The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Decision Making for Model Based Design by Reinforcement Learning

Tatsuhide Sakai[*1] Takahiro Inabe[*2]

[*1] Great Wall Motor [*2] Sakiyomi AI labo

The Model Based Design is identified that hierarchy is structured on functions of each part to achieve a competitiveness in a product design. As the hierarchy becomes complicated, design variables have a huge data space, so it is difficult to properly make decisions in a short time even if a designer has extensive experience. It is verified whether Reinforcement Learning is effective for the design of electric vehicles. When applied to the vehicle performance of the top of hierarchy, the design limit of energy consumption was derived from the variables space of 128 to the 17th power and the optimal solution for Package was learned from the variables space of 10 to the 77th power.

1. Introduction

In the product design of automobiles etc., a method based on model base design (hereinafter described as MBD) is often used. The product performance is made hierarchically related to the part function, this method is effective for securing competitiveness. Since it is structured on logical expressions such as design calculations, quantitative consideration is possible, but it is difficult to decide an appropriate design specification because choices are enormous. It is verified whether Reinforcement Learning is effective for decision making of design variables in MBD by using vehicle performance design.

2. Model Base Design

2.1 Electric Vehicle Concept Design

The design objects of vehicle performance are environment, power, dynamic quality, package, strength endurance, and correlation of related functions is complicated and intertwined.

- Level 0: Phenomenon of customer experience
- Level 1: Combination of multiple components which has difficulty with each individual change
- Level 2: Single component which can be changed

It is tried applying Reinforcement Learning to EV design calculation of environmental performance and Package, to realize a long mileage with less battery & a large cabin. (Fig.1)

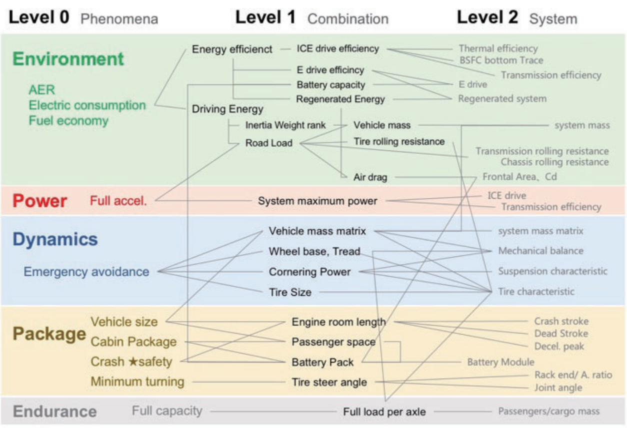

Fig.1 Model Base Design for Vehicle performance

Contact: Tatsuhide Sakai, Great Wall Motor, 2076Chaoyang Nan Da Jie Baoding Hebei China, jiujinglongying@gwm.cn

2.2 Vehicle Energy Design

The evaluation values are energy efficiency, electricity consumption and mileage. It is the design goal to reduce the vehicle driving energy and improve the electric drive efficiency and increase the regenerative energy to increase the mileage by suppressing the battery capacity.

$$E_{Effi} = W/(W_c + W_{re}) \quad (1)$$

$$E_{Cons} = (W_c + W_{re})/L_{all} \quad (2)$$

$$AER = Batt_{capa}/E_{Cons} \quad (3)$$

$E_{Effi}: energy\ efficiency$

$E_{Cons}: electric\ consumption$

$AER: electric\ driven\ mileage$

$W: vehicle\ driven\ Energy$

$W_c: battery\ consumption\ energy$

$W_{re}: regenerated\ energy$

$Batt_{capa}: battery\ electric\ capacity$

Evaluation values are set the three performances on left side of Equation (1), (2) and (3). Design variables are set 17 functions which is related to rolling resistance, aerodynamic resistance, accelerating resistance, electric drive efficiency and regeneration efficiency which are components of evaluation values.

2.3 2.3 Vehicle Package Design

37 design variables are set from the C.G., mass, and geometries of each component and so on. (Fig.2) 12 evaluation values are set as their performances of environment, dynamics, package.

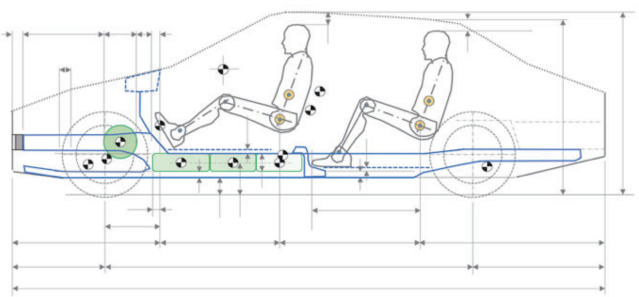

Fig.2 Design variables for vehicle package

The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

3. Reinforcement Learning

3.1 Modeling

Neural Network is individually set for each of Energy Design and Package Design, design variables of Model Based Design are set as input, and performance evaluated values are set as output. (Fig.3)

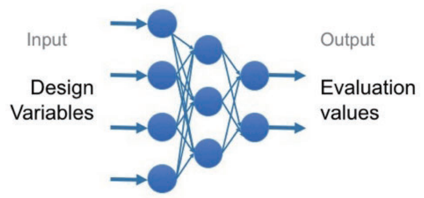

Fig.3 Neural Network for Design Calculations

In order to learn the Neural Network weighting parameters, a value-based function and a policy-based function in the Markov decision process are used.

$$(S_t, a_t) = r_t + \gamma \max_{a_t} Q(S_{t+1}, a_{t+1}) \qquad (4)$$

$$V(S_t) = \max_{a_t} r_t^a + \gamma V(S_{t+1}) \qquad (5)$$

$Q: Value\ based\ function, V: policy\ based\ function$

$S: situation, a: action, r: reward, \gamma: learning\ rate, t: time$

3.2 Variable and Evaluation

The range of the design variable is set from the minimum value and the maximum value. The variable selection can be made between the minimum value and the maximum value.

$$x_{i,n} = x_{i,min} + n \cdot \frac{x_{i,max} - x_{i,min}}{n_{max}} \qquad (6)$$

$i = 1, \dots, i_{max}, n = 1, \dots, n_{max}$

$x: design\ variable, i: number\ of\ variables, n: division$

A reward is made determined depends on an evaluation which is calculated from the priority of performance evaluated values and sensitivity of design variable ranges to them.

$$Eva_t = function_1(w_j, y_j, b_j, c_i) \qquad (7)$$

$$r_t = function_2(Eva_t) \qquad (8)$$

$Eva: evaluation, r: reward,$

$w: priority, y: performance\ evaluated\ value, b: constant,$

$c: sensitivity, j: number\ of\ performance\ evaluated\ value,$

$i: number\ of\ design\ variable\ ranges$

4. Verification Result

The experimental programs are coded with DQN, MCT, A3C.

4.1 Theoretical Limit of the Design Calculation

Design limits of vehicle energy performance can be calculated. It is already known that the performance limit can be calculated in case of the loss related variables are minimized and the electric efficiency related variables are maximized, when the Battery capacity become maximum in its design variable range. The results of Reinforcement Learning are shown in Figure 4. The range of design variable is normalized to be 0.7 to 1.0 It was verified that the design limit was perfectly found by DQN in 128^{17} data space (17variables, 128choices). For MCT, only one variable result differently. This variable is a source of another variable, but it turns out not such a difference as to change another. There is almost no influence on performance evaluated value due to so small contribution.

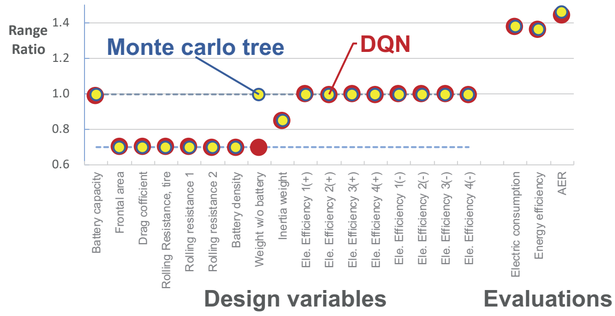

Fig.4 Learning Results for Vehicle Energy Design

4.2 Solutions in the threshold

In the case of Vehicle Package, since each of the 37 variables is set to 128 choices, the data space of the design variable is 128^{37} (10^{77} more). From the huge Data space, a solution will be searched for that matches competitiveness (a data set combination of 37 variables). A threshold value worthy of design consideration was set to the performance evaluated value and learned by A3C to achieve it.

Solutions reaching the threshold can be learned to 61, 101 set. In this report, the optimum solution is selected for the design concept (the long mileage, the large cabin, the small battery, more than current competitors on the market). The standard deviation and the optimal solution are shown in Fig.5. The optimal solution exists within the deviation, and it can be understood that competitiveness cannot be secured unless it is designed within this range.

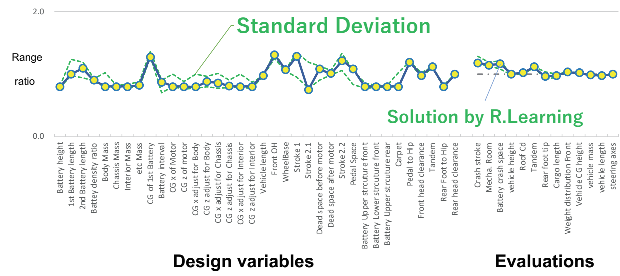

Fig.5 Learning Results of Vehicle Package Design

4.3 The optimal solution versus an expert design

Figure 6 shows a comparison of the optimal solutions by Reinforcement Learning and the result by authors. It can be known that vehicle package by Reinforcement Learning has better frontal collision performance than the design by author. On the design by Reinforcement Learning, the rear passenger space and cargo capacity which have trade off relationship to frontal collision are worse but kept on the threshold. It can be said that the design by Reinforcement Learning is better.

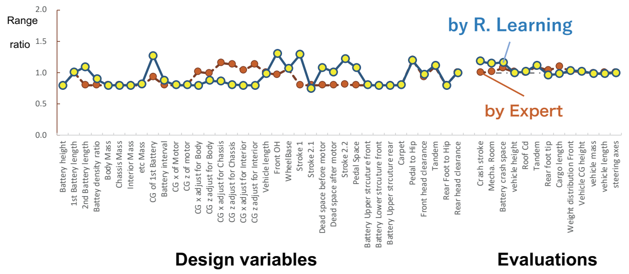

Fig.6 Comparison of Learning result with an expert's

It can be considered that the design to make use of the space efficiency of the rear part of the vehicle to the front part of the vehicle was obtained because the setting of the priority of performance and sensitivity of design range in Reinforcement Learning was appropriate. The performance evaluated values of the vehicle package are the following items.

- Vehicle height, Roof geometry (Air drag)
- Frontal collision stroke
- Driving position, driver space
- Rear passenger space
- Cargo capacity
- C.G., Vehicle mass

It turns out that each of them forms a continuous space. (Fig.7) It is a multi-disciplinary relationship, and even if a designer has lots of experience, it will not be a reason to get the optimal solution. In the case that the design variables have a complex trade off to the evaluation values, the Reinforcement Learning can search the design variables widely and deeply according to the priority of the performance evaluated value and the sensitivity of the design variable ranges. The depth & width of the search range is an advantage superior which the designer cannot have.

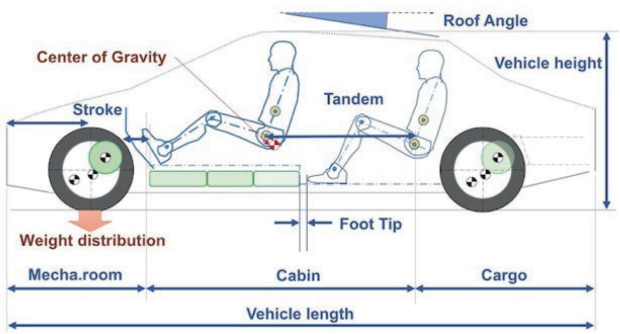

Fig.7 Evaluation values for vehicle package

5. Conclusion

When Model Based Design has a hierarchical structure that is multilayered and the design variable has a huge data space, Reinforcement Learning has superiority in determining optimal design specifications.

- The greatest advantage of Reinforcement Learning is the breadth and depth of search for design variables, even skilled mature people can never catch up.
- In order to make full use of advantage, it is necessary to enrich the logical formulas to build MBD, such as design calculations, Database and so-on.
- The role of the designer changes to making the priority of the evaluation value and the sensitivity of the variable faithfully conform to the design purpose.
- It will not be the role of the designer to search for variables that reach the performance threshold.
- The learning performance of A3C is very high, various choices of reaching the threshold are obtained and it is possible to design with deviation taken into account.

In this report, the priority of the performance evaluated values and sensitivity of the variable range for the value-based function and the policy-based function are set by trial and error. Their grounds are considered by an author through the way to overtake the competitor and the balance of performance targets. In the next stage they will be made objects of Reinforcement Learning.

References

[Sakai 2018] Tatsuhide Sakai: An Automatic Search to EV design variables using Reinforcement Learning, EVS31 Kobe Japan, 2018.

[Mnih 2016] Volodymyr Mnih: Asynchronous Methods for Deep Reinforcement Learning, ICML arXiv:1602.01783, 2016.

[Mnih 2015] Volodymyr Mnih: Human-level control through deep reinforcement learning, nature142236, NATURE VOL518, 2015.

[Mnih 2013] Volodymyr Mnih: Playing Atari with Deep Reinforcement Learning, NIPS arXiv:1312.5602v1, 2013.

[Silver 2017] David Silver: Mastering the game of Go without human knowledge, nature24270 NATURE VOL550, 2017.

[Silver 2016] David Silver: Mastering the game of Go with deep neural networks and tree search, nature16961 NATURE VOL529, 2016.

[Szepesvari, 2017] Csaba Szepesvari: Algorithm of Reinforcement Learning, Kyoritsu Publishing, 2017.

[Gruslys 2018] Audrunas Gruslys: THE REACTOR: A FAST AND SAMPLE-EFFICIENT ACTOR-CRITIC AGENT FOR REINFORCEMENT LEARNING, International Conference on Learning Representations, 2018.

[Hasselt 2016] Hado van Hasselt: Deep Reinforcement Learning with Double Q-learning, AAAI arXiv:1509.06461, 2016.

[Brockman 2016] Greg Brockman: OpenAI Gym, arXiv:1606.01540, 2016.

[Lillicrap 2015] Timothy P. Lillicrap: Continuous control with deep reinforcement learning, arXiv:1509.0297v1, 2015.

[Bergstra 2012] James Bergstra: Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research 13, 2012.

[Diaconis 2009] Persi Diaconis: The Markov chain Monte Carlo revolution, BULLETIN of the AMERICAN MATHEMATICAL SOCIETY 46, p179-205, 2009.

[Patterson 2017] Josh Patterson: Deep Learning A Practitioner's Approach, O'Reilly Media, p.1-403, 2017.

[Sugomori 2017] Yusuke Sugomori: Deep Learning: Practical Neural Networks with Java, Packt Publishing, 2017.

[Sugomori 2017] Yusuke Sugomori: 詳解ディープラーニング, MyNavi publishing, p.023-208, 2017.

[Sugomori 2016] Yusuke Sugomori: Deep Learning Java programming, Packt publishing, 2016.

[Sugomori 2016] Yusuke Sugomori: Deep Learning Java プログラミング 深層学習の理論と実装, Impress corporation, 2016.

[Henrik 2017] Henrik B.: Machine Learning, Impress corporation, 2017.

[Bostjan 2016] Kaluza Bostjan: Machine Learning in Java, Packt Publishing, 2016.

[Fujita 2016] K. Fujita, A.Takahara: 実装ディープラーニング, Ohmsha, 2016.

[Makino 2016] T. Makino: これからの強化学習, Morikita publishing, 2016.

[Saito 2016] Y. Saito: Deep Learning–Python, O'Reilly Japan, 2016.

[Okatani 2015] Takayuki Okatani: Deep Learning Machine Learning Professional Series, Kodansha, 2015.

[Ohlsson 2011] Stellan Ohlsson: Deep Learning: How the Mind Overrides Experience, Cambridge University Press, 2011.

[Otsuki 2017] K. Otsuki: 最強囲碁 AI アルファ碁解体新書, Shoeisha, 2017.

[Sutton 2000] Richard S. Sutton: Reinforcement Learning, Morikita publishing, 2000.

[Karim2018] Md. Rezaul Karim: Scala Machine Learning Projects, Packt Publishing, 2018.

[Shimada 2017] N. Shimada: Chainer with Deep Learning, Gijyutsu Hyoron, 2017.

[Chiusano 2015] Paul Chiusano: Scala Functional type design & programming, Impress corporation, 2015.

[Gibson 2017] Adam Gibson. "Deep Learning for Java". DL4J. https://deeplearning4j.org, accessed 2017-12-10.

[Silver 2017] David Silver. "Lecture 1: Introduction to Reinforcement Learning". 2017-12-12. http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching_files/intro_RL.pdf, accessed 2017-12-26.

[Silver 2017] David Silver. "UCL Course on RL". University College London. 2015-12-31. http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html, accessed 2017-12-11.

[Sutton 2018] Richard S. Sutton. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". Neural Information Proceeding Systems Foundation Inc. https://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf, accessed 2018-03-20.

[Bergstra 2018] James Bergstra. "Algorithms for Hyper-Parameter Optimization". Neural Information Proceeding Systems Foundation Inc.. https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf, accessed 2018-1-18.

[Kim 2018] Hyunsoo Kim. "Awesome Reinforcement Learning". Awesome-rl. 2018-1-8. https://github.com/aikorea/awesome-rl, accessed 2018–1-24.

[Szepesvari 2009] Csaba Szepesvari. "Algorithms for Reinforcement Learning". Department of Computing Science University of Alberta. 2009-6-9. https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf, accessed 2017-12-20.

[WU 2017] YUHUAI WU. "OpenAI Baselines: ACKTR & A2C". OpenAI. 2017-8-18. https://blog.openai.com/baselines-acktr-a2c/, accessed 2017-12-14.

[KARPATHY 2017] ANDREJ KARPATHY. "Evolution Strategies as a Scalable Alternative to Reinforcement Learning". OpenAI. 2017-3-24. https://blog.openai.com/evolution-strategies/, accessed 2017-12-22.

[Benesse 2017] "強化学習とは？ALPHAGO でも使われている強化学習を丁寧に解説". みんなの AI 講座 ゼロから Python で 学 ぶ 人 工 知 能 と 機 械 学 習 . 2017-12-11. https://udemy.benesse.co.jp/ai/reinforcement-learning.html, accessed 2017-12-23.

[DeepAge 2017] "これさえ読めばすぐに理解できる強化学習の 導 入 と 実 践 ". DeepAge. 2017-8-10. https://deepage.net/machine_learning/2017/08/10/reinforcement-learning.html, accessed 2017-12-16.

[BrainPad 2017] BrainPad. "強化学習入門 ～これから強化学習を学びたい人のための基礎知識～". Platinum Data Blog. 2017-2-24. http://blog.brainpad.co.jp/entry/2017/02/24/121500, accessed 2017-12-15.

[Kanakubo 2018] Masaaki Kanakubo. "強化学習". laboratory of Intelligent interaction Department of computer science Shizuoka Institute of Science and Technology. http://www.sist.ac.jp/~kanakubo/research/reinforcement_learning.html, accessed 2018-1-8.

[saltcooky 2018] saltcooky. "強化学習を強化学習（1）". Qiita. 2018-02-25. https://qiita.com/saltcooky/items/312e5a4919cf197a9307, accessed 2018-3-12.

[pytry3g 2018] pytry3g. "強化学習をしてみた". Qiita. 2018-2-1. https://qiita.com/pytry3g/items/024d06c52a5dfd681d94, accessed 2018-02-18.

[sugulu 2017] sugulu. "これから強化学習を勉強する人のための「強化学習アルゴリズム・マップ」と、実装例まとめ". Qiita. 2017-10-30. https://qiita.com/sugulu/items/3c7d6cbe600d455e853b, accessed 2017-12-21.

[daisuke 2017] daisuke-team-ai. "強化学習事例集 by Team AI". Qiita. 2017-10-25. https://qiita.com/daisuke-team-ai/items/ace0e1b098ee0bced127, accessed 2017-12-11.

[icoxfog417 2017] icoxfog417. "ゼロから Deep まで学ぶ強化学習". Qiita. 2017-6-6. https://qiita.com/icoxfog417/items/242439ecd1a477ece312, accessed 2017-12-13.

[evk-yk 2017] evk-yk. "強化学習で参考になったサイトまとめ". Qiita. 2017-3-27. https://qiita.com/eve_yk/items/2ace6d4c1dad7e912df1, accessed 2017-12-10.