Optimized Cooperative Car-Following Through Lightweight Vehicle-to-Vehicle Intent Sharing

Hangyu Li[1,†], Juyoung Oh[2,†], Ke Ma[1,*], Zhaohui Liang[1], Peng Zhang[1], and Xiaopeng Li[1]

***Abstract*— Cooperative driving systems are expected to enhance safety, mobility, and efficiency through vehicle connectivity technologies. Lower-level vehicle-to-vehicle (V2V) communication transmits high-frequency status information, such as location, velocity, and acceleration, between vehicles. This approach contributes limitedly to prediction accuracy, requires high-frequency hardware, and is sensitive to communication delays. Recent studies have shown that intent sharing, which conveys planning trajectories, significantly improves prediction accuracy and control performance but requires higher bandwidth. However, mainstream vehicle communication methods struggle to balance cost and bandwidth for effective intent sharing. High-bandwidth wireless communication methods such as dedicated short-range communication (DSRC) and cellular vehicle-to-everything (C-V2X) cost much for devices, while low-cost visible light communication (VLC) can hardly support the necessary bandwidth. To address this challenge, we propose a lightweight intent sharing approach that reduces data transmission volume while maintaining prediction accuracy. Specifically, intended velocity trajectories are represented using regressed polynomial functions over a fixed time period, requiring only the transmission of polynomial coefficients and a timestamp for synchronization. The feasibility of this approach is demonstrated through simulations of car-following behavior using a Linear–Quadratic Regulator (LQR). Additionally, real vehicle experiments using a designated velocity cycle further validate the method. Results show that both planned and actual trajectories of the following vehicle closely align with those using ideal intent sharing approaches under significantly reduced communication data volume.**

I. Introduction

Although the integration of autonomous vehicles (AVs) into human-driven traffic remains controversial due to their unpredictable behavior and conservative strategies adopted to ensure safety [1], [2], cooperative driving is widely expected to enhance AV safety, mobility, and efficiency through vehicle connectivity technologies [3]. Among these, Cooperative Adaptive Cruise Control (CACC) plays a key role in improving AV car-following behavior by leveraging vehicle-to-vehicle (V2V) communication for longitudinal control [4]. Unlike conventional Adaptive Cruise Control (ACC), where the following AV relies solely on distance and velocity observations, CACC enables the following AV to receive information from the preceding vehicle, allowing it to achieve optimal spacing and velocity, thereby improving road capacity and reducing energy consumption [5], [6].

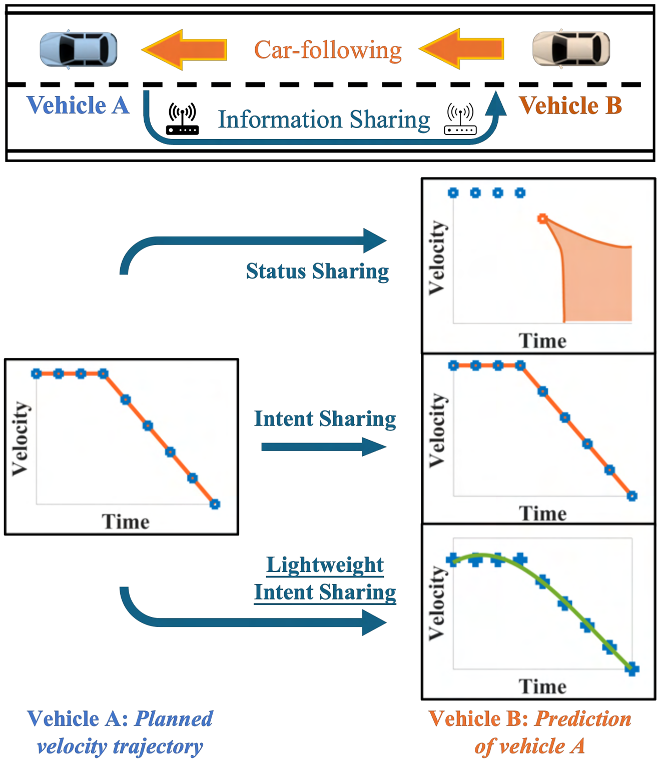

Car-following
Vehicle A
Information Sharing
Vehicle B
Status Sharing
Intent Sharing
Lightweight Intent Sharing
Velocity
Time
Vehicle A: Planned velocity trajectory
Vehicle B: Prediction of vehicle A

Fig. 1. Information sharing structure and V2V approaches for CACC. Our lightweight intent sharing approach significantly reduces data transmission volume while keeping the prediction accuracy of the preceding vehicle.

Early-stage CACC systems primarily rely on status sharing, where connected and automated vehicles (CAVs) exchange low-level status information such as location, velocity, and acceleration [7]. These data are used to estimate inter-vehicle distances and time headway, and therefore predict the preceding vehicle's future motion [8]. However, status sharing only enhances the following vehicle's perception rather than providing an explicit prediction on the preceding AV, leading to significant accuracy loss over time. Moreover, this method is highly dependent on real-time transmission without delays, making it difficult to achieve its ideal utilities under existing networks and hardware. Recent research has demonstrated that intent sharing, where CAVs transmit planned trajectories as intents, significantly improves safety, mobility, and efficiency compared to status sharing [9], [10], [11]. For CACC, the preceding vehicle transmits the planned trajectory (i.e., intent) to the following vehicle through the V2X network. The ego vehicle thus can optimize its long-term control to achieve smoother, safer,

†Equal contribution

*Corresponding author, email: kma62@wisc.edu

[1]Department of Civil and Environmental Engineering, University of Wisconsin-Madison, Madison, United States.

[2]Department of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, United States.

and more efficient car-following behavior [12]. In addition, intent sharing can also enhance the benefits of CAVs in conflicting traffic scenarios such as lane changes, mergings, and intersections [13].

Despite its advantages, intent sharing faces major challenges in balancing the cost and bandwidth of mainstream vehicle communication methods [14], [15]. High-bandwidth wireless communication methods include Dedicated Short-Range Communication (DSRC) and Cellular Vehicle-to-Everything (C-V2X) [16]. DSRC has been in development for over two decades and enables vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, but requires massive infrastructure investments. Furthermore, the complex process of matching intent information to specific surrounding vehicles introduces deception and cyberattacks, limiting large-scale deployment [17]. C-V2X is becoming more popular in a global manner, while PC5 mode (direct communication) suffers from similar security and matching concerns with DSRC. The other Uu mode (using cellular base stations) introduces high latency, making it challenging to support real-time intent sharing [18]. Additionally, both DSRC and C-V2X cost hundreds to thousands of dollars per unit with marginal benefits at low market penetration rates, posing a high barrier to widespread CAV adoption.

On the other hand, low-cost visible light communication (VLC) has been proposed as an alternative for V2V communications in CACC [19], [20], including our previous work [21]. VLC transmits dynamic information via an LED panel using binary encoding, which is then captured by the following vehicle's camera. Although VLC is limited to direct line-of-sight communication and cannot support long-range cooperation, field tests have shown that it is sufficient for over 99.9% of traffic scenarios [22]. Moreover, VLC enables simultaneous information sharing and vehicle matching, while being more resistant to wireless interference and cyberattacks. However, due to resolution constraints in camera-based recognition at standard car-following distances, VLC-based intent sharing is limited to transmitting only a few dozen binary bits per frame, making it inadequate for regular intent sharing.

To address this challenge, a new lightweight intent sharing approach is proposed in this paper to efficiently improve the cooperative car-following performance in terms of safety, mobility, and efficiency. Unlike regular intent sharing, which transmits complete planned velocity trajectories [23], our approach applies polynomial regression to encode velocity trajectories over fixed time intervals. This segmentation balances flexibility and accuracy, aligning with the motion prediction horizons used in model predictive control (MPC) methods [24]. By transmitting only the polynomial coefficients and a timestamp for time synchronization, our method enables CACC with low-frequency, low-rate communication, improving robustness to delays while maintaining high prediction accuracy of the preceding vehicle's trajectory. Figure 1 illustrates the overall structure of the proposed lightweight intent sharing framework, comparing it to status sharing and conventional intent sharing. Additionally, a qualitative comparison of different information-sharing approaches is presented in Table I.

In this study, we assume the following vehicle drives under a constant time headway policy [25]. After receiving the preceding vehicle's intended velocity trajectory, the following vehicle employs optimal control to generate its own velocity trajectory. We investigate the effectiveness of the proposed lightweight intent sharing approach in accurately representing the preceding vehicle's velocity trajectory and assess the optimality of the following vehicle's planned velocity trajectory using a Linear–Quadratic Regulator (LQR) control strategy. A predefined velocity cycle is used for the preceding vehicle in simulation experiments to evaluate our approach. Additionally, real-world vehicle experiments are conducted to account for practical communication, control, and environmental constraints. In these experiments, two CAVs are deployed, where the preceding AV follows a designated velocity cycle and transmits intent segments via V2V communication at fixed intervals. The following AV then generates its motion plan using LQR control, integrating the received intent information. Experiment results demonstrate that the proposed lightweight intent sharing approach substantially reduces data transmission volume while achieving cooperative driving performance comparable to regular intent sharing methods. The Normalized Euclidean Distance (NED) between the following vehicle's velocity trajectories obtained through regular and lightweight intent sharing is 0.002 m/s in simulation, 0.050 m/s for real vehicle planning, and 0.053 m/s for actual behavior.

The rest of this paper is organized as follows. Section II describes the car-following model and LQR used for CACC and provides theoretical validation for the lightweight intent sharing approach by polynomial regression. In Section III, we introduce our velocity cycle and verify the proposed lightweight intent sharing approach for CACC through simulation experiments. Then we demonstrate a real vehicle experiment conducted in Section IV to validate our lightweight intent sharing's feasibility in a two-CAV scenario. Finally, Section V concludes the paper.

TABLE I

COMPARISON OF DIFFERENT INFORMATION SHARING METHODS IN CACC SCENARIO

Approaches	Frequency	Data Volume	Delay Sensitivity	Prediction Accuracy
Status sharing	High	**Low**	High	Low
Intent sharing	**Low**	High	**Low**	**High**
(ours) Lightweight intent sharing	**Low**	**Low**	**Low**	**High**

II. Method

In this section, we will first state the car-following scenario as mathematical formulations. We then introduce the optimal control method for the following AV through LQR with intent sharing from the preceding AV. In addition, the polynomial regression for lightweight intent sharing will also be demonstrated and compared.

A. Car-Following Scenario

Prior to the specifics of the car-following control model, we first introduce definitions and notations for description.

Define $i \in \mathscr{I} := \{0, 1, 2, \ldots, I\}$ denoting the vehicle index, where I is the total number of car-following CAVs in a platoon. CAV with $i = 0$ refers to the lead vehicle, which is not in the car-following state.

As discrete systems in the real world, each CAV operates at consecutive time steps with a fixed interval τ, in line with the system frequency of CAVs from perception to control. The trajectory data at time t of vehicle i then corresponds to a state vector:

$$s_i(t) = [p_i(t), v_i(t)]^T . \tag{1}$$

Here, $p_i(t)$ and $v_i(t)$ denote the spatial location and the longitudinal velocity of vehicle i at time t, respectively.

We then define $d_i(t)$ as:

$$d_i(t) = p_{i-1}(t) - p_i(t), \quad \forall i \in \mathscr{I} \setminus 0, \tag{2}$$

to represent the spacing between the two consecutive vehicles $i-1$ and i at time t. Here $i \geq 1$ as the lead CAV is not in the car-following state.

Similarly, we have:

$$\Delta v_i(t) = v_{i-1}(t) - v_i(t), \quad \forall i \in \mathscr{I} \setminus 0, \tag{3}$$

which is defined as the relative velocity between two consecutive vehicles $i-1$ and i at time t.

Therefore,

$$[d_i(t), \Delta v_i(t)]^T = s_{i-1}(t) - s_i(t), \quad \forall i \in \mathscr{I} \setminus 0. \tag{4}$$

A corresponding response $a_i(t) = g\left((d_i(t), \Delta v_i(t), v_i(t)\right)$ is then given to each vehicle i at time t. Here, $g(\cdot)$ refers to an arbitrary control law denoting the desired acceleration between time steps based on the current states.

Similar to the widely adopted ACC/CACC model [7], [26], a linear control law [27] is given to the system:

$$a_i(t) = g^d\left(d_i(t) - d_s\right) + g^{\Delta v} \Delta v_i(t) + g^v v_i(t), \tag{5}$$

where d_s represents the minimum safe distance, related to velocity and vehicle length. Additionally, g^d, $g^{\Delta v}$, and g^v are three linear parameters forming the model.

We can then establish the state space function for the vehicle dynamics system according to its state vector $s_i(t)$ and response vector $u_i(t) = [v_i(t), a_i(t)]^T$:

$$s_i(t + \tau) = C s_i(t) + D u_i(t), \tag{6a}$$

with the output matrix C and the feedthrough matrix D are:

$$C = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & \tau \end{bmatrix}. \tag{6b}$$

B. Linear–Quadratic Regulator

As we have derived the general formulation, a specific simplified longitudinal car-following control model can be set in a two-CAV CACC scenario. In addition, the constant time headway (CTH) principle [25] is adopted to maintain a fixed time headway between two CAVs:

$$a(t) = g^d\left(d(t) - v(t)T - d_s\right) + g^{\Delta v} \Delta v(t), \tag{7}$$

where $a(t)$, $d(t)$, $v(t)$, and $\Delta v(t)$ represent the following CAV's acceleration, distance gap to the preceding vehicle, velocity, and velocity difference with the preceding CAV, respectively. Additionally, g^d and $g^{\Delta v}$ are two linear parameters similar to those in Equation (5). Furthermore, T is the safe time headway between two CAVs, and becomes the desired time headway if d_s is set to zero.

Assume a planned velocity trajectory is transmitted from the preceding vehicle to the following vehicle as intent sharing, denoted as $v_0^{\psi}(t), t \in \mathscr{T}_{\psi} := \{t_c, t_c + \tau, \ldots, t_c + \psi\}$. It consists of the planned velocity of the preceding vehicle at consecutive time steps from cooperating time t_c, with ψ representing the planning period. The following CAV can then utilize the preceding vehicle's planned velocity trajectory $v_0^{\psi}(t)$ to optimize its own velocity in order to follow the given trajectory[1].

With an initial distance gap being measured as d_c when cooperating, the expected distance gap between two CAVs at each time step can be derived as:

$$d(t) = d_c + \sum_{\theta = t_c}^{t} \left(v_0^{\psi}(\theta) - v(\theta)\right) \tau, \quad \forall t \in \mathscr{T}_{\psi}. \tag{8}$$

The state error $e_{\psi}(t)$ can then be derived as:

$$\begin{aligned} e(t) &= s(t) - s_0^{\psi}(t) - \left[v(t)T + d_s\right]^T , \\ &= \left[d(t) - v(t)T - d_s, v(t) - v_0^{\psi}(t)\right]^T , \quad \forall t \in \mathscr{T}_{\psi}, \end{aligned} \tag{9}$$

which shows the deviation from the ideal distance gap and velocity difference of the expected CACC process.

The LQR optimal control algorithm aims to optimize two linear control parameters g^d and $g^{\Delta v}$ to minimize a quadratic cost function over a given time period $\mathscr{T}_{\psi}$. The cost function is defined as:

$$J = \sum_{t \in \mathscr{T}_{\psi}} \left(\frac{1}{2} e(t)^T Q e(t) + \frac{1}{2} u(t)^T R u(t)\right), \tag{10}$$

where Q and R are the weighting matrices for the states and inputs $u(t) = [v(t), a(t)]^T$, respectively. These matrices are chosen to balance the trade-off between achieving the desired state and avoiding the rapid changes in vehicle motion.

[1] For the received intent information of the preceding vehicle, there are many other objectives that can be optimized for the following vehicle, such as safety, energy, comfort, etc. Here, we only consider following the trajectory of the preceding vehicle. For more information, please refer to our previous work [21] or other related studies.

C. Polynomial Regression

In a fixed time range $[t_c, t_c + \psi]$, the total data transmission volume required by regular intent sharing is about $\frac{\psi}{\tau}$ velocity data points, challenging the cost for bandwidth during communications between multiple vehicles in complex environments.

We therefore propose a lightweight communication approach for intent sharing, with modeling the velocity trajectory $v_0^{\psi}(t)$ as a polynomial function over time under the fixed period ψ.

Taking a cubic polynomial regression as an example, the cubic function for $v_0^{\psi}(t)$ is defined as:

$$v_0^{\psi}(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3 + \varepsilon(t), \qquad (11)$$

where β_0, β_1, β_2, and β_3 are the polynomial coefficients that need to be estimated, and $\varepsilon(t)$ represents the error term at each time step.

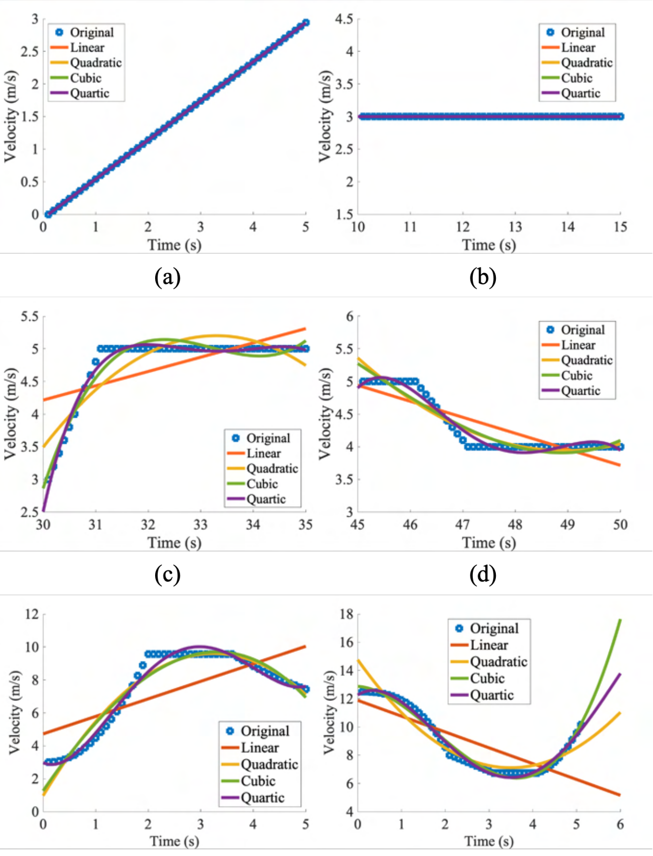
The estimation of the coefficients is performed by minimizing the sum of squared errors (SSE) between the planned velocities and those generated by the cubic polynomial function. This can be expressed as:

$$\min_{\beta_{1,2,3,4}} \sum_{t=t_c}^{t_c+\psi} \varepsilon(t)^2 = \left(v_0^{\psi}(t) - (\beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 t^3)\right)^2. \qquad (12)$$

After determining the coefficients, it is crucial to evaluate the model's performance of velocity predictions. Common metrics for this purpose include the coefficient of determination R-squared, and Root Mean Squared Error (RMSE). R-squared is used to measure the proportion of the variance in the velocity variable that is predictable from the independent variable (time). Also, RMSE provides a measure of the differences between velocity values predicted by the model and the velocity values actually observed.

Six typical scenarios are shown in Figure 2, including constant acceleration, constant velocity, linear velocity variation and nonlinear velocity variation, etc. We utilized polynomial regression to fit the velocity trajectories in such five-second pieces, from linear to quartic.

The performance metrics of these regressions, such as R-squared (R^2) and Root Mean Squared Error (RMSE), are recorded in Table II, demonstrating the accuracy of the corresponding method for conveying dynamic information in a simplified format.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 2. Six typical scenarios for polynomial regression. (a) Constant acceleration. (b) Constant velocity. (c) Linear acceleration and maintenance. (d) Linear deceleration between constant velocities. (e) Constant velocity between nonlinear changes. (f) Nonlinear velocity changes.

It can be seen from Figure 2 and Table II that all polynomial regression methods perform 100% well in constant acceleration and velocity scenarios (with linear velocity-time relationships). However, due to the lack of flexibility, it is difficult to describe complex dynamic processes for linear and quadratic regression methods, especially when accelerating first and then driving at a constant velocity. For more complex processes such as constant velocity, deceleration, and constant velocity again, the performance is actually better, partly because the continuous changes within a short period of five seconds create smaller differences endogenously. The R-squared value for linear regression in the third typical data piece is only 0.4068, with an RMSE

TABLE II

PERFORMANCE METRICS OF DIFFERENT POLYNOMIAL REGRESSIONS TO VELOCITY TRAJECTORIES.

Data piece	Linear		Quadratic		Cubic		Quartic	
	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE
(a) Constant acceleration	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(b) Constant velocity	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000	1.0000	0.0000
(c) Linear acceleration and maintenance	0.4068	0.3971	0.7695	0.2501	0.9459	0.1225	0.9831	0.0692
(d) Linear deceleration between constant velocities	0.7272	0.2256	0.9027	0.1361	0.9076	0.1341	0.9663	0.0818
(e) Constant velocity between nonlinear changes	0.4641	1.7164	0.9376	0.5919	0.9395	0.5891	0.9808	0.3350
(f) Nonlinear velocity changes	0.5822	1.4259	0.8970	0.7155	0.9827	0.2964	0.9888	0.2410

equal to 0.3971, showing significant deviations. Quadratic regression is better by a few, but the R-squared value of 0.7695 and the RMSE of 0.2501 are far from accurate representations of the original trajectory.

Despite the quality of the fit varying among pieces, the intended velocities and the trajectory fitted by the cubic and quartic polynomial function closely align for each piece. Cubic regression methods achieve R-square values above 0.90 for all of the six typical cases presented, demonstrating very good tracking performance. It is not beyond expectation that quartic regression performs better in all situations, achieving R-square values above 0.95 for all. However, its advantage requires an additional 33% number of coefficients for intent sharing data transmission.

III. Simulation

In the previous section, we demonstrated the data compression and intent reproduction capabilities of our lightweight intent sharing for the original velocity trajectories. In this section, we demonstrate our simulation experiments to verify the feasibility of the proposed lightweight intent sharing approach in the CACC scenario. We start from the introduction of a velocity cycle, then represent lightweight intent sharing by using cubic polynomial regression, followed by the performance analysis of our lightweight intent sharing and comparison with regular intent sharing.

A. Velocity Cycle

A velocity cycle will be first introduced, which serves as the base of our experiments, as shown in Figure 3.

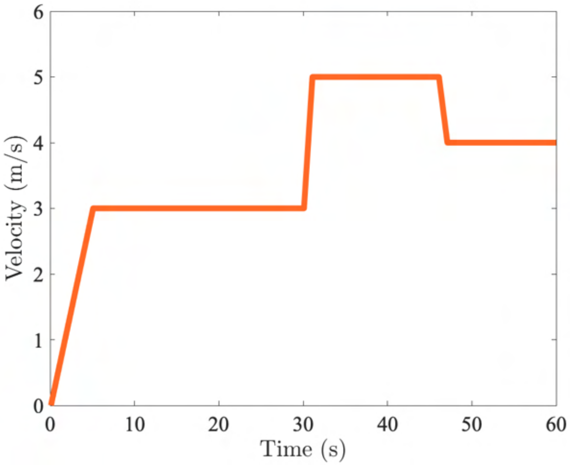


Fig. 3. A designed velocity cycle, including two acceleration processes, a deceleration process, and three constant velocity processes.

The designed velocity trajectory starts from stillness. It increases to $3m/s$ at the time of $5s$, and then maintains the constant velocity for 25 seconds. At the time of $30s$, the vehicle is set to velocity up again, and ends at $5m/s$ within 1 second. After 15 seconds of driving without velocity change, the trajectory decreases the velocity to $4m/s$ in 1 second. Finally, the vehicle was designed to travel at a constant velocity for the last 13 seconds.

To sum up, besides constant velocity driving, two velocity-increasing processes with different accelerations are included, while a deceleration is also covered in this velocity trajectory to build a complete velocity change process.

B. Lightweight Intent Sharing

The whole velocity cycle is split into 12 segments, of which the time interval is five seconds. Each piece of the five-second velocity trajectory is utilized for transmission as intent information of the preceding vehicle. The length of the time interval is consistent with the motion prediction duration used in predictive control methods [28], [29].

Subsequently, we employ polynomial regression to transform the original velocity trajectory into a series of polynomial coefficients. We utilized *cubic polynomial functions* to fit the velocity trajectory into twelve five-second segments. This polynomial function possesses the highest compression rate among the viable data-representative approaches[2].

The time step of the velocity cycle and each piece of velocity trajectories is 0.1 second, which is consistent with the normal frequency of ordinary CAVs. Therefore, to transmit the intent information through a regular approach, each piece requires a volume of 51 data points (including one timestamp). On the contrary, our lightweight intent sharing approach needs only 5 data points (four coefficients and one timestamp). In such a comparison, our approach provides over 90% reduction in the required transmitted data volume, and a foundation for CACC that supports low-cost VLC.

C. Simulated CACC

We used the velocity cycle as the preceding vehicle's intent, and transmitted the intent information to the following CAV. We conducted simulation tests using the LQR controller mentioned in section II.

Specifically, the Q and R matrices we have selected for simulation are:

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{13}$$

which gives distance gap more weight than velocity difference, and a balance between tracking performance and smooth changes.

For the car-following model, we keep using a time interval of 0.1 seconds, set a headway of two seconds, and let the minimum safe distance to be five meters. These data are derived from empirical analysis and are highly capable of expressing typical traffic characteristics.

For the initial state, we set both vehicles to zero velocity, that is, starting from a standstill. The distance between the preceding and the following CAV is set to five meters, aligning with the expected distance gap at the standstill (equals minimum safe distance), preventing unexpected chaos and oscillations.

The results of the simulation experiment are shown in Figure 4. The upper half of the figure shows the velocity

[2] Other polynomial functions can also be adaptively utilized referring to the bandwidth limit, and will be studied in the future.

cycle (also the velocity trajectory of the preceding vehicle) and the result of regular intent sharing, that is, transmitting original velocity trajectories without any compressions. The lower half of the figure shows the regressed velocity cycle and the simulated velocity trajectory of the following CAV using our lightweight intent sharing.

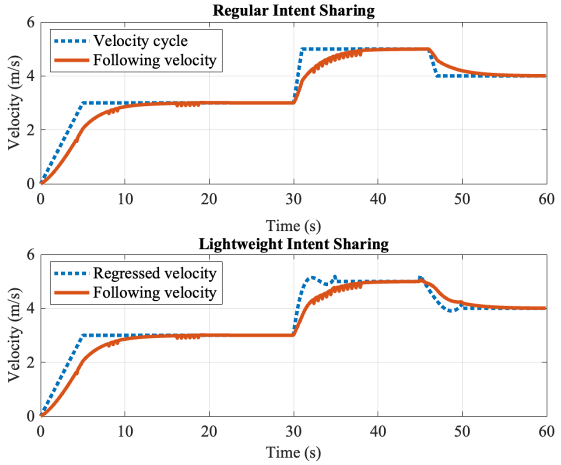


Fig. 4. Simulation results on LQR control with (up) regular intent sharing and (bottom) our lightweight intent sharing in a two-CAV CACC scenario.

By comparing the velocity cycle and the following CAV's velocity trajectories, it can be observed that during the initial start-up, the following CAV exhibited significant acceleration. However as the preceding vehicle continued to increase the velocity, the following CAV gradually began to fall behind. At about 15 seconds, it showed a stable state following the preceding vehicle, and two CAVs both drive at constant velocity. Afterward, the velocity of the preceding vehicle changed twice, and the following CAV caught up with the preceding CAV after a certain delay, forming a stable state. This is in line with the constant time headway car-following criterion, especially when we set the Q and R matrices to give greater weight to the distance gap.

In addition, comparing the regular intent sharing with our lightweight intent sharing, we found that before 30 seconds, due to the linear trajectory of the velocity cycle, our lightweight intent sharing method compressed the transmission data without any information loss, thus maintaining complete consistency with the control after regular intent sharing.

After 30 seconds, there is an obvious deviation between lightweight intent sharing and regular intent sharing, especially when there are changes in the velocity cycle (e.g., from acceleration to constant velocity, from constant velocity to deceleration, from deceleration to uniform speed, etc.). In such cases, the speed trajectories of the following vehicle after LQR control also show certain differences. For information loss that occurs at sudden changes in the original velocity cycle, such as 31 seconds and 46 seconds, the following CAV after LQR control does not show significant trajectory differences compared to that with regular intent sharing, possibly because the control algorithm already has certain robustness to state changes. On the contrary, the information loss at a constant state resulted in a deviation between the two, such as 47 seconds and 50 seconds.

In this study, Normalized Euclidean Distance (NED) will be used as a performance metric to evaluate the similarity between trajectories and velocity profiles generated by regular and lightweight intent sharing. This metric, derived from the standard Euclidean distance, quantifies the deviation between two vectors and is normalized to eliminate the influence of scale.

$$\mathrm{NED} = \frac{\|\mathbf{v}_1 - \mathbf{v}_2\|_2}{N} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{v}_{1i} - \mathbf{v}_{2i})^2 \qquad (14)$$

Here $\mathbf{v}_1$ and $\mathbf{v}_2$ represents two velocity trajectories, and $\|\cdot\|_2$ indicates the 2-norm, which is the Euclidean distance between them. In addition, N represents the total number of data points used for normalization of Euclidean distances to derive NED.

Based on the simulation result, the NED between the following velocities obtained using regular and lightweight intent sharing is 0.002 m/s, indicating an almost negligible difference.

Overall, the results reflect the excellent tracking ability of the LQR controller under intent sharing, and the excellent performance of our lightweight intent sharing for CACC while significantly reducing the data transmission volume.

Certainly, the real-world conditions will be more complicated, including time delays, unpredictable controls, and unexpected vehicle dynamics influenced by the environment. To further substantiate the validity of our lightweight intent sharing approach, we will provide a detailed account of our real-world vehicle experiments in the subsequent section.

IV. Real Vehicle Experiment

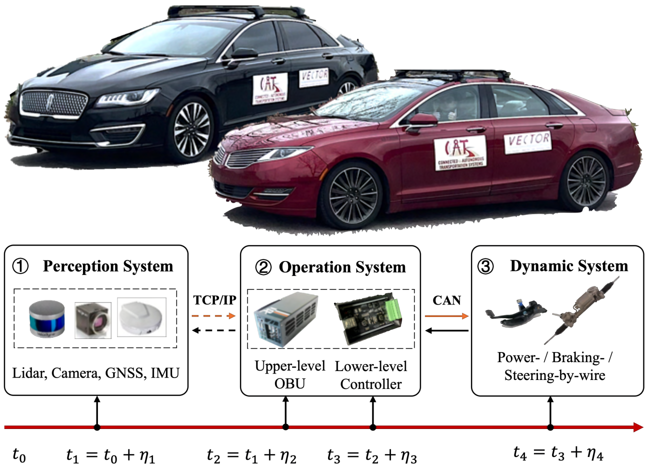


Fig. 5. Connected and autonomous vehicles with sensors and communication devices for real world experiment.

In addition, to validate the feasibility and effectiveness of our lightweight intent sharing approach, we conducted real vehicle experiments for verification. Two CAVs with advanced sensors, in-vehicle computation platforms, and

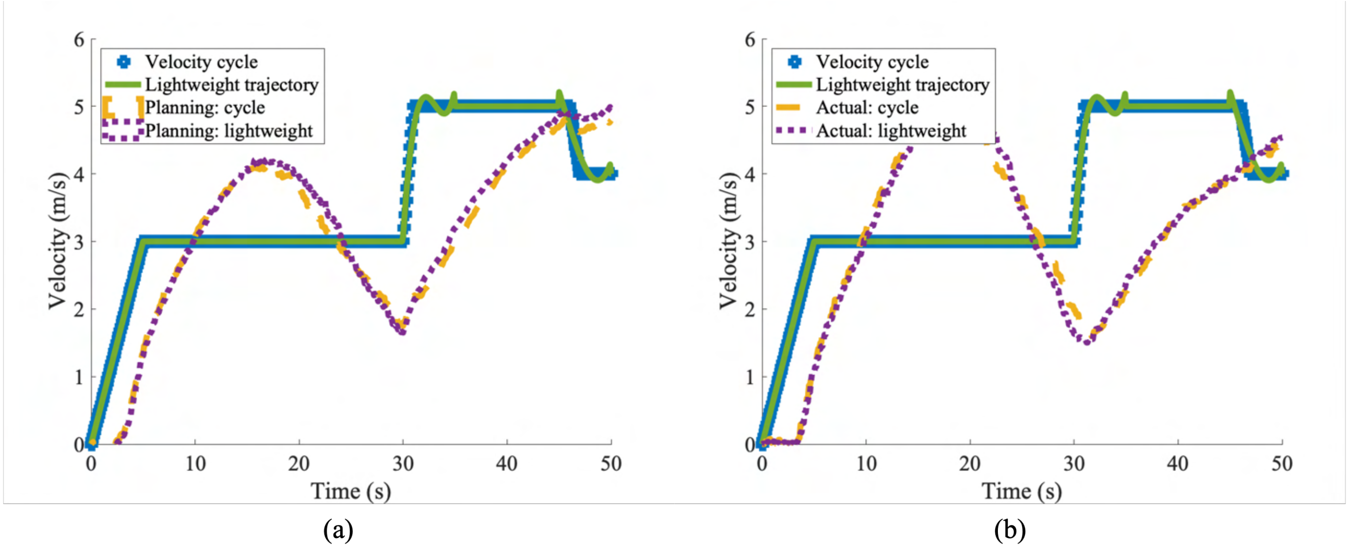


Fig. 6. Comparison of results between two control experiments. (a) Comparison of planned velocities adopting regular intent sharing and our lightweight intent sharing. (b) Comparison of actual velocities adopting regular intent sharing and our lightweight intent sharing.

drive-by-wire system are utilized for experiments, as shown in Figure 5.

We use the black vehicle as the preceding vehicle while the red one is set to follow it. The preceding black vehicle is assumed to drive with the velocity trajectory in line with our cycle shown in Figure 3. The whole planned velocity trajectory is shared with the red following vehicle when starting from still at the time 0 second. The location, distance, and velocity of the preceding black vehicle are also perceived by the following red vehicle to the computing unit for reference by the control algorithm, which has also experienced these time delays. Therefore, the time delay of shared information in the sensors and communication, computing unit, and control actuators are taken into account, including η_1, η_2, η_3, and η_4 in Figure 5.

We conducted two control experiments. One of which transmits the original velocity cycle as the regular intent shared by the black preceding vehicle and received by the red following vehicle. On the other hand, the other group adopted the lightweight intent sharing approach proposed in this paper and transmitted the polynomial regressed velocity trajectory as the preceding vehicle's intent.

With the received information, the red following vehicle took the control method demonstrated in Section II and followed the black preceding vehicle's velocities. The control method generated the planning velocity at each time and gave out the acceleration value for the actuators to refer. The comparison of planning velocities and actual velocities between the two groups of experiments can be seen in the Figure 6.

From Figure 6, it can be seen that the velocity trajectory adopting our proposed lightweight intent sharing approach maintains a significantly high consistency with the original velocity cycle, especially in the constant velocity cases that take over the most time in the cycle, with only a small amount of deviation at occasional accelerations and decelerations, demonstrating the high prediction accuracy of our proposed approach.

As a result, planning trajectories supported by original velocity cycle and lightweight intent sharing show a slight difference (see Figure 6(a)). Moreover, the actual velocity trajectories do not amplify the differences between planning trajectories. On the contrast, the two trajectories show a precise overlap during the $30 \sim 50$ second period. This period coincides with the time when polynomial regressed velocity slightly deviates from the original velocity cycle (see Figure 6(b)).

Specifically, the NED between the two velocity trajectories achieved by regular and lightweight intent sharing is 0.050 m/s for the planning trajectory and 0.053 m/s for the actual trajectory, indicating minimal deviation.

Therefore, it can be verified that our proposed lightweight intent sharing approach has the ability to simultaneously keep the prediction accuracy of preceding vehicle while significantly reduce the requirements on the volume of data transmission in-between vehicles.

V. Conclusion

In this paper, we propose a lightweight intent sharing communication approach in-between vehicles, aiming at reducing the data volume requirements under the limited bandwidth while keeping the prediction accuracy of surrounding vehicles' velocity trajectories. This methodology provides a practical framework to enhance cooperative driving. Specifically, we adopt a polynomial regression to represent the velocity trajectory of the preceding vehicle on a fixed period. Then, the coefficients of the polynomial function are transmitted from the preceding vehicle to the following vehicle. Then, the following vehicle obtains the intended velocity of the preceding vehicle based on these coefficients. Also, we verified that the cubic polynomial regression method is the lightest for an acceptable data representation. Furthermore, we verified this lightweight intent sharing communication approach not only through theoretical derivation but also by simulation and real vehicle experiments. The experiment results demonstrate that the planning and actual vehicle trajectories supported by regular intent sharing and our lightweight one are precisely consistent, with a NED of 0.002 m/s in simulation and 0.050 m/s for planned trajectories and 0.053 m/s for actual behaviors in the real vehicle experiment. This proves that our approach is effective in decreasing transmission data to reduce communication bandwidth requirements, while maintaining the expected improvement of intent sharing in cooperative driving for traffic utilities.

As for consequent work and future directions, sensor and information sharing misalignment will be taken into account. When the sensing and communication delays are inconsistent, time misalignment might have some slight negative effects on the cooperative driving system. Subsequent experiments will try to cover this and attempting to verify the feasibility and effectiveness of our lightweight intent sharing approach in a more complete way. In addition, it promotes low-cost and easy-to-use VLC to quickly improve the traffic utility through connected and automated vehicles.

ACKNOWLEDGMENT

The authors would like to express their gratitude as this paper is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Energy Efficiency and Renewable Energy under Award Number DE-FOA-0002420.

REFERENCES

[1] T. Seo and Y. Asakura, "Endogenous market penetration dynamics of automated and connected vehicles: Transport-oriented model and its paradox," *Transportation Research Procedia*, vol. 27, pp. 238–245, 2017.

[2] H. Li, X. Sun, C. Zhuang, and X. Li, "On the robotic uncertainty of fully autonomous traffic," *arXiv preprint arXiv:2309.12611*, 2024.

[3] X. Qu, L. Zhong, Z. Zeng, H. Tu, and X. Li, "Automation and connectivity of electric vehicles: Energy boon or bane?," *Cell Reports Physical Science*, vol. 3, no. 8, 2022.

[4] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (cacc) systems: Architectures, controls, and applications," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2884–2891, IEEE, 2018.

[5] K. C. Dey, L. Yan, X. Wang, Y. Wang, H. Shen, M. Chowdhury, L. Yu, C. Qiu, and V. Soundararaj, "A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (cacc)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 491–509, 2015.

[6] S. E. Li, Y. Zheng, K. Li, L.-Y. Wang, and H. Zhang, "Platoon control of connected vehicles from a networked control perspective: Literature review, component modeling, and controller synthesis," *IEEE Transactions on Vehicular Technology*, 2017.

[7] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on intelligent transportation systems*, vol. 15, no. 1, pp. 296–305, 2013.

[8] D. Lang, T. Stanger, R. Schmied, and L. d. Re, "Predictive cooperative adaptive cruise control: Fuel consumption benefits and implementability," *Optimization and optimal control in automotive systems*, pp. 163–178, 2014.

[9] Y. Tan and K. Zhang, "Real-time distributed cooperative adaptive cruise control model considering time delays and actuator lag," *Transportation Research Record*, vol. 2676, no. 11, pp. 93–111, 2022.

[10] A. Moradipari, S. S. Avedisov, and H. Lu, "Benefits of intent sharing in cooperative platooning," in *2024 IEEE Vehicular Networking Conference (VNC)*, pp. 195–202, IEEE, 2024.

[11] H. M. Wang, S. S. Avedisov, O. Altintas, and G. Orosz, "Intent sharing in cooperative maneuvering: Theory and experimental evaluation," *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[12] L. Kanipakam, A. H. Sakr, S. S. Avedisov, and A. Moradipari, "Cooperative adaptive cruise control based on intent sharing messages and reinforcement learning," in *2024 IEEE Vehicular Networking Conference (VNC)*, pp. 188–194, IEEE, 2024.

[13] H. M. Wang, S. S. Avedisov, T. G. Molnár, A. H. Sakr, O. Altintas, and G. Orosz, "Conflict analysis for cooperative maneuvering with status and intent sharing via v2x communication," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1105–1118, 2022.

[14] S. Öncü, N. Van de Wouw, W. M. H. Heemels, and H. Nijmeijer, "String stability of interconnected vehicles under communication constraints," in *2012 IEEE 51st IEEE conference on decision and control (cdc)*, pp. 2459–2464, IEEE, 2012.

[15] M. Razzaghpour, S. Shahram, R. Valiente, and Y. P. Fallah, "Impact of communication loss on mpc based cooperative adaptive cruise control and platooning," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pp. 01–07, IEEE, 2021.

[16] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, pp. 3–20, 2009.

[17] H. Li, H. Huang, X. Sun, and X. Li, "Deception for advantage in connected and automated vehicle decision-making games," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 2234–2241, IEEE, 2024.

[18] R. Kawasaki, H. Onishi, and T. Murase, "Performance evaluation on v2x communication with pc5-based and uu-based lte in crash warning application," in *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*, pp. 1–2, IEEE, 2017.

[19] M. Y. Abualhoul, O. Shagdar, and F. Nashashibi, "Visible light inter-vehicle communication for platooning of autonomous vehicles," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 508–513, IEEE, 2016.

[20] L. E. M. Matheus, A. B. Vieira, L. F. Vieira, M. A. Vieira, and O. Gnawali, "Visible light communication: concepts, applications and challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3204–3237, 2019.

[21] K. Ma, X. Li, P. Zhang, and Z. Liang, "A new vehicle-to-vehicle communication technology based on binary light code," Univ. of South Florida, Tampa, FL (United States), 01 2024.

[22] C. Ma, H. Li, K. Long, H. Zhou, Z. Liang, P. Li, H. Yu, and X. Li, "Real-time identification of cooperative perception necessity in road traffic scenarios," *Available at SSRN 4973353*, 2024.

[23] Y. Zhou, M. Wang, and S. Ahn, "Distributed model predictive control approach for cooperative car-following with guaranteed local and string stability," *Transportation research part B: methodological*, vol. 128, pp. 69–86, 2019.

[24] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *ROBOMECH journal*, vol. 1, pp. 1–14, 2014.

[25] D. Swaroop and K. Rajagopal, "A review of constant time headway policy for automatic vehicle following," *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pp. 65–69, 2001.

[26] L. Xiao, M. Wang, and B. Van Arem, "Realistic car-following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles," *Transportation Research Record*, vol. 2623, no. 1, pp. 1–9, 2017.

[27] K. Ma, H. Wang, Z. Zuo, Y. Hou, X. Li, and R. Jiang, "String stability of automated vehicles based on experimental analysis of feedback delay and parasitic lag," *Transportation research part C: emerging technologies*, vol. 145, p. 103927, 2022.

[28] A. Jain, H. S. Koppula, B. Raghavan, S. Soh, and A. Saxena, "Car that knows before you do: Anticipating maneuvers via learning temporal driving models," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3182–3190, 2015.

[29] D. Zhou, H. Liu, H. Ma, X. Wang, X. Zhang, and Y. Dong, "Driving behavior prediction considering cognitive prior and driving context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2669–2678, 2020.