Daily Assistive Modular Robot Design Based on Multi-Objective Black-Box Optimization

Kento Kawaharazuka[1], Tasuku Makabe[1], Kei Okada[1], and Masayuki Inaba[1]

***Abstract*— The range of robot activities is expanding from industries with fixed environments to diverse and changing environments, such as nursing care support and daily life support. In particular, autonomous construction of robots that are personalized for each user and task is required. Therefore, we develop an actuator module that can be reconfigured to various link configurations, can carry heavy objects using a locking mechanism, and can be easily operated by human teaching using a releasing mechanism. Given multiple target coordinates, a modular robot configuration that satisfies these coordinates and minimizes the required torque is automatically generated by Tree-structured Parzen Estimator (TPE), a type of black-box optimization. Based on the obtained results, we show that the robot can be reconfigured to perform various functions such as moving monitors and lights, serving food, and so on.**

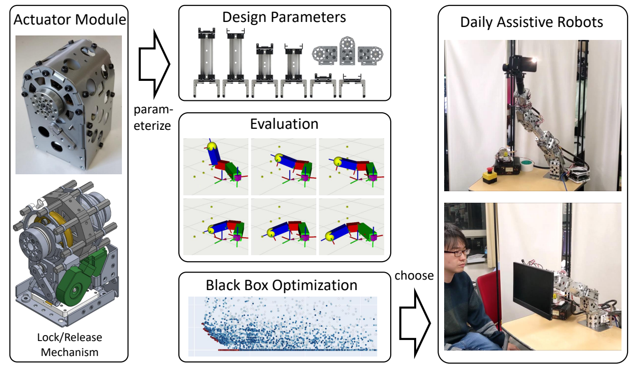

Fig. 1. The concept of this study: automatic design optimization of robots with actuator modules with a lock-release mechanism through multi-objective black-box optimization.

I. INTRODUCTION

Robots are expanding their field of activities from industrial fields, where the environment is fixed, to diverse and changeable environments including nursing care support and daily life support [1], [2]. In this context, instead of introducing a large number of robots that are exactly the same, robots that can be personalized for each user and task are needed. When such robots perform daily assistive tasks, e.g. move monitors, whiteboards, fans, and lights, serve food, and position tables for smaller robots, the actions vary greatly depending on the user, environment, and task. Of course, a general-purpose six-axis arm robot can be used, but if the robot can be configured for a given task with fewer joints and minimized torque requirements, the task can be carried out more continuously and efficiently. In addition to actually performing a task, the robot can be used as a table or chair by locking its posture when it is not moving, or can be manually operated by leaving its joints released.

Therefore, we propose an actuator module with a lock-release mechanism that can be reconfigured to various link configurations, and a robot design optimization method based on this module. By freely reconfiguring this module, the robot can perform personalized movements with a small number of appropriate joints. Also, by using the lock-release mechanism the robot can carry heavy objects and be operated by direct teaching. Here, multiple target coordinates or trajectories for a task are given, and a robot configuration that satisfies both the minimization of the control errors and the minimization of the required torque is automatically generated. Many robot models are generated by randomizing joint module types, joint orientations, and link lengths, and these parameters are optimized by using multi-objective black-box optimization. We analyze the robot configurations obtained under various conditions and show that the desired task can actually be performed by reconfiguring the actuator modules.

Modular robot design and its design optimization have been studied extensively. For industrial robots, the number and type of modules and the relative positions among the modules satisfying the target coordinates are optimized by a genetic algorithm [3]. In [4], though not a modular-type robot, the motor type and gear ratio of a general six-axis manipulator are optimized based on the minimization of weight and the maximization of manipulability. [5] has optimized the joint arrangements and link lengths of a modular robot that can run on uneven terrain. [6] has performed design optimization of a modular robot that can run on uneven terrain using Generative Adversarial Network (GAN) [7]. On the other hand, previous studies are not conducted on actual robots [3]–[6], do not perform optimization for continuous values but only for discrete values [3], do not vary the number of modules [4], or do not perform multi-objective optimization [3], [6]. In addition, there are few examples of modular robots that include a lock-release mechanism for the purpose of personalization in daily assistive robots. In this study, various Unified Robot Description Format (URDF) models are automatically generated from the defined constraints, the feasibility of target coordinates and necessary torques are evaluated, and the design parameters are optimized by using Tree-Structured Parzen Estimator (TPE)

[1] The authors are with the Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan. [kawaharazuka, makabe, k-okada, inaba]@jsk.t.u-tokyo.ac.jp

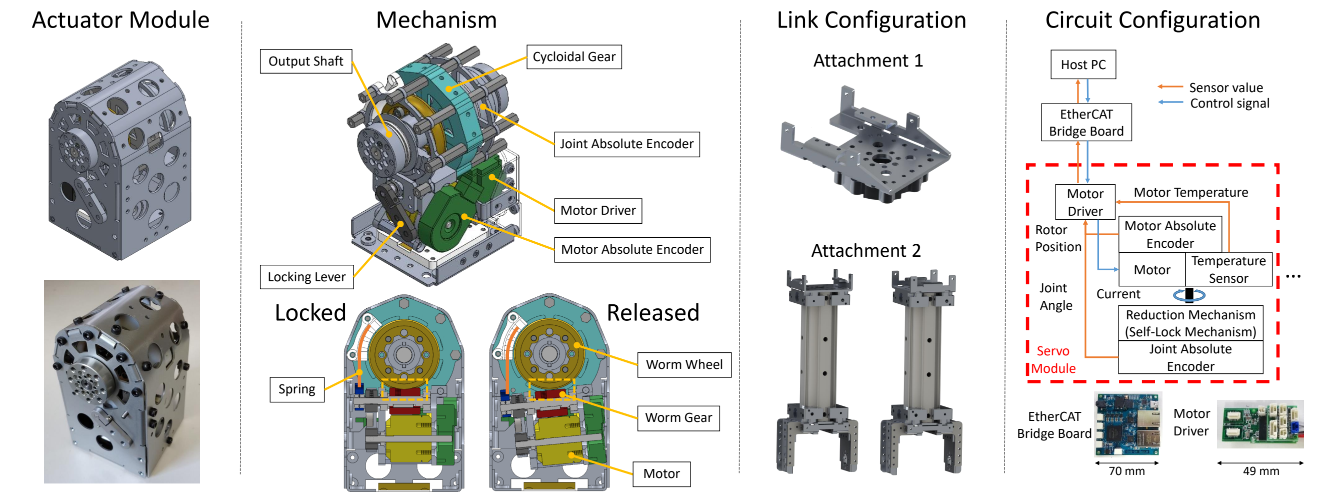

Fig. 2. The modular actuator design with a lock-release mechanism. This module includes a worm gear and cycloidal gear that can switch between locked and released states by a locking lever. By using two joint attachments, various kinematic configurations can be generated.

[8], a type of black-box optimization.

The structure of this study is as follows. In Section II, we describe an actuator module which can be reconfigured to various link configurations and has a lock-release mechanism, as well as other link structures and circuit configurations. In Section III, a black-box optimization procedure is presented to automatically output the link configuration that can satisfy the target coordinates with minimum torque. In Section IV, the link structure obtained by the optimization is analyzed in simulation, and the actual robot experiments with the developed actuator modules are conducted. In Section V, we discuss the experimental results and some limitations of this study, and conclude in Section VI.

II. Modular Actuator Design with Lock-Release Mechanism

The configuration of the actuator module in this study is shown in Fig. 2. This is a $0.07\times0.07\times0.115$ [m] actuator module with a square bottom. The motor inside is a T-MOTOR MN2212, and a worm gear and cycloid gear are used. The gear ratio of the worm gear is fixed at 50:1, which satisfies the self-lock condition. The gear ratio of the cycloid gear is variable, but basically set to 47. The actuator module has a lock-release mechanism. The worm gear and the worm wheel can be disengaged by rotating a locking lever, and the state can be switched between lock (the motor is connected to the output shaft) and release. The locked state enables the robot to carry heavy objects without back drive, and the released state enables the robot to be operated by direct teaching.

The link configuration is described below. Two types of attachments can be attached to one or both ends of the rotation axis of the actuator module. The bottom of the actuator module and the attachments, and their screws are arranged in a square shape, so that various connections can be realized. Although the link length of Attachment 2 of Fig. 2 can be changed continuously, it is handled as a discrete value in this study for the sake of practicality.

The circuit configuration is described below. HOST PC can be daisy-chained to Motor Driver through EtherCAT Bridge Board. The motor driver receives information from a temperature sensor, motor-side absolute encoder, and joint-side absolute encoder, and calculates motor current commands. Since the motor drivers are very small and each of them is equipped with an IMU, they can be used as redundant sensors. This compact module includes a motor, reduction gears, encoders on the motor and joint side, a temperature sensor, an inertial sensor, and a motor driver. The maximum current for the motor driver is 10A.

III. Automatic Robot Design with Multi-Objective Black-Box Optimization

The overall system of this study is shown in Fig. 3. Design parameters are defined, URDF models are automatically generated and evaluated, and black-box optimization is performed.

A. Robot Design Parameters

In this study, the design parameters are varied between the simulations and the actual actuator modules. In the simulation, the optimization results are verified by using general design parameters, and in the actual robot, a body that can be configured with the actuator modules is designed and tested. All joints are single-axis joints that rotate in the pitch direction. As shown in the left figure of Fig. 3, Link i is connected to Joint i ($0 \leq i \leq N_{joint}$, where N_{joint} denotes the number of joints). Let $\boldsymbol{d}_i$ denote the direction vector of Link i relative to the coordinates of Joint i, and l_i denote the length of the link. Let R_i be the rotation matrix of Joint i with respect to the coordinates of Joint $i-1$. Note that there exists a fixed joint Joint 0 at the origin of the world coordinates, which does not rotate. Joint 1 and Link 1 are the root of the robot body, and the origin of the robot is at the tip of Link 0.

First, we describe the parameters for the general configuration in a simulation. The joint angle limit of each Joint i

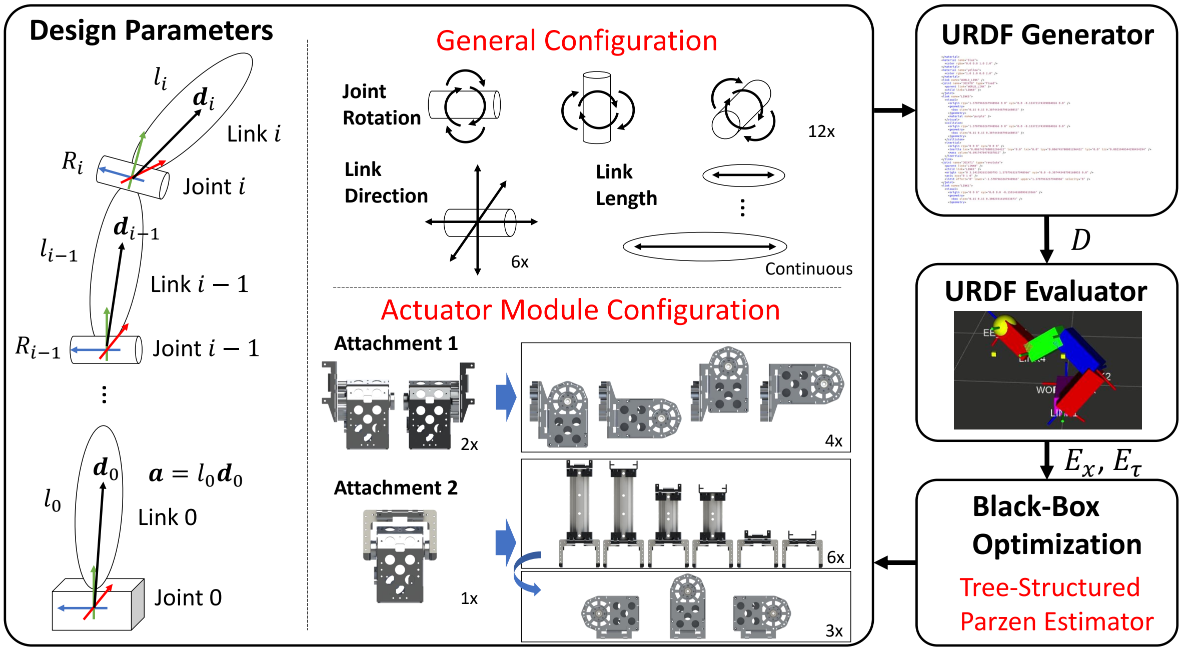

Fig. 3. The automatic robot design system with black-box optimization. We prepare different design parameters for general configuration and actuator module configuration, generate a URDF model automatically, evaluate it, and optimize the design parameters through Tree-Structured Parzen Estimator.

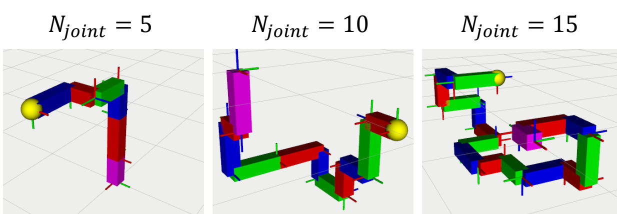

Fig. 4. Examples of robots generated with general configuration.

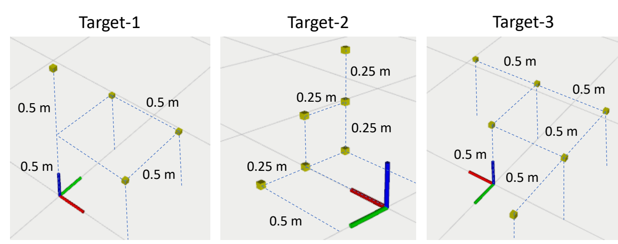

Fig. 5. The target positions to be realized for general configuration.

is assumed to be -90 to 90 [deg]. The rotation matrix R_i of Joint i has 12 discrete values in total, where the rotation axis of the joint faces each direction of xyz and the joint center faces each of the four directions apart by 90 degrees. For $\boldsymbol{d}_i$, the direction vector of Link i, we prepare a total of six discrete values considering positive and negative values for each direction of xyz. In other words, at the initial position where all joint angles $\boldsymbol{\theta}$ are $\boldsymbol{0}$, all link directions are placed at right angles. The length of Link i, l_i, is assumed to be a continuous value between 0.1 and 0.6 [m]. The links are constrained such that they do not overlap each other.

Next, we describe the parameters for the actuator module configuration. Attachment 1 is attached to the actuator module at either end (two patterns), while Attachment 2 is attached at both ends (one pattern). For Attachment 1, the actuator module can be connected in a total of eight different ways. For Attachment 2, there are a total of six different link lengths and joint orientations, and then the actuator module can be connected in three different ways. Therefore, there are 26 ($= 8 + 6 \times 3$) discrete ways of connection in total. Note that $\boldsymbol{d}_0$ and l_0 in both general configuration and actuator module configuration are different from the above patterns. We prepare a three-dimensional position $\boldsymbol{a}$ which is the product of $\boldsymbol{d}_0$ and l_0, and set the range of its xyz coordinates, $a_{\{x,y,z\}}$, as continuous values appropriate for each task.

Based on these design parameters, URDF models are automatically generated. For the general configuration, all links are rectangular with 0.15 $\times$ 0.15 $\times$ l_i [m]. The mass and inertia of the links are calculated assuming that the density of the link is 1.0 g/cm^3. The generated URDF models are shown in Fig. 4. These are examples when $N_{joint} = \{5, 10, 15\}$, and it can be seen that a variety of bodies are configured.

B. Black-Box Optimization of Design Parameters

Although various forms of evaluation functions for body design optimization are possible, in this study, we adopt a relatively simple form due to the characteristics of the task. The evaluation values are the control error at the target coordinates and the joint torque value when reaching the target. First, the target position and posture $\{(\boldsymbol{x}^{ref}_1, R^{ref}_1), \cdots, (\boldsymbol{x}^{ref}_{N_{ref}}, R^{ref}_{N_{ref}})$ (where N_{ref} is the number of target positions and postures) are given. Here, the

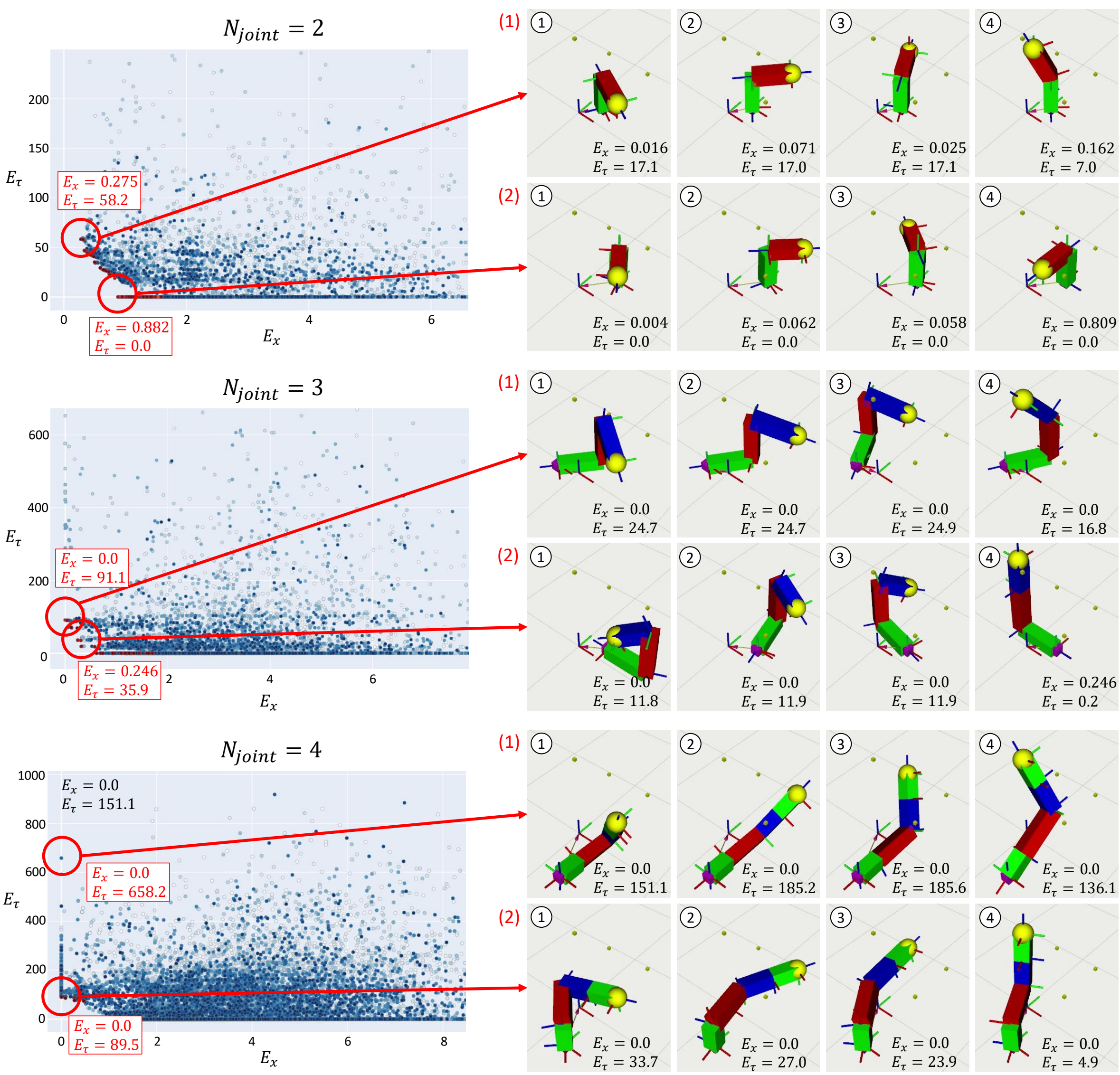

Fig. 6. The optimization results for Target-1 with general configuration. N_{joint} is changed to 2, 3, and 4, and two solutions are shown for each N_{joint}.

following values E_x and E_τ are calculated,

$$\boldsymbol{x}_i, \boldsymbol{\tau}_i = \mathrm{IK}(\boldsymbol{x}_i^{ref}, R_i^{ref}) \tag{1}$$

$$E_x = \sum_i^{N^{ref}} ||\boldsymbol{x}_i - \boldsymbol{x}_i^{ref}||_2 \tag{2}$$

$$E_\tau = \sum_i^{N^{ref}} ||\boldsymbol{\tau}_i||_2 \tag{3}$$

where IK is the inverse kinematics when the current URDF model is given, and $\boldsymbol{x}_i$ and $\boldsymbol{\tau}_i$ are the calculated end-effector position and joint torque. [9], [10] are used as the algorithm for the inverse kinematics.

Generally, a single value $E = E_x + wE_\tau$ is calculated using a certain weight coefficient w, and optimization is performed based on this E. On the other hand, in such a case, it is difficult to optimize parameters appropriately because the solution varies depending on the adjustment of w. Moreover, since only one optimal solution can be obtained, it is impossible to create a process in which the user selects his/her preferred personalized body design based on the obtained solutions. Therefore, in this study, we perform a multi-objective optimization problem to minimize both E_x and E_τ simultaneously, present several Pareto front solutions, and finally determine appropriate body parameters. We use Tree-Structured Parzen Estimator [8] in Optuna [11] as a library of black-box optimization.

IV. Experiments

Various experiments are first conducted in the general configuration to demonstrate the effectiveness of this study. Next, experiments are conducted on actual robots with the actuator module configuration to demonstrate the practical applications of this study.

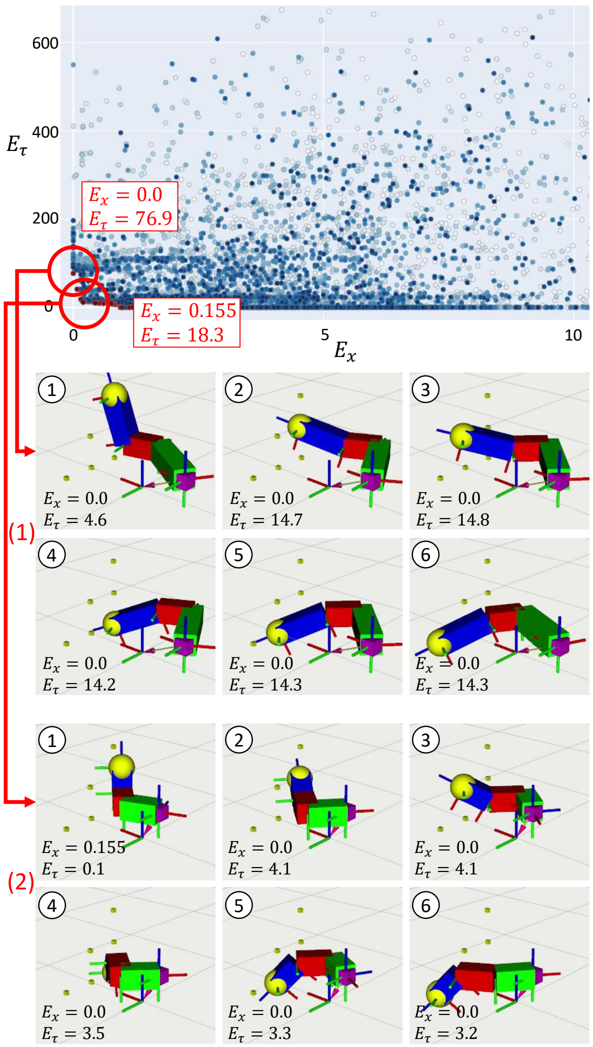

Fig. 7. The optimization results for Target-2 with general configuration. Two Pareto front solutions are shown.

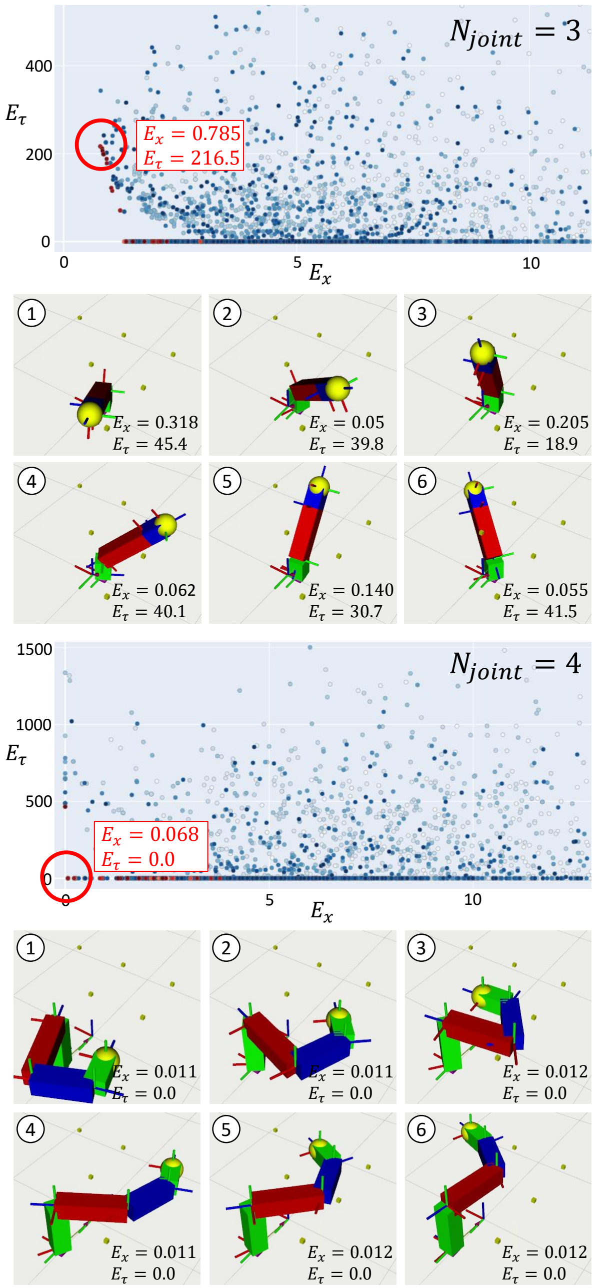

Fig. 8. The optimization results for Target-3 with general configuration. N_{joint} is changed to 3 and 4.

A. Automatic Design Optimization for General Configuration

Target-1, Target-2, and Target-3 shown in Fig. 5 are prepared as target positions (target postures are not given). $a_{\{x,y,z\}}$ is specified in the range of [-1.0, 1.0] [m], but a_z is set to [-0.1, 0.1] for Target-1, a_x is set to [-1.0, 0.0] for Target-2, and a_z is set to [-1.0, 0.0] for Target-3.

For Target-1, the sampling results when $N_{joint} = \{2, 3, 4\}$ are shown in the left figures of Fig. 6, and the inverse kinematics results for some of the solutions are shown in the right figure of Fig. 6. Here, the red dots in the graphs are Pareto front solutions. For $N_{joint} = 2$, solution (1) is a design in which E_x is small but E_τ is large, and solution (2) is a design in which E_x is large but E_τ is zero. With $N_{joint} = 2$, it is difficult to satisfy all the target positions, and E_x cannot be zero. In (1), E_x is reduced as much as possible by installing a yaw-axis joint in the first axis and a pitch-axis joint in the second axis, although the required torque increases. On the other hand, in (2), both the first and second axes are yaw-axis joints, so that the required torque is always zero. However, due to the two-dimensional motion of yaw-axis joints, only 3 out of 4 target positions can be supported, and E_x is significantly increased. Next, for $N_{joint} = 3$, solution (1) is a design in which E_x is zero, and solution (2) is a design in which E_τ is reduced as much as possible while maintaining accuracy. By increasing the number of joints, (1) is able to realize all target positions, while E_τ is higher than that of (1) with $N_{joint} = 2$. On the other hand, in (2), the length of the final link is shortened to prevent the increase of E_τ. Finally, for $N_{joint} = 4$, solution (1) is a design with the largest E_τ among $E_x = 0$ apart from the Pareto front, and solution (2) is a design with the smallest

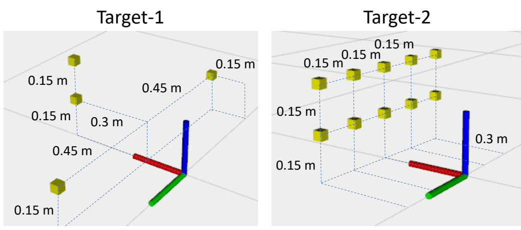

Fig. 9. The target positions to be realized for actuator module configuration.

E_τ among $E_x = 0$. (2) with $N_{joint} = 4$ is almost the same as (1) with $N_{joint} = 3$, which means that $N_{joint} = 3$ is sufficient for Target-1. The above mentioned solutions reduced the required torque by using yaw-axis joints for the root joints, but (1) with $N_{joint} = 4$ uses a pitch-axis joint for the root joint, resulting in a large E_τ. It can be seen that the required torque varies greatly depending on the joint arrangement.

For Target-2, the sampling result when $N_{joint} = 3$ is shown in the upper figure of Fig. 7, and the inverse kinematics results for two of the Pareto front solutions are shown in the lower figure of Fig. 7. Solution (1) is a design that satisfies $E_x = 0$ for all the target positions but has a large E_τ, while solution (2) is a design with reduced E_τ but with slightly lower accuracy. In both (1) and (2), the root joint is a yaw-axis joint that is not subject to gravity force, and only the tip joint is a pitch-axis joint. The link length of (2) is shorter than that of (1), which reduces the required torque significantly while somewhat decreasing the accuracy.

For Target-3, the sampling results when $N_{joint} = \{3, 4\}$ and the inverse kinematics results for one Pareto front solution for each are shown in Fig. 8. The solution with the smallest E_x is shown for $N_{joint} = 3$, but E_x is not zero and E_τ is large. The target coordinates of Target-3 are similar to those of Target-2, and $N_{joint} = 3$ seems to be able to achieve $E_x = 0$, but the widths of the target positions are wider than that of Target-2, and the link length is insufficient to achieve $E_x = 0$. On the other hand, for $N_{joint} = 4$, a design where E_x is almost zero and E_τ is also zero is generated. The target positions of Target-3 are aligned on the plane where $z = 0.5$, and thus a so-called SCARA type robot is generated.

B. Automatic Design Optimization for Actuator Module Configuration

Target-1 and Target-2 shown in Fig. 9 are prepared as the target positions. Note that $a_{\{x,z\}}$ and a_y are specified in the range [-1.0, 0.0] [m] and [-1.0, 1.0] [m], respectively.

For Target-1, the sampling results and inverse kinematics results for one Pareto front solution are shown in the upper figures of Fig. 10. Since the target positions are distributed over a wide range, the range is covered by using a yaw-axis joint at the root of the body and two pitch-axis joints next to it. The actual motions of the modular robot built with the automatically designed parameters are shown in Fig. 10. As an example, the task of moving a monitor over a wide range is conducted by attaching a monitor to the end of the arm. By using the lock-release mechanism of the module, the robot is operated by direct teaching in the released state and successfully moves a heavy monitor in the locked state by playing back the taught motion. Since all modules are equipped with IMUs, it is possible to detect vibrations from humans to the monitor and replay the taught motion automatically.

For Target-2, the sampling results and the inverse kinematics results for one Pareto front solution are shown in the upper figures of Fig. 11. In order to satisfy a large number of target positions in the upper front of the robot, a pitch-axis joint is used at the root of the body and two yaw-axis joints are used next to it. The actual motions of the modular robot built with the automatically designed parameters are shown in Fig. 11. As an example, the task of lighting operation is conducted by attaching a light to the end of the arm. By using the lock-release mechanism of the module, the robot is operated by direct teaching in the released state and successfully operates the light in the locked state. In a similar manner, it is also possible for the robot to operate a camera to take pictures or serve food.

V. DISCUSSION

The results obtained are summarized below. In this study, several target coordinates that should be realized by a robot are given, body structures with minimum control error and minimum torque are generated by simulations, and one of them is selected to construct an actual modular robot. From the simulation experiments, it is found that the obtained Pareto front solutions have various reasonable joint arrangements and link lengths. In order to minimize the torque, the yaw-axis joints are arranged at the root of the body to avoid the effect of gravity, and the link lengths are shortened as much as possible. The optimal number of joints can be obtained by running the optimization while changing the number of joints. Depending on the target coordinates, a general configuration such as a SCARA type robot can also be obtained. Next, from the actual robot experiments, the obtained optimal solution is actually constructed and the desired performance is successfully obtained. In particular, the actuator module with a lock-release mechanism enables the robot to carry heavy objects without back drive and to be operated by direct teaching. The actual robots have successfully moved monitors and performed lighting operations. Similar operations such as food delivery, tool delivery, and camera operation are also possible.

The use of Tree-Structured Parzen Estimator is versatile, and it can solve various problems of handling continuous and discrete parameters. Therefore, it is possible to solve a very wide range of optimization problems including kinematic branching and body dynamics in a unified manner by parameterizing parent links, gear ratios, and so on. On the other hand, the number of parameters is fixed due to the nature of black-box optimization. Of course, it is possible to define a large number of parameters and add another variable that expresses the range of the parameters to be used, but at present, each optimization is performed while changing the numbers of joints. Although we did not handle complicated

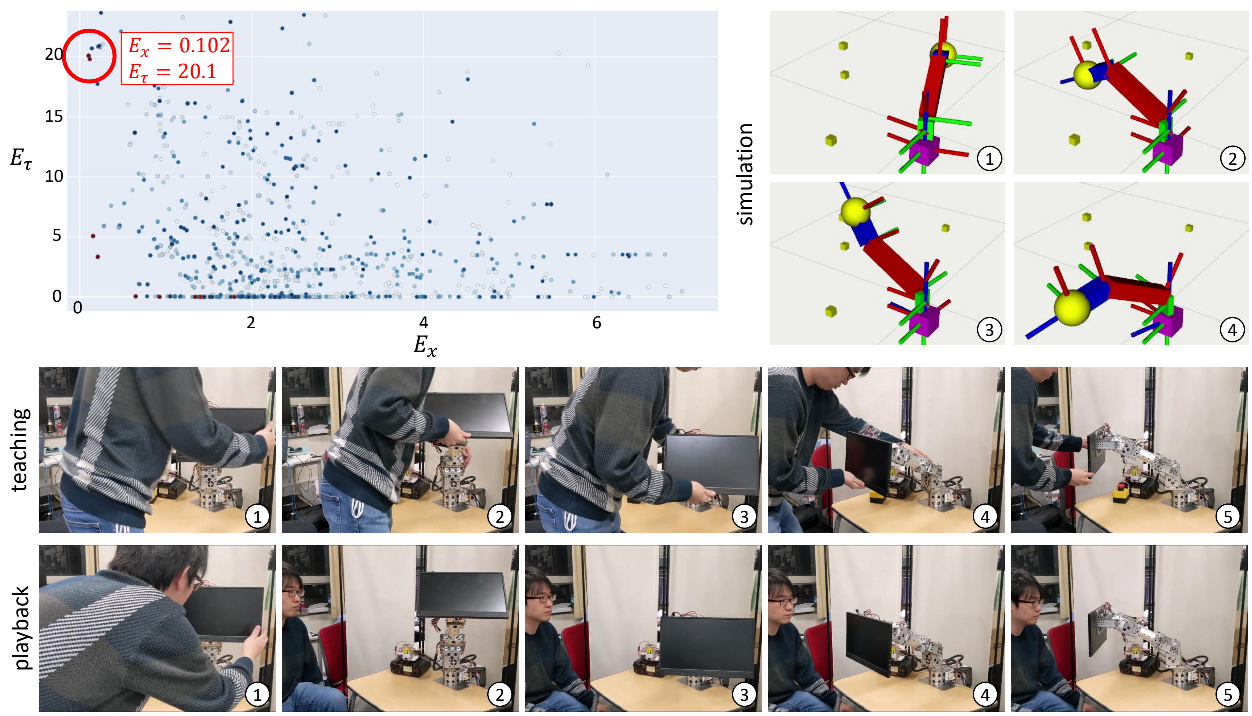

Fig. 10. The optimization results for Target-1 with actuator module configuration. One Pareto front solution is shown in simulation. The lower figures show the teaching and playback motions of the actual optimized modular robot for monitor movement.

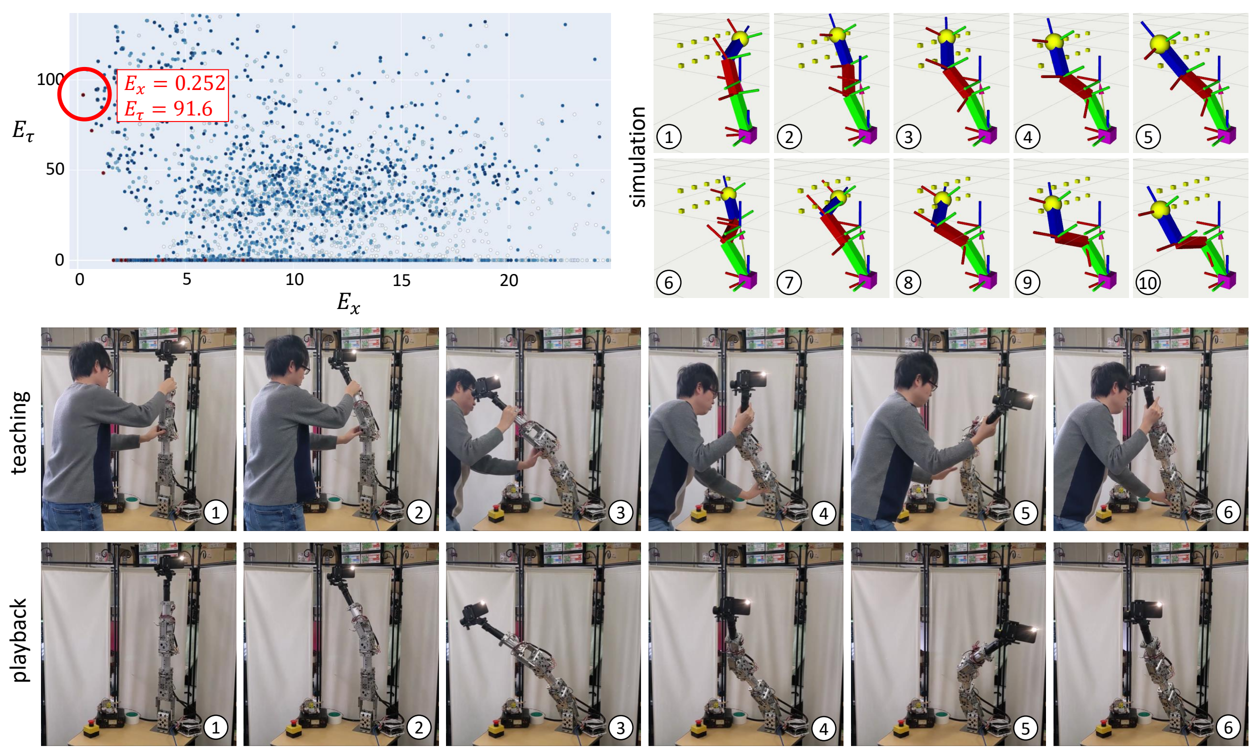

Fig. 11. The optimization results for Target-2 with actuator module configuration. One Pareto front solution is shown in simulation. The lower figures show the teaching and playback motions of the actual optimized modular robot for a lighting operation.

body parameters or evaluation functions in this study due to the nature of daily assistive robots, we would like to increase the degrees of freedom and handle more complicated parameters in the future. Also, an actuator module with a lock-release mechanism was used to carry heavy objects and be operated by direct teaching, but these characteristics are not directly incorporated into the optimization. In the future, we would like to study which part of the module should have the lock-release mechanism in combination with general actuator modules. Another important issue to be addressed is how to construct a novel body configuration combined with linear joints and tendon-driven actuators through black-box optimization.

VI. CONCLUSION

In this study, we have developed a modular robot for daily life support that is personalized to each user and task. The actuator module has a lock-release mechanism and allows various configurations of links and joints, which enables manipulation of heavy objects and direct teaching. By changing the orientation of the two types of attachments and the length of the links, a variety of bodies can be constructed. By optimizing these body design parameters with Tree-structured Parzen Estimator, a type of black-box optimization, a body that can achieve the desired motion with a small required torque can be constructed automatically. The optimized body was actually constructed and successfully used to operate monitors and lights. In the future, we would like to expand the range of the actuator design, target motions, and evaluation functions, toward developing robots that can adapt to various environments.

ACKNOWLEDGEMENT

This work was partially supported by JST Moonshot R&D under Grant Number JPMJMS2033.

REFERENCES

[1] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and M. Inaba, "Humanoid Motion Generation System on HRP2-JSK for Daily Life Environment," in *Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation*, 2005, pp. 1772–1777.

[2] M. Saito, H. Chen, K. Okada, M. Inaba, L. Kunze, and M. Beetz, "Semantic Object Search in Large-scale Indoor Environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Active Semantic Perception and Object Search in the Real World*, 2011.

[3] G. Yang and I. Chen, "Task-based optimization of modular robot configurations: minimized degree-of-freedom approach," *Mechanism and Machine Theory*, vol. 35, no. 4, pp. 517–540, 2000.

[4] Y. Xiao, Z. Fan, W. Li, S. Chen, L. Zhao, and H. Xie, "A Manipulator Design Optimization Based on Constrained Multi-objective Evolutionary Algorithms," in *Proceedings of the 2016 International Conference on Industrial Informatics*, 2016, pp. 199–205.

[5] A. Zhao, J. Xu, M. K. Luković, J. Hughes, A. Speilberg, D. Rus, and W. Matusik, "RoboGrammar: Graph Grammar for Terrain-Optimized Robot Design," *ACM Transactions on Graphics*, vol. 39, no. 6, pp. 1–16, 2020.

[6] J. Hu, J. Whitman, M. Travers, and H. Choset, "Modular Robot Design Optimization with Generative Adversarial Networks," in *Proceedings of the 2022 IEEE International Conference on Robotics and Automation*, 2022, pp. 4282–4288.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Proceedings of the 2014 Neural Information Processing Systems*, 2014, pp. 2672–2680.

[8] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Proceedings of the 2011 Neural Information Processing Systems*, 2011, pp. 2546–2554.

[9] T. F. Chan and R. V. Dubey, "A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 286–292, 1995.

[10] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time collision avoidance with whole body motion control for humanoid robots," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 2053–2058.

[11] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.