一般社団法人 電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report

Stochastic Discrete Event Simulation Environment for Autonomous Cart Fleet for Artificial Intelligent Training and Reinforcement Learning Algorithms

Naohisa HASHIMOTO‡ Ali BOYALI† Shin KATO†

Takao OTSUKA‡ Kazuhisa MIZUSHIMA‡ Manabu OMAE‡

†National Institute of Advanced Industrial Science and Technology 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan

‡Keio University 144-8 Ogura, Saiwai, Kawasaki, Kanagawa, 212-0054, Japan

E-mail: †{naohisa-hashimoto, ari-boyari, shin.kato}@aist.go.jp ‡{otk,mizkaz,omae}@sfc.keio.ac.jp

Abstract In this report we give details of a Discrete Event Simulation (DES) framework coded in Python environment for simulation and analysis of a customized Personal Rapid Transport (PRT) with passenger behavior. The prior analysis of the system is a must before deployment of the autonomous PRT cars (carts) to make decision of initial investment, number of carts required and to design supervisory control algorithms that reduces the defined cost functionals in the optimization. The simulation program coded in consideration of training Artificial Intelligent (AI) agents by Deep Reinforcement Learning methods similar to OpenAI Gym environment. The basic requirements for modeling of discreet stochastic simulation and analyses are summarized in the report.

Keywords Rapid Transit System Simulation, Empty Vehicle Relocation, Reinforcement Learning based Supervisory Control

1. INTRODUCTION

Although dates back to mid 1950's [1], deployment of Personal Rapid Transport Systems (PRTs) in which many small or medium-sized autonomous vehicles employed has gained impetus in the recent years. The main motive under the revival of autonomous PRTs are the improvements in Electric Vehicle (EV) and battery technologies and the Artificial Intelligence studies with a growing body of research.

In PRTs, depending on the scale and topology of the systems, the control of the many autonomous cars (carts or minibuses); finding an optimal strategy for the next action of the individual cars is by no means of trivial task.

The configuration of the PRT systems vary depending on the aim, system constraints and where they are deployed. These issues make the design of generalized control systems hard with an additional burdens coming from the stochastic nature of the passenger and cart interaction, passenger arrival pattern spread out of the year and passenger group behavior [2].

During the design phases of PRTs, from concept to deployment, many questions arise that demand answers. The typical questions are;

1. How to keep the maintenance and operation cost minimum without compromising passenger satisfaction by keeping the waiting durations minimal?,
2. How should the individual cars behave when there is a demand or cars are empty?
3. What will be the average waiting duration of the passengers in the system given the passenger arrival patterns?,
4. How many cars should be deployed for the optimization targets?,
5. What will be the average queue length, throughput or system (server) utilizations?,
6. What will be the exceedance probabilities (given the system, probability of passengers waiting more than a given threshold) [3].

These questions are not easy to answer before constructing a simulation environment and performing rigorous analyses on the realistic scenarios. The response to these questions can be NP-hard as the state space of the system grows exponentially which we will detail in the proceeding sections.

In this study, we build a Discrete Event Simulation (DES) system to obtain answers to our design questions for the passenger - autonomous cart system and implement and Artificial Intelligence (AI) environment similar to OpenAI Gym [4] which is used to train and test various Reinforcement Learning algorithms. The discrete event simulations have broad range of application area in which there is no single answer can be obtained by analytical methods. In general the design questions fall under the queuing theory. In simulation based methods, the system model is constructed then the various scenarios tested by the large amount of simulations.

The rest of the paper is organized as follows; In Section 2, we briefly detail on the structure and topology of the passenger - autonomous cart and the constraint of the system, then we present how we coded this environment to train the recent deep reinforcement learning algorithms, implement various analyses on the simulation environment and our finding on system bottlenecks that require design modifications. We conclude paper with proposals to model the passenger arrival patterns of the given geographical location with the burgeoning probabilistic programming frameworks and libraries with flexible hierarchical Bayesian models.

2. PRT System Description

In Japan, several researches for personal mobility has been done[5]. One of the several projects, the new PRT systems are planned in three locations in Japan. We report the simulation results using the system topology of the Okinawa district where eight stops are defined to pick-up and drop-off the passengers. The design problem is to make decision about how many autonomous cars will be used given the upper level of design parameters such as the minimum level of perceived waiting duration from the passenger side, shown in Fig.1.

Perceived waiting duration, stop arrival means, inter-stop transfer probability distributions will be determined by additional studies and are not reported in this study. However, we will summarize a road-map for approximating passenger arrival means in the conclusion section.

Our aim is to construct a simulation platform to be able to make decisions prior to deployment of the system and prepare a playground for artificial intelligent and machine learning algorithms.

The autonomous cars are modified from four and six seater golf carts. The maximum speed of the carts is adjusted as 12 km/h with the 1 m/s^2 acceleration and deceleration profile. All the carts are launched from the first stop (in programming environment 0th stop) where carts can park indefinitely. This stop is also called capacitor [6]. The carts move autonomously in the PRT system ring shaped path on which the eight cart stops are defined. The carts are autonomous and follow a magnetic track on the ring path. The total length of the track length is 2184 m. It takes around 11 minutes to complete the ring with acceleration and deceleration profile non-stop to complete the ring for a cart.

The distance between the cart stops are given in Table I. In the simulation model, the capacity of the golf carts is coded as a variable. In the reported simulations we set the number of seats that can carry passengers as four which include the driver seat. The system is highly constrained in the initial setup phase. For instance, the active carts (which are on patrol picking up and dropping of the passengers) cannot take-over the proceeding carts in front, therefore must wait in the previous stop for the next station if it is occupied. The PRT track is one-directional and circular. The carts check if there is a cart following or waiting at the previous stop, if there exists it has to proceed in order to allow the traffic flows. This rules do not apply at the 0th stop where the passive carts can wait indefinitely and parked.

Table I. Inter-stop Distances

Stops	0-1	1-2	2-3	3-4	4-5	6-7	7-0
Dist. - [m]	435	352	180	170	190	393	435

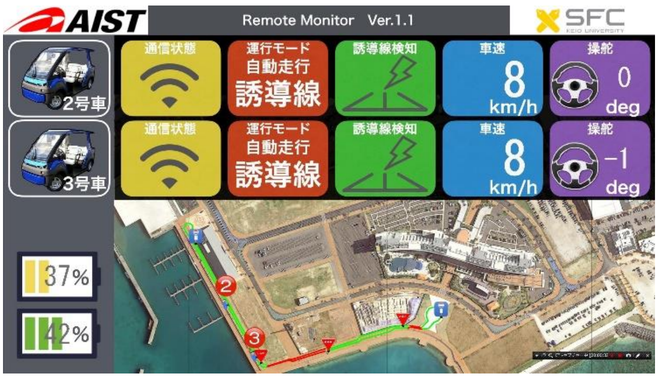

Figure1 System Interface Image

3. Simulator Design

3.1. Discrete Event Simulations

Simulation is a tool when the system models are complicated to get answers on the behavior of the whole system when the analytical tools such as linear and dynamic programming are not sufficient [7]. Building a Discrete Event System (DES) simulation environment allows the designers to test the system behavior under different conditions and modifications prior to deployment of the actual system into the service.

Even with a simple topological structure and low number of cart stop, our Okinawa model is a complex system in which the passenger arrival mean of the stations (stops) are unknown. In transportation systems models, the passenger arrival rates are modeled as the Poisson Arrival process [8-9] which vary for time of the day, day of the week and week of the year. On the other hand, the passenger group behavior brings additional complexity into the systems which requires compound Poisson arrival distributions in which passengers may arrive not only individually but also in groups of two, three and four or more. The carts in the system pick passengers accordingly and the passenger get-in and alighting durations are also drawn from a probability distributions.

There are commercial simulation software in the market are costly and might not provide freedom for adding special mathematical formulations such as complex cost functions for optimization, highly constraining rules and customized environments. A naive approach for the simulation would be using flexible coding environments from scratch for these kind of requirements. In this study we modeled the PRT system in Python environment to obtain an Artificial Intelligence (AI) playground to train Reinforcement Learning (RL) agents for optimal decision making.

In Discrete Event Simulation (DES), events are scheduled in an event calendar and the required actions at the event time are executed [10]. Instead of advancing with a fixed time interval, the time of the simulation advanced to the next event time allowing faster simulations. In Python programming, the event calendar can be constructed by priority queues [11]. At the current event time, after executing the necessary actions in the system such as picking-up passengers, dropping-off, and the next events are scheduled and put in the event calendar. The simulations end when the pre-specified conditions are met.

3.2. Event Processes

The DES system is coded in Python by using Object Oriented Programming (OOP) framework. The system object components in the PRT system are the passenger, autonomous cart and cart stop objects.

The environment object is composed of these three components. The event calendar is composed of a priority queue in Python which store the events according to time schedule, retrieve events from the queue when the event time is reached. The events are produced by Python co-routines in which generators are used to generate the next event at the event execution time of the related process. The co-routines use processes in the simulations. There are two processes in our system. The first process is the process of the cart events by which the next events such as decision, passengers boarding and alighting events are brought from the queue at the current event time and the new events are put in the event calendar. For example, when the carts approach a cart stop, the next event will be stopping or passing the stop without stopping. If the stopping event is put in the event calendar at the previous event execution time, the cart is going to stop at the next stop and execute the necessary actions related to passenger's interactions. The event, actions and cart state space variables are given in Fig.2. The cart stop process only produces passenger arrival event and put the passengers in the stop queue. These two processes and decision constraints are summarized in Fig.3.

In the processes random distributions are used. In queue and probability theory arrival processes are the modeled by Poisson and the time duration between the arrivals by Exponential distributions.

$$p(k) = \lambda^k \frac{e^{-\lambda}}{k!} \quad \text{and f(x, } \lambda = \begin{cases} \lambda e^{-\lambda x} if x \geq 0 \\ 0 else \end{cases} \tag{1}$$

where k is the number of arrivals in the given interval and λ is the mean number of arrivals and x is the random time interval. The time unit is flexible and mean number of arrivals can be normalized with the chosen time units. In our simulation we choose an upper and lower limit for the mean number of arrivals for each cart stops. At the beginning of the each simulation epoch, a random vector of the mean arrivals are generated in the interval of [0, 6] and the generated mean arrival rates are used for all hours of the day. In realistic simulations the mean arrival rate for the hours must be approximated. This can be done by the probabilistic programs using the promising probabilistic programming frameworks such as PYMC3, Edward and Bambi libraries in Python [12-14].

The PRT system model is semi-Markov processes [8] with the random time interval arrival process the random variables of which are given in Fig.4. In Fig.4., S is the time of the arrivals, $N(t)$ is the counting process showing the number of arrivals and X is the arrival interval generated by the exponential distribution [8].

The other events that depend on the random numbers are the destination of the passengers generated by the stops, get-in and alighting times. When the passengers are generated, the destination stop of the passengers are drawn from the Uniform distribution. At the passenger generation time, an additional probability distribution is used to decide the number of passengers to be generated who will act and travel together. The maximum number of group is chosen as four which represent a nuclear family with the probability 5% of the total number of passengers. In addition, a group of three people and couples are also included with the probabilities 5 and 10%. The rest of the passengers are the individuals with the probability of 80%. The Poisson process in which group behaviors are included is called as compound Poisson process [15]. As the autonomous carts have four door exits, the get-in and alighting time for each passengers can be produced by truncated exponential distribution. In sequential get-in and alighting situation such as in the bus queues passengers get-in the bus from a single door or alight. For this situations the distribution of independent get-in times for each passengers are generated by Erlang distribution, however there is no get-in and alighting queue for our cars. We truncated the exponential distribution in the interval [2, 8] seconds. The lowest time unit of the simulation is second. Environment object is a collection of passenger, stops and cart objects in which the start and end time of the simulations are defined. The PRT system starts to operate at 07:00 AM and end the operations at 9:00 PM. There is no new event is scheduled after this hour.

```
_cartEvent = namedtuple('cartEvent', 'time cartID event_type info')

_cart_event_space = {0: 'decision_time_stopped', 1: 'alighting', 2: 'boarding',
                     3: 'decision_time_motion', 4: 'car_arrival_stop', 5: 'car_arrival_past'}

_cart_state_space = {0: 'Stopped', 1: 'In_Motion'}
_cart_inverse_state = {'Stopped': 0, 'In_Motion': 1}
_cart_action_space = {'Stopped': ['move', 'wait'],
                      'In_Motion': ['continue_past_next', 'stop']}
```

Figure 2. Event, action and cart state space Python variables

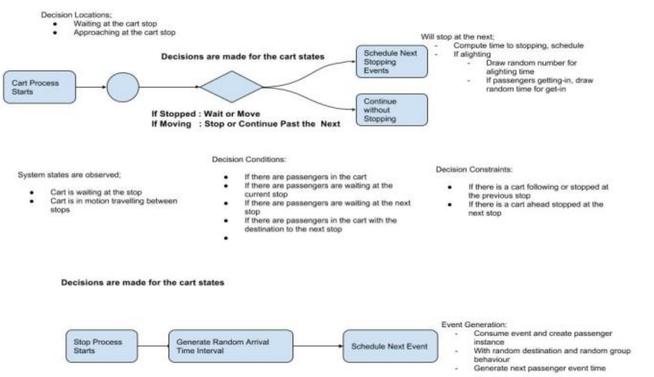

Figure 3. Event Calendar Processes

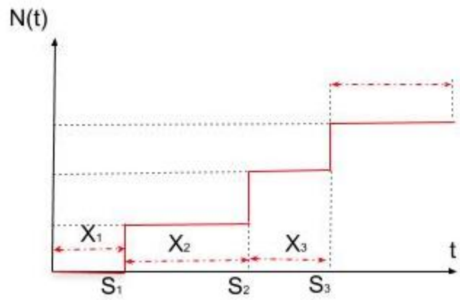

Figure 4. Arrival Process (Semi-Markov)

4. Simulations and Results

The PRT system is composed by stochastic processes. In order to assess the variables, performance measures of the system single simulation does not give reliable assessment as the whole system is random process. Therefore, a large number of simulations must be run and averages of the variables of the interest should be reported. In this study we report the results typical to queuing theory such as average cart occupation, average waiting durations of the passengers, queue lengths of the stops, maximum and minimum values of these variables and some other statistical measures. In the Python environment we keep statistics using the Pandas (Python Data Analysis) library for each objects in the program. Various tables and figures can be generated using Pandas in Python.

In Fig.5, average passenger waiting times of the passengers originated from each stops are shown for reactive action selection for single simulation run which corresponds to one day of the PRT system. The figures are grouped by the group behavior of the passengers. The overall mean waiting time of the passengers are given in Fig.6. The average waiting time of the all passengers in the system is realized as 13.66 minutes for this simulation. There are many other statistics, related tables and figures are generated for the simulation, however to the page limitation, it is not possible to summarize all here.

We run a large number of simulations (400 days) for all the action selection mechanisms and summarized the results in the following table for two active carts in the system for three action selection mechanisms. Please note that the number of cart is a variable in the simulations and can be changed flexibly.

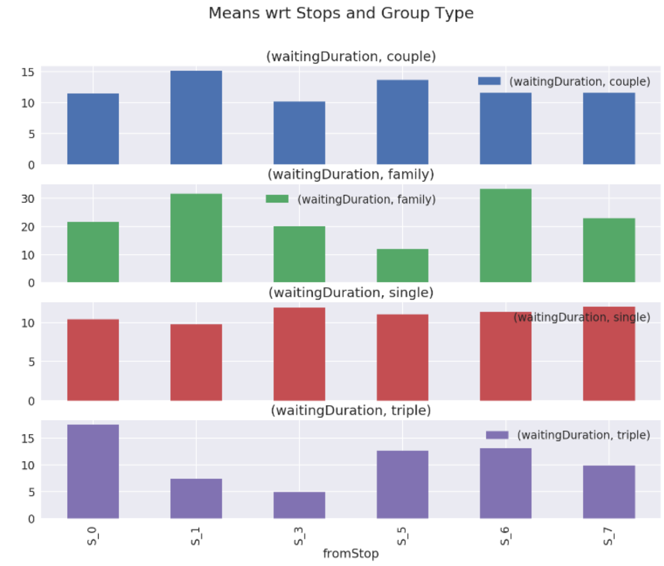

Figure 5. Average Passenger Waiting Times at each Stop with Groups

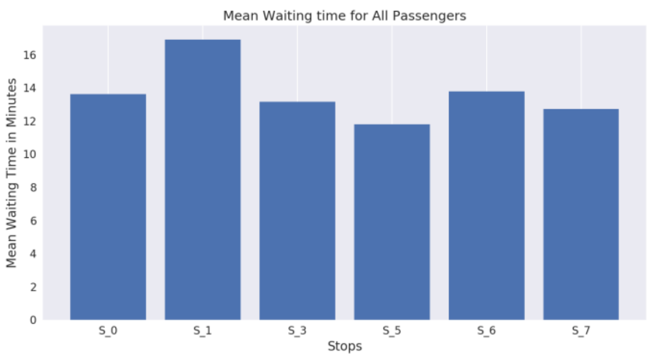

Figure 6. Overall Average Passenger Waiting Times at each Stop

Table II. Overall Averages for Different Action Selection Methods

	Avg. Wait. [min]	Avg. Occupancy %
Reactive	14.13	63
Random	14.14	56
Reinf. Learning	10.55	81

5. Conclusion

In this report, we presented a PRT system simulation framework which can be used as a training environment for AI algorithms and give brief summary of the statistical results. We run a large number of results to asses the effects of the variables for different action selection mechanisms in the highly constrained initial setup of the PRT system. As given in Table II, the RL action selection, the details of which are not discussed here, reduce the overall average waiting time of the passengers up to 3.6 minutes.

文　献

[1] Anderson, J. Edward. "A review of the state of the art of personal rapid transit." Journal of Advanced Transportation34.1 (2000): 3-29.

[2] Daszczuk, Wiktor B. "Discrete event simulation of Personal Rapid Transit (PRT) systems." arXiv preprint arXiv:1705.05237 (2017).

[3] Fishman, G.S, Discrete-event simulation: modeling, programming, and analysis. Springer Science & Business Media, 2013

[4] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. and Zaremba, W., 2016. OpenAI gym. arXiv preprint arXiv:1606.01540

[5] N.Hashimoto, K.Tomita, A.Boyali, Y.Takinami, O.Matsumoto, “Experimental Study of the Human Factors when Riding an Automated Wheelchair –Supervision and Acceptability of the Automated System”, IET Journal of Intelligent Transport Systems, 2017.

[6] Daszczuk, Wiktor B. "Discrete event simulation of Personal Rapid Transit (PRT) systems."arXiv preprint arXiv:1705.05237 (2017).

[7] Hira, D.S., 2008. Operations research. S. Chand Publishing.

[8] Gallager, Robert G. Stochastic processes: theory for applications. Cambridge University Press, 2013.

[9] Stewart, William J. Probability, Markov chains, queues, and simulation: the mathematical basis of performance modeling. Princeton University Press, 2009.

[10] Tyszer, Jerzy.Object-oriented computer simulation of discrete-event systems. Vol. 10. Springer Science & Business Media, 2012.

[11] Ramalho, Luciano.Fluent Python: clear, concise, and effective programming. " O'Reilly Media, Inc.", 2015.

[12] Salvatier, John, Thomas V. Wiecki, and Christopher Fonnesbeck. "Probabilistic programming in Python using PyMC3." PeerJ Computer Science 2 (2016): e55.

[13] Tran, Dustin, et al. "Edward: A library for probabilistic modeling, inference, and criticism." arXiv preprint arXiv:1610.09787 (2016).

[14] Yarkoni, Tal, and Jake Westfall. "Bambi: A simple interface for fitting Bayesian mixed effects models." (2016).

[15] Karlin, Samuel. A first course in stochastic processes. Academic press, 2014.

[16] Lees-Miller, John D. "Minimizing average passenger waiting time in personal rapid transit systems." Annals of Operations Research 236.2 (2016): 405-424.

[17] Lees-Miller, John D., and R. Eddie Wilson. "Proactive empty vehicle redistribution for personal rapid transit and taxis." Transportation Planning and Technology 35.1 (2012): 17-30.

[18] Crites, Robert H., and Andrew G. Barto. "Elevator group control using multiple reinforcement learning agents."Machine learning 33.2 (1998): 235-262.