Transactions on Engineering and Computing Sciences - Vol. 11, No. 4
Publication Date: August 25, 2023
DOI:10.14738/tecs.114.15355.
Yan, H., Zohdy, M. A., Shaout, A., & Mahmoud, A. (2023). Recurrent Proximal Policy Optimization Based Tractor-Trailer Wheeled Robot Automatic Parking Algorithm. Transactions on Engineering and Computing Sciences, 11(4). 124-142.



Recurrent Proximal Policy Optimization Based Tractor-Trailer Wheeled Robot Automatic Parking Algorithm

Hao Yan
ORCID: 0009-0007-6773-5246
Department of Electrical and Computer Engineering, Oakland University, USA

Mohamed A Zohdy
Department of Electrical and Computer Engineering, Oakland University, USA

Abdelrahman Shaout
Department of Electrical and Computer Engineering, Oakland University, USA

Amr Mahmoud
Department of Electrical and Computer Engineering, Oakland University, USA

ABSTRACT

Truck-trailer reverse parking poses significant challenges due to the system's inherent instability, complex road geometry, and collision avoidance requirement. Traditional approaches for trailer control rely on manually designed control policies, which may have limited applicability and scalability. This paper presents a novel approach for autonomous reverse parking of a tractor-trailer wheeled robot (TTWR) system in tight and complex environments. Utilizing the Proximal Policy Optimization (PPO) algorithm, the controller is trained, with a long short-term memory (LSTM) network enhancing the handling of sequential observation data. To improve safety and reliability, ultrasonic sensors are installed on the trailer to detect nearby obstacles and ensure safety. Furthermore, a novel reward function is introduced that encourages the TTWR to maintain a safe distance from surrounding obstacles while minimizing the parking trajectory distance and the steering maneuver. The effectiveness of the proposed LSTM-PPO algorithm is compared with the original PPO and other trending reinforcement learning algorithms. The results demonstrate the improved convergence stability and speed of the proposed approach, as well as its successful execution of end-to-end reverse parking maneuvers in unknown environments.

Keywords: Tractor–trailer Wheeled Robot (TTWR), Proximal Policy Optimization, Recurrent Neural Network, Trajectory Planning, Obstacle Avoidance

INTRODUCTION

ADAS, or Advanced Driver Assistance Systems, has gained significant attention and considerable interest as an innovative and revolutionary technology for enhancing guidance experiences, aiming to enhance safety, provide active assistance to drivers, and facilitate the development of fully autonomous driving technologies. Machine learning algorithms have been extensively incorporated into ADAS to effectively manage intricate road geometry and

topology, handle multi-agent interactions, and accurately follow high-level commands like routing information. As a result, these algorithms have gained exponential growth of popularity in the ADAS field over the past few decades [1] [2].

Among the various features of ADAS, the control and auto-parking assist systems for tractor-trailer wheeled robots (TTWR) have gained significant attention in recent years, because of trailer vehicles' ability to significantly increase load capacity at a relatively low cost, resulting in more efficient use of resources and reduced transportation costs for achieving cooperative tasks [3]. However, the nonlinear, nonholonomic, unstable, and under-actuated system behaviors, along with the large blind zone behind the trailer make TTWR reverse parking a challenging and stressful experience. Another difficult part of the system is the phenomenon called jackknife, which occurs when the angle between the tractor and trailer is beyond a certain threshold, which prevents the driver from steering or aligning the trailer properly during driving backward. When the TTWR system is in the jackknife state, the steering maneuver has little effect on controlling the trailer's states.



Figure 1: Several real-world application scenarios for tractor–trailer wheeled robot system.

Extensive research, for decades now, has been dedicated to addressing the challenges associated with TTWR systems, resulting in the development of various controlling methods that use controllers such as fuzzy logic, neural networks, and nonlinear controllers. For example, traditional control techniques, including Proportional-Integral-Derivative (PID) and model predictive controllers, have been widely utilized in Jackknife prevention, as well as other systems such as steer-by-knob. Xu et. al [3] built a yaw rate-based trailer hitch angle observer which improved the estimation accuracy of the hitch angle, and simplified trailer towing and parking. Jing et. al [4] developed a generic control safety governor to monitor tracking control algorithms and take over the control signal when necessary to avoid safety issues, especially Jackknifing.

As technology evolves, especially in recent years, the application of machine learning algorithms for ADAS has emerged as a key element in the automotive industry for autonomous driving, as well as driver comfort and safety. Deep reinforcement learning has shown great potential for handling decision-making and controlling problems [5]. It has become a popular research topic, in the field of continuous control development, for its flexibility and potential for improved control performance through trial and error. Unlike traditional supervised learning methods, deep reinforcement learning agents learn control policies through interactions within that environment, making them suitable for dynamic and unpredictable environments that are encountered in various applications such as autonomous driving and robotics [7]. Several ADAS problems have been tackled using reinforcement learning algorithms [14] [8] [9], but the development of end-to-end autonomous reverse parking systems for TTWR is still an ongoing area of investigation and growth. To overcome the limitations of rule-based models, which are prone to failures in unknown environments, Du et. al [9] developed a Deep Q-Network based trajectory planner for automated parking systems. Wang et. al [13] proposed a backward motion controller for tractor-trailer mobile robots by using a policy gradient (PG) deep reinforcement learning algorithm to generate continuous action space. Some previous research also integrates reinforcement learning methods with other control algorithms, Bejar et.al [6] used a deep deterministic policy gradient algorithm to train the truck-trailer vehicle and integrate a fuzzy logic rule-based algorithm system into the controller to avoid jackknife which improved safety performance.

The remaining sections of the paper are structured as follows: The second section provides an in-depth explanation of the LSTM and PPO algorithms, highlighting their mechanisms and functionalities. The third section focuses on the kinematic characteristics of the TTWR system during reverse driving, presenting an analysis of the system and establishing mathematical models for both the host vehicle and the trailer. The fourth section introduces the principle of the controller based on LSTM-PPO algorithm and the improvement of the system to achieve minimal maneuvering, safe distance maintenance, and jackknife prevention. In the last section, the algorithm is transplanted to the TTWR platform modeled in Matlab. Furthermore, a comprehensive comparative analysis is conducted among the proposed algorithm and other popular deep reinforcement learning methods, including Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG) and Twin-Delayed Deep Deterministic Policy Gradient Agents (TD3), the TTWR parking trajectory is analyzed, and the effectiveness of the algorithm is evaluated and discussed.

POLICY-BASED REINFORCEMENT LEARNING AND LONG SHORT-TERM MEMORY NETWORKS

The problem of autonomous trailer parking with collision avoidance can be effectively formulated as a sequential decision-making problem within the reinforcement learning framework. There are two main categories of reinforcement learning: value-based and policy-based methods. The value-based methods estimate the value of a given state, denoted as $V^{\pi}(s)$, and then learn the optimal policy by estimating the values of actions and selecting the action with the highest estimated value. Instead, the policy-based methods do not need to maintain a value function model but directly search for an optimal policy π^{*} [19]. In this study, the policy-based reinforcement learning algorithm, PPO, is implemented which directly learns the policy

function, and also LSTM module is integrated with PPO algorithm to map TTWR's observation states to vehicle action inputs.

Proximal Policy Optimization Algorithms

The PPO algorithm is an improved policy gradient (PG) reinforcement learning algorithm proposed in 2017 by OpenAI [20], which is formulated as an unconstrained optimization problem including two terms: accumulative discount return and Kullback–Leibler (KL) divergence [29]. Compared to its predecessors, the PPO algorithm showed improved sample efficiency, less computation burden, and risk mitigation of policy divergence or catastrophic forgetting.

The PG algorithm is founded on the computation of policy gradient estimator and employing it within a stochastic gradient ascent algorithm. A commonly utilized gradient estimator is defined as:

$$\hat{g} = \hat{E}_t\left[\nabla_\theta log\pi_\theta(a_t \mid s_t)\hat{A}_t\right] \tag{1}$$

where π_θ is a stochastic policy and $\hat{A}_t$ is an estimator at timestep t, and $\hat{E}_t[\ldots]$ is empirical average over a finite batch of samples. The estimator $\hat{g}$ can be computed by differentiating the objective:

$$L^{PG}(\theta) = \hat{E}_t\left[log\,\pi_\theta(a_t \mid s_t)\hat{A}_t\right] \tag{2}$$

where $\hat{A}_t$ is the estimator of the advantage function at timestep t, andπ_θ is a stochastic policy.

One issue in PG algorithm is its sensitive to step size, which brings difficulty when choosing a suitable step size. To solve the unconstrained optimization problem, inspired by Trust Region Policy Optimization (TRPO) algorithm [30], the objective function is maximized by using the ratio of old strategies $\pi_{\theta_{\text{old}}}$ and new strategies π_θ, and penalising the size of the policy update using coefficient β:

$$\underset{\theta}{\text{maximize}}\ \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t - \beta\text{KL}\left[\pi_{\theta_{\text{oll}}}(\cdot|\, s_t), \pi_\theta(\cdot|\, s_t)\right]\right] \tag{3}$$

where the probability ratio of old and new strategies is denoted as $r_t(\theta)$ in following equations. To penalize the excessively large policy update caused when maximizing the objective L^{CPI}, the PPO algorithm penalizes changes to the policy that move $r_t(\theta)$ away from 1 , and get the main objective function:

$$L^{\text{clip}}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{cli p}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{4}$$

where the first term L^{CPI} is the "surrogate" objective from TRPO algorithm, the second term, $\text{cli p}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_l$, is the clip function which limits the value of the old and new policy parameters $r_t(\theta)$ to be inside the range of $[1-\epsilon, 1+\epsilon]$. The last step is applying the min function to take the smaller value of the clipped and unclipped objective.

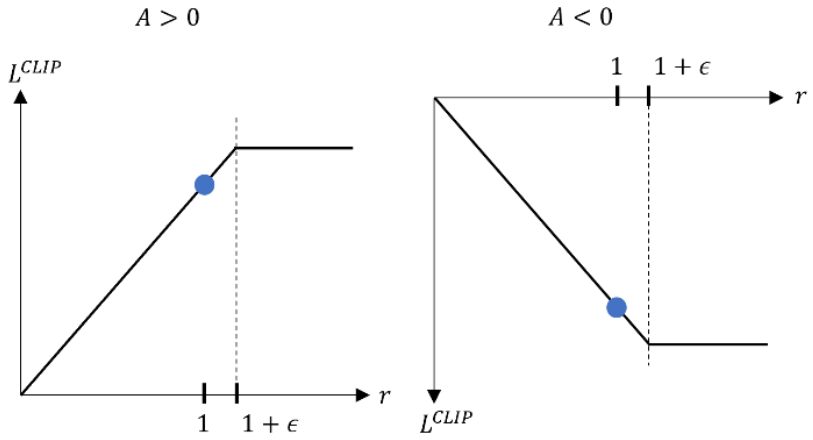


Figure 2: Illustration of the clip function, where the probability ratio r is clipped at $1-\epsilon$ or $1+\epsilon$ depending on the polarity of the advantage.

The core philosophy of the PPO algorithm is to mitigate the challenges caused by large policy updates in the PG algorithm, which can make it difficult for step sizes determination and low data efficiency. By adopting a clipping mechanism to avoid imposing hard constraint completely, the PPO is proven to be very effective in dealing with a wide range of challenging tasks, while being simple to implement and tune [31].

Long-short Term Memory

Long Short-Term Memory is a recurrent neural network architecture that has gained significant attention in the field of deep learning due to its ability to effectively model and predict sequences of data. Compared with traditional RNN structure, the main improvement in LSTM is introducing gate functions including the input gate, forget gate, memory cell, and output gate, which allow the network to capture and retain information over longer periods of time. This unique architecture enables LSTM to overcome the vanishing gradient problem, which hinders the learning process in traditional RNNs.

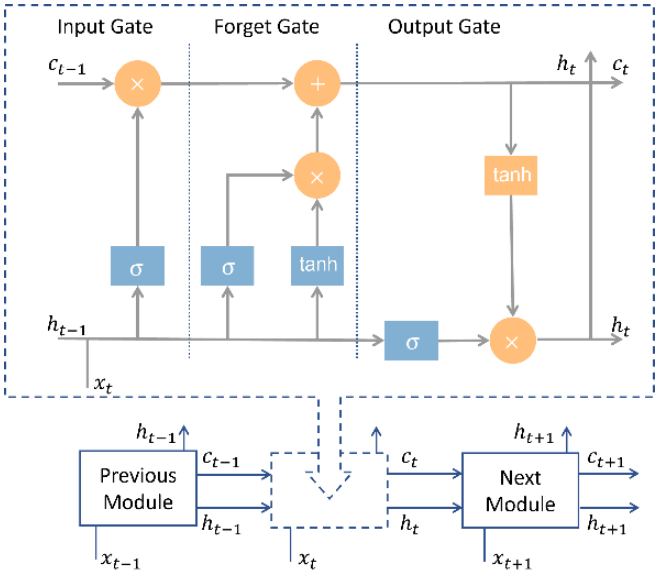


Figure 3: Internal structure of an LSTM cell.

The forget gate decides whether the output h_{t-1} and cell state C_{t-1} of the network at the previous step shall be cleared from the current cell state C_t. The equation of the forget gate's activation vector can be expressed as [23]:

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{5}$$

where $x_t \in \mathbb{R}^d$ is the input vector to the LSTM unit, $h_t \in (-1,1)^h$ is the hidden state vector also known as the output vector of the LSTM unit, $W \in \mathbb{R}^{h\times d}$, $U \in \mathbb{R}^{h\times h}$ and $b \in \mathbb{R}^h$ are weight matrices and bias vector parameters which need to be learned during training where the superscripts d and h refer to the number of input features and the number of hidden units, respectively.

The input gate controls the flow of the current input vector x_t to the cell state C_t. In this module, the input vector x_t and the previously hidden vector h_t are processed by the sigmoid function similar to forget gate, acting as a gatekeeper to decide how much of the current input should be transferred to the cell state, while the tanh function scales the values of the input vector:

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{6}$$

The last part of the LSTM network is the output gate, which determines the output of the LSTM unit based on the current input and the information stored in the cell state C_t, which is defined as:

$$h_t = o_t \odot \sigma_h(c_t) \tag{7}$$

where the $o_t \in (0,1)^h$ is the output gate's activation vector.

Selectively outputting relevant information from the current state is a key feature of the LSTM network, enabling it to capture and maintain useful long-term dependencies in many control tasks.

TTWR State Space and Control Algorithm

Previous research [24] [25] has provided various physical models for vehicle state analysis, including both kinematic and dynamic state models. The choice of model depends on the desired level of accuracy and the specific operating conditions. For high-speed driving tasks, the dynamic factors play a significant role in motion planning and vehicle control; however, in low-speed driving scenarios, the kinematic constraints have a greater influence on the motion behavior of the TTWR system. Since the TTWR reverse parking task is operated in low-speed driving conditions, the system model is established based on the kinematic relations. In this paper, we utilized the Ackermann chassis vehicle as the host vehicle, and the trailer employed a single-axle chassis for the TTWR system.

The two vehicles in the system are linked by the hitch ball, which is a unique pivoting point. The pivot point enables the interconnected vehicles to execute relative rotational motion between each other. From a mechanical structure perspective, the tractor and the trailer can be regarded

as two separate rigid bodies. The vehicle body structure exhibits symmetry with respect to the axis, with the left and right sides being symmetrical. On flat non-slippery ground and under low-speed driving, the sideslip caused by tires and control effect delay caused by mechanical or electrical units can be neglected [26]. In Table 1, the major parameters of the TTWR system are listed.

Table 1: TTWR system states.

Parameters	Descriptions	Parameters	Descriptions
x_1	The lateral position of the host vehicle	ϕ	The host-trailer angle
y_1	The longitudinal position of the host vehicle	v	The longitudinal velocity of host vehicle
θ_1	The yaw angle of the trailer	δ	The steering angle of host vehicle
x_2	The lateral position of the trailer	L_1	The host vehicle wheelbase length
y_2	The longitudinal position of the trailer	L_2	The rear overhang of the host vehicle from the rear axle to the hitch point
θ_2	The yaw angle of the host vehicle	L_3	The overhang of the trailer from the the hitch to trailer axle

KINEMATIC MODEL OF THE HOST AND TRAILER

This state-space model of TTWR in this paper is illustrated in Figure 4, where the trailer has a single axle and a conventional overrun braking system. The assumptions for the TTWR system are as summarized as follows:

- The system has a unique pivoting point, known as the hitch, which is essential in yielding the unstable equilibrium point during reverse driving.
- The system is defined on a flat surface and without slide slip in the kinematics model and simulation environment.
- The trailer has a uniformly distributed mass.

Because all the perception values are received in the vehicle coordinate, it is essential to build the reference point coordinate transformation between the vehicle coordinate and the world coordinate. In this paper, only the translation from trailer coordinate to world coordinate is explained, because host coordinate transformation also follows a similar computation.

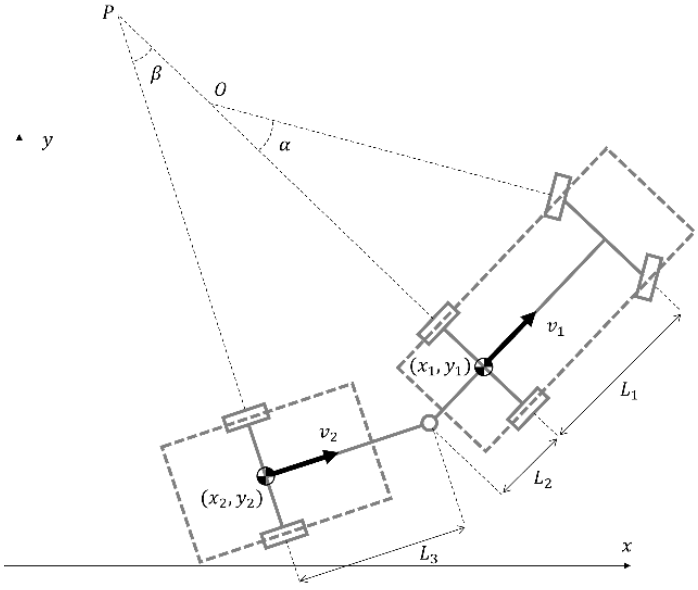


Figure 4: Kinematic full-track model of TTWR.

Under the trailer coordinate, the reference points coordinates can be expressed as:

$$P_2 = \begin{bmatrix} x_{21} & x_{22} & \cdots & x_{2n} \\ y_{21} & y_{22} & \cdots & y_{2n} \end{bmatrix} \tag{8}$$

where P_2 is the coordinate vector matrix, each column vector of the matrix represents a reference point (e.g. surrounding obstacle detections, trailer corner coordinate and et al.) under the trailer coordinate system. $(x_{21}, x_{22}, \ldots, x_{2n})$ and $(y_{21}, y_{22}, \ldots, y_{2n})$ are X-coordinate and the Y-coordinate values respectively.

In linear algebra, the rotation matrix R_{world} and the translation matrix T_{world} can be known as:

$$R_{\text{world}} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 \\ \sin\theta_2 & \cos\theta_2 \end{bmatrix} \tag{9}$$

$$T_{\text{world}} = [t_{x2} \quad t_{y2}]^T \tag{10}$$

where the (x_2, y_2) are the origin point of the trailer coordinate on the trailer rear axle center, and then the coordinate vector translation from trailer coordinate to world coordinate can be expressed as:

$$P_{\text{world}} = R_{\text{world}} * P_2 + T_{\text{world}} \tag{11}$$

In the world coordinate, the host vehicle state model is referred to as the front wheel Ackerman steering model, which allows control of the front wheel orientation relative to the heading of the vehicle. Due to limitations on the available computational power, vehicle dynamics are often highly simplified at the simulation phase [27]. Because the state model in this paper is carried out at low speed and with the assumption of an ideally flat environment, the full track model

represented in Figure 4 can be further simplified as the kinematic bicycle model in Figure 5, and the host vehicle kinematic equation can be expressed as:

$$\dot{x_1} = v\cos\theta_1 \tag{12}$$

$$\dot{y_1} = v\sin\theta_1 \tag{13}$$

$$\dot{\theta_1} = \frac{v}{L_1}\tan\delta \tag{14}$$

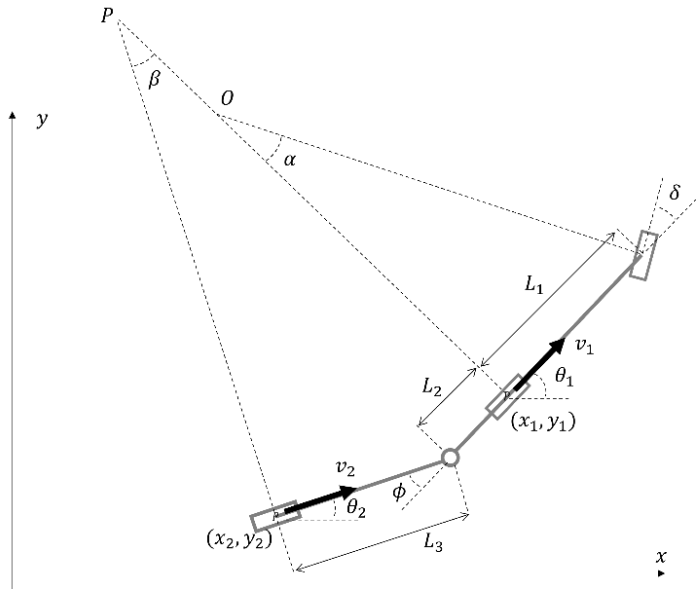


Figure 5: Simplified bicycle kinematic model of TTWR.

The kinematic model of the trailer can be derived by propagating the state space of the host vehicle, as the trailer's kinematics are influenced by the maneuver of the host vehicle. The longitudinal speed of the trailer can be estimated by projecting the normal and longitudinal speed of the host vehicle onto the longitudinal axis of the trailer at the hitch point. Following the bicycle kinematic model, the states $[x_2, y_2, \theta_2]^T$ of the trailer can be expressed as:

$$\dot{x_2} = v\cos\phi\left(1 - \frac{L_2}{L_1}\tan\phi\tan\delta\right)\cos\theta_2 \tag{15}$$

$$\dot{y_2} = v\cos\phi\left(1 - \frac{L_2}{L_1}\tan\phi\tan\delta\right)\sin\theta_2 \tag{16}$$

$$\dot{\theta_2} = -v\left(\frac{\sin\phi}{L_3} + \frac{L_2}{L_1 L_3}\cos\phi\tan\delta\right) \tag{17}$$

To evaluate the threat of jack-knifing, the host-trailer angle is also required for trailer state prediction, which is defined by the difference between Equation 17 and Equation 14:

$$\dot{\phi} = -\frac{v}{L_3}\sin\phi - \frac{v}{L_1}\left(1 + \frac{L_2\cos\phi}{L_3}\right)\tan\delta \tag{18}$$

During reverse driving, the host-trailer system is limited to one degree of freedom in control, which is the host vehicle steering input, the system control is designed to minimize mainly the trailer's tracking error. So, the control reference variables are defined by the trailer's rear axle center position (x_2, y_2), and heading θ_2 compared with the parking slot's reference position (x_{goal}, y_{goal}), and heading angle θ_{goal}.

SYSTEM ANALYSIS

To identify the properties and conditions of the truck trailer system, its equilibrium, observability, and controllability are analyzed. Let $X = [y_2, \theta_2, \phi]^T$ be the trailer states obtained from Equation 5-7 by neglecting the longitudinal component x_3, the state space model can be written as [17]:

$$\dot{\mathbf{X}} = v\big(\mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{X}; \delta)\big) \tag{19}$$

The equilibrium state of X is the origin $X_e = 0$, and we can derive the trailer equilibrium state from Equation 18:

$$\mathbf{X}_e = \{(y_2, \theta_2, \phi) \mid \delta = 0, y_2 \in \mathbb{R}, \theta_2 \in [-\pi, \pi), \phi = 0\} \tag{20}$$

The equilibrium set indicates during reverse driving, the trailer system is in an equilibrium state when the host vehicle and trailer are aligned in either direction, or there should be no steering angle input given by the host vehicle.

To examine the internal asymptotic stability, we substitute $\delta = 0$ into Equation 8, resulting in the following equation:

$$\dot{\mathrm{X}} = v(\mathcal{A}\mathcal{X} + \mathcal{B}\delta) \tag{21}$$

where

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{L_3} \\ 0 & 0 & -\frac{1}{L_3} \end{bmatrix} \mathcal{B} = \begin{bmatrix} 0 \\ \frac{L_2}{L_1 L_3} \\ -\frac{L_3 + L_2}{L_1 L_3} \end{bmatrix} \tag{22}$$

By using the Cayley-Hamilton theorem, the controllability matrix $C = [B, AB, AAB]$ is row full rank for state space in Equation 10, which shows the host-trailer system to be controllable out of the singular locus [18], and the origin of the nonlinear system can be made an asymptotically stable equilibrium by state feedback.

LSTM-PPO BASED TTWR PARKING CONTROLLER

The following presents the end-to-end TTWR autonomous reverse parking controller in tight free space, and to improve user comfort and collision avoidance, a novel reward function has been designed to incorporate historical system states by combining LSTM with the PPO algorithm. The autonomous parking training process contains four stages as shown in Figure 6. The first step is to initialize the TTWR system and parking environment, and then the TTWR

system receives perception data including vehicle states, sonar obstacle list, and parking position. Then the observation state vector is sent to the reward function for the PPO algorithm to train, and the optimal action is generated and sent to the TTWR agent for interacting with the environment.

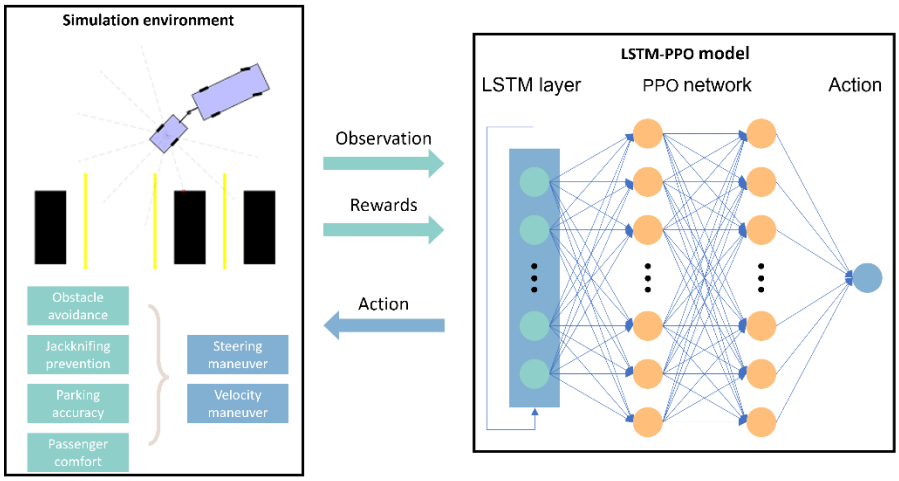


Figure 6: TTWR autonomous parking system architecture.

TTWR Parking System Architecture

The system architecture for the TTWR autonomous parking system is shown in Figure 6. The system consists of two modules: a deep reinforcement learning model and a simulation environment. In particular, the deep reinforcement learning model employs a LSTM-PPO algorithm to train the agent in learning an optimal control policy for autonomous parking and obstacle avoidance, while interacting with the surrounding objects inside the parking environment. The simulation environment includes the kinematics model of TTWR, stationary vehicle obstacles, parking slots, and sonar-based distance detection, which is implemented in MATLAB to interact with the training agent.

Due to the high non-linearity of the TTWR system and the complex interactions with the surrounding environment, relying solely on the current state vector may not adequately capture the environmental complexity and the coordination required for tasks such as autonomous parking, jackknife prevention, and obstacle avoidance. This paper solves this challenge by incorporating LSTM layers in both the actor network and the critic network. The proposed model architecture, referred to as LSTM-PPO, is depicted in Figure 6. The LSTM network extracts relevant features from the high-dimensional environment observation, enables the system to capture valuable perception information and enhance the learning ability when dealing with sequential environment data. The PPO network takes the observations from the actor and passes them through the LSTM layer, and then the wrapped states are then processed by fully connected layers for further feature extraction and policy optimization. Finally, the output layer of the policy network employs the softmax activation function to determine the action distribution and generate the final action output through sampling. This approach ensures that the system can effectively process and utilize sequential information in order to make informed decisions and improve overall performance.

Observation and Action Space

The TTWR system considers the parking target position, TTWR system states, and the detected obstacle information to perform the autonomous parking task. To solve the dimension curse caused by complex observation information, the algorithm discretizes the obstacle detection information around the agent and takes sample points within a threshold, therefore the observation vector can be simplified as follows:

$$S_t = \left[S_{\text{TTWR}}, S_{\text{sonar}}, S_{\text{goal}}\right]^T \tag{23}$$

where the $S_{\text{TTWR}} = [x_1, y_1, \theta_1, x_2, y_2, \theta_2, \phi]^T$ is the TTWR state states defined by the kinematics model, and the parking goal $S_{\text{goal}} = \left[x_{\text{goal}}, y_{\text{goal}}, \theta_{\text{goal}}\right]^T$ is position and yaw angle states in world coordinate where the TTWR shall park, and the value is given by the HMI input.

Moreover, the S_{sonar} is the detection from sonar devices installed on the trailer, instead of sensing 360° environments around the trailer, the sonar returns the distance detection inside its detection FOV, and then this value can be converted from polar coordinate to Cartesian coordinate to represent obstacles around the trailer vehicle. Specifically, the detection of S_{sonar} can be written as the following:

$$S_{\text{sonar}} = [\,\text{dist}_1 \quad \text{dist}_2 \quad \cdots \quad \text{dist}_n] \tag{24}$$

where dist_n is the distance detection of the n-th sonar on the trailer.

To improve the driver experience and smooth the control maneuver of the vehicle, this paper designs the action module representing the host vehicle's steering input for lateral control, and a piece-wise function for the host vehicle's longitudinal velocity, so that the agent can learn when and how to perform steering. The steering controller requires the agent to select the optimal action value from a continuous action space, which is generated by the LSTM-PPO algorithm and is defined as the interval of $[-\delta_{\max}, \delta_{\max}]$, where $\delta_{\max}$ is the maximum steering angle depending on the host vehicle's manufacturing.

This paper also developed a low-level longitudinal velocity controller to perform the movement directed by the PPO decision-maker at a higher level. The host velocity piece-wise function is defined as:

$$\text{v} = \begin{cases} -v_{max} & if\ \text{dist}_{park} \leq \text{dist} \\ -v_{max} & if\ \text{dist}_{tol} \leq \text{dist} \leq \text{dist}_{park} \\ 0 & if\ \text{dist} \leq \text{dist}_{tol} \end{cases} \tag{25}$$

where dist is the distance between the trailer and goal position, dist_{tol} is the tolerance of parking error and the TTWR system should brake when it arrives at the tolerance range, and dist_{park} indicates fine-tuning parking range which requires a slower velocity, and the TTWR system drives under $-v_{\max}$ when it is approaching the parking position until reached the fine-tuning range.

Reward Function

The reward function serves as the feedback mechanism for the controller and maps the current state and action of the TTWR to a scalar value that evaluates potential actions. Specifically, the reward function is designed to ensure that the controller learns how to park the trailer with a focus on collision avoidance and providing a smooth user experience.

- Distance reward R_{dist}: evaluation of distance between the trailer position and parking slot.
- Parking reward R_{parking}: reward when the trailer is parked inside the slot within given tolerance.
- Steering penalty R_{δ}: penalty of large steering maneuver to improve passenger comfort.
- Collision penalty $R_{\text{collision}}$: penalty when TTWR system is too close to surrounding objects.
- Jackknife prevention penalty R_{ϕ}: evaluation of the risk when approaching jack knifing.

As shown in Figure 6, the distance reward R_d is given in every iteration to incentives the TTWR to approach the selected parking slot, which is defined as:

$$R_{\text{dist}}(s_t, a_t, s_{t+1}) = e^{-\left(s_t^2 + s_{\text{gual}}^2\right)} \tag{26}$$

where s_t and s_{goal} denote the current TTWR state and goal state respectively.

The parking reward R_{park} is assigned when the trailer successfully parked itself inside the parking slot with state tolerance s_{ϵ} and the extra accuracy reward is measured by the distance between the final state s_{final} and goal state s_{goal} :

$$R_{\text{parking}} = \begin{cases} c - \frac{\|s_{\text{linal}} - s_{\text{gyual}}\|}{s_{\epsilon}}, \text{ if } \|s_{\text{final}} - s_{\text{goal}}\| \leq s_{\epsilon} \\ 0, \text{ otherwise} \end{cases} \tag{27}$$

Moreover, the collision penalty R_c is assigned when the distance between TTWR and surrounding obstacles is within the threshold, and it will also end the ongoing episode to show a collision; the jackknife penalty $\boldsymbol{R}_{\phi}$ is proportional to the ϕ value to encourage minor maneuver, and steering penalty is proportional to the steering value to avoid large δ input. Therefore, the complete form of parking reward consisting of all 4 parts, as expressed in Equation 28:

$$R = R_{\text{collision}} + R_{\text{parking}} + R_{\text{collision}} + R_{\delta} + R_{\phi} \tag{28}$$

EXPERIMENT RESULTS

Simulation Setup

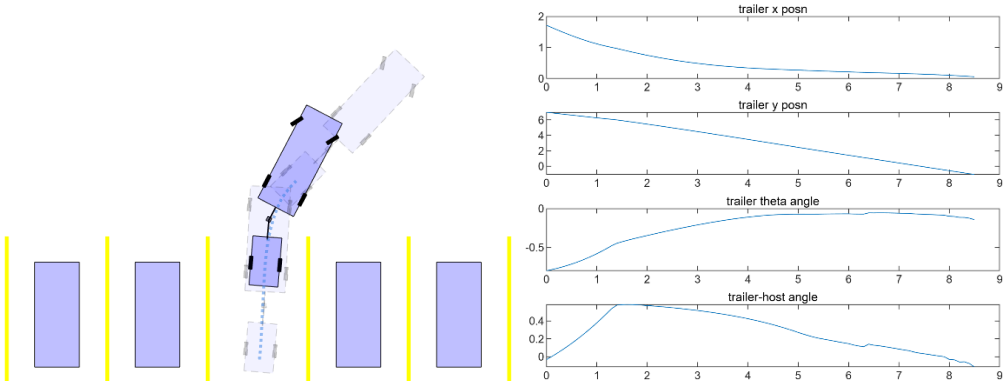
The simulation network was designed to replicate a real-world parking scenario including both perpendicular parking and oblique parking, as shown in Figure 7 and Figure 8. The simulation was implemented using MATLAB/Simulink. The parking area consists of multiple parking slots, each spot is $5m$ long and $3m$ wide. The selected parking slot is marked as the goal parking spot, featured by the yellow solid lines. The remaining parking slots are occupied by vehicle obstacles,

indicated by solid boxes. The objective of the TTWR agent is to reverse into the selected parking spot while avoiding collisions with the surrounding obstacles and preventing jackknifing. The maximum acceleration and emergency braking deceleration of vehicles is set as 2.5 m/s^2. The distance threshold for near-collision penalty is set to 0.2 m. The major PPO algorithm hyper-parameters are provided in Table 2.

During the training process, the neural network receives sequential observations as input, and the TTWR agent explores the environment based on the method described in the previous chapter, continuously learning from its experiences. If an episode is done, which can occur when the TTWR agent exceeds the training boundaries, reaches the selected goal position, experiences a collision, or encounters jackknifing, the simulation environment is reset.

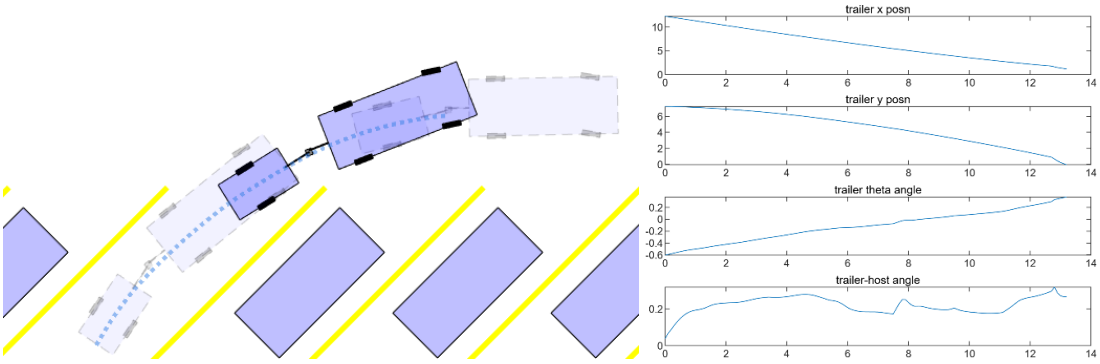
Multiple Parking User Cases Simulation

Figure 7a illustrates the trajectory of perpendicular parking simulation results, which proves that proposed LSTM-PPO controller can adjust the TTWR system's input including host velocity and host steering angle to approach the selected parking spot during perpendicular parking task. The starting positions are randomly initialized with certain noise, and the goal position is selected by the user input. The dotted line represents the trajectory of the trailer, and two faded TTWR shows its initial and final position. In Figure 7b, the trailer states during parking task are shown in the world coordinate, which aligns with the selected parking spot's position and direction.



a. Perpendicular parking simulation b. Perpendicular parking system states

Figure 7: TTWR perpendicular parking.



a. Oblique parking system states b. Oblique parking system states

Figure 8: TTWR oblique parking.

In Figure 8, the TTWR trajectory during 45° oblique parking simulation is shown. Similarly, the starting positions are randomly initialized with certain noise and the proposed methods successfully parked the TTWR system with jackknifing prevention and obstacle avoidance.

LSTM-PPO Controller Simulation Analysis and Comparison

To assess the proposed LSTM+PPO framework, several evaluation metrics have been employed, including the average episode reward, success rate of jackknifing prevention, and collision avoidance. Specifically, the jackknifing rate quantifies the frequency of simulation failures attributed to jackknifing within the prior averaging window. Conversely, the collision rate evaluates the likelihood of encountering a collision with adjacent vehicles during the execution of the reverse parking maneuver. A successful simulation is defined as a successful parking maneuver inside the selected parking slot without jackknifing or collisions with other vehicles.

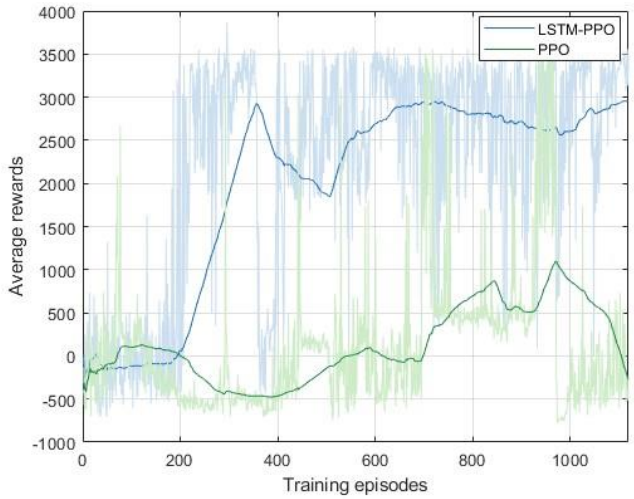


Figure 9: Learning curve comparison between LSTM+PPO and original PPO.

Figure 9 compares the average reward for every 100 episodes between the modified LSTM-PPO algorithm and the original PPO algorithm. The learning curve of LSTM-PPO is represented by the blue line while the original PPO algorithm is shown in the green. It can be seen from the Figure 9 that the blue line achieves a better reward value in fewer episodes, and the average reward reaches 1500 and above within 300 episodes, which represents a successful parking maneuver, and the LSTM-PPO keeps improving parking accuracy and then stables the average reward around 2500 after 600 episodes. In comparison, the original PPO algorithm showed a slower convergence rate and experienced catastrophic forgetting after 1000 episodes during training process. This phenomenon can be caused when many of the weights of neural network are changed, and it is unlikely for prior knowledge to be kept intact. During sequential learning, the inputs become mixed, with the new inputs being superimposed on top of the old ones [28]. Figure 10 showed the learning curve of jackknife prevention, which is the rate of jackknife-caused parking failure in the past 50 training episodes, and the light blue curve is the jackknife occurrence in each episode. Because the TTWR is an unstable system, the jackknife is the main reason failed the parking maneuver. After exploration and exploitation, the agent learns to avoid the occurrence of jackknifing at the same time as learns how to park itself, and at the end

of the training stage, the occurrence rate of jackknifing is close to0, which means the agent already learned the control maneuver to avoid jackknifing.

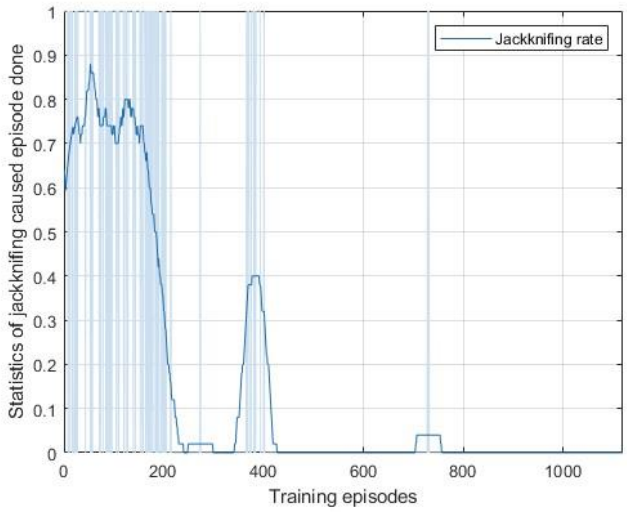


Figure 10: Occurrence of jackknifing and evaluation results of jackknife rate.

Figure 11 showed the learning curve of collision avoidance. Similar to the learning curve for jackknife prevention, the agent learns to avoid obstacles during the parking maneuver. Towards the end of the training stage, the occurrence rate of collision rate dropped from 0.3 to around 0.05. These learning curves demonstrate the ability of the TTWR system to learn safety-critical tasks, guided by the designed reward function, using the LSTM-PPO model proposed in this study.

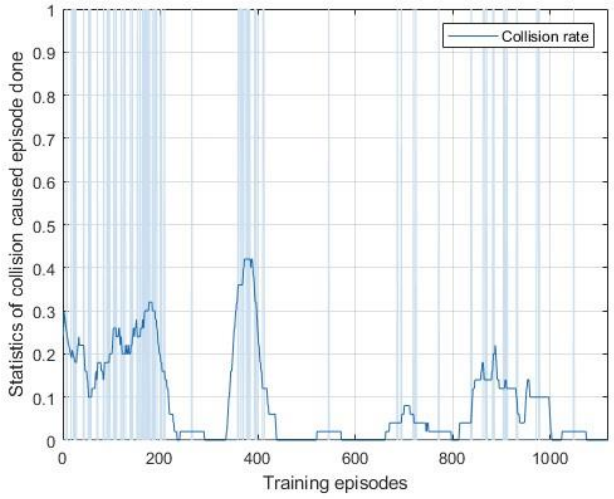


Figure 11: Occurrence of jackknifing and evaluation results of jackknife rate.

Figure 12 presents a comparison of the learning curves between the proposed LSTM-PPO algorithm and several popular RL algorithms in a perpendicular parking scenario. The LSTM-PPO algorithm shows reduced fluctuation in the early stages of training process compared to the DQN, DDPG, and TD3 algorithms. Among the comparable algorithms, TD3 shows

convergence after approximately 300 training episodes and achieves similar control results as the proposed method. However, the PPO-based algorithm in this paper, benefiting from its on-policy nature, shows a faster learning speed overall. Additionally, in the later stages of the training process, the algorithm leverages the memory function of the LSTM network to accumulate higher rewards by effectively incorporating collision avoidance and jackknifing prevention mechanisms.

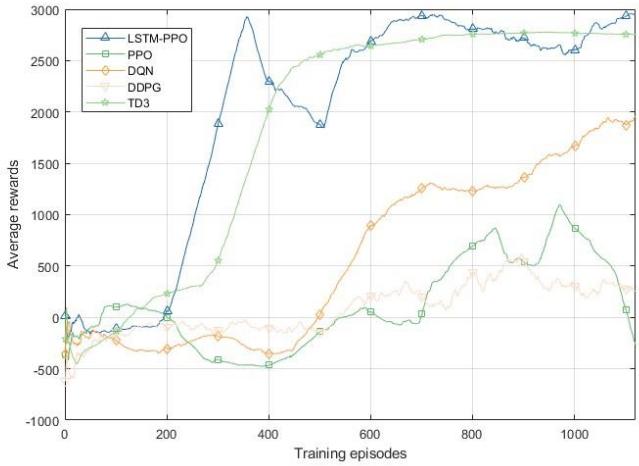


Figure 12: Comparison of average rewards of RPPO among different approaches.

CONCLUSION AND FUTURE WORK

This paper proposed an automated TTWR parking algorithm using a combination of Long short-term memory (LSTM) and Proximal Policy Optimization (PPO) based deep reinforcement learning algorithm, which features collision free, trajectory optimization, and passenger comfort to improve the model's performance. By integrating the LSTM network into the PPO algorithm, the learning rate is enhanced, and the accumulation of higher rewards is facilitated through the effective transformation and processing of high-dimensional observation data. This integration allows the network to effectively learn from and adapt to time-series data, resulting in improved performance and the ability to handle complex and dynamic parking scenarios. The simulation results show the efficacy of the proposed LSTM-PPO algorithm in training the TTWR system, the trained system demonstrates the ability to make appropriate decisions that maximize the cumulative reward while achieving less than 5% jackknifing and obstacle collision rate during the parking process, without forward driving for vehicle states adjustment. The next step of this study will focus on enhancing the proposed TTWR parking algorithm by incorporating forward driving adjustments.

This additional feature will enable the system to dynamically adjust and realign the position and angle of the trailer as it approaches the critical points in tight parking scenarios. Model-based reinforcement learning algorithms will also be studied to reduce sample complexity while maintaining high predictive power, thereby improving the overall performance and efficiency of the system.

References

[1]. J. Chen, B. Yuan, and M. Tomizuka, "Model-free Deep Reinforcement Learning for Urban Autonomous Driving." arXiv, Oct. 21, 2019. Accessed: Jun. 03, 2023. [Online]. Available: http://arxiv.org/abs/1904.09503

[2]. A. Shaout, A. Shaout and S. A. Varagiri, "ADAS Technology Classifications Using a Modified Agile Process," 2022 International Arab Conference on Information Technology (ACIT), Abu Dhabi, United Arab Emirates, 2022, pp. 1-7, doi: 10.1109/ACIT57182.2022.9994175.

[3]. Q. Liu, X. Li, Y. Zhu, and S. Li, "A Review of Tracking Control Method for Tractor-Trailer Vehicles," in Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics, Nanjing China: ACM, Jun. 2022, pp. 863–867. DOI: 10.1145/3548608.3559323.

[4]. L. Xu, E. Tseng, T. Pilutti, and S. Schondorf, "Yaw Rate Based Trailer Hitch Angle Estimation for Trailer Backup Assist," presented at the WCX™ 17: SAE World Congress Experience, Mar. 2017, pp. 2017-01–0027. doi: 10.4271/2017-01-0027.

[5]. J. Jing, J. M. Maroli, Y. B. Salamah, M. Hejase, L. Fiorentini, and U. Ozguner, "Control Method Designs and Comparisons for Tractor-Trailer Vehicle Backward Path Tracking," in 2019 American Control Conference (ACC), Philadelphia, PA, USA: IEEE, Jul. 2019, pp. 5531–5537. doi: 10.23919/ACC.2019.8815188.

[6]. F. Ye, X. Cheng, P. Wang, C.-Y. Chan, and J. Zhang, "Automated Lane Change Strategy using Proximal Policy Optimization-based Deep Reinforcement Learning," in 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 1746–1752. doi: 10.1109/IV47402.2020.9304668.

[7]. E. Bejar and A. Moran, "A Preview Neuro-Fuzzy Controller Based on Deep Reinforcement Learning for Backing Up a Truck-Trailer Vehicle," in 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada: IEEE, May 2019, pp. 1–4. doi: 10.1109/CCECE.2019.8861534.

[8]. S. L. Brunton and J. N. Kutz, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge, UK etc.: Cambridge University Press, 2022.

[9]. C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-Robot Interaction: Crowd-Aware Robot Navigation With Attention-Based Deep Reinforcement Learning," in 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada: IEEE, May 2019, pp. 6015–6022. DOI: 10.1109/ICRA.2019.8794134.

[10]. Z. Du, Q. Miao, and C. Zong, "Trajectory Planning for Automated Parking Systems Using Deep Reinforcement Learning," Int.J Automot. Technol., vol. 21, no. 4, pp. 881–887, Aug. 2020, doi: 10.1007/s12239-020-0085-9.

[11]. E. Bejar and A. Moran, "Reverse Parking a Car-Like Mobile Robot with Deep Reinforcement Learning and Preview Control," in 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA: IEEE, Jan. 2019, pp. 0377–0383. DOI: 10.1109/CCWC.2019.8666613.

[12]. E. E. Reber, R. L. Michell, and C. J. Carter, ``Oxygen absorption in the earth's atmosphere,'' Aerospace Corp., Los Angeles, CA, USA, Tech. Rep. TR-0200 (4230-46)-3, Nov. 1988.

[13]. S. Albeaik, T. Wu, G. Vurimi, F.-C. Chou, X.-Y. Lu, and A. M. Bayen, "Deep truck cruise control," Control Engineering Practice, vol. 121, p. 105026, Apr. 2022, doi: 10.1016/j.conengprac.2021.105026.

[14]. Q. Wang, J. Cheng, and H. Zhang, "Policy Gradient Reinforcement Learning Method for Backward Motion Control of Tractor-Trailer Mobile Robot," in Proceedings of the 11th International Conference on

Computer Engineering and Networks, vol. 808, Q. Liu, X. Liu, B. Chen, Y. Zhang, and J. Peng, Eds. Singapore: Springer Nature Singapore, 2022, pp. 303–311. doi: 10.1007/978-981-16-6554-7_35.

[15]. M. Jaritz, R. de Charette, M. Toromanoff, E. Perot, and F. Nashashibi, "End-to-End Race Driving with Deep Reinforcement Learning," in 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, May 2018, pp. 2070–2075. doi: 10.1109/ICRA.2018.8460934.

[16]. C. Altafini, A. Speranzon, and B. Wahlberg, "A feedback control scheme for reversing a truck and trailer vehicle," IEEE Trans. Robot. Automat., vol. 17, no. 6, pp. 915–922, Dec. 2001, doi: 10.1109/70.976025.

[17]. C. Altafini, "Controllability and singularities in the n-trailer system with kingpin hitching," IFAC Proceedings Volumes, vol. 32, no. 2, pp. 2169–2174, Jul. 1999, doi: 10.1016/S1474-6670(17)56368-5.

[18]. Arulkumaran, Kai, et al. "Deep reinforcement learning: A brief survey." IEEE Signal Processing Magazine 34.6 (2017): 26-38.

[19]. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms." arXiv, Aug. 28, 2017. Accessed: Apr. 25, 2023. [Online]. Available: http://arxiv.org/abs/1707.06347

[20]. Williams, Ronald J. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Reinforcement learning (1992): 5-32.

[21]. Lee, Kyowoon, et al. "Deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling." International conference on machine learning. PMLR, 2018.

[22]. Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," Neural Computation, vol. 31, no. 7, pp. 1235–1270, Jul. 2019

[23]. F. Sorge, "Motion and force analysis of slow-speed multi-trailer systems," Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 233, no. 8, pp. 2615–2635, Apr. 2019, doi: 10.1177/0954406218792589.

[24]. X. Liu, A. K. Madhusudhanan, and D. Cebon, "Minimum Swept-Path Control for Autonomous Reversing of a Tractor Semi-Trailer," IEEE Trans. Veh. Technol., vol. 68, no. 5, pp. 4367–4376, May 2019, doi: 10.1109/TVT.2019.2895513.

[25]. G. Lei and Y. Zheng, "Research on Cooperative Trajectory Planning Algorithm Based on Tractor-Trailer Wheeled Robot," IEEE Access, vol. 10, pp. 64209–64221, 2022, doi: 10.1109/ACCESS.2021.3062392.

[26]. P. Polack, F. Altche, B. d'Andrea-Novel, and A. De La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?," in 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA: IEEE, Jun. 2017, pp. 812–818. doi: 10.1109/IVS.2017.7995816.

[27]. K. McRae and P. Hetherington. Catastrophic interference is eliminated in pre-trained networks. In CogSci, 1993. 1

[28]. H. Xu, Z. Yan, J. Xuan, G. Zhang, and J. Lu, "Improving proximal policy optimization with alpha divergence," Neurocomputing, vol. 534, pp. 94–105, May 2023, doi: 10.1016/j.neucom.2023.02.008.

[29]. J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust Region Policy Optimization." arXiv, Apr. 20, 2017. Accessed: Jun. 04, 2023. [Online]. Available: http://arxiv.org/abs/1502.05477

[30]. Y. Wang, H. He, and X. Tan, "Truly Proximal Policy Optimization" Uncertainty in Artificial Intelligence. PMLR, 2020.