doi:10.3311/BMEZalaZONE2022-011

Initiation and Stabilization of Drifting Motion of a Self-driving Vehicle with a Reinforcement Learning Agent

Szilárd Hunor Tóth[1,2], Ádám Bárdos[1a], Zsolt János Viharos[2,3]

[1] Department of Automotive Technologies, Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics

[2] Research Laboratory on Engineering and Management Intelligence, Intelligent Processes Research Group, Institute of Computer Science and Control

[3] Department of Management and Business Law, Faculty of Economics, John von Neumann University

[a] Corresponding author: bardos.adam@kjk.bme.hu

Abstract

Performing special driving techniques like drifting can be challenging even for professional human drivers. However, such maneuvers can be essential for avoiding accidents in critical road scenarios like evasive maneuvers. This paper reports novel research results whose main goal is to develop a self-driving agent for drift motion control based on vehicle simulation in MATLAB/Simulink. The state representation of the vehicle includes the longitudinal and lateral velocities with the yaw rate. The agent action space consists of two actuators: the throttle position and the roadwheel angle. The goal of the agent is twofold: first, it needs to jump into a drifting state; second, it has to keep the vehicle in drift. The simulation results show that the proposed RL agent is capable of learning to approach a predetermined drift equilibrium from cornering and staying in this drift situation as well. For the training, the solution excluded using any prior data. It only works with information gained from the simulation model, which is a remarkable difference from the actual state-of-the-art RL-based solutions.

Keywords: *reinforcement learning, vehicle drifting, vehicle motion control*

1 Introduction

The research and development of the autonomous vehicles has increased in pace and precision in the recent years [1]. In the field of vehicle control, more and more challenges have arisen regarding lateral vehicle control, including motion control at handling limits.

Drifting is a kind of cornering motion, where the driver is constantly counter-steering to maintain a high side-slip angle, which most ordinary drivers are unable to control, and usually leads to the vehicle spinning uncontrollably. This maneuver is mostly seen in motorsports (for example in rally), but the application of this motion has notable potential on the public roads as well. According to the GES (General Estimates System) crash reports of 2013, a study [2] showed that the relative frequency of control loss-related accidents is around 8.32%. Although, there is a potential possibility to decrease this significant value by using self-driving vehicles which can maintain and adjust the motions of the vehicle when it becomes unstable. In addition, the increased value of automated vehicles in motorsports [3] and various potential entertainment purposes (like implementing a "drift button" for hobby drivers) also grants a motivation for research.

As for previous examples and results for the implementation of self-drifting, using linear control methods seem to work successfully for steady-state drift problems in simulation [4] [5] and also in real-life applications [6]. The simulation application of the MPC (Model Predictive Control) controller for drift stabilization tasks has also been successful so far [7]. In each of these cases, a car with a high rear traction force considered to be essential for achieving satisfying performance, based on real-life observations and measurements data.

The idea of using reinforcement learning (RL) methods to solve self-drifting problems promise better generalization abilities in continuously changing driving situations than the previously mentioned control methods. In paper [8] a model-based policy search algorithm was used to solve a steady-state problem with good results, validated on a radio-controlled car. In the work of [9] and [10] the goal was to achieve high side slip angles at high speeds while following a pre-defined trajectory using actor-critic algorithms, the result showing good generalization abilities on previously unseen trajectories. Although, all of these research achievements incorporated some kind of prior knowledge when training the agent. These indicate the challenge of how to apply reinforcement learning without any preliminary setup or knowledge to control and initiate drifting. In addition, previous work [11] proves the potential of applying RL for automatization problems.

In this paper, novel results on reinforcement learning aided autonomous drift are introduced. The task is the initiation and stabilization of a steady-state drift without using any kind of prior knowledge for training the operating RL agent. A single-track dynamic vehicle model was implemented in MATLAB/Simulink, and a Soft Actor-Critic (SAC) [12] algorithm for training was designed. The target drift state was calculated by solving a system of equilibrium equations for the vehicle model. Also, a particular drift indicator was defined for identifying vehicle drifting.

The next section describes the drift state of vehicles established by equation-based solutions while describes the applied vehicle simulation model. The subsequent section presents the coupled reinforcement learning method. After the section about the experiments and results, conclusions and references close the paper.

2 Vehicle Modelling

The basic definition of drift with the state-of-the-art solutions of finding drift equilibriums based on theoretical models (equations), and the integrated vehicle model are described in the following paragraphs.

2.1 The Drifting Motion

To initiate a drift, the driver needs to apply high enough torque input for the rear wheels to increase the rear tire slip angle, so the rear end of the car can "drift" off the arc of the corner. This is easier to achieve using rear-wheel-drive vehicles because they can directly increase the longitudinal forces applied to the rear wheels and saturate them. To control the now unstable car, the driver needs to counter-steer to compensate for the high rear-slip angle, so the car can stay on the corner's arc, otherwise the vehicle would spin out.

A method for identifying drifting operation points is called the equation-based (or model-based) solution which involve using the vehicle model to describe the drift equilibrium points. Previous work [13] show this can be done by solving a system of algebraic equilibrium equations, based on the Newtonian laws of motion. This calculation can be seen in detail in the aforementioned works.

2.2 Dynamic Vehicle Model

The model used for the simulations presented in this research is a single-track dynamic model with two different tire models for the front (steered) and the rear (driven) wheel. The advantage of this model is its simplicity: it ignores the roll dynamics and aerodynamics, which are far less important than the tire model for accurately representing the motions during drifting [14].

The most important equations for the model are the longitudinal (1), lateral (2), and yaw (3) motions of the vehicle's body frame [14]:

$$\dot{v_x} = \frac{1}{\mathrm{m}} F_x + r v_y \tag{1}$$

$$\dot{v_y} = \frac{1}{\mathrm{m}} F_y - r v_x \tag{2}$$

$$\dot{r} = \frac{1}{\mathrm{I_z}} M_z \tag{3}$$

These are the derivatives of the v_x the longitudinal velocity, the v_y lateral velocity and the r yaw rate. For the detailed explanation of the force components in the above equations, see [14].

The model of the front tires is an analytic hybrid model based on the brush tire model, which is purely a lateral slip model [15]. Because this model has a rear-wheel drive, longitudinal forces are only needed to consider in the case of the rear wheels, which would also generate longitudinal slip, so a combined slip tire model is needed. This means that consideration of wheel speed dynamics is also required if the brush tire model is being used. However, the longitudinal rear tire force (F_{x_r}) is handled here as an input, a simpler approach can be used based on the friction circle approximation as proposed in [15]. For the formulas, please see the referenced paper.

3 Drifting with Reinforcement Learning

The description of proposed soft actor-critic reinforcement learning algorithm and the structure of the proposed RL problem are described in this section.

3.1 Soft Actor-Critic Method

The learning process of the RL algorithms is based on the interaction between an agent and an environment. The agent takes an action based on the current state of the environment and receives a reward signal which informs the agent on the effectiveness of the selected action. The agent's mission is to maximize the expected value of the next reward signal with exploring which actions give the maximum cumulative reward in the various states [16].

Actor-Critic methods are state-of-the-art RL algorithms which usually use neural networks to operate within a continuous environment. The actor is responsible for deciding on the actions while the critic estimates the values of the states and produces a critique to update both networks, so they can maximize the cumulative expected reward. The Soft Actor-Critic (SAC) algorithm [12] operates with two separate critic networks to maximize the effectiveness of value approximation and uses a stochastic actor to ensure the exploration, which has an adaptive feature as well. These abilities made this method the most ideal solution above the other state-of-the-art algorithms to solve a complex continuous control problem like autonomous drifting.

3.2 RL Agent Architecture

To represent the control of drifting as a reinforcement learning problem, it's needed to define the environment's state and action spaces and the reward signal.

The velocities defined in (1), (2) and (3) are enough to obviously determine the state of the vehicle, so the continuous state space was chosen to be $\boldsymbol{S} = (v_x, v_y, r)$.

For the agent's actions, drift is achievable by just adjusting the longitudinal force applied on the rear wheel through the pedal input ped_{acc} and the front wheel angle δ by changing the steering wheel angle δ_{steer}. So, the continuous action space is $\boldsymbol{A} = (ped_{acc}, \delta_{steer})$. These values are defined to be in the intervals $ped_{acc} \in [0,1]$ and $\delta_{steer} \in [100°, -200°]$, meaning $ped_{acc} = 1$ is full throttle, and $\delta_{steer} > 0$ is the left-hand side domain of the steering wheel.

By defining $v_x = 10\, {}^{m}/_{s}$ and $\delta = -10°$ for this problem (which means a given speed and a fixed steering wheel position), based on the solution method described in [13] the equilibrium point received for the target drift state is $S_{drift} = (10\, {}^{m}/_{s}, -3.48\, {}^{m}/_{s}, 0.8334\, rad)$. The goal of the agent is to reach and maintain this state, which is a point in the state space. The defined reward function (4) is the negative of the relative Euclidian distance of the state vector from the target drift state, so

$$r(S_t) = -\frac{1}{3}\sum_{i=1}^{3}\left(\frac{S_{t_i}}{S_{drift_i}} - 1\right)^2. \quad (4)$$

4 Experiments and Results

To validate the implemented agent and the Simulink vehicle environment, the following preliminary task was defined: after starting the simulation from the target drift state, the agent's designated task is to stay in a very narrow proximity of the drift equilibrium, and with that, it's able to control the pre-initialized drift. After completing the above-mentioned test successfully, experiments were made to see if it can learn how to approach the target drift state then stay close to it.

To identify if the vehicle satisfies the required drift conditions, the following indicator function (5) was created:

$$Isdrift(S_t) = \begin{cases} 1, \; if \; \frac{S_{t_i} - S_{drift_i}}{S_{drift_i}} < 0.1 \; \forall i \in \{1,2,3\} \\ 0, \; otherwise \end{cases}, \quad (5)$$

meaning it returns 1 if each state variable is in a close relative distance to the target variable at the same time, and 0 otherwise. The training was done by using episodes, which are start from an initial state and lets the agent operate until some simulation termination time T. In the case of the preliminary tests the initial state was set to be the target drift state ($S_0 = S_{drift}$), and for the following experiments the $S_0 = (9\, {}^{m}/_{s}, 0.825\, {}^{m}/_{s}, 0.8334\, rad)$ starting point was defined (a medium-speed left cornering non-drift equilibrium).

For first, $T = 5s$ termination time was set which is a little more than the required amount of time for the vehicle to enter drifting from the initial state under a perfect action selection policy. The SAC agent was successfully trained to accomplish reaching the drift state and to stay in it until the end of the episode, although when the episode length $T > 5$ was set, it was found the agent loses control over the drift after a while. The found solution for this problem was to use a consecutive training technique where the training session was repeated on the same agent with extending the duration of the episodes every time by a small step. After reaching the setup $T = 10s$ using this method, the agent improved significantly enough to hold the target drift for even long-drawn-out episodes ($T \geq 120s$).

The analysis and the identification of desired drifting was done by monitoring the $Isdrift$ indicator (5). On Fig. 1, a scope of a 15-second-long simulation is shown with the vehicle's motion trajectory pictured on Fig. 2. It can be seen that the indicator only "lights up" after approx. 8 seconds of simulation, though drifting occurs much before around 2 seconds. While the role of the indicator was to capture when the vehicle is close to the target, these observations might suggest defining a new or an additional indicator in the future for more effective training and analysis.

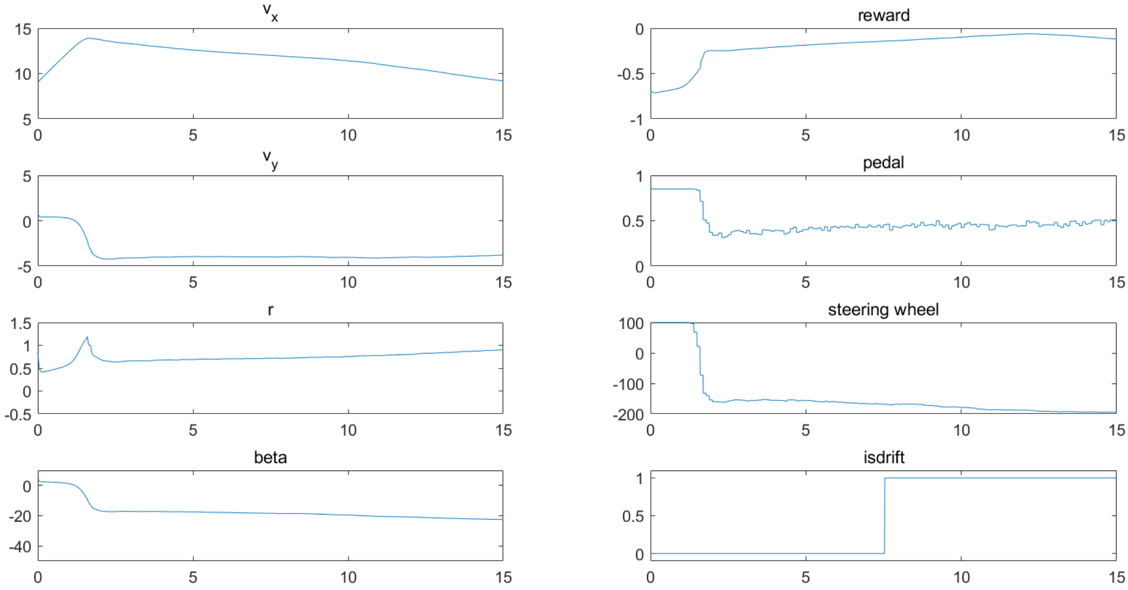

Fig. 1 *Scope graph of a simulation showing 15 seconds of the agent's operation. In addition to looking at the indicator (bottom right), the state variables (left side) and the reward signal (top right) also shows the good performance of the agent, which has chosen actions similarly as a human driver would in practice.*

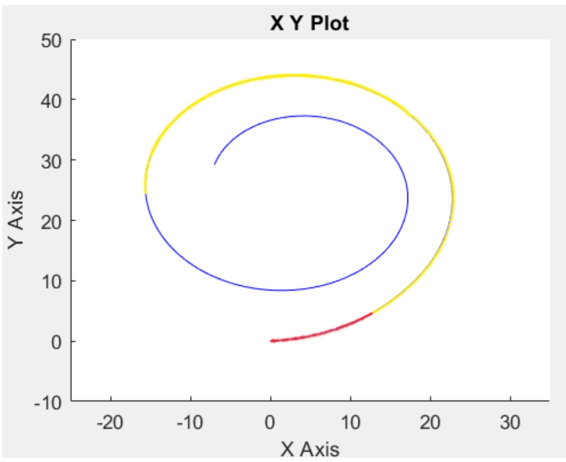

Fig. 2 *Trajectory of the simulation shown on Fig. 1 The colors describe the stages of drifting: red means no drift, yellow shows drifting, which changes to blue when Isdrift = 1. It can be observed that the trajectory's arc bends differently in each of these sections.*

5 Conclusion

The paper presented a reinforcement learning agent trained on a single-track vehicle model to approach and maintain a target drift state in a MATLAB/Simulink simulation environment. The results and the analysis of the defined drift indicator function indicate that the agent is capable of learning to approach a target drift equilibrium point from various initial states. The next steps of this research include evaluating more RL algorithms on this problem (e.g., discrete Q-learning, DQN) to find the most beneficial solution in practice. In the future, the goal is to test this approach in real conditions on a commercial test car on the ZalaZONE proving ground preceded by a validation on the freely available ZalaZONE simulated environment [17]. Also, this research will set its focus on more complex problems and situations later, like adding noise to the environment (e.g., varying road surfaces) and performing more challenging maneuvers in form of trajectory following problems.

Acknowledgement

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

[1] Di, X., & Shi, R. (2021). A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning. Transportation research part C: emerging technologies, 125, 103008.

[2] Li, T., & Kockelman, K. M. (2016, January). Valuing the safety benefits of connected and automated vehicle technologies. In Transportation Research Board 95th Annual Meeting (Vol. 1).

[3] Yang, F. (2021, June). Research on the course, form and strategy of autonomous driving competition. In Journal of Physics: Conference Series (Vol. 1948, No. 1, p. 012093). IOP Publishing.

[4] Bárdos, Á., Domina, Á., Szalay, Z., Tihanyi, V., & Palkovics, L. (2019, November). MIMO controller design for stabilizing vehicle drifting. In 2019 IEEE 19th International Symposium on Computational Intelligence and Informatics and 7th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics (CINTI-MACRo) (pp. 000187-000192). IEEE.

[5] Velenis, E., Katzourakis, D., Frazzoli, E., Tsiotras, P., & Happee, R. (2011). Steady-state drifting stabilization of RWD vehicles. Control Engineering Practice, 19(11), 1363-1376.

[6] Bárdos, Á., Domina, Á., Tihanyi, V., Szalay, Z., & Palkovics, L. (2020, October). Implementation and experimental evaluation of a MIMO drifting controller on a test vehicle. In 2020 IEEE Intelligent Vehicles Symposium (IV) (pp. 1472-1478). IEEE.

[7] Czibere, S., Domina, Á., Bárdos, Á., & Szalay, Z. (2021). Model Predictive Controller Design for Vehicle Motion Control at Handling Limits in Multiple Equilibria on Varying Road Surfaces. Energies, 14(20), 6667.

[8] Cutler, M., & How, J. P. (2016, May). Autonomous drifting using simulation-aided reinforcement learning. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 5442-5448). IEEE.

[9] Cai, P., Mei, X., Tai, L., Sun, Y., & Liu, M. (2020). High-speed autonomous drifting with deep reinforcement learning. IEEE Robotics and Automation Letters, 5(2), 1247-1254.

[10] Orgován, L., Bécsi, T., & Aradi, S. (2021). Autonomous Drifting Using Reinforcement Learning. Periodica Polytechnica Transportation Engineering, 49(3), 292-300.

[11] Viharos Z. J.; Jakab, R. B. (2021). Reinforcement Learning for Statistical Process Control in Manufacturing. Measurement, Vol. 182.

[12] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., ... & Levine, S. (2018). Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.

[13] Hindiyeh, R. Y., & Gerdes, J. C. (2009, January). Equilibrium analysis of drifting vehicles for control design. In Dynamic Systems and Control Conference (Vol. 48920, pp. 181-188).

[14] Jazar, R. N. (2019). Pp. 115-161. Advanced vehicle dynamics. Springer.

[15] Hindiyeh, R. Y. (2013). Dynamics and control of drifting in automobiles (Doctoral dissertation, Stanford University). Pp. 20-28.

[16] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[17] Gangel, K.; Hamar, Z.; Háry, A.; Horváth, Á.; Jandó, G.; Könyves, B.; Panker, D.; Pintér, K.; Pataki, M.; Szalai, M.; Szalay, Zs.; Tettamanti, T.; Tihanyi, V. R.; Tóth, B.; Varga, B.; Viharos, Zs. J. (2021). Modelling the ZalaZONE Proving Ground: a benchmark of State-of-the-art Automotive Simulators PreScan, IPG CarMaker, and VTD Vires. Acta Technica Jaurinensis, 14(4), 488-507.