state trajectories for all velocity patterns. In Algorithm 2, the optimized parameter for this centralized controller is denoted as W^*_{G,n_L}. The algorithm proceeds by constructing a decentralized controller for the subsequent vehicles (i.e., $n \in \{n_L, \ldots, N\}$-th vehicles) iteratively through the implementation of both the forward and the backward processes. As with the centralized case, for every trial $o \in \{1, \ldots N_{\text{batch}}\}$, one of the velocity patterns is randomly selected that is denoted as $v_0^{(m_o)}(0), \ldots, v_0^{(m_o)}(T)$, $m_o \in \{1, \ldots, M\}$, and then compute the state trajectory for the n-th vehicle according to the dynamics and the corresponding loss, given the current parameter $W_{L,n}$ and previously optimized parameters for the preceding vehicles $W^*_{L,n_L}, \ldots, W^*_{L,n-1}$ (line 10). In addition, as will be detailed below, we periodically adjust parameters based on the trajectories of the preceding ($n-1$-th) vehicle (lines 15–22). The decentralized controller parameters are updated in each iteration based on the total loss (line 14 or line 21). This sequential learning continues until the final vehicle is reached.

It is remarked that we aim at enhancing training speed and generalization performance by incorporating two key strategies. Firstly, we expedite the training of decentralized controllers for vehicles from n_L to N by initializing their parameters with those optimized for the immediately preceding vehicle, as given in line 4. This approach draws from transfer learning [25] principles, leveraging pre-trained models to speed up training for new tasks. Secondly, to prevent overfitting to the training data of n-th vehicle's state trajectories, we periodically adjust parameters based on the trajectories of the preceding ($n-1$-th) vehicle, as detailed in lines 15–22. This ensures our controller model generalizes well across different state trajectories.

ALGORITHM 2 | Learning decentralized controllers.

Input : $x(0)$ (initial state); $v_0^{(m)}(0), \ldots, v_0^{(m)}(T)$, $m = 1, \ldots, M$ (the set of velocity patterns of the leading vehicle); W^*_{G,n_L} (parameter of the centralized controller for the first $n_L - 1$ vehicles); N_{ite} (number of iterations); N_{batch} (batch size); n_{ite} (periods to update the parameter from the $n-1$-th vehicle); $W_{L,n}$, $n = n_L, \ldots, N$ (initial weight parameters);

Output: Optimized weight parameters $W^*_{L,n}, n = n_L, \ldots, N$

1 For all $m = 1, \ldots, M$, by applying the centralized controller for the first $n_L - 1$ vehicles (with the parameter given by W^*_{G,n_L}),compute the state trajectory for the first $n_L - 1$ vehicle as $x_n^{(m)}(0), \ldots, x_n^{(m)}(T)$ for all $n = 1, \ldots, n_L$;
2 **for** $n = n_L : N$ **do**
3 **if** $n > n_L$ **then**
4 $W_{L,n} \leftarrow W^*_{L,n-1}$ (otherwise, $W_{L,n}$ is randomly chosen);
5 **end**
6 **for** $l = 1 : N_{\text{ite}}$ **do**
7 $\text{Loss}_{total} \leftarrow 0$;
8 **for** $o = 1 : N_{\text{batch}}$ **do**
9 $m_o \leftarrow$ randomly select from $\{1, \ldots, M\}$;
10 Given $x(0), \{x_n^{(m_o)}(0), \ldots, x_n^{(m_o)}(T)\}_{n=0}^{n_L-1}, W^*_{L,n_L}, \ldots W^*_{L,n-1}$ and $W_{L,n}$, compute the state trajectory $x_{n-\ell}^{(m_o)}(0), \ldots, x_{n-\ell}^{(m_o)}(T)$ for all $\ell = 0, \ldots, n_L - 1$ (i.e., the trajectories of n_L vehicles ahead of the n-th vehicle).
11 Compute the corresponding loss as $\text{Loss}_o = \sum_{\ell=0}^{n_L-1} \sum_{k=0}^{T} \{\text{Loss}(x_{n-\ell}^{(m_o)}(k)),$ where $\text{Loss}(x_{n-\ell}^{(m_o)}(k)) = \text{Loss}_E(x_{n-\ell}^{(m_o)}(k)) + \alpha \text{Loss}_{ST}(x_{n-\ell}^{(m_o)}(k))\}$;
12 $\text{Loss}_{total} \leftarrow \text{Loss}_{total} + \text{Loss}_o$;
13 **end**
14 $W_{L,n} \leftarrow W_{L,n} - \beta \partial \text{Loss} / \partial W_{L,n}$;
15 **if** $(l \bmod N_{\text{ite}}) == 0$ **then**
16 **for** $o = 1 : N_{\text{batch}}$ **do**
17 Given $x(0), \{x_n^{(m_o)}(0), \ldots, x_n^{(m_o)}(T)\}_{n=0}^{n_L-1}, W^*_{G,n_L}, W^*_{L,n_L}, \ldots W^*_{L,n-1}$ and $W_{L,n}$, compute the state trajectory $x_{n-\ell}^{(m_o)}(0), \ldots, x_{n-\ell}^{(m_o)}(T)$ for all $\ell = 1, \ldots, n_L$ (i.e., the trajectories of n_L vehicles ahead of the $n-1$-th vehicle).
18 Compute the corresponding loss as $\text{Loss}_o = \sum_{\ell=1}^{n_L} \sum_{k=0}^{T} \{\text{Loss}(x_{n-\ell}^{(m_o)}(k)),$ where $\text{Loss}(x_{n-\ell}^{(m_o)}(k)) = \text{Loss}_E(x_{n-\ell}^{(m_o)}(k)) + \alpha \text{Loss}_{ST}(x_{n-\ell}^{(m_o)}(k))\}$;
19 $\text{Loss}_{total} \leftarrow \text{Loss}_{total} + \text{Loss}_o$;
20 **end**
21 $W_{L,n} \leftarrow W_{L,n} - \beta \partial \text{Loss} / \partial W_{L,n}$;
22 **end**
23 **end**
24 $W^*_{L,n} \leftarrow W_{L,n}$;
25 **end**

4 | Simulation Results

In this section, we illustrate the effectiveness of the proposed approach given in the previous section. All experiments were conducted on a PC with Windows Intel i7 CPU, 32 GB RAM. The code is available on https://github.com/naka-ir/vehicle-data.

4.1 | Parameter Setting

With $M = 100$, we generate speed patterns of the leading vehicle in the same way as Figure 1. More precisely, we assume that the initial speed was set to 25 m s^{-1}, and with sampling period $\Delta t = 0.1$, the leader's speed evolves as

TABLE 1 | Simulation parameters.

Δ	v_0	ν_1	ν_2	T	$d_{\min}$	w_0	c_w	n_L
1 s	50 N	2 N s m^{-1}	0.1 N s^2 m^{-2}	100	3 m	0.1 s	0.2	3

$$v_0(k+1) = v_0(k) + a(k)\,\Delta t,$$

where the acceleration is piecewise constant with a single switch:

$$a(k) = \begin{cases} a_1, & 0 \le k < k_{\text{sw}}, \\ a_2, & k \ge k_{\text{sw}}. \end{cases}$$

The values a_1 and a_2 are drawn uniformly from $[-0.1, 0.1]$ m s^{-2} (with opposite signs to induce a deceleration → acceleration or acceleration → deceleration transition), and the switch index k_{sw} is sampled within the simulation horizon to ensure a mid-course change. This setup produces abrupt yet bounded variations in the leader's speed, enabling the controller to be tested under nontrivial transient conditions. The initial error for all vehicles is set to zero. The horizon for all simulations and controller evaluations is set to $T = 100$, aligning with the time horizon used in the training loss computation (8) and consistent with the simulation duration.

We consider building a platooning controller that comprises $N = 6$ following vehicles. The dynamics of the following vehicles are given by,

$$x_i(k+1) = x_i(k) + \Delta \begin{bmatrix} 0 \\ u_i(k) \\ x_{i-1}(k+1) - x_i(k) \end{bmatrix} + \Delta \begin{bmatrix} -(\nu_0 + \nu_1 x_{i-1}(k+1) + \nu_2 x_{i-1}^2(k+1))/M_{i-1} \\ -(\nu_0 + \nu_1 x_i(k) + \nu_2 x_i^2(k))/M_i \\ 0 \end{bmatrix} \tag{13}$$

for all $i = 1, \ldots, N$, where Δ represents the sampling time period, M_i is the mass of the i-th vehicle and ν_0, ν_1, ν_2 are the air drag coefficients. The mass of the following vehicle is randomly selected from a range $M_i \in [1000, 1800]$ for all $i = 1, \ldots, N$. Other parameters, including those for VTH, are shown in Table 1.

4.2 | Synthesis of Centralized Controller

In this section, we illustrate simulation results of a centralized controller. We particularly conduct a comparative analysis between a centralized DNN controller with LSTM and without LSTM. For training a centralized controller *without* LSTM, we synthesize a controller that does not utilize the history of the state information:

$$u(k) = \pi_G(x(k); W_G), \tag{14}$$

that is, only the current state information $x(k)$ is used to generate the control input. On the other hand, for the case where we use the LSTM, a controller is synthesized according to (6). The hyperparameters employed during the learning process of the model with LSTM and without LSTM are manually tuned to obtain the best performance and these are shown in Tables 2 and 3, respectively.

TABLE 2 | Hyper parameters for centralized controller with LSTM.

Hidden state	128
Layer	6
Learning rate	0.0005
Epoch	70
Iteration	100
Batch size	8
α	0.1

TABLE 3 | Hyper parameters for centralized controller *without* LSTM.

Hidden state	64 128 128 128 128 64
Layer	8
Learning rate	0.0005
Epoch	100
Iteration	50
Batch size	8
α	1

Figure 8 illustrates the simulation results by applying the centralized controller trained with LSTM and without LSTM. Figure 8a represents velocity, error and distance of the control outcomes with LSTM and Figure 8c represents the average of cumulative errors computed by $\frac{1}{T}\sum_{k=1}^{T} e_i^2(k)$ against the vehicle index i. Figure 8b,d are the outcomes without LSTM. As illustrated in Figure 8a,c, velocity of the leading vehicle undergoes a sudden transition from acceleration to deceleration at some time instance. Consequently, the error of the first following vehicle increases (blue solid line). When the LSTM is incorporated (Figure 8a), the error of the following vehicles behind remains consistently close to zero, and the average cumulative error decreases over the rear vehicle, showing the satisfaction of string stability. On the other hand, when the LSTM is *not* incorporated (Figure 8b), the error does not sufficiently decrease over time. Importantly, this escalation in error is not limited to the first vehicle; errors for subsequent vehicles also emerge, preventing them from maintaining proximity to zero. Figure 8d shows that the string stability is not achieved as the cumulative error does not decrease for some i. Such a failure is due to the fact that the controller without LSTM is trained to generate control inputs based only on the current state and so does not capture temporal behaviours of the leading vehicles. Hence, the proposed LSTM approach demonstrates the potential to design a controller, such that not only the stabilization to the target distance but also the string stability.

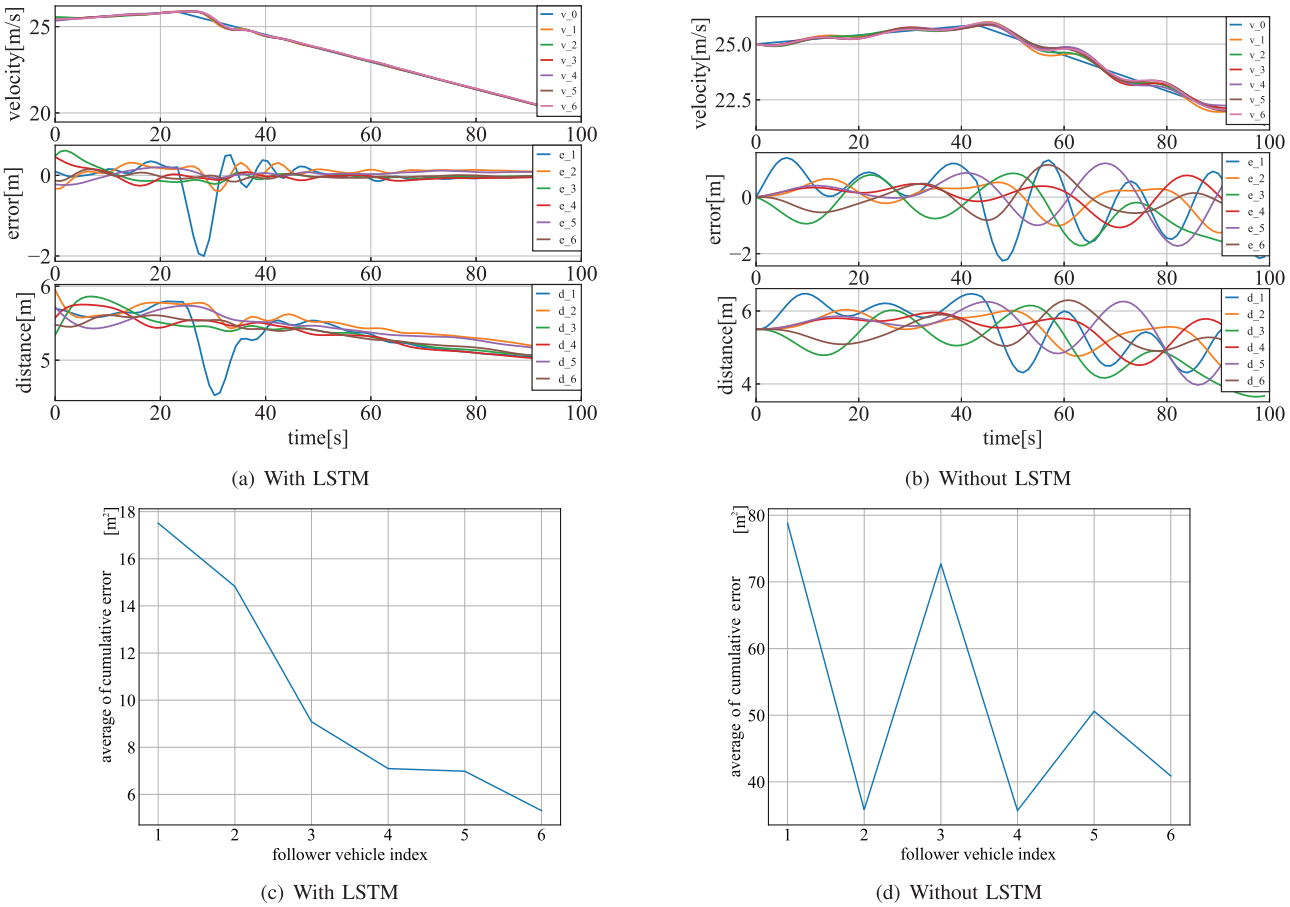


FIGURE 8 | Simulation results by applying the learned centralized controller equipped with LSTM and without LSTM. (a) Velocity, error and distance of the control outcomes with LSTM. (c) Average of cumulative errors $\frac{1}{T}\sum_{k=1}^{T} e_i^2(k)$ against the vehicle index i. (b,d) Outcomes without LSTM.

TABLE 4 | Hyper parameters for decentralized controller.

Hidden state	256
Layer	8
Learning rate	0.0005
Epoch	300
Iteration	50
Batch size	4
α	1

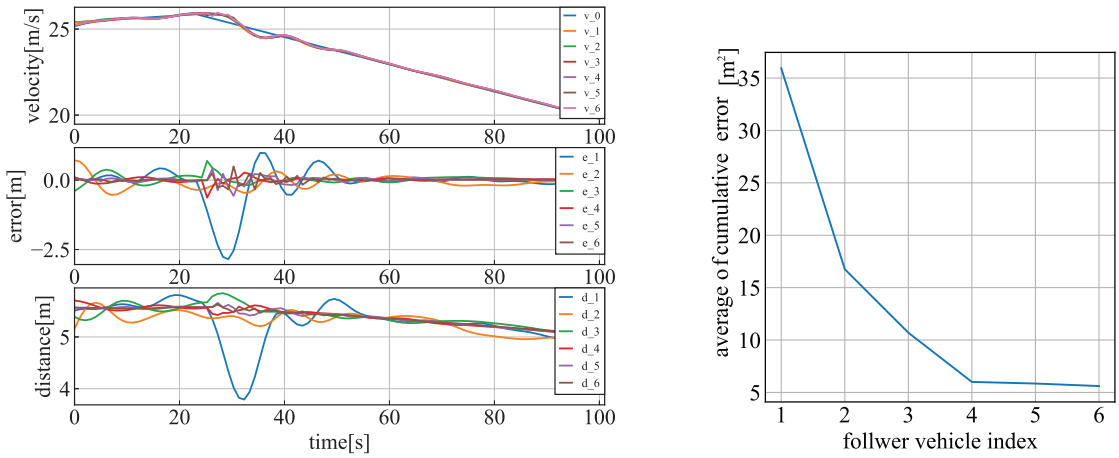
4.3 | Synthesis of Decentralized Controller

The hyperparameters of the decentralized controller are presented in Table 4, and the simulation results are visualized in Figure 9. As with the centralized case, Figure 9a represents velocity, error and distance by applying the trained decentralized controller, and Figure 9b represents the average of cumulative errors $\frac{1}{T}\sum_{k=1}^{T} e_i^2(k)$. Although the error of the first following vehicle exhibits an increase due to an abrupt change of the leading vehicle (blue solid line), the error of the following vehicles behind remains consistently close to zero. Figure 9b shows that the average cumulative error decreases and so satisfies string stability.

TABLE 5 | Performance comparisons between decentralized and centralized controllers.

	Execution time (in min)	**Averaged error (in m²)**
Decentralized	350	10.26
Centralized	748	8.35

To provide comparisons between the centralized and the decentralized controllers obtained by the proposed approach, we illustrate in Table 5 the total computational time (in min) to train the controllers and the averaged cumulative square error over time and all vehicles $\frac{1}{N}\frac{1}{T}\sum_{i=1}^{N}\sum_{k=1}^{T} e_i^2(k)$. First, the table shows that the control performance for the centralized case is superior to the decentralized case. This superiority stems from the centralized controller's ability to leverage state information from all vehicles for generating control inputs, as opposed to the decentralized controller, which only accesses state information from a select number of preceding vehicles. Conversely, decentralized controllers demonstrate a reduction in computational time as it requires lower dimensional inputs (states) and outputs (control inputs), as well as the employment of transfer learning, thereby offering better scalability in training compared to the centralized approach.



(a) Trajectories (b) Cumulative error

FIGURE 9 | Simulation results by applying the learned decentralized controller equipped with LSTM. (a) Velocity, error and distance of the control outcomes with LSTM. (b) Average of cumulative errors $\frac{1}{T}\sum_{k=1}^{T} e_i^2(k)$ against the vehicle index i.

5 | Conclusion and Future Work

In this study, we introduce a new machine learning-based controller for managing vehicle platoons. Our goal is to achieve string stability and accurately reflect the temporal dynamics of the leading vehicle. To this end, we design a controller based on LSTM neural networks, with a loss function that minimizes the cumulative error differences between adjacent vehicles. Moreover, we explore the development of two types of platoon controllers—centralized and decentralized—to address and analyse the corresponding performance and computational complexity. By conducting a comparative analysis of these control strategies through numerical simulations, particularly comparing scenarios with and without the LSTM integration, we demonstrate the superior performance of the LSTM-based controller.

The limitation of this study lies in its reliance on synthetic simulation data, which may not fully capture the complexities of real-world driving behaviours, such as vehicle interactions at intersections, effects of downhill grades, traffic-light starts and mixed-autonomy environments. Additionally, future research will aim to validate the proposed controllers using real-world driving datasets and to enhance the framework by incorporating advanced temporal models, such as attention mechanisms or bidirectional LSTMs (bi-LSTMs). Furthermore, exploring hybrid approaches that integrate LSTM networks with traditional control-theoretic guarantees represents a promising direction for future work. While we focus on standard LSTM networks for their proven ability to capture temporal dependencies, more advanced recurrent architectures such as bi-LSTM and attention-based models could further enhance predictive accuracy by leveraging both past and future state information. Exploring these extensions is left for future work.

Author Contributions

Ryota Nakai: conceptualization, investigation, methodology, validation, visualization, writing – original draft. **Kazumune Hashimoto:** investigation, methodology, supervision, funding acquisition. **Xun Shen:** supervision, writing – review and editing. **Shigemasa Takai:** supervision, writing – review and editing.

Acknowledgements

This work was supported by JST CREST JPMJCR201, JST ACT-X JPMJAX23CK, JSPS KAKENHI Grants 25K07794 and 22KK0155.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Research data beyond the contents of the article are not shared.

References

1. T. Tank and J. Linnartz, "Vehicle-to-Vehicle Communications for AVCS Platooning," *IEEE Transactions on Vehicular Technology* 46, no. 2 (1997): 528–536.

2. I. Sen and D. W. Matolak, "Vehicle-Vehicle Channel Models for the 5-GHz Band," *IEEE Transactions on Intelligent Transportation Systems* 9, no. 2 (2008): 235–245.

3. K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three Decades of Driver Assistance Systems: Review and Future Perspectives," *IEEE Intelligent Transportation Systems Magazine* 6, no. 4 (2014): 6–22.

4. A. A. Alam, A. Gattami, and K. H. Johansson, "An Experimental Study on the Fuel Reduction Potential of Heavy Duty Vehicle Platooning," in *13th International IEEE Conference on Intelligent Transportation Systems* (IEEE, 2010), 306–311.

5. E. Coelingh and S. Solyom, "All Aboard the Robotic Road Train," *IEEE Spectrum* 49, no. 11 (2012): 34–39.

6. K.-Y. Liang, J. Martensson, and K. H. Johansson, "Heavy-Duty Vehicle Platoon Formation for Fuel Efficiency," *IEEE Transactions on Intelligent Transportation Systems* 17, no. 4 (2016): 1051–1061.

7. Z. Wang, Y. Gao, C. Fang, L. Liu, S. Guo, and P. Li, "Optimal Connected Cruise Control With Arbitrary Communication Delays," *IEEE Systems Journal* 14, no. 2 (2020): 2913–2924.

8. K. Santhanakrishnan and R. Rajamani, "On Spacing Policies for Highway Vehicle Automation," *IEEE Transactions on Intelligent Transportation Systems* 4, no. 12 (2003): 198–204.

9. M. D. Bernardo, A. Salvi, and S. Santini, "Distributed Consensus Strategy for Platooning of Vehicles in the Presence of Time-Varying Heterogeneous Communication Delays," *IEEE Transactions on Intelligent Transportation Systems* 16 (2015): 102–112.

10. A. Petrillo, A. Salvi, S. Santini, and A. S. Valente, "Adaptive Multi-Agents synchronization for Collaborative Driving of Autonomous Vehicles With Multiple Communication Delays," *Transportation Research Part C: Emerging Technologies* 86 (2018): 372–392.

11. D. Yanakiev and L. Kanellakopoulos, "Nonlinear Spacing Policies for Automated Heavy-Duty Vehicles," *IEEE Transactions on Vehicular Technology* 47 (1998): 1365–1377.

12. J. Chen, H. Liang, J. Li, and Z. Lv, "Connected Automated Vehicle Platoon Control With Input Saturation and Variable Time Headway Strategy," *IEEE Transactions on Intelligent Transportation Systems* 22, no. 8 (2021): 4929–4940.

13. J. Chen, Y. Zhou, and H. Liang, "Effects of ACC and CACC Vehicles on Traffic Flow Based on an Improved Variable Time Headway Spacing Strategy," *IET Intelligent Transport Systems* 13 (2019): 1365–1373.

14. L. E. Peppard, "String Stability of Relative-Motion PID Vehicle Control Systems," *IEEE Transactions on Automatic Control* 19, no. 5 (1974): 579–581.

15. G. J. L. Naus, R. P. A. Vugts, J. Ploeg, M. J. G. van de Molengraft, and M. Steinbuch, "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach," *IEEE Transactions on Vehicular Technology* 59, no. 9 (2010): 4268–4279.

16. G. J. Naus, R. P. Vugts, J. Ploeg, M. J. V. D. Molengraft, and M. Steinbuch, "String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach," *IEEE Transactions on Vehicular Technology* 59, no. 11 (2010): 4268–4279.

17. D. Swaroop and J. Hedrick, "String Stability of Interconnected Systems," *IEEE Transactions on Automatic Control* 41, no. 3 (1996): 349–357.

18. Y. Liu, H. Gao, C. Zhai, and W. Xie, "Internal Stability and String Stability of Connected Vehicle Systems With Time Delays," *IEEE Transactions on Intelligent Transportation Systems* 22, no. 10 (2021): 6162–6174.

19. Z. Qiang, L. Dai, B. Chen, and Y. Xia, "Distributed Model Predictive Control for Heterogeneous Vehicle Platoon With Inter-Vehicular Spacing Constraints," *IEEE Transactions on Intelligent Transportation Systems* 24, no. 3 (2023): 3339–3351.

20. Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed Model Predictive Control for Heterogeneous Vehicle Platoons Under Unidirectional Topologies," *IEEE Transactions on Control Systems Technology* 25, no. 3 (2017): 899–910.

21. S. Graffione, C. Bersani, R. Sacile, and E. Zero, "Model Predictive Control of a Vehicle Platoon," in *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)* (IEEE, 2020), 513–518.

22. P. Wang, H. Deng, J. Zhang, L. Wang, M. Zhang, and Y. Li, "Model Predictive Control for Connected Vehicle Platoon Under Switching Communication Topology," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 7 (2022): 7817–7830.

23. S. Feng, Y. Zhang, S. E. Li, Z. Cao, H. X. Liu, and L. Li, "String Stability for Vehicular Platoon Control: Definitions and Analysis Methods," *Annual Reviews in Control* 47 (2019): 81–97.

24. A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," *Physica D: Nonlinear Phenomena* 404 (2020): 132306.

25. S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering* 22, no. 10 (2010): 1345–1359.

26. Y. Lin, P. Wang, Y. Zhou, F. Ding, C. Wang, and H. Tan, "Platoon Trajectories Generation: A Unidirectional Interconnected LSTM-Based Car-Following Model," *IEEE Transactions on Intelligent Transportation Systems* 23, no. 3 (2022): 2071–2081.

27. M. Zhou, X. Qu, and X. Li, "A Recurrent Neural Network Based Microscopic Car Following Model to Predict Traffic Oscillation," *Transportation Research Part C: Emerging Technologies* 84 (2017): 245–264.

28. Y. Liu, D. Yao, H. Li, and R. Lu, "Distributed Cooperative Compound Tracking Control for a Platoon of Vehicles With Adaptive NN," *IEEE Transactions on Cybernetics* 52, no. 7 (2022): 7039–7048.

29. A. Almatar, "Increasing Electric Vehicles Infrastructure in Urban Areas for Efficiently Employing Renewable Energy," *Environment, Development and Sustainability* 26, no. 10 (2024): 26183–26204.

30. R. Ahasan, M. N. Hoda, M. S. Alam, Y. R. Nirzhar, and A. Kabir, "Changing Institutional Landscape and Transportation Development in Dhaka, Bangladesh," *Heliyon* 9, no. 7 (2023): e17887.

31. K. M. Almatar, "Transit-Oriented Development in Saudi Arabia: Riyadh as a Case Study," *Sustainability* 14, no. 23 (2022): 16129.

32. K. M. Almatar, "Traffic Congestion Patterns in the Urban Road Network: Dammam City," *Ain Shams Engineering Journal* 14, no. 3 (2022): 101886.

33. K. M. Almatar, "Investigating the New Public Transportation System in Riyadh City," *Journal of Urban Planning and Development* 150, no. 1 (2024): 05023051.

34. K. M. Almatar, "Smart Transportation Planning and Its Challenges in the Kingdom of Saudi Arabia," *Sustainable Futures* 8 (2024): 100238.

35. D. Swaroop, J. Hedrick, C. C. Chien, and P. Ioannou, "A Comparison of Spacing and Headway Control Laws for Automatically Controlled Vehicles," *Vehicle System Dynamics* 23, no. 1 (1994): 597–625.