Bulletin of Networking, Computing, Systems, and Software – www.bncss.org, ISSN 2186-5140
Volume 9, Number 1, pages 23–24, January 2020

Learning course features with Mini 4WD simulator

Yuji Onoo
Faculty of Science and Engineering
Saga University
Saga, Japan
18573005@edu.cc.saga-u.ac.jp

Nobuhiko Yamaguchi
Faculty of Science and Engineering
Saga University
Saga, Japan
yamag@ is.saga-u.ac.jp

Hiroshi Wakuya
Faculty of Science and Engineering
Saga University
Saga, Japan
wakuya@cc.saga-u.ac.jp

Suguru Ueda
Faculty of Science and Engineering
Saga University
Saga, Japan
sgrueda@cc.saga-u.ac.jp

Osamu Fukuda
Faculty of Science and Engineering
Saga University
Saga, Japan
fukudao@cc.saga-u.ac.jp

Hiroshi Okumura
Faculty of Science and Engineering
Saga University
Saga, Japan
okumurah@cc.saga-u.ac.jp

***Abstract*— While several sensor data are required for Mini 4WD AI to learn, acquiring sensor data is time-consuming and takes a very high cost. Therefore, in this study, we developed a simulator for Mini 4WD AI, and learn Mini 4WD AI with the Mini 4WD simulator.**

***Keywords*—Mini 4WD, simulator, Reinforcement learning**

I. Introduction

In recent years, AI(Artificial Intelligence) technology has been developed, and various experiments and researches are being conducted to create and evaluate AI robots. However, creating and evaluating AI robots is generally expensive. Therefore, for the purpose of making inexpensive and familiar things more useful with the power of AI, Mini 4WD(four Wheel Drive) AI competitions have been held [1].

Mini 4WD AI is low strength, and there is therefore a risk of it breaking in the experiments to actually run the Mini 4WD AI. Also, repeating experiments with the Mini 4WD AI requires several manual operations, such as changing the battery or returning to the course when going out of the course. To overcome these problems, we develop a simulator for the Mini 4WD AI and let the Mini 4WD AI learn the course features by the Mini 4WD simulator.

II. Learning environment

In this paper, we develop a simulation environment by using Unity [2] and Blender [3]. Unity is a game engine and is used in various fields. Unity incorporates PhysX as a physics engine, which enables physical operations such as gravity, collision, and friction. Blender is an open source 3D computer graphics software. The created model by Blender can be exported to various software. We use Blender to model Mini 4WD body, wheels, courses, etc. Blender can express complex surfaces and objects that Unity alone cannot express. In this paper, we use Unity and Blender to develop a simulator for Mini 4WD AI. Mini 4WD AI model parameters used in Unity are summarized in TABLE I.

TABLE I MINI 4WD MODEL PARAMETER

Mass	158
Torque	74
Dynamic Friction (roller)	0
Static Friction (roller)	0
Extremum Slip (tire)	0.4
Extremum Value (tire)	1
Asymptote Slip (tire)	0.8
Asymptote Value (tire)	0.5

III. ML-Agents

ML-Agents [5] is a library for machine learning published by Unity, for example, reinforcement learning using PPO [6], imitation learning to imitate players, and curriculum learning that gradually increases the difficulty of training. The ML-Agents is composed of three objects: Academy, Brain, and Agents. Academy is an object that manages the learning environment, and can set the maximum number of steps of episodes, rendering quality, reset parameters, etc. Brain is an object that determines the behavior of Agents according to the state observed by Agents, and can determine the behavior of multiple Agents by one Brain. Agents are objects that observe the state and determine the action by Brain. The environment gives rewards to the Agents, and the rewards are used to determine Brain's next action [7].

IV. Experiment

In this experiment, we use a Mini 4WD course shown in Fig. 1. This course is a simplified version of the course of the Mini 4WD AI competition in FSS2016 [8], which is changed from lane change to straight parts. We put a jump ramp in the course in such a way so as to go out of the course if the Mini 4WD AI runs at full speed.

This work was supported by JSPS KAKENHI Grant Number 18K11439.

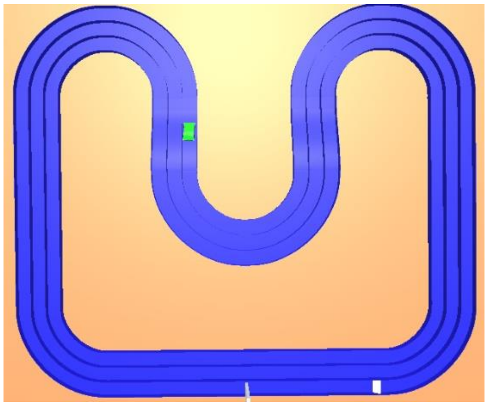

Fig 1 The course layout of Mini 4WD

TABLE II PARAMETERS USED FOR LEARNING

State	X coordinate Z coordinate
Action	-1.0~1.0
Reward	Checkpoint (+1) Course-out (-50) Roll over (-50)

Table II shows parameters used in reinforcement learning. The state consists of (x, z), where x and z are plane coordinates without height information. The action takes continuous values from -1 to 1, and is determined by Brain. The Mini 4WD AI model is controlled by multiplying this action value by torque. For positive rewards, we set 14 check points on the course and give 1 reward when passing each check point. For negative rewards, we give -50 reward when the Mini 4WD AI goes out of the course or rolls over.

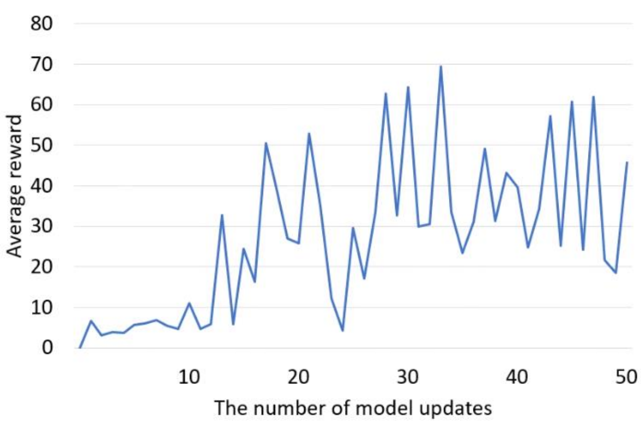

Fig 2 Reward average

Fig. 2 shows the average reward of 8 trials. The horizontal axis represents the number of model updates, and the vertical axis represents the average reward. In this experiment, the model update is performed 50 times per trial when the Mini 4WD AI goes out of the course, rolls over, or time step exceeds a certain number. In the initial phase of the reinforcement learning, the Mini 4WD AI does not know how to get positive rewards and sometimes goes backward. As the Mini 4WD AI model is updated, the Mini 4WD AI can go forward and get more positive rewards. The average reward falls when the number of model updates is over around 20. The reason is that the Mini 4WD AI goes out of the course or rolls over due to excessive speed. However as the Mini 4WD AI model is updated, the Mini 4WD AI can be learned so as to go forward at appropriate speed and so as not to go out of the course.

In order to compare between model after and without learning, we evaluate course-out rate and impossibility rate of running. The impossibility of running occurs when the Mini 4WD AI is caught by the wall of the course or stops a slope due to the lack of speed. The comparison between after and without learning is summarized in TABLE III. From TABLE III, we can see that the model after learning produces lower course-out rate and impossibility rate of running. The course-out rate and the impossibility rate of running decreases by 40% and 8%, respectively.

TABLE III COMPARISON BETWEEN AFTER AND WITHOUT LEARNING

	Course-out rate	Impossibility rate of running
Without learning	46%	22%
After learning	6%	16%

V. CONCLUSIONS

In this study, we developed a simulator for Mini 4WD AI, and learn Mini 4WD AI by the simulator. Our simulator can produce the Mini 4WD AI model that can run at appropriate speed without going out of the course. Experimental result shows that the Mini 4WD AI model learned by our simulator provides better course-out rate and impossibility rate of running.

Our future work is to parallelize learning, that is, to speed up the learning time by learning and integrating multiple Mini 4WD AI models. Another future work is to use the completely same course used in a Mini 4WD AI competition. In this experiment, we use a simplified version of the Mini 4WD course used in a Mini 4WD AI competition, which is changed from lane change to straight parts. We would like to use the completely same course.

REFERENCES

[1] Mini 4WD AI Competition, https://sites.google.com/site/ai4wdcar/

[2] Unity, https://unity3d.com/jp/

[3] Blender, https://www.blender.org/

[4] ML-Agents, https://unity3d.com/jp/machine-learning

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, "Proximal Policy Optimization Algorithms", arXiv:1707, 06347v2, 2017, https://arxiv.org/abs/1707.06347

[6] Eiichi Hurukawa, "Unity machine learning and reinforcement learning starting with Unity ML-Agents practice game programming", 2018, bone digital.

[7] Mini 4WD AI Competition Executive Committee, Mini 4WD AI Conference in FSS2016 Saga

https://sites.google.com/site/ai4wdcar/home/taikai/fss2016/