CRLLK: Constrained Reinforcement Learning for Lane Keeping in Autonomous Driving

Demonstration Track

Xinwei Gao
Nanyang Technological University
Singapore
xinwei.gao@ntu.edu.sg

Arambam James Singh
Indian Institute of Technological
Delhi, India
jamesa@iitd.ac.in

Gangadhar Royyuru
Indian Institute of Technology
Madras, India
rspgangadharroyyuru@gmail.com

Michael Yuhas
Nanyang Technological University
Singapore
michaelj004@e.ntu.edu.sg

Arvind Easwaran
Nanyang Technological University
Singapore
arvinde@ntu.edu.sg

ABSTRACT

Lane keeping in autonomous driving systems requires scenario-specific weight tuning for different objectives. We formulate lane-keeping as a constrained reinforcement learning problem, where weight coefficients are automatically learned along with the policy, eliminating the need for scenario-specific tuning. Empirically, our approach outperforms traditional RL in efficiency and reliability. Additionally, real-world demonstrations validate its practical value for real-world autonomous driving.

KEYWORDS

Lane Keeping; Autonomous Driving; Reinforcement Learning

ACM Reference Format:
Xinwei Gao, Arambam James Singh, Gangadhar Royyuru, Michael Yuhas, and Arvind Easwaran. 2025. CRLLK: Constrained Reinforcement Learning for Lane Keeping in Autonomous Driving: Demonstration Track. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

The problem of lane keeping (LK) is an instance of a challenging real-time sequential decision-making problem in the domain of self-driving cars or autonomous driving systems [5, 15, 16]. Traditional model-free reinforcement learning (RL) based approaches to the LK problem face the challenge of defining a reward function that manages trade-offs between multiple objectives [6]. Prior RL approaches use fixed-weighted combinations for objectives such as driving distance [1], minimizing yaw angle [8, 11], crash avoidance [2, 23], and lateral/longitudinal control [7, 14]. Traditional multi-objective reinforcement learning (MORL) approaches [9, 21] tackle such problems by learning a set of optimal policies or applying scalarized reward schemes that rely on static weighting. However, a fundamental limitation of such approaches is that the weight coefficients are typically obtained using scenario-specific tuning and extensive grid searches, which is both time-consuming and computationally expensive for high-fidelity physical simulators.

In response to this challenge, this work introduces a constrained reinforcement learning based formulation and learning approach for the LK problem, which dynamically adjusts the weight coefficients of different objectives. By leveraging this constrained formulation, our system significantly outperforms traditional approaches in terms of efficiency (travel distance) and reliability (lane deviations and avoidance of collisions). We present the weight coefficient learning process and validate our framework in both simulation and real-world settings.[1][2]

2 CONSTRAINED RL FOR LANE KEEPING

2.1 Problem Formulation

Lane keeping is considered as a multi-objective problem with reward and cost functions defined at each time step. The *travel distance reward* is $r(s_t, a_t) = \mathrm{d}^{\mathrm{trv}}$, where $\mathrm{d}^{\mathrm{trv}}$ denotes the forward distance. The *lane deviation cost*, $c_{\mathrm{lane}}(s_t, a_t) = \mathrm{d}^{\mathrm{lane}}$, penalizes horizontal deviation from the lane center, with the average performance given by $J^{\pi_\theta}_{c_{\mathrm{lane}}} = \mathbb{E}\big[\frac{1}{H}\sum_{t=0}^{H-1} c_{\mathrm{lane}}(s_t, a_t)\big]$. The *collision cost*, $c_{\mathrm{coll}}(s_t, a_t) = \mathbf{1}^{\mathrm{col}}$, takes the value 1 if the agent collides with obstacles or boundaries, with performance $J^{\pi_\theta}_{c_{\mathrm{coll}}} = \mathbb{E}\big[\sum_{t=0}^{H-1} c_{\mathrm{coll}}(s_t, a_t)\big]$ across multiple episodes. To ensure safe and efficient driving, we formulate lane-keeping as a constrained optimization problem:

$$\max_{\pi_\theta \in \Pi} J^{\pi_\theta}_R, \quad \text{s.t.} \quad J^{\pi_\theta}_{c_{\mathrm{lane}}} \le \alpha_1,\ J^{\pi_\theta}_{c_{\mathrm{coll}}} \le \alpha_2, \tag{1}$$

where α_1 and α_2 are non-negative thresholds for costs. α_1 corresponds to a real-world distance in decimeters ($10^{-1}m$).

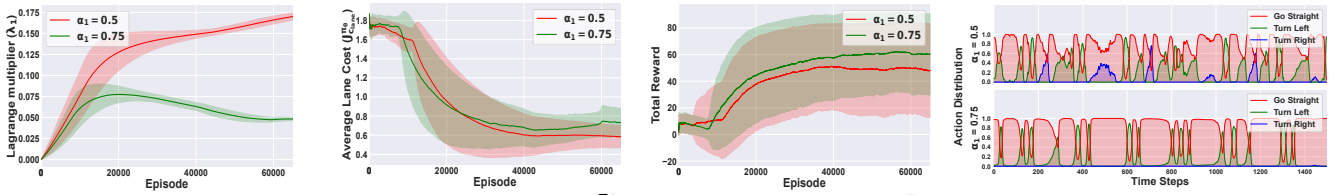
We apply Lagrangian relaxation (LR) technique [4] to convert the constrained optimization in Equation (1) into an equivalent unconstrained optimization problem as follows:

$$\min_{\lambda_i \ge 0} \max_{\theta} L(\lambda_1, \lambda_2, \theta) = \min_{\lambda_i \ge 0} \max_{\theta} \left[J^{\pi_\theta}_R - \sum_{i=1}^{2} \lambda_i \left(J^{\pi_\theta}_{C_i} - \alpha_i \right) \right], \tag{2}$$

where $i \in \{1, 2\}$, L is Lagrangian, λ_i are Lagrangian multipliers for lane and collision costs. λ_i and θ are updated following the gradient

[1] Demonstration video: youtu.be/1BlwJOIUaGM
[2] Source code: github.com/CPS-research-group/CPS-NTU-Public/tree/AAMAS2025

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).



(a) Langrangian multiplier λ_1 (b) Average lane deviation $J^{\pi_\theta}_{c_{\text{lane}}}$ (c) Total distance reward $J^{\pi_\theta}_R$ (d) Action distribution in evaluation.

Figure 1: (a-c) Learning curves for different constraint threshold values (α_1). (d) Action distribution for one test episode.

descent and ascent approach as follows:

$$\lambda_i^{\text{new}} = \max\left(0, \lambda_i^{\text{old}} - \eta_i\left(J^{\pi_\theta}_{C_i} - \alpha_i\right)\right), \quad i \in \{1, 2\}, \tag{3}$$

$$\theta^{\text{new}} = \theta^{\text{old}} + \eta_3 \nabla_\theta \mathbb{E}\Big[\log \pi_\theta(a|s)\Big(J^{\pi_\theta}_R - \lambda_1 J^{\pi_\theta}_{C_{\text{lane}}} - \lambda_2 J^{\pi_\theta}_{C_{\text{coll}}}\Big)\Big], \tag{4}$$

where η_1, η_2 and η_3 are learning rates. Equations (3) updates λ_i to satisfy the original cost constraints. The policy parameters θ is updated with the discounted cost version of $J^{\pi_\theta}_{c_{\text{lane}}}$ and $J^{\pi_\theta}_{c_{\text{coll}}}$ to track the recursive property of Bellman equation [3], where $J^{\pi_\theta}_{c_{\text{lane}\cdot\gamma}} \triangleq \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c_{\text{lane}}(s_t, a_t)\right]$ and $J^{\pi_\theta}_{c_{\text{coll}\cdot\gamma}} \triangleq \mathbb{E}\left[\sum_{t=0}^{H-1} \gamma^t c_{\text{coll}}(s_t, a_t)\right]$. Using the discounted cost constraints above, we define a modified reward function that applied in our approach:

$$\hat{r}(s_t, a_t, \lambda_1, \lambda_2) \triangleq r(s_t, a_t) - \lambda_1 c_{\text{lane}}(s_t, a_t) - \lambda_2 c_{\text{coll}}(s_t, a_t). \tag{5}$$

In the reward function $\hat{r}(\cdot)$, parameters λ_1 and λ_2 act as the adaptive weight coefficients of objectives.

2.2 Implementation

Existing constraint RL solutions like RCPO [20] update policy with truncated samples, failing to track collision cost constraints across multiple episodes, while two-timescale frameworks are impracticable for high-fidelity physical simulators due to computational costs. We adopt a one-timescale framework, updating both θ and (λ_1, λ_2) simultaneously, akin to the simplifications of Actor-Critic [10] made by A3C [13] and DDPG [18]. We implement this approach on the Duckietown platform with PPO [17] algorithm. The system observes the vehicle's state through camera and computes actions while minimizing lane deviation and avoiding obstacles.

3 EXPERIMENT AND DEMONSTRATION

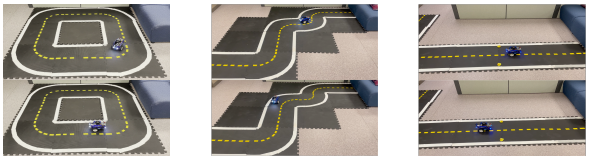
3.1 Simulation Evaluation

The simulation evaluation results on a *small loop* scenario are listed in Table 1. CRLLK-D, CRLLK-C are implementations of our approach on discrete and continuous setting, while PPO-D, PPO-C are the traditional approach with a fixed reward in [17]. Each method is evaluated on 100 test episodes and averaged over two different seeds. Further evaluation results for baselines and scenarios are listed in the video. This evaluation result shows significant performance improvement by adapting our constrained RL framework in terms of efficiency (higher J_R for travel distance), and reliability (lower $J_{c_{\text{lane}}}$ for lane deviations and lower $J_{c_{\text{coll}}}$ for collision).

We performed a demonstration on the training convergence and testing behavior for CRLLK-D on a *small loop* map in Figure 1, with two different constraint levels: tight constraint ($\alpha_1 = 0.5$) and loosened constraint ($\alpha_1 = 0.75$). For tight constraint, the agent policy has a higher probability to choose turn-left and turn-right

Table 1: Performance comparison on the small loop scenario with $\alpha_1 = 0.5$ and $\alpha_2 = 0.02$.

	CRLLK-D	PPO-D	CRLLK-C	PPO-C
$J_{c_{\text{lane}}}$	0.66±0.02	0.99±0.07	**0.31±0.00**	1.04±0.25
$J_{c_{\text{coll}}}$	0.17±0.00	0.38±0.04	**0.05±0.01**	0.34±0.20
J_R	**69.0±0.6**	46.6±3.2	62.4±22.2	51.3±14.8



(a) Small loop (b) Zig-zag (c) Obstacle loop

Figure 2: Real-world evaluation in scenarios.

actions than in the loose constraint, showing fewer lane deviations but covering a shorter total distance. Through this study, we gain insight into how weight coefficients are dynamically learned and how different constraint thresholds affect the lane-keeping trade-off between the lane deviation and travel distance.

3.2 Real-World Demonstration

We evaluate the constrained RL-based policy for lane-keeping tasks using a Duckiebot in real-world scenarios after training in simulation. Testing covered three scenarios: low-difficulty (*small loop*), high-difficulty (*zig-zag*), and complex (*obstacle loop*). The Duckiebot, equipped with Jetson Nano [19] hardware and ROS2 [12], uses an autonomous driving architecture [22] for perception, decision-making, and control. Despite differences between simulation and real-world conditions (e.g. lighting, obstacles), the Duckiebot adapts and navigates smoothly. Figures 2 showcase snapshots of the experiments, with results detailed in the demonstration video.

4 CONCLUSION

We formulate lane keeping as a constrained optimization problem and propose a constrained RL-based solution. The weight coefficients are adaptively learned, eliminating the need for scenario-specific tuning. Empirically, our approach outperforms traditional RL. We analyze the impact of constraint thresholds on policy behavior and convergence, while validating our method through real-world demonstrations across various scenarios.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-017), and MoE, Singapore, Tier-2 grant number MOE2019-T2-2-040.

REFERENCES

[1] Péter Almási, Róbert Moni, and Bálint Gyires-Tóth. 2020. Robust reinforcement learning-based autonomous driving agent for simulation and real world. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[2] SH Ashwin and Rashmi Naveen Raj. 2023. Deep reinforcement learning for autonomous vehicles: lane keep and overtaking scenarios with collision avoidance. *International Journal of Information Technology* 15, 7 (2023), 3541–3553.

[3] Richard Bellman. 1952. On the theory of dynamic programming. *Proceedings of the national Academy of Sciences* 38, 8 (1952), 716–719.

[4] D Bertesekas. 1999. Nonlinear programming. athena scientific. *Belmont, Massachusetts* (1999).

[5] Weiwei Chen, Weixing Wang, Kevin Wang, Zhaoying Li, Huan Li, and Sheng Liu. 2020. Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: A review. *Journal of traffic and transportation engineering (English edition)* 7, 6 (2020), 748–774.

[6] Laurene Claussmann, Marc Revilloud, Dominique Gruyer, and Sébastien Glaser. 2019. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems* 21, 5 (2019), 1826–1848.

[7] Jingliang Duan, Shengbo Eben Li, Yang Guan, Qi Sun, and Bo Cheng. 2020. Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data. *IET Intelligent Transport Systems* 14, 5 (2020), 297–305.

[8] Arpad Feher, Szilard Aradi, and Tamas Becsi. 2018. Q-learning based reinforcement learning approach for lane keeping. In *2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI)*. IEEE, 000031–000036.

[9] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 26.

[10] Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. *Advances in neural information processing systems* 12 (1999).

[11] Bálint Kővári, Ferenc Hegedüs, and Tamás Bécsi. 2020. Design of a reinforcement learning-based lane keeping planning agent for automated vehicles. *Applied Sciences* 10, 20 (2020), 7171.

[12] Steven Macenski, Tully Foote, Brian Gerkey, Chris Lalancette, and William Woodall. 2022. Robot Operating System 2: Design, architecture, and uses in the wild. *Science Robotics* 7, 66 (2022), eabm6074.

[13] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.

[14] Khan Muhammad, Amin Ullah, Jaime Lloret, Javier Del Ser, and Victor Hugo C de Albuquerque. 2020. Deep learning for safe autonomous driving: Current challenges and future directions. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2020), 4316–4336.

[15] Manana S Netto, Salim Chaib, and Said Mammar. 2004. Lateral adaptive control for vehicle lane keeping. In *Proceedings of the 2004 American Control Conference*, Vol. 3. IEEE, 2693–2698.

[16] Ahmad El Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. 2016. End-to-end deep reinforcement learning for lane keeping assist. *arXiv preprint arXiv:1612.04340* (2016).

[17] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[18] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*. Pmlr, 387–395.

[19] Ahmet Ali Süzen, Burhan Duman, and Betül Şen. 2020. Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 1–5.

[20] Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2019. Reward Constrained Policy Optimization. In *7th International Conference on Learning Representations, ICLR, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[21] Kristof Van Moffaert and Ann Nowé. 2014. Multi-objective reinforcement learning using sets of pareto dominating policies. *The Journal of Machine Learning Research* 15, 1 (2014), 3483–3512.

[22] Gustavo Velasco-Hernandez, John Barry, Joseph Walsh, et al. 2020. Autonomous driving architectures, perception and data fusion: A review. In *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 315–321.

[23] Yali Yuan, Robert Tasik, Sripriya Srikant Adhatarao, Yachao Yuan, Zheli Liu, and Xiaoming Fu. 2020. RACE: Reinforced cooperative autonomous vehicle collision avoidance. *IEEE transactions on vehicular technology* 69, 9 (2020), 9279–9291.