2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Evolvable Motion-planning Method using Deep Reinforcement Learning

Kaichiro Nishi[1] and Nobuaki Nakasu[1]

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561602

***Abstract*— A motion-planning method that can adapt to changes in the surrounding environment is proposed and evaluated. Automation of work is progressing in factories and distribution warehouses due to labor shortages. However, utilizing robots for transport operations in a distribution warehouse faces a problem; that is, tasks for setting up a robot, such as adjustment of acceleration for stabilization of the transportation operation, are time consuming. To solve that problem, we developed an "evolvable robot motion-planning method." The aim of this method is to reduce the preparation cost by allowing the robot to automatically learn the optimized acceleration according to the weight and center of gravity of the objects to be transported. It was experimentally demonstrated that the proposed method can learn the optimized acceleration control from time-series data such as sensor information. The proposed method was evaluated in a simulator environment, and the results of the evaluation demonstrate that the learned model reduced the inertial force due to the acceleration of robot motion and shortened the transport time by 35% compared with the conventional method of manual adjustment. The proposed method was also evaluated in a real machine environment, and the evaluation results demonstrate that the method can be applied to a real robot. Since the speed of the robot does not need to be adjusted in the case of the proposed method, the adjustment man-hours can be reduced.**

I. Introduction

The percentage of the elderly in the world increases every year [1], and developed countries are going to suffer from the so-called "aging society;" in particular, it is reported that the over-sixty population will be one third of the total global population in 2050, and that situation will cause labor shortages in industrial fields. Under this circumstance, automation of work in factories and distribution warehouses by using robot is therefore progressing [2]. Conventional applications of robots have mainly been routine work that repeats the same operations, such as welding [3], [4], [5] and painting [6], [7]. However, in recent years, robots have been promoted to perform so-called "unsteady" work, in which operations such as assembly [8], [9] and unloading [10], [11] change in various ways, so their movements must be changed each time accordingly.

A general "depalletizing" robot system for unloading is overviewed in Fig. 1. The robot unloads the objects to be transported (i.e., boxes stacked on a pallet) onto the conveyor. Information such as appearance, size, weight, and center of gravity of each transported object is registered in the database. The system uses a camera mounted above the pallet to take a picture of the object to be transported,

[1]Kaichiro Nishi and Nobuaki Nakasu are with Production Systems Research Department, Center for Technology Innovation - Production Engineering, Hitachi, Ltd. Research and Development Group, Yokohama, Kanagawa, Japan. kaichiro.nishi.kt@hitachi.com

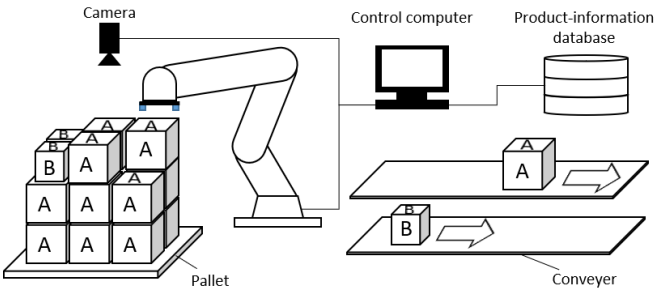
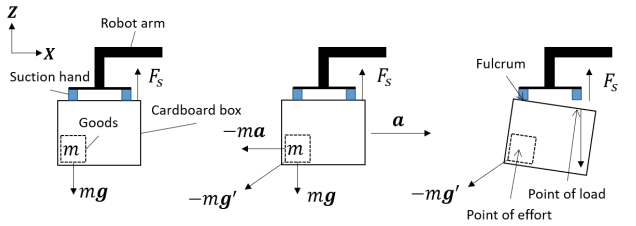


Fig. 1: General depalletizing robot system

and it obtains information about the object by collating the information obtained from the logo printed on the object and pattern information about the object with information stored in a database. Then, the acceleration of the robot arm is adjusted according to the obtained information about the object to be transported, and an operation appropriate for the object to be transported is planned and executed to transport the object. As for this method, pre-registration of the transportation target is a prerequisite, and if the transportation target is unknown, no information can be obtained, and motion planning cannot be performed. Even if the object to be transported is unknown, information about the size of the object to be transported can be obtained by using a device for measuring appearance, such as a 3D camera; however, the weight and center of gravity of the object cannot be obtained by such appearance measurement.

The force relationship between the suction hand attached to the tip of the robot and the transported object is shown in Fig. 2. The stationary state, in which the robot holds the object to be transported and stands still, is shown in 2. (a). In the stationary state, gravity acts in the $-\mathbf{Z}$ direction according to weight m of the object to be transported and gravitational acceleration $\boldsymbol{g}$. The suction hand generates suction force F_s in the $+\mathbf{Z}$ direction so that the object to be transported does not fall. Since F_s is larger than the gravitational force due to $m\boldsymbol{g}$, the transported object is gripped without falling. The moving state (small acceleration), in which the robot grips and transports the object to be transported, is shown in Fig. 2. (b). In the moving state, inertial force $-m\boldsymbol{a}$ is generated in the direction opposite to the direction of acceleration because the robot starts moving at acceleration $\boldsymbol{a}$. At that time, a force that combines gravity and inertial force acts on the suction hand. An example in which the object to be transported falls during transportation is shown in Fig. 2. (c). If the goods are packed in a biased manner or if the weight of the object to be transported or the acceleration of the robot to be



(a) Stationary state (b) Moving state (small acceleration) (c) Moving state (large acceleration)

Fig. 2: Behavior of object transferred by robot

transported is too large, the object to be transported tends to fall. Therefore, when transporting an unknown object, the robot must be moved at low acceleration, and under the assumed maximum weight and center of gravity of the object to be transported, throughput is significantly reduced.

The weight of an unknown object can be measured by attaching a force sensor to the hand of the robot. However, to prevent the object falling, it is necessary to adjust the acceleration during transportation by estimating not only the weight of the object but also the force generated on the object to be transported. The force during transportation can be estimated by physical simulation, but the throughput deteriorates because it takes several seconds to plan a stable transportation operation.

Even if the weight and center of gravity of the object to be transported are unknown, if the acceleration can be controlled in consideration of the force applied to the object to be transported, the preparatory man-hours concerning transporting the object (such as adjusting transport speed and trajectory) can be reduced. In this research, we aimed to reduce the preparatory man-hours while maintaining the throughput by developing a planning for motion method that can control the acceleration according to the state of the object to be transported.

II. Related works

In recent years, with the advent of deep learning [12], performance that surpasses humans has been confirmed in regard to image processing [13], [14] and natural-language processing [15], [16]. In response to this performance trend, application of deep learning in robot tasks as well is being promoted, and the performance of each task is being improved. These tasks are classified into two types according to two different learning approaches.

The first approach is learning without physicality [17], [18], [19], [20]. The feature of this approach is that it cuts out image-recognition and voice-recognition processing as subtasks and adapts them to robots. For example, in the case of a picking task, the task of estimating the picking position from the input image and the task of transporting the picked object are performed by different processes. By cutting out only the task to be estimated from the input image, the existing research results can be applied well, and the success rate of grasping can be increased. As for the picking task of a robot, not only grasping the object but also transporting the object is important. However, these operations are not taken into consideration, and the transport speed is manually adjusted for each transport object.

The second approach is learning with physicality [8], [9], [21], [22]. The feature of this approach is that it uses reinforcement learning approach [23], in which the robot learns through trial and error. In the approach using reinforcement learning, the behavior of the robot at the beginning of learning is unstable. It is therefore often applied in a real machine environment after learning in a simulator environment. However, in a real machine environment, the states of robots and transported objects, for example, the trajectory of the manipulator and the force applied to the object to be transported, change. These changes are difficult to predict from the appearance of the object, and the system must be able to make corrections automatically.

In consideration of the above-described requirements, we propose a method for solving the problem concerning the transport task by learning with physicality based on the reinforcement learning approach.

III. Methods

A. Overview of proposed method

The proposed "evolvable robot motion-planning method" is overviewed in Fig. 3. This system is operated in two phases. The first phase is to learn the motion plan in the simulator. The system inputs sensor information and trajectory information, and learns the operation for which the force applied to the robot's hand to transport the object is smallest and the movement time is shortest. During the learning on the simulator, the operation of randomly changing the weight of the transported object and the transit point is repeatedly executed. By randomly changing the state, the system can learn robust behavior for unknown objects and trajectories.

The second phase is to execute the depalletizing operation in a real machine environment. As for this operation by an actual machine, the operation can be divided into two types depending on the state of the object to be transported. If the system already has the information about the transported object and has been manually adjusted, the robot is controlled according to the adjusted motion planning. If the system does not have the information about the object to be transported, the robot is controlled according to the learned model. By simultaneously executing the control adjusted by hand and the control by the learning model, it is possible to improve the stability and throughput of the entire transport system at the same time.

B. Learning approach

"Reinforcement learning" [23] is a known method of learning the optimum behavior in each state of a learning target through trial and error. Q-learning [24] is a typical method of reinforcement learning, but it suffers a problem that it becomes difficult for the learning of the action-value function to converge as the number of states increases. In recent years, DQN (Deep Q-Network) [25] has been

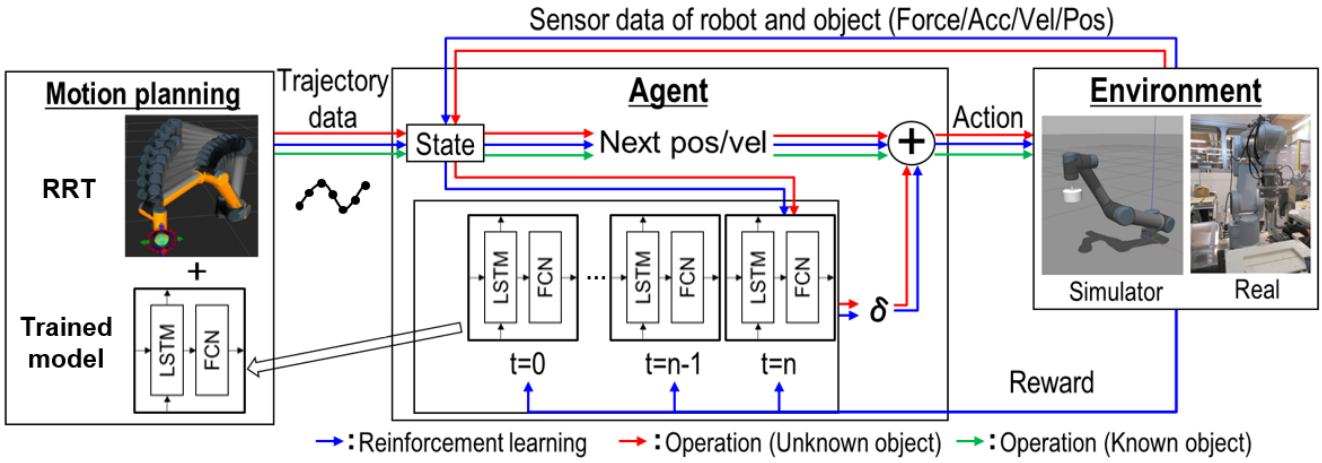


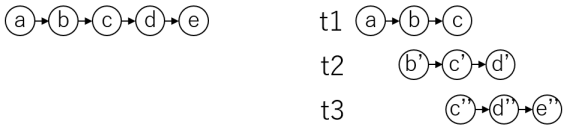
Fig. 3: Overview of evolvable robot motion-planning method

proposed as a method to solve this problem. As for DQN, replacing the action-value function with a deep learning model makes it possible for the learning to converge even when the number of states is large.

As for the proposed evolvable motion-planning method, the acceleration of the robot arm at the next waypoint is controlled on the basis of sensor information at each waypoint on the transport trajectory. It is difficult to determine the acceleration to be controlled only from the sensor information of a certain waypoint. This is because even if the sensor information and the next waypoint position are in agreement, the control acceleration to be taken next differs according to whether the sensor is gradually accelerated or decelerated. The system must therefore predict the sensor information at the next waypoint from the sensor information in the past time series and select the optimum control speed. As for the proposed method, LSTM (long short-term memory) [26] (used for time-series data such as speech recognition and video recognition) is applied to the action-value function of reinforcement learning. By using LSTM as an action-value function, it is possible to learn the control motion on the basis of past time-series sensor information by trial and error.

C. Real-time trajectory data change

The ROS (Robot Operation System) [27] is used to control the robot. As for ROS, motion-planning systems such as MoveIt [28] are used for generating trajectory data. As for RRT (rapidly exploring random trees) [29], which is a typical method of motion planning, multiple waypoints are generated for the input coordinate information of the start point and the end point. At each waypoint, angle, velocity, acceleration, and arrival time of each joint of the robot are stored. The waypoints from the start point to the end point are sent to the controller manager as trajectory data. The controller manager outputs the joint angle values, etc. by PID control from the trajectory data and the information about the current posture state of the robot arm. The outputted joint-angle value is used to control the robot in the case of the simulator or the actual machine via the hardware resource interface layer.



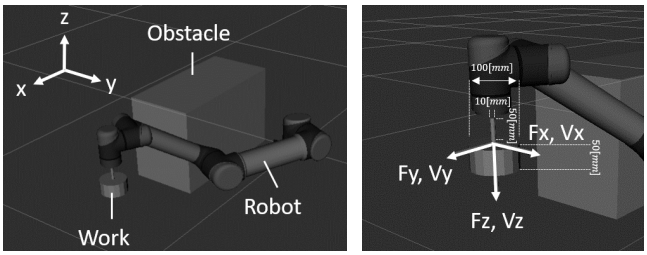
(a) Original trajectory (b) Modified trajectory

Fig. 4: Output of trajectory data by the proposed method

As for the proposed method, it is necessary to change the trajectory data in real time. The trajectory-change process by the proposed method is therefore executed before the trajectory data is sent to the controller manager. The original trajectory data output by the motion-planning system and the trajectory data modified by the proposed method are shown in Fig. 4. First, the position information about the start point and end point are entered into the conventional motion-planning system, and a waypoint from point "b" to point "d" is generated. At time $t1$ before the operation starts, the proposed system extracts points "a" to "c" as positional information and sends that information to the controller manager. At time $t2$ when the operation starts and the manipulator reaches point "b," waypoints "b" to "d" are extracted, the values are modified, and the modified values are sent to the controller manager. This process is performed until the robot manipulator reaches the end point.

D. Learning by simulator

In Phase 1, the motion plan is learned by the simulator. Gazebo [30] was used as an open-source 3D robot simulator. The learning environment built in the simulator is shown in Fig. 5. In the simulation environment, a 6-DOF robot manipulator (UR5 [31], developed by Universal Robots) was placed as a learning target. To randomly change the learning environment, objects to be transported and obstacles were placed. The object to be transported is a cylinder with a diameter of $100mm$ and a height of $50mm$, and it is attached to the tip of the robot arm by a support with a diameter of



(a) Overview (b) Work

Fig. 5: 6-DoF-arm robot in simulator environment

$10mm$ and a length of $50mm$. The weight of the object to be transported is randomly changed in the range of 0 to $5kg$ for each transfer operation so that the force applied to the object to be transported changes. Similarly, by randomly changing the height of the obstacle in the range of $0mm$ to $1000mm$, the trajectory planned by the RRT changes, and the force applied to the object to be transported also changes.

The following information is used as input to the learning model: force (F_x, F_y, F_z), torque (T_x, T_y, T_z), velocity (V_x, V_y, V_z), acceleration (A_x, A_y, A_z) applied to the attachment part of the object to be transported, and joint angle ($j_1\theta$, $j_2\theta$, $j_3\theta$, $j_4\theta$, $j_5\theta$, $j_6\theta$) and angular velocity ($j_1\omega$, $j_2\omega$, $j_3\omega$, $j_4\omega$, $j_5\omega$, $j_6\omega$) in relation to each axis of the robot manipulator. The action at each waypoint was to reduce the acceleration by 0% to 50%. The reward was designed differently for the waypoint and the final point. At each waypoint, -1 is given as a negative reward when the magnitude of the force applied to the strut mounting exceeds $\pm 15N$. At the final point, if the arrival time is later than the time initially planned by RRT, a negative reward is given in proportion to the arrival time.

$$r = \frac{RRT\ trajectory\ execution\ time}{Learned\ trajectory\ execution\ time} - 1 \quad (1)$$

Setting these rewards allows the robot to learn acceleration control that can suppress the increase in transport time while suppressing the force applied to the object to be transported.

IV. Evaluation in simulation environment

The results of an evaluation in the simulation environment after executing phase-1 learning for 24 hours are presented hereafter.

A. Evaluation method

The trajectory used for the evaluation is shown in Fig. 6. For simplification, this evaluation trajectory meanders on the XY plane without obstacles. Accordingly, forces F_x and F_y generated in the sensor can be evaluated as the forces applied to the transported object. The weight of the object to be transported was randomly varied in the range of 0 to $5kg$, and 50 sets were tried with the center of gravity of the robot arm fixed. The robustness of the proposed method in relation to the trajectory can also be evaluated by executing a trajectory that is not used during learning.

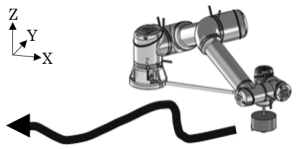


Fig. 6: Trajectory for evaluation

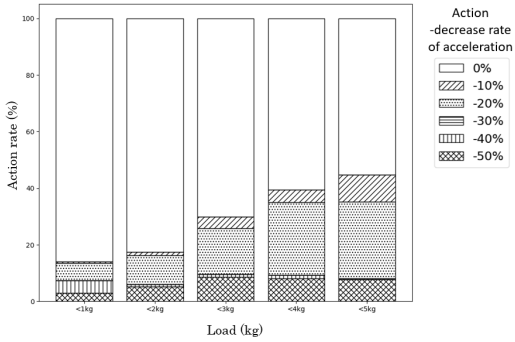
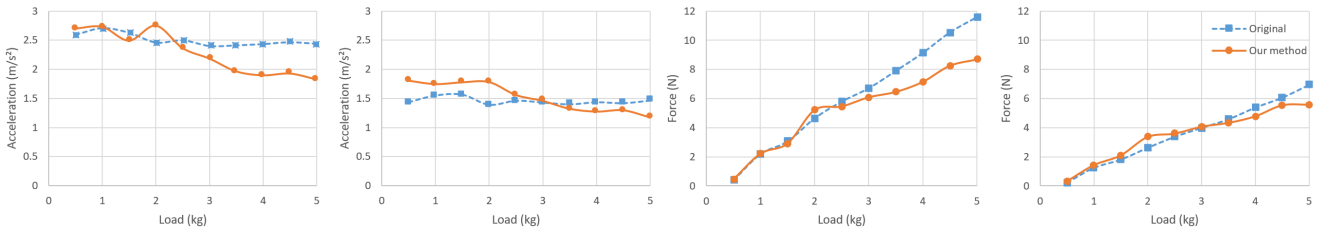


Fig. 7: Selected action rate by weight

B. Result

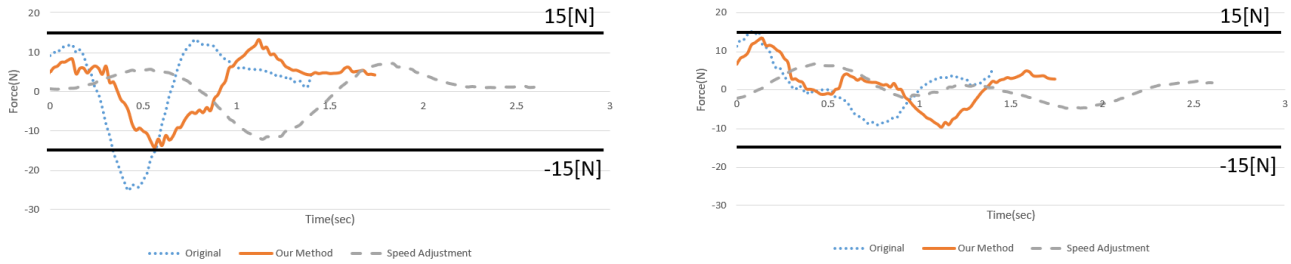
The tendency to select a certain action learned by the proposed method was evaluated first. The rate of actions selected by the proposed method for each weight of the object to be transported is shown in Fig. 7. It can be seen that the proposed method learns to select more actions that reduce the acceleration as the weight of the transported object increases. In general, the heavier the weight of the object to be transported, the greater the force generated during transport. Therefore, as for the proposed method, when it is predicted that the force applied to the object to be transported will increase, the system tends to take many actions to reduce the acceleration of the robot arm.

Change in the force applied to the object to be transported by the proposed method was evaluated next. Output values of the acceleration sensor and force sensor averaged for each weight of the object to be transported are shown in Fig. 8. The method that executes the evaluation trajectory as is is referred to as "Original method," and the method that executes the trajectory modified by the proposed mehod is referred to as "Proposed method." The acceleration of the x-axis and y-axis are constant until the weight of the object to be transported is $2.0kg$. However, when the weight of the object to be transported exceeds $2.5kg$, it can be seen that the acceleration decreases as the weight of the object to be transported increases. Similarly, it can be seen that the increase in forces acting on the x-axis and y-axis are suppressed when the weight of the object to be transported exceeds $2.5kg$. These results indicate that the proposed method learned to reduce the acceleration in order to suppress the increase in the force applied to the transported object. In addition, it is probable that when the weight of the object to be transported is $2.5kg$ or less, the proposed method (i) judged that the object could be stably transported without reducing acceleration of the robot arm and (ii) learned not



(a) Acceleration of X-axis (b) Acceleration of Y-axis (c) Force acting on X-axis (d) Force acting on Y-axis

Fig. 8: Data output from sensors by weight



(a) Force acting on X-axis (b) Force acting on Y-axis

Fig. 9: Output from force sensor at 4.89 kg

to reduce the acceleration in order to maintain transport efficiency.

Whether the proposed method could both suppress the force applied to the transported object and improve transport efficiency was evaluated last. The relationship between the force applied to the transported object and the transport time when the weight of the transported object is $4.89kg$ is shown in Fig. 9. The three evaluation targets are labeled "Original method," the evaluation trajectory was executed as is, "Proposed method," the trajectory modified by the proposed method was executed, and "Speed adjustment," the trajectory was executed by manually adjusting the acceleration of the robot. As for the original method, the force generated on the X-axis of the object to be transported 0.4 seconds after the start of the operation exceeds the threshold value of $-25N$, so the object to be transported falls. As for the proposed method, the force generated in the transported object is suppressed within $\pm 15N$ by suppressing the acceleration 0.3 seconds after the start of operation, and the transported object is prevented from falling. As for speed adjustment, the overall acceleration value is manually adjusted so that the peak value is $\pm 15N$ or less, so the force generated on the transported object is suppressed to $\pm 15N$ or less. However, as a result of the overall decrease in acceleration, transfer speed decreased, and transfer time became 2.6 seconds. On the contrary, as for the proposed method, transport time is 1.7 seconds, and transport efficiency is 35% higher than that in the case of the manual speed adjustment.

These results confirm that the proposed evolvable motion-planning method can learn the motion according to differences in weight and trajectory of the transported object even if the object is in an unknown state. They also confirm that the learned operation achieved both stability and transport efficiency according to the weight of the transport object and the trajectory.

V. Execution in Real Machine Environment

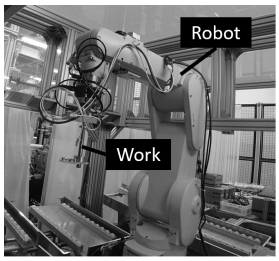
The results of an evaluation applying the proposed method to an actual robot are presented hereafter.

A. Evaluation method

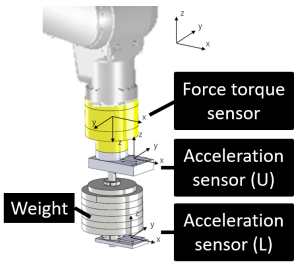
It is an important evaluation target to confirm whether the system learned on the simulator can operate on an actual machine. It is also important that the system can learn without depending on the structure of a specific robot. This is because when building a robot system, robots from different manufacturers may be used. Therefore, in this evaluation, a different robot from that use in Section IV was used, and both its applicability and versatility were confirmed.

The evaluation environment of the actual machine is shown in Fig. 10. In the real machine environment, a 6-DOF robot arm (RV-7FL [32], developed by Mitsubishi Electric) was placed as an application target. An object to be transported is attached to the tip of the robot arm. The object to be transported weighs $5.0kg$ and consists of seven $0.5kg$ weights, namely, one weighing $0.25kg$, and the rest weighing $1.25kg$. The weights can be attached and detached manually, and the weight of the transported object can be changed in the range of $1.25kg$ to $5.0kg$.

As for learning movement of the robot, the robot in the simulator was changed from the UR5 to the RV-7FL, and learning operation was executed for five hours. When the movement of the robot was being learned, inputting the values of accelerometer into the learning model was stopped. This is because the accelerometer attached to the actual



(a) Overview



(b) Work

Fig. 10: 6-DoF-arm robot in real machine environment

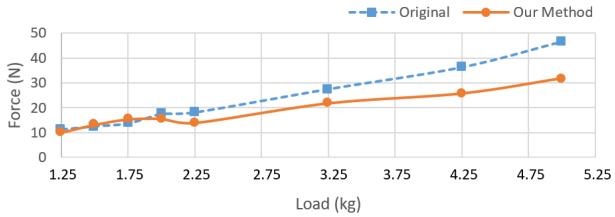
machine has low responsiveness, so it only outputs sensor values with noise to the movement of the robot, which has a negative influence on the learning model. Other than that problem, the robot movement under the same settings as used for UR5 was learned.

The proposed method was evaluated by using the same trajectory as that described in Section IV. In the real machine environment, the weight of the object to be transported cannot be changed freely. The weight of the transported object was therefore changed by manually changing the number of weights attached to the tip of the robot arm. The evaluation trajectory was run five times for each number of weights, and changes in sensor value and transport time were evaluated.

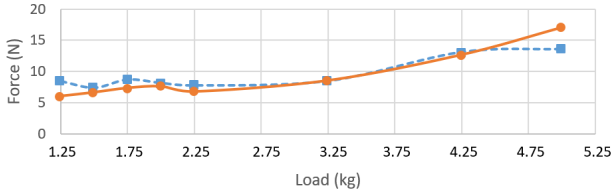
B. Result

Force applied to the transported object and transport time measured for each number of weights are shown in Fig. 11. Two targets were evaluated: "Original method," the evaluation trajectory was executed as is, and "Proposed method," the trajectory modified by the proposed method was executed. Weight of the transported object is $1.25kg$ when the number of weights is zero and $5.0kg$ when the number of weights is 7.5. When the number of weights is changed by one, the weight of the transported object changes by $0.5kg$.

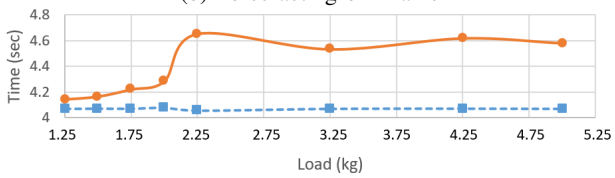
When the weight of the transported object is $2.0kg$ or less, the force applied to the transported object increases in the same ways in the cases of both the original method and the proposed method. This result is explained by the fact that the proposed method judged that it was not necessary to change the acceleration if the weight of the transported object was $2.0kg$ or less in the case of the evaluation trajectory. On the other hand, when the weight of the transported object is $2.25kg$ or more, how the force applied to the transported object increases differs in the cases of the original method and proposed method. As for the force acting on the X-axis, it differs by up to 30% in the case of the original and proposed methods. This is because it was determined that when the weight of the object to be transported is $2.25kg$ or more in the case of the given or evaluation trajectory of the proposed method, it is necessary to reduce the acceleration in order to suppress the force applied to the transported object. Therefore, transport time is increased by 15% in the case of the proposed method when the weight of the transported



(a) Force acting on X-axis



(b) Force acting on Y-axis



(c) Transport time

Fig. 11: Data output from sensors and transport time

object is $2.25kg$ or more. However, when the weight of the transported object is $2.0kg$ or less, the increase in transport time is suppressed to 5% or less. It can be confirmed that the proposed method also considers transportation efficiency.

The results presented above demonstrate that the proposed method can learn model without depending on a specific robot structure, and that the model learned in the simulator environment can be operated in a real machine environment.

VI. Conclusions

An "evolvable motion-planning method" for reducing preparatory man-hours (such as adjusting transport speed and trajectory in the automation of transport of objects with various packing forms, weights, and centers of gravity) was proposed. The proposed method enables motion planning that can handle various weights and centers of gravity of the object to be transported by controlling the acceleration of a robot arm on the basis of time-series sensor information and trajectory information. The results of evaluating the proposed method in a simulation environment demonstrate that transport time can be reduced by 35% while the force applied to the transported object is suppressed by controlling the acceleration of the robot arm according to the weight of the object to be transported and its trajectory. In addition, the results of an evaluation in a real-machine environment demonstrated that (i) learning is possible without depending on a specific robot and (ii) the system learned in the simulator environment can be operated in a real machine environment. Since the speed of the robot arm does not need to be adjusted in the case of the proposed method, the man-hours needed for adjustment can be reduced.

REFERENCES

[1] United Nations, Department of Economic and Social Affairs, Population Division, "World population ageing 2015 (st/esa/ser.a/390)."

[2] J. Manyika, S. Lund, M. Chui, J. Bughin, J. Woetzel, P. Batra, R. Ko, and S. Sanghvi, "Jobs lost, jobs gained: Workforce transitions in a time of automation," *McKinsey Global Institute*, vol. 150, 2017.

[3] J. N. Pires, T. Godinho, and P. Ferreira, "Cad interface for automatic robot welding programming," *Industrial Robot: An International Journal*, 2004.

[4] N. Larkin, A. Short, Z. Pan, and S. van Duin, "Automatic program generation for welding robots from cad," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2016, pp. 560–565.

[5] N. Larkin, A. Short, Z. Pan, and S. Van Duin, "Task space motion planning decomposition," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1688–1694.

[6] A. Klein, "Cad-based off-line programming of painting robots," *Robotica*, vol. 5, no. 4, pp. 267–271, 1987.

[7] S.-H. Suh, I.-K. Woo, and S.-K. Noh, "Development of an automatic trajectory planning system (atps) for spray painting robots," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*. IEEE Computer Society, 1991, pp. 1948–1949.

[8] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel, "Learning robotic assembly from cad," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2018.8460696

[9] K. Ota, D. K. Jha, T. Oiki, M. Miura, T. Nammoto, D. Nikovski, and T. Mariyama, "Trajectory optimization for unknown constrained systems using reinforcement learning," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019. [Online]. Available: http://dx.doi.org/10.1109/IROS40897.2019.8968010

[10] R. Krug, T. Stoyanov, V. Tincani, H. Andreasson, R. Mosberger, G. Fantoni, and A. J. Lilienthal, "The next step in robot commissioning: Autonomous picking and palletizing," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 546–553, 2016.

[11] E. Lamon, M. Leonori, W. Kim, and A. Ajoudani, "Towards an intelligent collaborative robotic system for mixed case palletizing," in *Proc. Int. Conf. Robot. Autom.(ICRA)*, 2020, pp. 9128–9134.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7291–7299.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423

[17] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, Jun 2017. [Online]. Available: http://dx.doi.org/10.1177/0278364917710318

[18] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively picking real-world objects with unconstrained spoken language instructions," in *Proceedings of International Conference on Robotics and Automation*, 2018.

[19] A. Zeng, S. Song, J. Lee, A. Rodriquez, and T. A.Funkouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *Robotics: Science and Systems XV*, Jun 2019. [Online]. Available: http://dx.doi.org/10.15607/RSS.2019.XV.004

[20] L. Berscheid, T. Ruhr, and T. Kroger, "Improving data efficiency of self-supervised learning for robotic grasping," *2019 International Conference on Robotics and Automation (ICRA)*, May 2019. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2019.8793952

[21] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," *Robotics: Science and Systems XIV*, Jun 2018. [Online]. Available: http://dx.doi.org/10.15607/RSS.2018.XIV.008

[22] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," *2019 International Conference on Robotics and Automation (ICRA)*, May 2019. [Online]. Available: http://dx.doi.org/10.1109/ICRA.2019.8793789

[23] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[24] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[28] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *arXiv preprint arXiv:1404.3785*, 2014.

[29] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning. technical report 98-11, computer science dept., iowa state university," 1998.

[30] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[31] "UR5," (accessed 20.10.13). [Online]. Available: https://www.universal-robots.com/

[32] "RV-7FL," (accessed 20.10.13). [Online]. Available: https://www.mitsubishielectric.com/fa/products/rbt/robot/