Multi-Agent Reinforcement Learning for Cooperative Adaptive Cruise Control

Ashley Peake*, Joe McCalmon*, Benjamin Raiford, Tongtong Liu, Sarra Alqahtani
Computer Science Department
Wake Forest University
Winston-Salem, NC, USA
{peakaa19, mccajl18, raifbh18, liut18, Sarra-alqahtani}@wfu.edu

***Abstract*— A growing trend in the field of autonomous vehicles is the use of platooning. The design of control algorithms for platoons is challenging considering that coordination among vehicles is obtained through diverse communication channels. Currently, Adaptive Cruise Control (ACC) is used in individual vehicles to regulate certain driving functions. ACC can be extended to leverage inter-vehicle communication, creating a tightly coupled vehicle stream in the form of a platoon. This extension, Cooperative Adaptive Cruise Control (CACC), typically assumes full communication among vehicles. In this paper, we develop a deep reinforcement learning based CACC that allows platooning vehicles to learn a robust communication protocol alongside their coordination policies. LSTM is used to implement ACC for each vehicle and is trained using policy gradient. To coordinate driving, the vehicle's LSTM adapts to exchange relevant information with the other vehicles, creating the CACC. We simulate two platoons of 3 and 5 vehicles, respectively. We test our CACC with the learned communication protocol against full and inhibited communication baselines with and without a jamming attack. We also train our approach with local and global reward systems. Results suggest that models with individual rewards and the learned communication protocol achieve higher performance and faster convergence.**

Keywords; autonomous vehicles, reinforcement learning, coordination, jamming, LSTM

I. INTRODUCTION

Autonomous Vehicles (AVs) are one of a range of applications and concepts within the field of Intelligent Transportation Systems (ITS) [1]. To operate as truly autonomous ITSs, AVs must be able to process a large volume of data collected via sensors and communication links. These links can effectively construct a self-organized group of close-following AVs, otherwise known as a platoon. Platooned vehicles maintain a smaller headway compared to normal vehicles with the same speed, which improves traffic throughput as well as homogeneity [2]. Many modern vehicles are already equipped with Adaptive Cruise Control (ACC), i.e., a radar-based system which automatically maintains a safe distance from surrounding vehicles. However, ACC alone does not improve road traffic efficiency as we will show in our experiment. For this reason, projects like PATH and SARTRE have developed Cooperative Adaptive Cruise Control (CACC) [3, 4] as an extension to ACC. CACC leverages inter-vehicle communication to create a tightly coupled vehicle stream achieving the platooning objective of string stability [5]. String stability is the attenuation of perturbations introduced by an arbitrary vehicle in the platoon along the string in upstream direction to keep the inter-vehicles distance as small as required.

Developing CACC algorithms is a complex task. Each AV must form an individual, local driving plan that is adapted through coordination with the other AVs to avoid potential collisions and maintain the string stability. Cooperative driving and safe operation are among the most challenging tasks for platoons, particularly since delays and/or loss of information from communication impairments can result in poor performance. Often, reliable communication is simply assumed when CACC is developed for platoon applications [6,7]. The impact of communication impairments on CACC coordination is not considered, or considered only partially [4], in most current works in the field.

Recently, Reinforcement Learning (RL) has had rapid and significant progress in the development of autonomous driving systems [8-11]. Majority of this research assumes full communication among vehicles. This assumption is problematic as it could lead to safety and robustness concerns if communication is disrupted. Moreover, the previous research focus on platoons of only 2 vehicles (i.e. leader-follower architecture), which are too simplistic for practical application. In this paper, we propose a platooning architecture using multi-agent reinforcement learning MARL-CACC in which cooperative vehicles learn a communication protocol alongside their coordination policies. Instead of simply adhering to a predetermined communication protocol, the specification and format of the communication in the MARL-CACC is not predetermined, but the vehicles collaboratively learn when and what to communicate.

In MARL-CACC, we develop each vehicle's ACC using a policy gradient RL with an LSTM network. During training, the LSTM has access to a continuous communication channel to enable vehicles to exchange information about their current states. The LSTM has a built-in "forget gate" to keep track of the important information and forget the rest to better utilize the communication bandwidth. Because the communication between vehicles' controllers is continuous, the model is trained via back-propagation. We test our approach using two simulated platoons of 3 and 5 vehicles with three different communication protocols: our learned protocol, full communication, and inhibited communication. To study the limitations of multi-agent credit assignment in our approach, we design two different reward functions: a global average reward shared among all vehicles versus an individual reward for each vehicle. The results show that models with individual

*Equal contribution.

2375-0197/20/$31.00 ©2020 IEEE
DOI 10.1109/ICTAI50040.2020.00013

rewards outperform all other models in terms of convergence and trajectory length, regardless of the communication setting. However, the results also suggest using the learned protocol improves the string stability of the platoons. We also analyze the performance of our MARL-CACC under a simulated jamming attack. In this scenario, the learned communication protocol increases robustness and helps the vehicles avoid collisions, compared to a full communication model.

We make the following contributions:

1. We develop Multi-Agent Reinforcement Learning Cooperative Adaptive Cruise Control Model (MARL-CACC) for autonomous driving in platoons.
2. We empirically show that in the proposed MARL-CACC the vehicles learn when and what to communicate in order to improve the string stability of their cooperative driving in the platoon.
3. We conduct experiments on platoons with 2 different sizes under a variety of communication settings, reward systems, and under a communication jamming attack. We show that MARL-CACC with the learned communication protocol outperforms the baselines with performance gaps that increase with scale. The results also show that individual rewards converge faster and better than global rewards.

The remainder of this paper is organized as follows. The next section further discusses related work in autonomous platoons. In section III, we provide general overviews of MARL and LSTM. Section IV formulates the problem, and the proposed approach is introduced in Section V. Finally, Section VI presents the experiments and discusses the results.

II. Related work

The idea of using platoons to improve traffic was originally proposed in [13] by PATH for *Intelligent Vehicle Highway System (IVHS)*. The control tasks of their system are organized in a five-layer hierarchy. Physical, regulation, and coordination layers are distributed among controllers on each vehicle, whereas link and network layers control groups of vehicles. Our proposed CACC approach resides in the coordination layer.

Michaud et al. [14] discuss different coordination strategies for automated vehicles in platoons, primarily focusing on communication patterns between vehicles in either a centralized or a decentralized fashion. However, they do not consider some important aspects of platoon control such as ensuring string stability. Segata et al. [3] develop an integrated simulator called PLEXE for studying strategies and protocols in platooning scenarios. This is the first attempt at designing a high-level platoon management protocol leveraging wireless vehicle-to-vehicle (V2V) communication with IEEE 802.11p in VANET-enabled vehicles.

With the development of deep learning, the domain of RL has become a powerful learning framework for autonomous driving. To our knowledge, Yu [8] was the first researcher to suggest using RL for steering control. His approach uses neural networks to analyze the vision sensor input and generate the steering control while maintaining vehicle movement within road boundaries. This method uses the RL model both to eliminate the need for external supervision and to provide the system with continuous learning ability similar to human driving practice. RL has also been used by Se-Young et al to investigate road-following [9]. Through RL, the control system indirectly learns the vehicle–road interaction dynamics, the knowledge of which is essential to staying on the road in high-speed road tracking.

In another attempt, Ng *et al.* [10] propose an adaptive control system using gain scheduling learned by RL. The proposed controller performs well on a one vehicle system, but when it is deployed in a platoon, the performance becomes less smooth. In particular, as the second car attempts to track the leader, slight oscillations result. This oscillation is passed onto the subsequent vehicle, but for the vehicles at the end of the platoon, the oscillations do decrease, implying stability. Nevertheless, our approach is more convenient for platoon control as it does not engender such high oscillations.

In his work, Pendrith [11] presents a distributed Q-Learning (DQL) framework applied to a lane change advisory system. His approach uses a local perspective representation state, which represents the relative velocities of the surrounding vehicles. Unlike our algorithms, DQL does not consider the actual actions of the proximate vehicles, which eventually results in a lack of learning stability.

In [15], a policy iteration method is utilized to learn parameters of a classical proportional-integral (PI) controller instead of direct longitudinal control. Researchers in [16], propose an informative reward design to ensure the safety and robustness of the Q-learning method applied to ACC. The proposed approach in [17] applies deep deterministic policy gradient (DDPG) to learn the continuous longitudinal control with predicted leading vehicle trajectories. Most of these works consider a predecessor-following problem in a two-vehicle system with full communication rather than a multi-vehicle setting like a platoon.

Although some attempts have been directed toward CACC and ACC using RL, no research has yet used RL for learning the communication protocols between vehicles in CACC. This paper attempts to fill this gap by developing a coordination approach that learns a communication protocol to enhance the stability of the platoon. We also present a detailed study of different communication protocols and their effects on our approach.

III. System model

A. ACC Model and Vehicle Dynamics

In this paper, we define the platoon as a set of connected autonomous vehicles N, traveling on a single lane freeway. We use Markov decision processes (MDP) [23] to model the ACC decision making process for each vehicle. An MDP is represented by $M = (S, A, P, \gamma, r)$, where S is the set of the possible states for the vehicle, A is the set of actions that can be taken by the vehicle, P is the transition probability function modeling the subsequent state after taking a particular action in a particular state, γ is a discount factor that ranges from [0,1], and r is the reward function $r: S \times A \rightarrow R$. A policy $\pi: S \rightarrow A$ is a mapping from states to actions. An optimal policy π^* is one such that $\sum_{t=0}^{N} \gamma^t r(s_t, a_t)$ is

maximized, where the MDP has a finite horizon of N steps and $\pi^*(s_t) = a_t$. The states S include the positions that the vehicle can be in during the trajectory while the actions A are; maintain the speed, accelerate, decelerate, and brake.

Given a vehicle i ∈ V, we denote its headway, i.e., bumper-to-bumper distance between i and its preceding vehicle i -1, by d_i, its velocity by v_i and its acceleration by u_i. The vehicle dynamics are given by [19],

$$\dot{d}_i = v_{i-1} - v_i \quad (1)$$

$$\dot{v}_i = u_i \quad (2)$$

Which we discretize as

$$d_{i,t+1} = d_{i,t} + \int_t^{t+\Delta t} (v_{i-1,\tau} - v_{i,\tau}) d\tau \quad (3)$$

where Δt represents the time sampling step.

The string stability constraint is defined for each vehicle as:

$$d_{min} = d_{i,t}$$

where d_{min} represents the preferred distance gap between any two successive vehicles. Speed, acceleration, and deceleration are constrained by:

$$0 \le v_{i,t} \le v_{max} \quad (4)$$

$$u_{min} \le u_{i,t} \le u_{max} \quad (5)$$

B. CACC Model

We consider the Markov stochastic games which are known in AI community as *multi-agent MDPs (MMDPs)* to design the MARL-CACC. In particular, we focus on partially observable Markov games [19]. We design our MARL-CACC as a Markov game for *N* vehicles with a set of states *S* describing the possible positions of all vehicles, a set of actions $\mathcal{A}_1 \dots \mathcal{A}_n$, and a set of observations $\mathcal{O}_1 \dots \mathcal{O}_n$ for each vehicle. To choose actions, each vehicle *i* uses a stochastic policy $\pi_{\theta_i}: \mathcal{O}_i \times \mathcal{A}_i \mapsto [0,1]$, which produces the next state according to the state transition function $\mathcal{T}: S \times \mathcal{A}_1 \times \dots \times \mathcal{A}_n \mapsto S$. Each vehicle *i* obtains rewards as a function of the state and vehicle's action $r_i: S \times \mathcal{A}_i \mapsto \mathbb{R}$, and receives a private observation correlated with the state $o_i: S \mapsto \mathcal{O}_i$. Each vehicle *i* aims to maximize its own total expected return $R_i = \sum_{t=0}^{T} \gamma^t r_i^t$ where γ is a discount factor and *T* is the trajectory time for the platoon. Without communication, each vehicle can act only based on local observations, reducing the CACC model to the individual ACC model. In this paper, we assume vehicles can communicate either with all vehicles in the platoon via the broadcasting channel or with selected vehicles via the V2V channel.

To maximize the expected return for each vehicle, we use a policy-based model free method where the policy for each vehicle $\pi_{\theta_i}\left({a_i}^t | {s_i}^t; {\theta_i}^t\right)$ is directly parameterized and the parameter θ_i is updated by performing, typically approximate, gradient ascent on $\mathbb{E}[R_i]$. We particularly use the REINFORCE algorithm [20]. Standard REINFORCE updates the policy parameters θ_i in the direction $\nabla_{\theta_i} \log \pi\left({a_i}^t | {s_i}^t; {\theta_i}^t\right) {R_i}^t$, which is an unbiased estimate of $\nabla_{\theta_i} \mathbb{E}[{R_i}^t]$.

The details of the REINFORCE algorithm are shown:

function REINFORCE
  Initialise θ arbitrarily
  for each episode $\{s_1, a_1, r_2, \dots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**
    for $t = 1$ to $T - 1$ **do**
      $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$
    end for
  end for
  return θ
end function

IV. MARL-CACC for Autonomous Driving In Platoons

In this section, we develop our MARL-CACC model that facilitates the cooperative driving of platoons. We develop an LSTM for each vehicle's ACC controller that coordinates with its neighbors to dynamically form the platoon's CACC.

A. LSTM-Based ACC Controller

The ACC for each vehicle is developed using an LSTM [21]. A single LSTM cell consists of three gates: an input gate, an output gate, and a forget gate. The cell itself has a state which can be used to remember information. Each gate alters the cell state by way of weights and transfer functions. These weights are updated using backpropagation. Refer to [22] for more details about the update rules, and the equations for the update of the cell state.

In this work, we use the cell s_t to represent the vehicle's state (i.e. position) in the freeway at time *t*. The input to the cell is the observation *obs* from the vehicle and the output is the expected next position for the vehicle. The forget gate controls the propagation of information from the previous state to the current state. This mechanism eventually helps the vehicle learning to communicate only important information to other vehicles. The structure of the LSTM for ACC controller is depicted in Fig.1. At any timestep *t*, the equations below describe the internal structure of the LSTM-ACC:

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ c_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} \begin{pmatrix} w_{obsi} obs_t + w_{si} h_{t-1} + b_i \\ w_{obsf} obs_t + w_{sf} h_{t-1} + b_f \\ w_{obso} obs_t + w_{so} h_{t-1} + b_o \\ w_{obsc} obs_t + w_{sc} h_{t-1} + b_c \end{pmatrix} \quad (6)$$

$$s_t = f_t \odot s_{t-1} + i_t \odot c_t \quad (7)$$

$$h_t = o_t \odot tanh\,(s_t) \quad (8)$$

where obs_t is the input to the LSTM block; i_t, f_t, o_t, s_t and h_t are the input gate, the forget gate, the output gate, the cell state, and the hidden state respectively. c_t is a vector of the external new information that is a candidate for addition to the next cell, as shown in Fig.1. We use c_t to represent the communication vector that each vehicle exchanges with other vehicles and we call it the communication gate. $w_{obsi}, w_{obsf}, w_{obsc}, and\ w_{obso}$ are the weights between the input gate, the forget gate, the communication gate, and the output gate respectively. $w_{si}, w_{sf}, w_{sc}, and\ w_{so}$ are the weights between the cell state and the input gate, the forget gate, the communication gate, and the output gate respectively. Finally, $b_i, b_f, b_c, and\ b_o$ are the additive biases of the input gate, the forget gate, the communication gate, and the output gate, respectively. The sigmoid function $\sigma(.)$ and the hyperbolic activation function *tanh*(.) are used as activation functions. In equation (7) and (8), the cell state s_t,

and the output of the LSTM block, h_t, are calculated using the outputs from the gates in equation (6), where ⊙ denotes and element-wise multiplication.

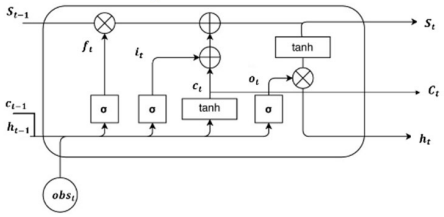

Figure 1: The structure of LSTM for ACC controller

The decision about the actions for each vehicle is modelled as an MDP and solved using the policy gradient RL in conjunction with LSTM [25,26] as in Fig. 2. The LSTM takes in at time *t* one input vector obs_t, the hidden state h_{t-1}, the communication vector from other vehicles c_{t-1}, and the previous state s_{t-1}. Then, it generates one feature output vector h_t for its hidden state, and the next state s_t. When the vehicle drives individually using its ACC, the LSTM will generate an empty communication vector c_t. For the *i*-th vehicle, its policy takes the following form:

$$h_t^i, s_t^i, c_t^i = LSTM(enc(obs_t^i), c_{t-1}^i, h_{t-1}^i, s_{t-1}^i) \quad (9)$$

$$a_t^i = \pi(h_t^i) \quad (10)$$

where c is an encoder function parameterized by a fully connected neural network as in [12], and π is the vehicle action policy.

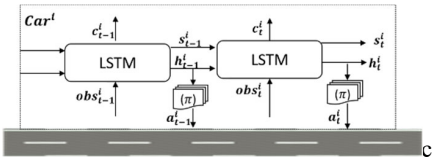

Figure 2: LSTM-based ACC for car *i* in 2 time steps

B. MARL- CACC

In order to coordinate the actions of the platooning vehicles in the MARL-CACC, we simply extend the independent ACC controller (equations 9,10), allowing vehicles to communicate their internal states. In our work, all vehicles use the same LSTM model, sharing parameters, which makes the MARL-CACC invariant to permutations of the vehicles and allows them to easily enter and leave the platoon. The equation (9) is extended as follows:

$$h_t^i, s_t^i, c_t^i = LSTM(enc(obs_t^i) + m_t^i, h_{t-1}^i, s_{t-1}^i) \quad (11)$$

where m_t^i is the communication vector received at each vehicle from other K vehicles computed as:

$$m_t^i = \frac{1}{K-1} M \sum_{k \neq i} c_{t-1}^k \quad (12)$$

M is a linear transformation matrix for transforming the average communication vector to a communication tensor, and K is the number of vehicles in the communication or vision ranges. A key point is that M is dynamically sized since the number of vehicles in the ranges varies at any point in time. This motivates the normalizing factor $\frac{1}{K-1}$ in equation (12), which rescales the communication tensor by the number of communicating vehicles. Fig.3 illustrates the details of our MARL-CACC approach.

I. Experiment Setup

This section details the experiments run to test our MARL-CACC system. This system integrates both sensors

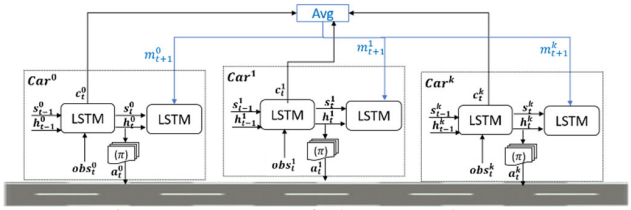

Figure 3: MARL-CACC for *k* cars over 2 timesteps

and inter-vehicle communication in a control loop to maintain secure, longitudinal, vehicle-following behavior. We here present the driving scenario, explain the communication protocols we test our system under, and describe the learning simulations. Through our experiments, we answer the following research questions:

RQ1: What is the impact of full communication via a broadcast channel compared to our learned comunication protocol via a V2V channel on MARL-CACC convergence and platoon's trajectory?

RQ2: What reward function structure, local or global, leads to more effective coordination of platoon's movement?

RQ3: Can ACC alone without any communication achieve stable strings in platoons?

A. Learning Task

The learning task considered in this work corresponds to a Stop&Go (SG) scenario. This scenario has been used by several researchers for the development of autonomous controllers and the assessment of their efficiency and effects on traffic flow [27,28]. In our simulation, we develop a platoon of 3 cars and another of 5 cars, each placed on a stretch of freeway of length 20 and 30 grid cells, respectively. Each individual car enters the freeway within 3 timesteps of the car preceding it. The leading vehicle starts at a velocity of 1 cell per timestep, takes an emergency brake, and then accelerates back to its initial cruising velocity as described in the scenario in [4]. The leading vehicle repeats this pattern until it reaches its destination, which is the end of the simulated freeway. The other vehicles must subsequently learn to follow the leading vehicle while keeping a desired headway of 3 cells.

B. Reward Design

The success of the RL agent depends on its reward function, as this function is used by the learning algorithm to direct the agent to areas of the state space where it can gather the maximum expected reward. As such, the reward function must be designed to appropriately encourage successful behavior. In this work, we consider rear-ended collisions, string stability, and traffic jams as metrics of platoon success (or lack thereof). Collision occurs when two vehicles are at the same location at the same time. Upon such a collision, we provide the vehicles involved in the collision with a negative reward r_{coll}. String stability is defined in terms of the inter-vehicle distance d_{min}, thus we provide a negative reward r_{string} when $d \neq d_{min}$. Finally, to reduce the travel time, we

provide a negative reward r_{time} on each time step to encourage the vehicles to keep moving forward. At timestep t, the individual reward for each vehicle (except the leading one which has hard-coded behavior) with C^t rear ended collisions and S^t error from d_{min} can be written as:

$$r(t) = r_{coll}C^t + r_{string}S^t + r_{time}t \quad (13)$$

The goal for each vehicle during training is to compute its policy π to maximize its reward r.

C. Training

To train the action policy π for each vehicle, we use the episodic REINFORCE algorithm as previously described. In our training architecture, we select the LSTM hyperparameters as shown in Table 1. Our model uses skip-connections [29]. Training was run on a 32-core CPU with 8 GB of memory and took between 22 to 55 hours based on the communication protocols and the size of the platoon as we will show.

Table 1: Parameters for LSTM model

Parameter	Value
Batch size	500
Epoch size	100 episodes
Kernel initializer	Uniform
Dropout rate	0.4
Learning rate	0.001
Hidden layer size	64
Activation function	Sigmoid
Optimizer	Adam

D. Baselines

We test our MARL- CACC approach under different reward and communication settings to answer the RQs. The models tested are:

Individual ACC (IACC): In this controller, an LSTM-based ACC model is applied individually to each car's observations to determine which action should be taken. As such, this model is essentially MARL-CACC without any communication. Each car solves its MDP using an LSTM as in Fig.2 and equations (9) and (10) in order to maximize its individual reward as defined in equation (13). This model partially adresses RQ3, and serves as the baseline for individual reward models for RQ2.

Global ACC (GACC): This model is equivalent to IACC, except that the cars are trained with a global average reward, instead of individual rewards. This global reward is defined as follows:

$$r_i(t) = \frac{\sum_{j=0}^{k}(r_{coll}C_j^t + r_{string}S_j^t + r_{time}T)}{k} \quad (14)$$

where k represents the total number of cars currently in the platoon. This model is designed to answer RQ2 and RQ3.

MARL-CACC with a Broadcast channel: This model represents a variation of our MARL-CACC approach such that vehicles communicate with each other and share their hidden states in a full and continous manner. Unlike our model, however, it is not modular and is inflexible with respect to communicaiton impairments, including the bandwidth. Each car maximizes its own individual reward as defined in equation (13). Platoon trajectories generated by this model help to answer RQ1 and RQ2.

MARL-CACC with a V2V communication channel: This model represents our learned communication approach such that each vehicle uses V2V channel to selectively communicate with other vehicles. In conjunction with the previous model, this model addresses primarily RQ1 and partially RQ2.

MARL-CACC-G with a Broadcast channel: This model is similar to MARL-CACC with a Broadcast channel, but it utilizes a global average reward as in equation (14) and similar to the model developed in [30]. *K* again represents the total number of cars in the platoon.

MARL-CACC-G with a V2V communication channel: This is similar to our approach but with a shared global reward. This model is designed to address RQ1 and RQ2.

II. Results and Discussion

A. Learning

Due to the stochastic nature of the policy gradient algorithms, each baseline model was executed 5 times. We present results from the most effective policies of these models. We show the mean success rate for each model in platoons of size 3 and 5 vehicles in Fig.4. The success rate represents the percentage of successful episodes in each epoch. We define "success" as all cars reaching the destination without any rear-ended collisions in the 100 timesteps of an episode.

Fig.4 clearly shows that the performance of models with individual rewards improves faster and ultimately reaches a higher level than that of models with a global reward. This finding begins to answer RQ2 and suggests that ACC and CACC models should be developed using an individual reward structure for safe platoons with fast convergence. We propose that this effect is primarily due to the credit assignment problem in multi-agent systems [31]. When operating based on a global reward, cars are given no feedback regarding their individual contribution (negative or positive) toward the reward of the overall platoon. Because of their rather limited ability to internalize the value of their actions, it is much more difficult for the cars to learn an optimal policy. As such, the models based on a global reward do not achieve near the success rate of those based on an individual reward.

Interestingly, in the platoon of 3 cars, we found that IACC converges faster than all other models. This is because the cars can individually, using only their local observations, avoid rear-ended collisions and reach the destination. There is no inherent need to coordinate or communicate with other vehicles to achieve the episode "success." However, the platoon trajectories generated by IACC do not achieve stable strings between the cars, as we will discuss later.

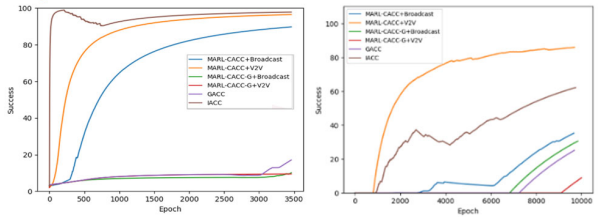

Figure 4: Mean success % for the most effective policy of each model in a platoon of 3 vehicles (left) and 5 vehicles (right)

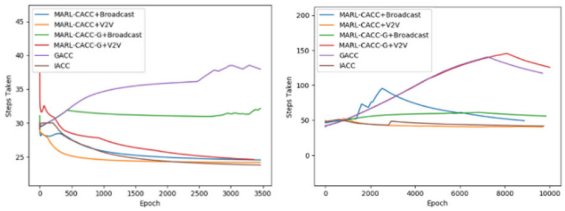

Figure 5: Average steps taken to complete an episode in a platoon with 3 vehicles (left) and 5 vehicles (right)

We now attempt to address the first part of RQ1 about the impact of communication protocol on model convergence. As shown in Fig.4, the learned communication over V2V improves the performance of the platoon compared to full communication and is scalable more than other communication protocols. This benefit is larger when coupled with an individual reward system, but it remains true even for models with a global reward. Therefore, we conclude that safe platoons do not need full communication. Instead, cars can efficiently and quickly learn what and when to communicate with each other in order to coordinate their driving decisions to successfully avoid rear-ended collisions.

To answer the second part of RQ1, how communication affects platoon trajectory, we consider the average number of steps takes each model to finish an episode (i.e. to safely reach the end of the freeway). Our results in Fig.5 suggest that communication only impacts the trajectory length when coupled with a global reward. In IACC, there is no need for a car to consider the actions of the other cars since its reward is local. Hence, the cars in the IACC-based platoon reach their destination relatively quickly. On the other hand, GACC, which also has no communication but uses a global reward, has a significantly longer trajectory. Similarly, we note that the both MARL-CACC-G models are slower than their local reward counterparts. Enabling communication via the broadcast channel does improve efficiency in a global reward model. In the 3-car platoon, the learned communication protocol improves it further still. Overall though, models with a local reward have shorter trajectory regardless of communication. Under a global reward, the cars learn to take the safest action (i.e. brake) more often, which leads to longer trajectories. Therefore, we conclude that a local reward, especially in conjunction with our learned communication protocol, can increase the efficiency of a platoon.

B. Testing

After training, we tested the policies generated for each model in a similar Stop&Go scenario. Following the convention in [4, 28], we quantify string instability using the root-mean-square error (RMSE) of the headways between cars. Since this metric is based on the size of the squared errors, it gives greater weight to values that are farther from the desired distance. Table 2 shows the average RMSE for the headways in platoons of 3 and 5 cars across all tested models. It is important to mention that we trained the three models with global rewards for an extra 5000 epochs to improve their convergence before testing.

Table 2 shows that both platoons are most stable using our MARL-CACC+V2V model with the learned communication.

Table 2: Headway Avg. RMSE values for the simulated platoons

	3-Cars	5-Cars
MARL-CACC+Broadcast	1.10	0.96
MARL-CACC+V2V	**0.54**	**0.69**
MARL-CACC-G+Broadcast	0.89	0.94
MARL-CACC-G+V2V	0.73	1.02
GACC	1.03	1.42
IACC	1.00	1.36

The 3-car platoon is least stable using the MARL-CACC+Broadcast,and the 5-car platoon using GACC. Since both the MARL-CACC+Broadcast and the MARL-CACC+V2V models use individual rewards, this drastic difference suggests that the communication setting plays a major role in determining string stability. In both platoons, the MARL-CACC+V2V model achieves better stability with less headway error compared to models with full communication. We propose that this is because cars communicating via broadcast are indiscriminately acting upon the information received from other cars, even when the sender is far away. In contrast, cars in MARL-CACC+V2V learn to only communicate the information that will help to improve their driving plans.

We attempt to understand what the cars learn to communicate in the MARL-CACC+V2V model by further analyzing the 5-car platoon. We start by recording the hidden state h_t^i of each car and the corresponding communication vectors c_t^i which represents the information that the car i sends to the hidden state of other cars at timestep t. Fig. 6 (top and bottom) shows the principle component analysis (PCA) of the communication and hidden state vectors respectively. Fig. 6 (top) shows a diverse range of hidden states, while (bottom) reveals far more clustered communication vectors, many of which are close to zero. This indicates that the two vectors carry different information, which suggests that the cars learn not to communicate all information in their hidden state unless necessary.

Fig.7-9 show the specific inter-vehicle distances over the first 17 and 29 timesteps for each model in 3-cars and 5-cars platoon scenarios, respectively. For reference, we define the optimal distance to be 3 cells. Interestingly, in both models with the learned communication protocol (Fig.7), we see that the headways between cars at the end of the string are generally more consistent than the ones at the beginning.

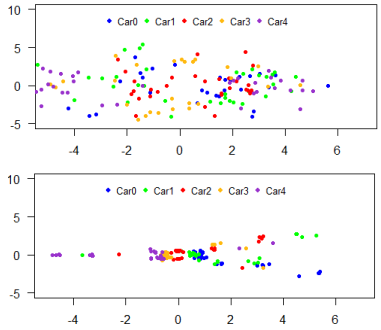

Figure 6: 2D PCA of hidden state vectors h (top) and communication vectors c (bottom)for MARL-CACC+V2V in the 5-car platoon

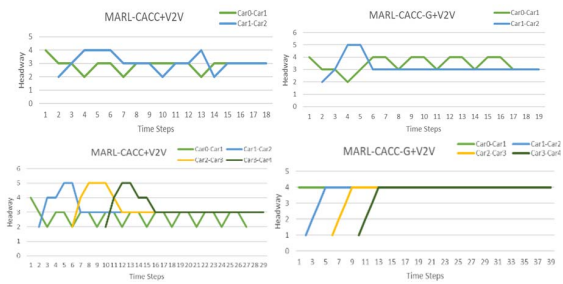

Figure 7: Headways in 3-car (top) and 5-car (bottom) platoons using MARL-CACC+V2V and MARL-CACC-G+V2V

This behavior indicates that the leader's accelerations and decelerations are not amplified down the line of the platoon. This result is promising as, in a large enough platoon, amplification could result in a complete halt of traffic flow. In the 5-cars platoon, the joining maneuver in MARL-CACC+V2V results in instability whenever a new car joins the platoon. Shortly after, however, the cars learn to keep the exact distance of 3 cells between one another (except the second car which is directly affected by the leader's braking). In MARL-CACC-G+V2V in the 5-cars platoon, cars learn to avoid collision by keeping 4 cells between each vehicle. This behavior explains why this model has a higher RMSE than its counterpart in the 3-car platoon.

Notably, full communication via the broadcast channel results in less stable platoons (Fig.8). Examining the headway trends in the 3-cars platoon for both the MARL-CACC and MARL-CACC-G models, it becomes apparent that, when the lead vehicle breaks, the second car reacts in an attempt to avoid collision. At the same time, we observe that the third car also abruptly reacts, as indicated by the second headway. Although the cars successfully avoid rear-end collisions, the distance error at the head is exacerbated at the last car, significantly decreasing the string-stability of the platoon. We propose that full communication between the cars is the source of this instability. The last car receives and incorporates information from the lead car, even if this information does not immediately affect it. Considering this, we conclude that communicated information enables the platoon to learn a better policy for string stability only when the cars coordinate in small groups and not necessarily when they coordinate as a one unit via a broadcast channel. This result once again emphasizes the importance of limiting the communication in platoons to efficiently achieve string stability. For the platoon with 5 cars, both models with full communication learn to keep a large, constant distance between vehicles to mitigate the impact of repetitive braking by the leader. Therefore, the RMSE of the headways in both models are relatively high, which ultimately defeats the purpose of using CACC instead of simply ACC.

The headways in the IACC model in both platoons (Fig.9 left) reveal that the leader's repeated braking has very little impact on other cars in the platoon. Since there is no communication between the cars, each vehicle simply learns to maintain a relatively large, constant distance from the vehicle ahead of it in order to avoid collisions. Although this strategy is effective to that end, it clearly reduces the string stability of the platoon.

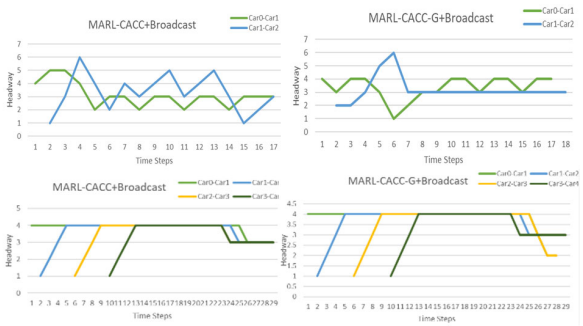

Figure 8: Headways in 3-car (top) and 5-car (bottom) platoons using MARL-CACC+Broadcast and MARL-CACC-G+Broadcast

On the other hand, the headways in the GACC models (Fig. 9 right) are more disturbed by the leader's behavior. Because all the cars share the same reward, they attempt to coordinate their actions. However, since communication is inhibited, the cars are unable to adequately stabilize their driving.

Figure 9: Headways in 3-car (top) and 5-car (bottom) platoons using IACC GACC models

C. Testing under Jamming Attack

We investigate the robustness of the learned communication protocol by testing MARL-CACC+V2V and MARL-CACC+Broadcast models on a freeway under a partial jamming attack. A channel jamming attack is a type of Denial of Service (DoS) attack that aims to block access to a communication channel by high power transmission on the channel or by injection of dummy messages. We simulated this attack by placing a simple, stationary, roadside jammer in a specific location on the freeway with a 6-cell range as described in [32]. We did not train the models on this environment to assess the difference between the robustness of the full communication and learned communication settings in sudden and dangerous situations. Additionally, we did not implement countermeasures against this attack to focus instead on developing a CACC approach that alleviates the risk of such attacks altogether. Our MARL-CACC+V2V model achieves this by learning only the necessary communication instead of constantly relying on information from each vehicle's hidden states.

The headways in the 5-cars platoon for both models are depicted in Fig. 10. The average RSME for MARL-CACC+Broadcast is 3.20 and, notably, cars 1, 2, and 3 all had rear-ended collisions (Fig. 10 left). In contrast, our learned communication approach in MARL-CACC+V2V has an average RMSE of only 1.07 without any collisions (right). Hence, we propose that the stability and safety of platooning vehicles can be increased by allowing them to learn how and

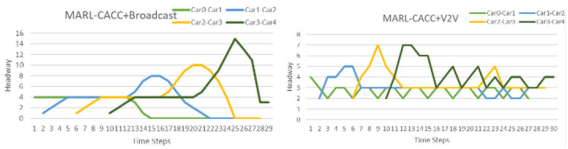

Figure 10: The impact of the jamming attack on the 5-cars platoon using MARL-CACC model

when to communicate with each other instead of continuously sharing their actions and observations at every time step.

III. Conclusion and Future Work

In this work, we developed a MARL approach to enhance the string stability of platoons of autonomous vehicles. The proposed approach uses LSTM and REINFORCE to coordinate driving in CACC controllers. We tested our approach under different communication protocols and two different reward function structures, individual and global. The results indicated that the model that uses individual rewards and the learned communication protocol via a V2V channel achieves the best convergence, stability, and shortest travel time. Our approach is thus practically significant as it does not require the assumption of full, stable communication among platoon vehicles as many contemporary models do. One limitation of this work is that the cars in some models have oscillatory behaviors due to a discrete state space. We plan to test our approach in a realistic, continuous state space in future work to address this oscillation.

References

[1] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon system," IEEE Transactions on Communications, 2019, p. 7907-7922

[2] E. Coelingh, and S. Solyom, "All aboard the robotic road train," IEEE Spectrum, 2012.

[3] M. Segata, F. Dressler, R.L Cigno, and M. Gerla, "A simulation tool for automated platooning in mixed highway scenarios," ACM SIGMOBILE Mobile Computing and Communications Review, 2012.

[4] M. Segata, et al. "Plexe: A platooning extension for Veins," IEEE Vehicular Networking Conference (VNC), 2014.

[5] Amoozadeh, M., et al., Platoon Management with Cooperative Adaptive Cruise Control Enabled by VANET, Vehicular Communications, 2015, p. 110-123.

[6] M. Segata, F. Dressler, R.L Cigno, Let's Talk in Groups: A Distributed Bursting Scheme for Cluster-Based Vehicular Applications, Vehicular Communications, 2017, Pages 2-12.

[7] G. Giordano, M. Segata, F. Blanchini, R.L. Cigno, The Joint Network/Control Design of Platooning Algorithms can Enforce Guaranteed Safety Constraints, Ad Hoc Networks, Volume 94, 2019.

[8] G. Yu, and I. K. Sethi, "Road-following with continuous learning," The Intelligent Vehicles Symposium, Detroit, 1995, pp. 412-417.

[9] S.Y. Oh, J.H Lee, and D.H. Choi "A new reinforcement learning vehicle control architecture for vision-based road following." IEEE Trans. Vehicular Technology 49 (2000), pp. 997-1005.

[10] L. Ng, C.M. Clark, and J.P Huissoon, Reinforcement Learning of Dynamic Collaborative Driving Part I: Longitudinal Adaptive Control. International Journal of Vehicle Information and Communication Systems, 2008.

[11] M.D. Pendrith, "Distributed reinforcement learning for a traffic engineering application," The fourth international conference on autonomous agents. 2000, p. 404–411.

[12] A. Singh, T. Jain, and S. Sukhbaatar, "Learning when to communicate at scale in multiagent cooperative and competitive tasks," International Conference on Learning Representations, 2019.

[13] M. Michaelian, and F. Browand, Field Experiments Demonstrate Fuel Savings for Close-Following, Institute of Transportation Studies, UC Berkeley, 2000.

[14] F. Michaud, et al., "Coordinated maneuvering of automated vehicles in platoons," IEEE Transactions on Intelligent Transportation Systems, 2006, p. 437-447.

[15] J. Wang, et al., "Self-learning cruise control using kernel-based least squares policy iteration," IEEE Transactions on Control Systems Technology, 2014, p. 1078-1087.

[16] Z. Li, T. Chu, I.V. Kolmanovsky, X. Yin, "Training drift counteraction optimal control policies using reinforcement learning: an adaptive cruise control example," IEEE Transactions on Intelligent Transportation Systems, 2018, p. 2903-2912.

[17] M. Buechel, and A. Knoll, Deep Reinforcement Learning for Predictive Longitudinal Control of Automated Vehicles. in 2018 21st International Conference on Intelligent Transportation Systems (ITSC). 2018.

[18] P. Xuan, V. Lesser, and S. Zilberstein, "Communication decisions in multi-agent cooperation: model and experiments," The International Conference on Autonomous Agents, 2001.

[19] M.L. Littman, "Markov games as a framework for multi-agent reinforcement learning," The 11th International Conference on Machine Learning (ICML'94), San Francisco, 1994, p. 157–163.

[20] R. J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning Journal, 1992, p 229–256.

[21] S. Hochreiter, and J. Schmidhuber. Long Short-Term Memory. Neural Computation, 1997, p. 1735–1780.

[22] F. A. Gers, J. Schmidhuber , and F. Cummins, "Learning to forget: continual prediction with LSTM," The 9th International Conference on Artificial Neural Networks (ICANN 99), 1999, pp. 850-855.

[23] M. Wiering, and M.V. Otterlo, Reinforcement Learning: State-of-the-Art. Springer-Verlag New York Incorporated, 2014.

[24] D.S. Bernstein, S. Zilberstein, and N. Immerman, "The complexity of decentralized control of Markov decision processes," The 16th conference on Uncertainty in artificial intelligence, 2002. p. 32–37.

[25] R.A. McCallum, Hidden State and Reinforcement Learning with Instance-Based State Identification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996, p. 464-473.

[26] B. Bakker, "Reinforcement learning with long short-term memory," The 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, 2001, p. 1475–1482.

[27] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "ACC+Stop&go maneuvers with throttle and brake fuzzy control," IEEE Transactions on Intelligent Transportation Systems, 2006, p. 213-225.

[28] C. Desjardins, and B. Chaib-draa, Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach, IEEE Transactions on Intelligent Transportation Systems, 2011, p. 1248-1260.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[30] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," The 30th International Conference on Neural Information Processing Systems, 2016, p. 2252–2260.

[31] D.T. Nguyen, A. Kumar, and H.C. Lau, "Credit assignment for collective multiagent RL with global rewards," The 32nd International Conference on Neural Information Processing Systems (NIPS'18), 2018, p.8113–8124.

[32] M. Amoozadeh et al., Security Vulnerabilities of Connected Vehicle Streams and Their Impact on Cooperative Driving, IEEE Communications Magazine, 2015, p. 126-132.