

Backstepping Time-Gap Control for Platooned Autonomous Vehicles: A Multi-Robot Virtually Decentralized Implementation

Nathan Bartley¹, Edward Cruz-Noble², Mark Dannemiller¹, Uwadiae Guobadia², and Shu-Xia Tang²

Abstract—This paper presents simulation and experimental validation of a backstepping-based time-gap control strategy for autonomous vehicle platooning using a small-scale multi-robot platform. The controller, originally developed in [1] for nonlinear vehicle platoon dynamics, is implemented on a three-robot e-puck2 platoon consisting of one leader and two followers. A Python simulation is first used to reproduce the theoretical controller behavior under idealized conditions, followed by a Webots physics-based simulation that incorporates realistic effects such as discrete sampling, sensor noise, and actuator limitations. Both simulations demonstrate accurate regulation of speed and time-gap errors and confirm the stability properties predicted by the theoretical controller. Physical experiments conducted using e-puck2 robots on a guided track further demonstrate that stable platoon formation can be maintained under realistic sensing, communication, and actuation constraints, although larger tracking errors are observed due to sensor inaccuracies, communication latency, and motor response limitations. These results demonstrate the feasibility of using small-scale robotic platforms for validating nonlinear platooning controllers and provide a reproducible pathway from theoretical design to physical implementation.

Index Terms—Vehicle platooning, backstepping control, time-gap regulation, multi-robot systems, experimental validation.

I. INTRODUCTION

Automated vehicle platooning is a key technology in intelligent transportation systems, offering potential improvements in fuel efficiency, roadway capacity, and traffic safety. Achieving these benefits requires control strategies capable of maintaining stable vehicle formations while accommodating nonlinear vehicle dynamics and external disturbances. Although numerous platooning controllers have been proposed, experimental validation remains an important step toward practical deployment [2]. Among these approaches, the backstepping-based time-gap regulation method developed in [1] has demonstrated promising performance in simulation studies. This paper investigates its validation using small-scale robotic platforms, bridging the gap between theoretical development and physical experimentation.

A. Current Traffic Control Algorithms and Autonomous Vehicle Integration

Modern traffic control systems increasingly employ adaptive algorithms that respond dynamically to real-time traf-

fic conditions. The emergence of autonomous vehicles (AVs) further enhances these capabilities through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure communication. Cooperative Adaptive Cruise Control strategies have been shown to improve traffic stability, increase roadway capacity, and reduce congestion in both autonomous and mixed traffic environments [3], [4]. As AV adoption continues to expand and is projected to reach 20% of global vehicle sales by 2030 [5], experimental validation of advanced platooning control strategies becomes increasingly important.

B. Swarm Robotics in System Validation

Small-scale robotic platforms provide a cost-effective environment for validating control algorithms under realistic operating conditions. Platforms such as the e-puck2 robot have been widely used for multi-robot coordination and formation control due to their sensing, communication, and mobility capabilities [6]. Unlike idealized simulations, robotic experiments naturally incorporate sensor noise, communication delays, and actuator limitations. In this work, e-puck2 robots are used to evaluate a backstepping-based time-gap regulation strategy for autonomous vehicle platooning.

C. Novelty and Contribution

The main contributions of this paper are:

- Validation of the backstepping-based time-gap regulation strategy in Python simulation, Webots simulation, and physical e-puck2 robot experiments.
- Development of a virtually decentralized implementation that accommodates the computational limitations of the e-puck2 platform.
- Evaluation of controller performance under realistic sensing, communication, and actuation constraints.

The remainder of the paper is organized as follows. Section II presents the platooning model and control formulation. Section III describes the simulation framework. Section IV presents the Webots validation results. Section V details the experimental implementation and physical validation. Section VI concludes the paper and discusses future work.

II. PROBLEM STATEMENT

Autonomous vehicle platooning requires vehicles to maintain a desired inter-vehicle spacing while tracking the motion of a preceding vehicle. Time-gap policies are widely adopted because they provide improved safety and string stability compared with constant-spacing approaches. In this work, we experimentally validate the backstepping-based time-gap regulation strategy proposed in [1] using a platoon of e-puck2

¹ Nathan Bartley and Mark Dannemiller are with the Department of Electrical & Computer Engineering, Texas Tech University, Texas, USA natbartl@ttu.edu, mdannemi@ttu.edu.

² Edward Cruz-Noble, Uwadiae Guobadia, and Shu-Xia Tang (corresponding author) are with the Department of Mechanical & Aerospace Engineering, Texas Tech University, Texas, USA eponcruz@ttu.edu, uguobadi@ttu.edu, shuxia.tang@ttu.edu.

robots. The objective is to investigate whether the theoretical controller can maintain stable platoon behavior under sensing uncertainty, communication delay, and actuator limitations encountered in physical robotic platforms.

Fig. 1 illustrates the desired time-gap policy used throughout this paper. The follower vehicles aim to maintain a desired spacing proportional to their velocity while tracking the motion of the preceding vehicle.

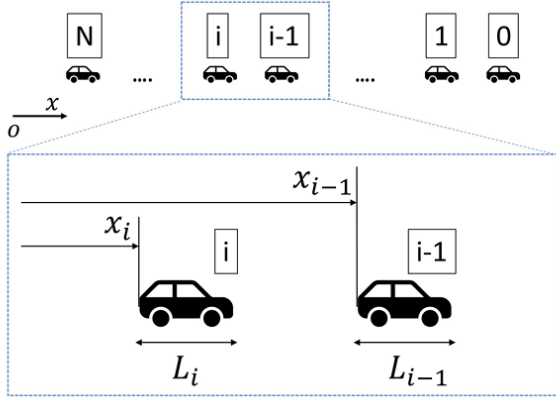


Fig. 1. Desired time-gap policy adopted from [1].

A. Vehicle Dynamics

The longitudinal dynamics of each vehicle are described by the third-order nonlinear model introduced in [1],

$$\dot{x}(t) = v_i(t), \quad (1)$$

$$\dot{v}_i(t) = a_i(t), \quad (2)$$

$$\dot{a}_i(t) = f_i(v_i(t), a_i(t)) + g_i(v_i(t))u_i(t), \quad (3)$$

where

$$f_i(v_i(t), a_i(t)) = -\frac{1}{\tau_i} \left(a_i(t) + \frac{A_{f,i} \rho C_{d,i} v_i(t)^2}{2m_i} + C_{r,i} \right) - \frac{A_{f,i} \rho C_{d,i} v_i(t) a_i(t)}{m_i}, \quad (4)$$

$$g_i(v_i(t)) \equiv \frac{1}{m_i \tau_i}. \quad (5)$$

Here, x_i , v_i , and a_i denote the position, velocity, and acceleration of vehicle i , respectively. τ_i represents the first-order actuator lag, $A_{f,i}$ is the frontal area, $C_{d,i}$ is the aerodynamic drag coefficient, $C_{r,i}$ is the rolling resistance coefficient, ρ is the air density, and m_i is the vehicle mass. The goal is to design control inputs $u_i(t)$ that regulate both time-gap and velocity errors while preserving platoon stability.

B. Backstepping-Based Time-Gap Regulation

To regulate inter-vehicle spacing and velocity synchronization within a platoon, we adopt the backstepping-based time-gap regulation framework developed in [1]. The controller enforces a desired time-gap policy while accounting for nonlinear vehicle dynamics, actuator lag, rolling resistance, and aerodynamic drag effects. In the following controller formulations, $k_{i,j}$ denotes a positive controller gain, $\epsilon_{i,j}$

denotes a positive scaling parameter, δ_0 denotes the acceleration bound, $e_{v,i}$ denotes the velocity tracking errors, respectively, $z_{i,j}$ denotes the backstepping error coordinates, and h denotes the desired time gap.

For the first follower vehicle, a compact representation of the nonlinear backstepping controller derived in [1] is

$$u_1(t) = g_1(v_1(t))^{-1} \left[-f_1(v_1(t), a_1(t)) + p_1 z_{1,1}(t) + (2 + p_1 q_1) e_{v,1}(t) - (p_1 + q_1) a_1(t) - k_{1,3} z_{1,3}(t) - |h + p_1 q_1 - p_1 - q_1| \frac{\delta_0}{2\epsilon_{1,3}} z_{1,3}(t) \right]. \quad (6)$$

For implementation on the e-puck2 platform, the intermediate backstepping variables are recursively evaluated using measurable state quantities:

$$\begin{aligned} u_1(t) = m_1 \tau_1 \left\{ \frac{1}{\tau_1} \left(a_1(t) + \frac{A_{f,1} \rho C_{d,1} v_1(t)^2}{2m_1} + C_{r,1} \right) + \frac{A_{f,1} \rho C_{d,1} v_1(t) a_1(t)}{m_1} + \left(k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} \right) (x_0(t) - x_1(t) - L_1 - h v_0(t)) + \left[2 + \left(k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} \right) \left(k_{1,2} + \left| 1 - k_{1,1} h - \frac{h^2 \delta_0}{2\epsilon_{1,1}} \right| \frac{\delta_0}{2\epsilon_{1,2}} \right) \right] \times (v_0(t) - v_1(t)) - \left[k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} + k_{1,2} + \left| 1 - k_{1,1} h - \frac{h^2 \delta_0}{2\epsilon_{1,1}} \right| \frac{\delta_0}{2\epsilon_{1,2}} \right] a_1(t) - \left[k_{1,3} + \left| h + \left(k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} \right) \left(k_{1,2} + \left| 1 - k_{1,1} h - \frac{h^2 \delta_0}{2\epsilon_{1,1}} \right| \frac{\delta_0}{2\epsilon_{1,2}} \right) \right] \frac{\delta_0}{2\epsilon_{1,3}} \right] \times \left[a_1(t) - [x_0(t) - x_1(t) - L_1 - h v_0(t)] - \left(k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} \right) (v_0(t) - v_1(t)) - \left(k_{1,2} + \left| 1 - h \left(k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} \right) \right| \frac{\delta_0}{2\epsilon_{1,2}} \right) \times (v_0(t) - v_1(t) + \left(k_{1,1} + \frac{h\delta_0}{2\epsilon_{1,1}} \right) \times [x_0(t) - x_1(t) - L_1 - h v_0(t)]) \right] \right\}. \quad (7) \end{aligned}$$

The control input for the second follower vehicle follows

$$\begin{aligned} u_{i+1}(t) = g_{i+1}(v_{i+1}(t))^{-1} \left\{ -f_{i+1}(v_{i+1}(t), a_{i+1}(t)) - [(2 - k_{i+1,1}^2) k_{i+1,1} + P_{i+1}] z_{i+1,1} + [2 - k_{i+1,1}^2 - (k_{i+1,1} + P_{i+1}) P_{i+1}] z_{i+1,2} - (k_{i+1,1} + P_{i+1} + Q_{i+1}) z_{i+1,3} + \sum_{j=1}^i M_{i+1,j} A_j X_j \right\}, \quad (8) \end{aligned}$$

where, unlike the controller for the first follower, the control law incorporates coupling terms that account for interactions

with neighboring vehicles and platoon dynamics. These terms are represented by P_{i+1} , Q_{i+1} , M_{i+1} , A_j , and X_j , as defined in [1]. Substituting the auxiliary variables and coupling terms into the control law yields the fully expanded expression for the controller of the second follower vehicle:

$$\begin{aligned}
u_2(t) = & m_2 \tau_2 \left\{ \frac{1}{\tau_2} \left(a_2(t) + \frac{A_{f,2} \rho C_{d,2} v_2(t)^2}{2m_2} + C_{r,2} \right) \right. \\
& + \frac{A_{f,2} \rho C_{d,2} v_2(t) a_2(t)}{m_2} \\
& - \left((2 - k_{2,1}^2) k_{2,1} + k_{2,2} + \frac{h\delta_0}{2\epsilon_{2,2}} |K_1^T B_1| \right) \\
& \times (x_1(t) - x_2(t) - L_2 - h v_1(t)) \\
& + \left[2 - k_{2,1}^2 - \left(k_{2,1} + k_{2,2} + \frac{h\delta_0}{2\epsilon_{2,2}} |K_1^T B_1| \right) \right. \\
& \times \left(k_{2,2} + \frac{h\delta_0}{2\epsilon_{2,2}} |K_1^T B_1| \right) \left. \right] \\
& \times \left(v_1(t) - v_2(t) - h a_1(t) + k_{2,1} (x_1(t) - x_2(t) - L_2 - h v_1(t)) \right) \\
& - \left[k_{2,1} + k_{2,2} + \frac{h\delta_0}{2\epsilon_{2,2}} |K_1^T B_1| + k_{2,3} + \left| (K_1^T - h K_1^T A_1) B_1 \right| \right. \\
& - h \left(k_{2,1} + k_{2,2} + \frac{h\delta_0}{2\epsilon_{2,2}} |K_1^T B_1| \right) K_1^T B_1 \left. \left| \frac{\delta_0}{2\epsilon_{2,3}} \right| \right] \\
& \times \left[a_2(t) - (1 - k_{2,1}^2) (x_1(t) - x_2(t) - L_2 - h v_1(t)) \right. \\
& - (k_{2,1} + k_{2,2} + \frac{h\delta_0}{2\epsilon_{2,2}} |K_1^T B_1|) \\
& \times [v_1(t) - v_2(t) - h a_1(t) + k_{2,1} (x_1(t) - x_2(t) - L_2 - h v_1(t))] \\
& - (K_1^T - h K_1^T A_1) X_1 \left. \right] \\
& + (K_1^T - h K_1^T A_1) A_1 X_1 \left. \right\}. \tag{9}
\end{aligned}$$

III. PYTHON NUMERICAL SIMULATION

To establish a baseline for experimental validation, the backstepping-based platooning controllers developed in Section II were first implemented in Python. The objectives were to verify the controller implementation and provide reference performance under idealized conditions prior to Webots and physical experiments. Following [1], the vehicle dynamics and controllers were evaluated using a three-vehicle platoon consisting of one leader and two followers.

In this study, the simulation parameters were selected according to the physical characteristics of the e-puck2 platform. The mass of each robot was taken as $m = 0.13\text{kg}$, and the frontal area was estimated as $A_{f,i} = 0.0028\text{m}^2$. The air density was set to $\rho = 1.225\text{kg/m}^3$, and the first-order actuator response lag was modeled using a time constant of $\tau = 0.5\text{s}$. The aerodynamic drag coefficient was estimated as $C_d = 2.76$ based on the geometric ratio between the robot height and diameter together with interpolation from standard drag-coefficient tables for bluff-body geometries.

Controller parameters were selected through iterative numerical tuning to achieve stable platoon operation while

minimizing transient oscillations and avoiding excessive control effort. For Controller 1, the gains were chosen as $k_{1,1} = k_{1,2} = k_{1,3} = 1.5$, while the boundary-layer parameters were selected as $\epsilon_{1,1} = \epsilon_{1,2} = \epsilon_{1,3} = 5$ to reduce high-frequency control chattering. The maximum allowable acceleration was set to $\delta_0 = 1.0\text{m/s}^2$. For Controller 2, the gains were similarly selected as $k_{2,1} = k_{2,2} = k_{2,3} = 1.5$, with $\epsilon_{2,1} = \epsilon_{2,2} = \epsilon_{2,3} = 5$. These parameter values were subsequently kept unchanged throughout the numerical, Webots, and experimental validations to enable consistent comparison across all platforms.

To emulate varying traffic conditions, the leader vehicle was commanded to follow a time-varying velocity profile generated using smooth half-cosine interpolations, as summarized in Algorithm 1. This approach avoids abrupt velocity transitions and produces bounded accelerations that more closely resemble realistic driving behavior. The resulting acceleration and deceleration phases introduce repeated disturbances to the platoon, enabling evaluation of the backstepping controllers under transient operating conditions.

Algorithm 1 Leader Robot Variable Speed Profile

Require: Simulation time $t \geq 0$

Ensure: Leader target velocity $v_{\text{target}}(t)$

```

1: function HALFCOSINE( $t, t_0, t_1, v_0, v_1$ )
2:   if  $t \leq t_0$  then
3:     return  $v_0$ 
4:   else if  $t \geq t_1$  then
5:     return  $v_1$ 
6:   else
7:      $s \leftarrow (t - t_0) / (t_1 - t_0)$ 
8:     return  $v_0 + \frac{v_1 - v_0}{2} [1 - \cos(\pi s)]$ 
9:   end if
10: end function

11: if  $0 \leq t < 5$  then
12:    $v_{\text{target}} \leftarrow \text{HALFCOSINE}(t, 0, 5, 0.15, 0.21)$ 
13: else if  $5 \leq t < 10$  then
14:    $v_{\text{target}} \leftarrow \text{HALFCOSINE}(t, 5, 10, 0.21, 0.14)$ 
15: else if  $10 \leq t < 15$  then
16:    $v_{\text{target}} \leftarrow \text{HALFCOSINE}(t, 10, 15, 0.14, 0.20)$ 
17:   ...  $\triangleright$  Additional 5-s and 10-s interpolations are
      defined similarly
18: else if  $60 \leq t < 65$  then
19:    $v_{\text{target}} \leftarrow \text{HALFCOSINE}(t, 60, 65, 0.13, 0.20)$ 
20: else
21:    $v_{\text{target}} \leftarrow 0.20$ 
22: end if
23: return  $v_{\text{target}}$ 

```

Fig. 2 shows the simulated time-gap errors of the follower vehicles. Although small transient deviations occur during changes in the leader velocity, the errors remain negligible throughout the simulation, demonstrating effective time-gap regulation and platoon stability.

Fig. 3 presents the corresponding speed tracking errors.

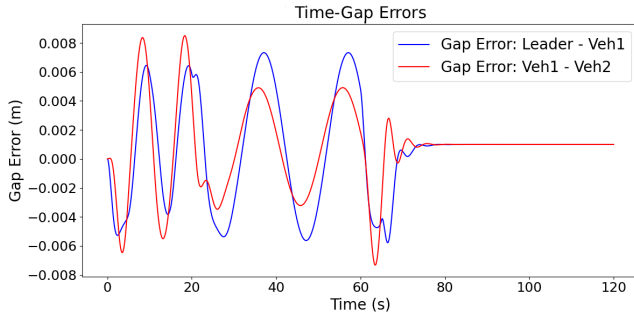


Fig. 2. e-puck2 time-gap errors.

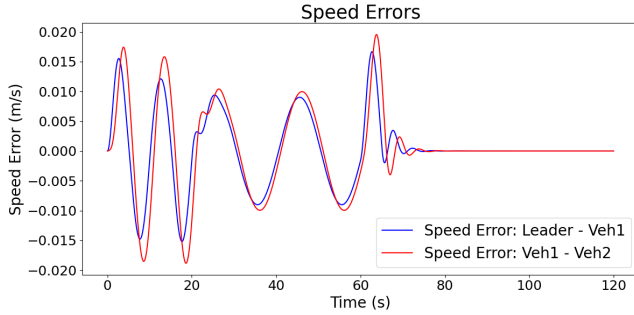


Fig. 3. e-puck2 speed errors.

Similar to the time-gap results, only minor deviations occur during acceleration and deceleration phases. These results indicate that the control inputs generated by the backstepping controllers provide sufficiently accurate velocity regulation for stable platooning operation.

Fig. 4 presents the velocity trajectories of the leader and follower vehicles. The follower vehicles closely track the leader throughout the simulation, exhibiting smooth responses and strong synchronization during both transient and steady-state operation. The close agreement among the velocity profiles demonstrates the effectiveness of the controller implementation and provides a benchmark for the subsequent Webots and experimental validations.

IV. WEBOTS VALIDATION

To further validate the backstepping controller under more realistic conditions, Webots, a physics-based 3D robot sim-

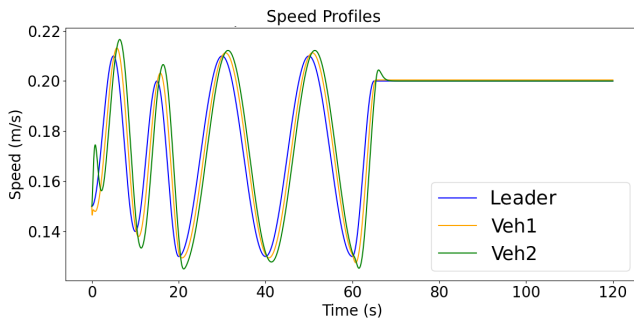


Fig. 4. e-puck2 speed profiles.

ulator developed by Cyberbotics, is employed [7]. Unlike the idealized Python simulation, Webots incorporates rigid-body dynamics, including inertia, mass, friction, collision detection, and sensor-based interactions.

A. Webots Setup

The Webots simulation consists of three e-puck2 robots arranged in a platoon with an initial spacing of 20 cm. The same controller parameters used in the Python simulation are retained for consistency. The leader robot follows a time-varying velocity profile, accelerating and decelerating between 0.075 m/s and 0.105 m/s over a 65-second interval before maintaining a constant speed of 0.1 m/s. The first follower uses its time-of-flight sensor to estimate the relative distance and speed of the leader and applies the backstepping controller to regulate time-gap and speed errors. The second follower receives position, speed, and acceleration information from the first follower and applies the corresponding controller to maintain the desired spacing within the platoon.

B. Validation Results

Fig. 5 presents the simulated time-gap errors obtained in Webots. Compared with the idealized Python simulation, the time-gap errors exhibit larger oscillations and a small steady-state bias caused by sensing uncertainty and discrete-time implementation. The error magnitudes remain bounded and the platoon formation is preserved throughout the simulation.

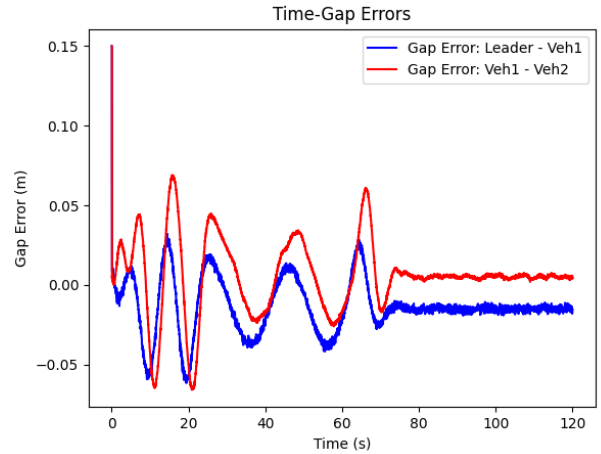


Fig. 5. Webots e-puck2 time gap errors.

Fig. 6 shows that the speed tracking errors remain bounded within approximately ± 0.04 m/s. The error peaks occur during leader speed transitions and are primarily caused by sensor noise and physics-engine effects. Despite these disturbances, the errors decay rapidly after each maneuver. Fig. 7 shows that the follower vehicles closely track the leader with a small phase lag relative to the idealized Python simulation, reflecting realistic sensing and actuation delays. The results remain consistent with the numerical simulation and demonstrate the robustness of the backstepping controller under more realistic operating conditions.

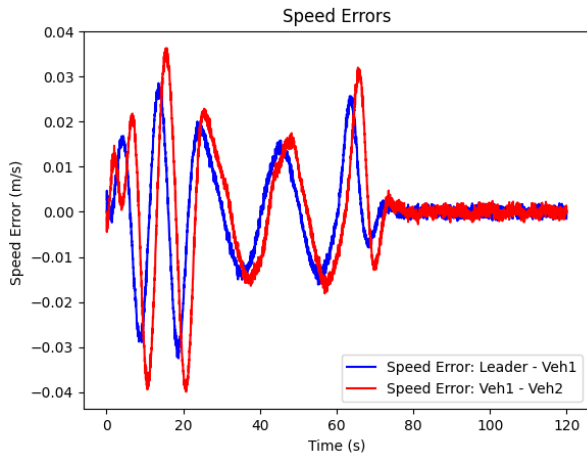


Fig. 6. Webots e-puck2 speed errors.

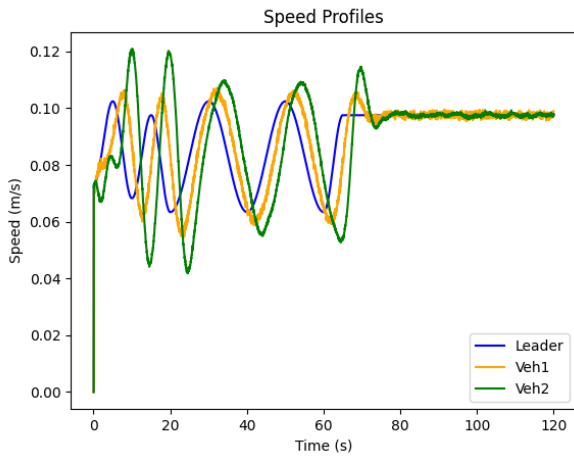


Fig. 7. Webots e-puck2 speed profiles.

V. PHYSICAL EXPERIMENTS

A. Experimental Setup

1) *Physical Setup:* The experimental validation employs three e-puck2 robots operating in a platoon configuration consisting of one leader and two followers, see Fig. 8. Each e-puck2 robot has a diameter of 7 cm, a height of 4.5 cm, and a mass of approximately 130 g [8]. The platform is equipped with a differential-drive system driven by stepper motors, providing a maximum speed of approximately 15.4 cm/s. The onboard sensing suite includes eight infrared proximity sensors for short-range obstacle detection, a front-facing camera, and a time-of-flight sensor with a maximum measurement range of 2 m, which is used for inter-vehicle distance estimation. Wireless communication is achieved through WiFi, enabling real-time transmission of control commands and sensor measurements [6], [9].

To ensure repeatable longitudinal motion, the robots were operated on a straight experimental track consisting of rigid metal guides spaced 20 cm apart over a distance of 3 m.

A smooth paper surface was used to reduce variations in wheel-ground interaction and sensor measurements.

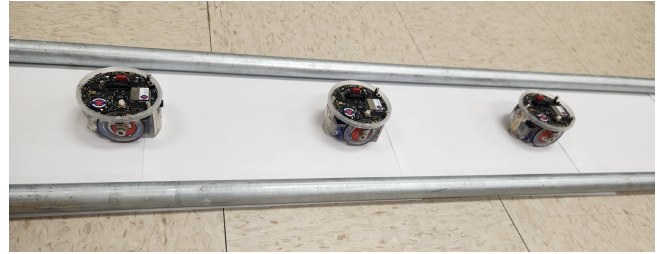


Fig. 8. Leader and follower e-puck2 robots on the experimental track.

2) *Control Implementation:* Unlike the numerical simulations, the physical experiments are subject to sensor noise, communication latency, actuator limitations, and discrete-time execution. Owing to the limited computational resources of the e-puck2 platform, the backstepping controllers were implemented in a virtually decentralized manner. A host computer executed the control algorithms and communicated wirelessly with the robots through Wi-Fi. Sensor measurements were collected from each robot, processed centrally, and converted into control commands that were transmitted back to the corresponding vehicle. This architecture preserves the information structure of the platoon controller while enabling real-time implementation on resource-constrained hardware. Consequently, the experiments provide a realistic assessment of controller performance in the presence of practical communication and actuation constraints that are not captured by the idealized simulation models.

B. Results

Fig. 9 compares the measured leader and follower speeds during the experiment. Although significant fluctuations are present due to sensing noise, communication delays, and motor nonlinearities, the follower generally responds to changes in the leader velocity and maintains coordinated motion without loss of platoon formation.

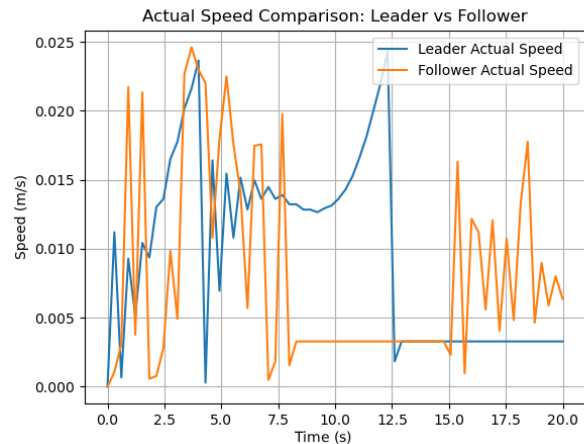


Fig. 9. e-puck2 speed comparison.

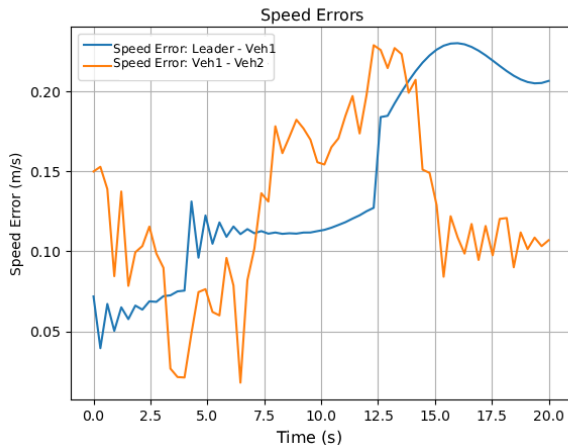


Fig. 10. e-puck2 speed errors.

Fig. 10 presents the speed tracking errors during the experiment. The observed fluctuations arise primarily from sensor uncertainty, communication delays, and motor response limitations. Although the experimental errors are larger than those observed in simulation, they remain bounded throughout the experiment. The results demonstrate that the proposed controller can maintain stable velocity coordination under realistic implementation constraints while preserving overall platoon operation.

Fig. 11 shows the measured time-gap errors during the physical experiments. Compared with the numerical and Webots simulations, the experimental responses exhibit substantially larger fluctuations due to sensing uncertainty, wireless communication delays, limited actuator resolution, and the short operating distance available on the experimental platform. In particular, intermittent variations in the distance measurements introduce abrupt changes in the computed time-gap error. Nevertheless, the follower robot remains responsive to the leader motion throughout the experiment, and no loss of platoon formation or instability is observed. These results indicate that the backstepping controller retains its ability to maintain coordinated platoon behavior under practical operating conditions despite significant sensing and actuation limitations.

VI. CONCLUSION AND FUTURE WORKS

This paper presented the implementation and validation of a backstepping-based time-gap regulation strategy for autonomous vehicle platooning using an e-puck2 robot platform. Numerical simulations and Webots validation demonstrate accurate regulation of time-gap and velocity errors. Physical experiments further show that stable platoon operation can be maintained under sensing, communication, and actuation constraints, although substantially larger tracking errors are observed than in simulation.

Future work will focus on extending the framework to larger platoons and more complex traffic scenarios. To improve experimental performance, future efforts will investi-

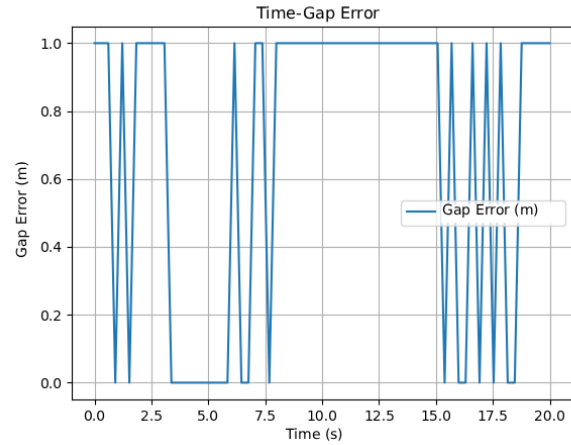


Fig. 11. e-puck2 time-gap errors.

gate enhanced state estimation and communication strategies, including sensor fusion, filtering techniques, and delay-compensation methods, to reduce the time-gap fluctuations observed in the physical experiments. Additional improvements such as wheel-encoder integration, observer-based velocity estimation, and more accurate distance sensing will also be explored. Validation using larger robotic platforms and eventually full-scale vehicles will further enable assessment of controller performance under realistic aerodynamic, powertrain, sensing, and V2V communication effects.

VII. ACKNOWLEDGMENT

The authors would like to thank Fang-Chieh Chou for sharing the MATLAB implementation of [1], which facilitated verification of the Python implementation used in this work.

REFERENCES

- [1] F.-C. Chou, S.-X. Tang, X.-Y. Lu, and A. Bayen, "Backstepping-based time-gap regulation for platoons," in *2019 American Control Conference (ACC)*, 2019, pp. 730–735.
- [2] C. Tomaselli, D. C. Guastella, G. Muscato, L. Minati, M. Frasca, and L. V. Gambuzza, "Synchronization of moving chaotic robots," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6496–6503, 2024.
- [3] A. Talebpour and H. S. Mahmassani, "Influence of connected and autonomous vehicles on traffic flow stability and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 143–163, 2016.
- [4] S. E. Shladover, D. Su, and X.-Y. Lu, "The benefits of cooperation in highly automated driving," *SAE International Journal of Connected and Automated Vehicles*, vol. 1, no. 1, pp. 141–158, 2018.
- [5] J. Deichmann, E. Ebel, K. Heineke, R. Heuss, M. Kellner, and F. Steiner, "Autonomous driving's future: Convenient and connected," Jan. 2023, accessed: Apr. 3, 2024. [Online]. Available: <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected>
- [6] F. Mondada *et al.*, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 2009, pp. 59–65.
- [7] Cyberbotics Ltd, "Webots: Open source robot simulator," <https://cyberbotics.com/cyberbotics>, accessed: 7 February 2026.
- [8] GCTronic, "e-puck2," <https://www.gctronic.com/doc/index.php?title=e-puck2>, accessed: 26 March 2025.
- [9] J. Guzzi *et al.*, "Human-friendly robot swarm navigation in indoor environments," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1785–1799, 2013.