SILESIAN UNIVERSITY OF TECHNOLOGY PUBLISHING HOUSE

SCIENTIFIC PAPERS OF SILESIAN UNIVERSITY OF TECHNOLOGY 2023
ORGANIZATION AND MANAGEMENT SERIES NO. 204

UTILIZING ARTIFICIAL INTELLIGENCE FOR ENERGY-EFFICIENT ROUTE PLANNING IN HYPERLOOP LOW-PRESSURE CAPSULE TRANSIT: A STUDY IN ALIGNMENT WITH SUSTAINABLE DEVELOPMENT GOALS

Rafał RUMIN[1*], Jerzy DUDA[2], Jędrzej BLAUT[3], Dawid PĘKAŁA[4], Thomas MEROLLA[5]

[1] AGH University of Krakow; rumin@agh.edu.pl, ORCID: 0000-0001-5852-0322
[2] AGH University of Krakow; jeduda@agh.edu.pl, ORCID: 0000-0002-9225-7123
[3] AGH University of Krakow; blaut@agh.edu.pl, ORCID: 0000-0002-9638-1287
[4] AGH University of Krakow; dawpek@student.agh.edu.pl, ORCID: 0009-0003-9091-1368
[5] University of Modena and Reggio Emilia-Unimore, Italy; thomasmerolla9@gmail.com
* Correspondence author

Purpose: The purpose of this paper is to explore the integration of reinforcement learning with simulation software to optimize Hyperloop transportation systems, aiming to enhance operational efficiency, reduce departure timetable variability, and improve station operations through dynamic dispatch prioritization.
Design/methodology/approach: The research utilizes reinforcement learning combined with discrete-event simulation to model and optimize Hyperloop system operations, focusing on departure schedules, delay reduction, and dispatch prioritization.
Findings: The research demonstrates that reinforcement learning significantly enhances Hyperloop system performance by optimizing departure schedules, reducing delays, and improving dispatch prioritization, leading to more efficient and reliable operations.
Research limitations/implications: The study's limitations include the reliance on simulated data and hypothetical scenarios, which may not fully capture real-world complexities, and future research should focus on testing the proposed methods in actual Hyperloop environments and addressing potential scalability issues.
Practical implications: The research identifies enhanced diagnostic methods for Hyperloop systems that could lead to more efficient and reliable operations, potentially reducing maintenance costs and downtime. The adoption of these methods can improve the safety and performance of Hyperloop services, thereby boosting commercial viability and economic benefits for businesses involved in the development and operation of this high-speed transportation technology.
Social implications: This research on Hyperloop diagnostics could significantly influence public attitudes towards the acceptance and adoption of high-speed vacuum transportation, highlighting its safety and efficiency. Improved diagnostic methods will enhance the reliability of Hyperloop systems, promoting sustainable and environmentally friendly transportation alternatives. Additionally, the findings could inform public and industry policy, encouraging investment in advanced transportation infrastructure, ultimately improving quality of life through reduced travel times and lower emissions.

Originality/value: This paper presents novel diagnostic methods tailored for the unique conditions of Hyperloop systems, such as high-speed vacuum environments. It offers valuable insights for engineers, researchers, and policymakers involved in the development and implementation of advanced transportation technologies.
Keywords: Artificial Intelligence, Hyperloop, FlexSim Simulation, Operational Efficiency, Reinforcement Learning.
Category of the paper: Research paper.

1. Introduction

The Hyperloop, initially conceptualized by George Medhurst and re-envisioned by Elon Musk in 2013, epitomizes a sustainable, high-speed transportation system utilizing electromagnetic propulsion within low-pressure tubes. This innovation promises drastic reductions in travel times and environmental impacts, aligning with the goals of the Green Industrial Revolution by decreasing greenhouse gas emissions and energy consumption (Premsagar, Kenworthy, 2022). With global research and development accelerating, Hyperloop technologies are poised to transform the transportation of passengers and freight, marking a shift towards a more efficient and sustainable future (Barbosa, 2020). Simultaneously, Artificial Intelligence (AI) has advanced from its nascent symbolic and rule-based frameworks to sophisticated neural networks and machine learning techniques, enriching problem-solving across various sectors, including education and healthcare (Zerilli et al., 2021). This progression is underscored by the emergence of Reinforcement Learning (RL), which through Markov decision processes, enables agents to learn and optimize behaviours via trial-and-error, thus enhancing decision-making in dynamic environments (Sutton, Barto, 1998). When combined with neural networks, RL—through deep reinforcement learning—has initiated groundbreaking improvements across diverse fields such as queue management and crisis resource allocation (Mnih et al., 2015). In the realm of transportation, AI's integration is further amplified by simulation technologies like FlexSim, which optimize logistics and potentially reduce environmental footprints. However, the deployment of such simulations raises critical concerns regarding overfitting and data privacy, necessitating robust AI governance to ensure fairness and transparency (Marquis et al., 2020). This paper examines the integration of an RL model into FlexSim for optimizing Hyperloop station operations, aiming to enhance operational efficiency and adjust departure schedules in response to fluctuating passenger demands.

2. Literature Review

Introduced by Elon Musk in 2013, the Hyperloop utilizes electromagnetic propulsion to transport passengers and goods at high speeds through low-pressure tubes, offering potential energy efficiency and minimal environmental impact. Despite its transformative promise, the Hyperloop confronts significant challenges including the necessity for full-scale testing facilities to emulate real-world conditions (Mitropoulos et al., 2021), safety concerns like motion sickness from rapid travel (Almujibah et al., 2020), and high capital costs which may limit access primarily to higher-income groups (Premsagar, Kenworthy, 2023). The system also requires the integration of vehicle and infrastructure design, demanding advanced optimization to meet complex parameters and secure stakeholder confidence (Kirschen, Burnell, 2021). Overcoming these technical and social hurdles through focused research and strategic policymaking is essential for successful Hyperloop implementation (Kupriyanovsky et al., 2020).

FlexSim, a discrete-event simulation software, effectively constructs three-dimensional models of systems, enhancing operations across industries by identifying bottlenecks and optimizing processes (Garrido, 2009). In manufacturing, it boosts productivity and efficiency by simplifying complex procedures (Yi-jun, 2011), and in logistics, it improves warehouse operations and system decision-making (Xing-hua, 2009). Moreover, FlexSim's educational applications provide practical operational insights, improving learning outcomes in logistics and industrial engineering (Ru, 2012).

AI revolutionizes transportation management through advanced route scheduling and dynamic response systems. For example in aviation, AI may optimize fleet routing to enhance profitability and service quality (Yan, Tseng, 2002), while in queue management, reinforcement learning (RL) techniques can reduce waiting times and optimize space usage (Sun et al., 2022). AI also refines simulation accuracy in transport logistics, crucial for applications like FlexSim (Marquis et al., 2020), and improves route and energy management through predictive analytics (Pal, 2023; Zhang et al., 2021).

3. Research methodology

The study employed the FlexSim simulation environment to design and evaluate a dynamic-flow process. This involved developing a code environment in Visual Studio, which was divided into four segments: environment, interface, training process and FlexSim process flow code directly related to the objective function adopted.

The environment designed in FlexSim to reflect a hypothetical Hyperloop capsule station is shown in Figure 1.

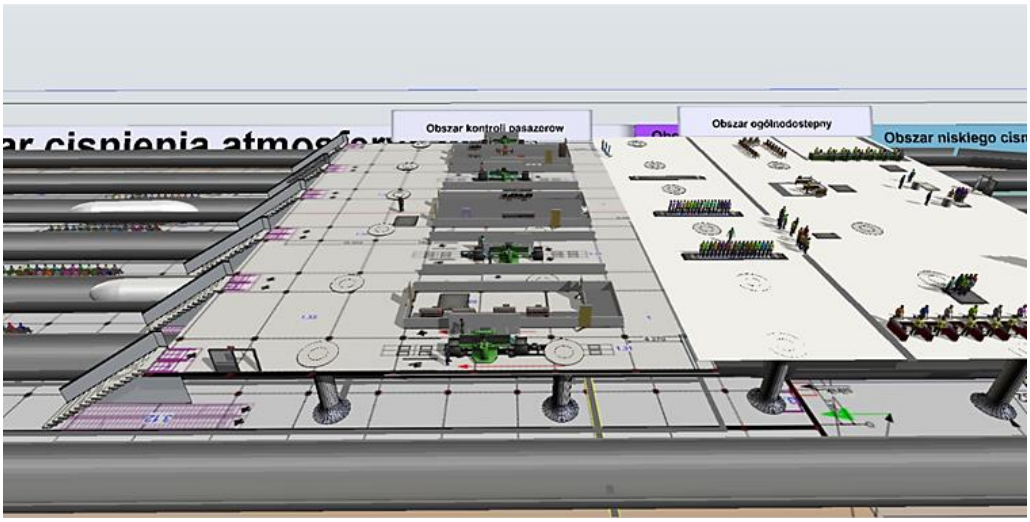

Figure 1. Presented concept of hyperloop station simulation made in Flexsim software.
Source: own elaboration.

Visual and interactive simulations made it possible to visualize the sequence of events, both temporally and spatially, thereby improving the identification of bottlenecks and the optimization of the passenger dispatch process. The adaptability and precision of FlexScript is crucial for testing various scenarios and refining the model to achieve more accurate and relevant outcomes (Nordgren, 2002).

3.1. Reinforcement learning integration

The framework, structured in multiple layers, leverages the Open AI Gym library (currently Gymnasium library) to develop, evaluate, and enhance reinforcement learning algorithms, supporting essential data transformations and complex array management through NumPy for real-time processing (see: Terry et al., 2020). It was than integrated with standard Python libraries such as os, subprocess, and socket to manage system processes and facilitate network communication, ensuring robust infrastructure for continuous data exchange and command operations with FlexSim. The interaction itself is enabled by the Python socket library, allowing real-time communication with the FlexSim application.

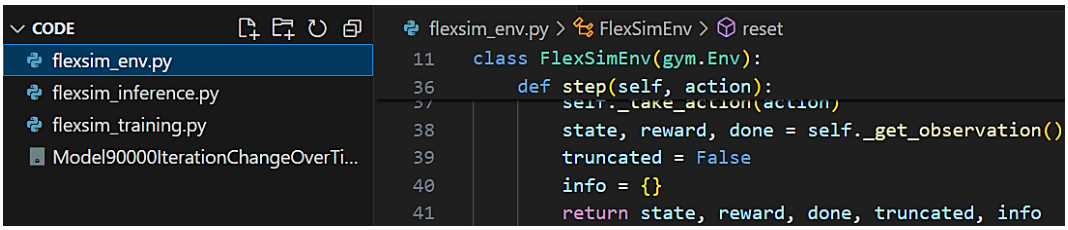

Figure 2. Code snippet from flexsim_env.py script in Visual Studio.
Source: own elaboration.

The architecture is based on FlexSimEnv (Figure 2), a custom Gym environment designed to interact seamlessly with FlexSim, employing methods like reset and step to adjust simulation states and manage interactions for consistent control (see: Yao, Chen, Zuo, 2014). In each simulation step, the function performing the step returns the state of the environment and the reward function. In addition, a rendering function is called, visualizing the progress of the simulation in the FlexSim environment.

The integration of the FlexSim environment with the OpenAI Gym API allows for increased productivity and efficiency of the system by employing reinforcement learning algorithms such as Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C), Deep Q-Network (DQN), and Soft Actor-Critic (SAC), which learn optimal policies through environmental interactions (Figure 3). The implementation of these algorithms is facilitated by Stable Baselines 3, which supports model training and comparison, leveraging PyTorch and TensorFlow to accelerate processing via GPU.

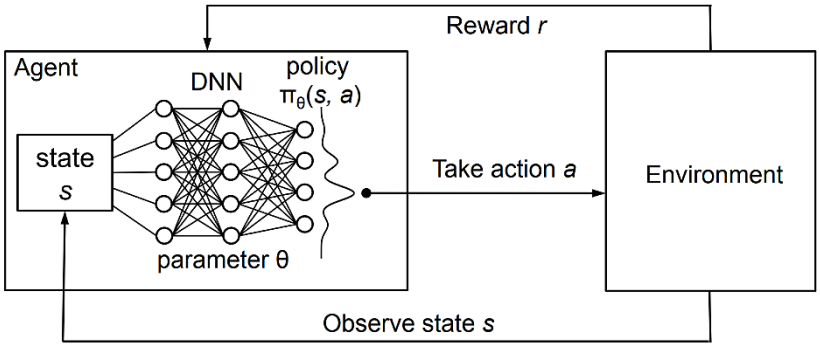

Figure 3. Reinforcement Learning with policy represented via DNN.
Source: Mao, Alizadeh, Menache, Kandula, 2016.

The training progress is monitored through Stable Baselines 3 custom callbacks using dynamically logging rewards. CUDA-enabled GPUs expedite computations through parallel processing, optimizing the handling of large models and datasets. The modularity of the code allows for seamless transitions between different RL algorithms by simply updating parameters in the setup function, significantly streamlining the testing process. Finally, training visualization plots average rewards against steps, providing critical insights into the effectiveness of the chosen algorithms and facilitating ongoing optimization

3.2. Model inference

Once the machine learning model has been trained it can be integrated with a Python-based HTTP server in order to enable dynamic, real-time interactions using data formatted in JSON. These data are then analyzed through Stable Baselines3, ensuring efficient predictions from incoming data streams (see: Seyidova, Shakhayev, 2023). Technically a special FlexSimInferenceServer class adeptly handles HTTP GET and POST requests and convert JSON formatted data into Python's numpy arrays using transformation functions, and subsequently generating predictions with a Stable Baselines3 model, thereby returning the

outcomes back in JSON to support ongoing machine learning processes (Raschka, see: Patterson, Nolet, 2020). The main function initializes the server with a pre-trained PPO model and manages its operations until a shutdown signal is received, ensuring continual system oversight and effective real-time response capabilities (see: Raffin et al., 2021).

3.3. Reward function calucation

This study developed a dynamic reward system through a sequence of steps, each crucial for system functionality. First the system initializes objects representing people in a given colour. Concurrently, passengers are assigned with colours as codes for their destination – red (1), green (2), blue (3), yellow (4), orange (5). The same goes for pallets functioning as pods (Figure 6). This gives quick identification and enables easy tracking of system conditions. The system calculates the number of people linked to a given colour (pallets). Then the capsule capacity difference (places left) is computed by subtracting number of people from the maximum capsule capacity, which in the simulation was 25.

Name	Value
LastItemType	3
People queue	28
Red people	11
Green people	12
Blue people	2
Yellow people	3
Orange people	0
Red pallet	58
Green pallet	69
Blue pallet	69
Yellow pallet	72
Orange pallet	78

Figure 4. Parameters and values in Flexsim Hyperloop Station simulation's observational space. Source: own elaboration.

The reward function model used in the training phase is defined as follows:

Having:

c – capacity difference (25 – number of people),

a – average number of people in the queue,

n_i – number of people of colour i in the queue, where $i \in$ {red, green, blue, yellow, orange},

n – total number of people,

C – capsule capacity (in the simulation $C = 10$),

v – value of the reward for having more people than the average (in the simulation $v = 5$),

A – the maximum reward function value (in the simulation $A = 10$),

The reward r is calculated as follows:

$$r = \begin{cases} A & \text{if } c = 0 \\ A - c & \text{if } 0 < c \leq 5 \\ \dfrac{1}{C + 1 - c} & \text{if } c > 5 \end{cases} \quad (1)$$

If the capacity difference c is 0, indicating full capacity, the reward is maximal (equal to A = 10). If the capacity difference c ranges from 1 to 5, the reward decreases linearly from A. If the capacity difference exceeds 5, the reward is inversely proportional to the capacity difference c beyond the threshold of the maximum capsule capacity C. The piecewise function plot of reward function calculation is shown in Figure 5.

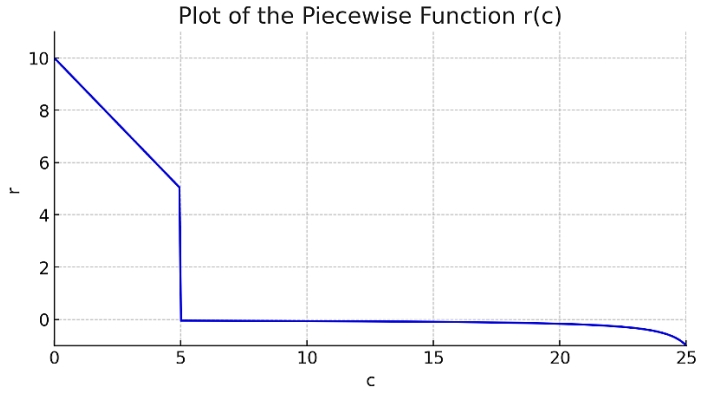

Figure 5. Plot of Piecewise Function r(c) of reward function calculation.

Source: own elaboration.

Reward adjustments are made based on action alignment and whether the combined total of number of people n and those associated with a specific colour n_i is within average number of people a.

For each colour $i \in$ {'red', 'green', 'blue', 'yellow', 'orange'} the reward function r is adjusted as follows:

$$r = \begin{cases} r - v & \text{if action} = color\ i \text{ and } n_i + n \leq \text{a} \\ r + v & \text{if action} = color\ i \text{ and } n_i + n > \text{a} \end{cases}$$

Rewards decrease by v = 5 if conditions are met, otherwise, they increase by the same amount. These calculated rewards update the node, influencing the system's overall reward value based on the evaluated parameters.

4. Results

The transition from conventional First In, First Out (FIFO) logistics, which lacked adaptability to variable demand, to the use of the Reinforcement Learning (RL) model represents a significant advancement. The RL model improves system efficiency by dynamically adjusting strategies based on real-time feedback, a necessary adaptation FIFO fails to make. Although initial RL applications demonstrated suboptimal results due to erratic behaviors, adjustments to the gamma parameter, which balances immediate versus future rewards, enhanced stability and performance. Early in the training phase, a sharp increase in average rewards indicates successful learning and performance improvements, despite fluctuations from strategy exploration. As training progresses, rewards continue to increase but at a slower rate, suggesting a stabilization in learning as strategies are refined towards optimal performance.

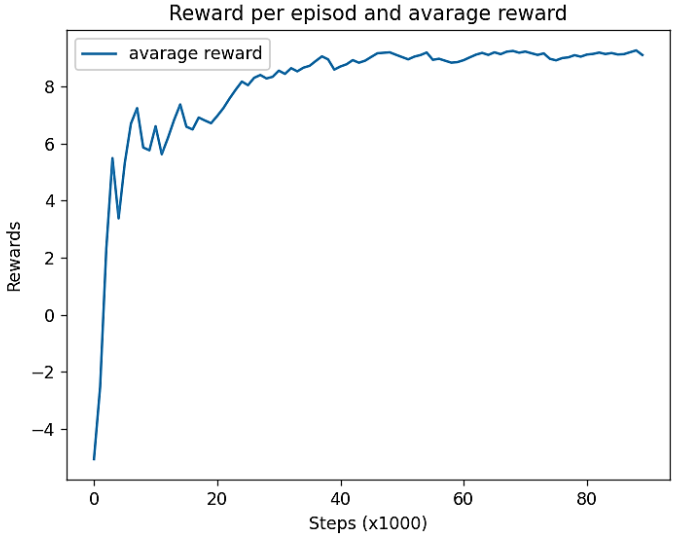

Figure 6. Reinforced Learning Step Count vs. Average Reward.

Source: own elaboration.

In the later stages of training, specifically between 60,000 and 90,000 steps, the agent's performance exhibits minimal reward increases, stabilizing at a high level and suggesting a plateau in learning where further improvements may be negligible. This stabilization, characterized by minor fluctuations, indicates that the model is approaching peak performance, rendering additional training potentially redundant. Through iterative parameter optimization, particularly adjustments to the gamma parameter that significantly influences the discounting of future rewards, the model's performance and stability have improved. This enhancement is evidenced by a consistent rise in capsule fill statistics, reflecting the model’s increased efficiency and accuracy in the simulation scenarios and a reduction in variability across iterations, underscoring the necessity of fine-tuning of the reinforcement learning models. The training, conducted over four stages: 2048; 10,000; 40,000 and 90,000 steps, shows that

the designed model, intended for capsules with a 25-passenger capacity, achieves an average fill of 21.26 passengers, markedly surpassing the efficiency of simpler FIFO method applications.

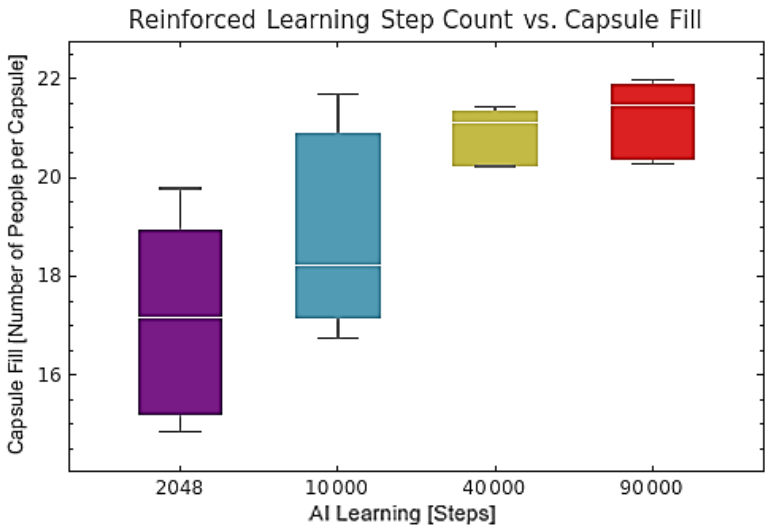

Figure 7. Box and whisker plot of RL Count vs. Capsule Fill.

Source: own elaboration.

In a research study examining reinforcement learning models, initial results showed a wide range in capsule fills with a lower median at 2048 steps. As the model progressed to 10,000 steps, both the median fill and its range increased, indicating improvements. By 40,000 steps, there was a noticeable refinement in prediction accuracy, evidenced by higher median fills and reduced variability. At 90,000 steps, the model's performance stabilized significantly, achieving a median fill close to the target of 25 passengers and minimal variability, suggesting that the learning was nearing a plateau (Figure 10). Throughout the learning process, significant enhancements were observed across various metrics such as minimum, first quartile, median, third quartile, and maximum, with a decrease in the interquartile range indicating a decrease in variability and a stabilization of results. These observations underscore the importance of parameter optimization in enhancing the performance of reinforcement learning models.

5. Discussion

This research explores enhancing Hyperloop station operations using a Reinforcement Learning (RL) model, employing the Stable Baselines3 library and Proximal Policy Optimization algorithm within a FlexSim simulation. The goal is to optimize efficiency and reduce departure timetable variability amid fluctuating passenger flows by programming the RL system to prioritize the dispatch of Hyperloop capsules, tailored to transport up to 25 passengers to five designated destinations based on passengers' shirt colours. Throughout

an 8-hour simulation, the model demonstrated rapid learning gains, achieving peak performance with consistent capsule fill rates and decreased variability. Furthermore, fine-tuning the gamma parameter significantly enhanced the stability and effectiveness of the model, highlighting the potential of RL to improve operational dynamics and environmental sustainability in complex transportation systems.

Proximal Policy Optimization (PPO) navigates the balance between exploration and exploitation within Reinforcement Learning (RL) algorithms, necessitating comparative analyses with other RL techniques like Advantage Actor-Critic and Deep Q-Networks to determine the most effective under varying conditions. Enhanced by automated tuning methods such as Bayesian optimization and genetic algorithms, these models gain robustness by minimizing biases and optimizing performance efficiently. Expanding simulation models to encompass additional destinations, processors, tracks, and capsules enables a deeper evaluation of system capacity and efficiency, particularly vital in high-speed transportation systems where optimizing passenger wait times and capsule utilization is critical for efficiency and satisfaction. Additionally, incorporating variability in passenger arrivals, such as during peak periods, improves the adaptability of these strategies to dynamic demands, increasing the realism and utility of the models. Future research should validate these strategies while exploring cargo logistics optimization using RL, which takes into account operational factors like delivery windows and cargo priorities, ensuring comprehensive improvements in both passenger and cargo transport sectors.

6. Conclusions

This study examines the transformation from a traditional First In, First Out (FIFO) logistics model to a dynamic, adaptive system using Reinforcement Learning (RL), which continually adjusts to fluctuating demands and priorities, thereby enhancing operational efficiency. Implementing RL within the FlexSim environment has markedly increased operational efficiency in hyperloop transportation systems by optimizing time management, reducing costs, and increasing throughput; notably, the average passenger capacity per vehicle has risen significantly. Initial training phases demonstrate rapid learning, with rewards peaking during early stages and stabilizing as the system approaches near-optimal performance. Further refinements in the gamma parameter have improved stability across various training milestones, underlining RL's potential in achieving highly efficient and adaptable transportation solutions.

This research not only contributes to the discourse on the potential applications of artificial intelligence, particularly reinforcement learning in transport systems, but also aims to improve the user experience of transport systems, as in the hyperloop under study, by optimizing operational indicators. Despite promising simulation results, the need for real-world validations

remains, highlighting the importance of further studies that extend these findings through practical applications and broader algorithm comparisons.

References

1. Almujibah, H., Kaduk, S., Preston, J. (2020). *Hyperloop – prediction of social and physiological costs, 6*, 43-59. https://doi.org/10.17816/TRANSSYST20206343-59.
2. Barbosa, F. (2020). *Hyperloop Concept Technological and Operational Review: The Potential to Fill Rail Niche Markets*. https://doi.org/10.1115/jrc2020-8033.
3. Chen, H., Yang, D., Sun, H. (2009). *Simulation research of the Mixed-Model production line based on Flexsim.* 16th International Conference on Industrial Engineering and Engineering Management, 1876-1881. https://doi.org/10.1109/ICIEEM.2009.5344303.
4. Garrido, J. (2009). *Introduction to Flexsim*, 31-42. https://doi.org/10.1007/978-1-4419-0516-1_3.
5. Kirschen, P., Burnell, E. (2021). Hyperloop system optimization. *Optimization and Engineering, 24*, 939-971. https://doi.org/10.1007/s11081-022-09714-7.
6. Kupriyanovsky, V., Klimov, A., Alenkov, V., Pokusaev, O., Dobrynin, A. (2020). Hyperloop - current status and future challenges. *International Journal of Open Information Technologies, 8,* 129-144.
7. Marquis, P., Papini, O., Prade, H. (2020). *Elements for a History of Artificial Intelligence*. 1-43. https://doi.org/10.1007/978-3-030-06164-7_1.
8. Mitropoulos, L., Kortsari, A., Koliatos, A., Ayfantopoulou, G. (2021). The Hyperloop System and Stakeholders: A Review and Future Directions. *Sustainability*. https://doi.org/10.3390/su13158430.
9. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529-533. https://doi.org/10.1038/nature14236.
10. Nordgren, W. (2002). Flexsim simulation environment. *Proceedings of the Winter Simulation Conference, 1*, 250-252, https://doi.org/10.1109/WSC.2002.1172892.
11. Pal, S. (2023). Steer Towards Sustainability: The roadmap to Cost and Eco-Efficient Transportation via AI-Enhanced Routing. *International Journal for Research in Applied Science and Engineering Technology*. https://doi.org/10.22214/ijraset.2023.57467.
12. Premsagar, S., Kenworthy, J. (2022). A Critical Review of Hyperloop (Ultra-High Speed Rail) Technology. *Urban and Transport Planning, Technical, Environmental, Economic, and Human Considerations, 4*. https://doi.org/10.3389/frsc.2022.842245.

13. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *J. Mach. Learn. Res., 22,* 268:1-268:8.
14. Raschka, S., Patterson, J., Nolet, C. (2020). *Machine Learning in Python: Main developments and technology trends in data science, machine learning, and artificial intelligence.* ArXiv, abs/2002.04803. https://doi.org/10.3390/info11040193.
15. Ru, S. (2012). Study on Flexsim-aided Logistics Engineering Course Teaching. *Logistics Technology*.
16. Seyidova, I., Shakhayev, S. (2023). Big data processing with Python in social networks. *ETM – Equipment, Technologies, Materials*. https://doi.org/10.36962/etm16042023-76.
17. Sun, Q., Han, S., Zhou, J., Chen, Y., Yao, K. (2022). *Deep Reinforcement-Learning-Based Adaptive Traffic Signal Control with Real-Time Queue Lengths*. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 1760-1765. https://doi.org/10.1109/SMC53654.2022.9945292.
18. Sutton, R., Barto, A. (1998). Reinforcement Learning: An Introduction. *IEEE Trans. Neural Networks, 9,* 1054-1054. https://doi.org/10.1109/TNN.1998.712192.
19. Terry, J., Black, B., Hari, A., Santos, L., Dieffendahl, C., Williams, N., Lokesh, Y., Horsch, C., Ravi, P. (2020). *PettingZoo: Gym for Multi-Agent Reinforcement Learning*. ArXiv.
20. Wakefield, C., Sonder, H., Lee, W. (2001). Installing and Configuring VB.NET, 91-143. https://doi.org/10.1016/B978-192899448-0/50008-2.
21. Xing-hua, W. (2009). Research on Simulation and Optimization of Warehouse System Based on Flexsim Software. *Logistics Technology*.
22. Yao, M., Chen, X., Zuo, L. (2014). Time Step Method of Computer Simulation Process Control. *Applied Mechanics and Materials*, 494-495, 1257-1261. https://doi.org/10.4028/www.scientific.net/AMM.494-495.1257.
23. Yi-jun, Z. (2011). Design of optimized product lines based on flexsim. *Journal of Beijing Information Science & Technology University*.
24. Zerilli, J., Danaher, J., MacLaurin, J., Gavaghan, C., Knott, A., Liddicoat, J., Noorman, M. (2021). What Is Artificial Intelligence, 1-19. https://doi.org/10.7551/MITPRESS/12518.001.0001.