CONFERENCE PRE-PRINT

AUTOMATED DESIGN RATIONALIZATION OF ROBOT COMPONENT CONFIGURATION FOR IN-VESSEL TASK OF ITER BLANKET REMOTE MAINTENANCE

T. IWAMOTO
National Institutes for Quantum Science and Technology
Naka, Ibaraki, Japan
Email: iwamoto.takuya@qst.go.jp

Y. NOGUCHI
National Institutes for Quantum Science and Technology
Naka, Ibaraki, Japan

N. TAKEDA
National Institutes for Quantum Science and Technology
Naka, Ibaraki, Japan

Abstract

The paper presents the development of an automated design rationalization method for a robotic component of the ITER Blanket Remote Handling System (BRHS). The proposed method integrates two key elements: automation of the robot design evaluation process and Bayesian optimization for efficient design parameter exploration. To demonstrate its effectiveness, the method was applied to rationalize the shield block gripper (SBG), one of the components of the BRHS, which requires rationalization of its geometric configuration. In the constrained and complex environment of the vacuum vessel, the SBG must satisfy multiple design requirements, including reachability, joint load reduction, and component compactness. The developed method successfully identified a geometric configuration of the SBG that reduces joint torque loads by 50% compared with the original SBG design, while satisfying all design requirements under the given constraints with only one-tenth the number of iterations required for an exhaustive grid search.

1. INTRODUCTION

This paper presents an automated design rationalization method for a robotic component of the ITER Blanket Remote Handling System (BRHS) [1]. The developed method consists of two key elements: the automation of the robot design evaluation process and the application of Bayesian optimization (BO) for design parameter exploration. This approach improves design quality while reducing the required time and effort compared to the conventional manual process.

We first provide an overview of the BRHS and the shield block gripper (SBG), the target of design rationalization in this study. The BRHS is a robotic system for maintaining ITER blanket modules and mainly consists of an arc-shaped articulated rail structure and a heavy-duty manipulator, as shown in Fig. 1. The manipulator replaces blanket modules inside the vacuum vessel (VV) using an interchangeable gripper connected to the wrist part of the manipulator via a tool changer. The SBG is used for replacing shield block (SB) type blanket modules. While most SBs can be replaced using standard SBGs, certain narrow areas inside the VV require a specially designed SBG with an offset structure due to limited reachability, as illustrated in Fig. 2.

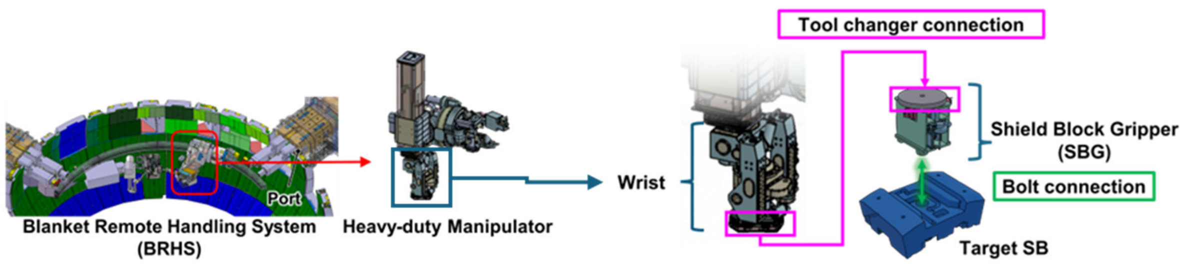
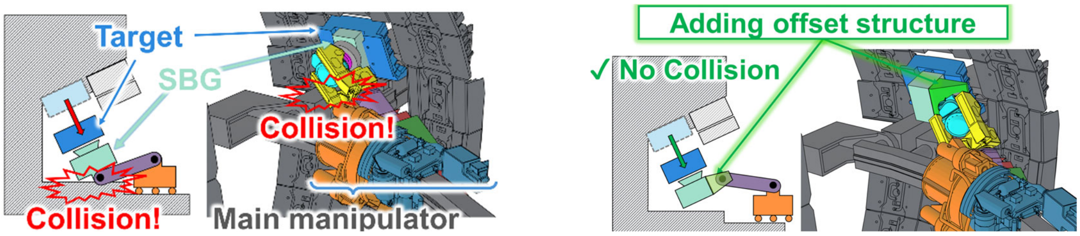


FIG. 1. BRHS configuration and SBG



(a) Narrow area for the BRHS (b) Adding an offset structure to the SBG

FIG. 2. Example of narrow area for the BRHS and concept of adding offset structure to the SBG

In the development of maintenance robotic components for nuclear fusion devices—such as the geometric design of SBG with offset structure—there are multiple challenging design constraints to address: ensuring reachability to the target components within the constrained environment of the VV, reducing the torque loads applied to robot joints, and ensuring component compactness. One effective strategy to meet such requirements is iterative design parameter exploration using virtual simulation. However, this process requires repeated tasks such as CAD modeling, construction of kinematic models, and simulation setup, which are time and effort consuming in the conventional manual method. As a result, only a limited number of design parameter conditions can be evaluated within a realistic timeframe, which may compromise design quality.

Therefore, to achieve both the reduction of design effort and the improvement of design quality under such challenging design constraints, the developed method integrates automation of the robot design evaluation process with BO for design parameter exploration. Regarding the automation of the robot design evaluation process, the developed method automates iterative updates of the 3D model and kinematic model of robotic components using Python scripts, and reachability analysis by utilizing Robot Operating System (ROS) [2] and MoveIt [3], which are widely used frameworks in the robotics field. For design parameter exploration, we regarded the design optimization problem in this study as a black-box optimization problem and applied BO, which is one of the effective approaches for solving such problems. With the developed method utilizing BO, a rationalized SBG design can be obtained—achieving required reachability, compactness, and joint load reduction—using approximately one-tenth the number of iterations compared to exhaustive grid search.

This paper is structured as follows: Section 2 presents details of the developed method, Section 3 shows demonstration of the developed method with the SBG case, and Section 4 concludes the paper.

2. DEVELOPED METHOD

This section describes the overview and individual implementation details of the developed method. The developed method serves as a tool for the basic design phase. The basic design phase, the process defining the overall geometry of the component based on the conceptual design, lies between the conceptualization and detailed design phases. As illustrated in Fig. 3, this process involves iteratively determining design parameters of the robot component geometry, performing 3D modeling, evaluating the robot's reachability, and modifying design parameters if reachability is not satisfied or further improvement is needed. As described in Section 1, these tasks have conventionally been performed manually, and increasing the number of design iterations to improve design quality requires time and effort. To address this, the proposed method automates the entire process, including model updates, reachability analysis, and design parameter iteration, thereby significantly reducing the human effort required for iterative design and enabling evaluation across a large number of design conditions. Furthermore, by integrating BO, the method enables efficient exploration of the design parameter space, achieving faster convergence to rationalized design than simple exhaustive evaluations.

In the following explanation, we present how the automation process and BO are implemented, using the actual design parameters of the robotic component that is rationalized in Section 3.

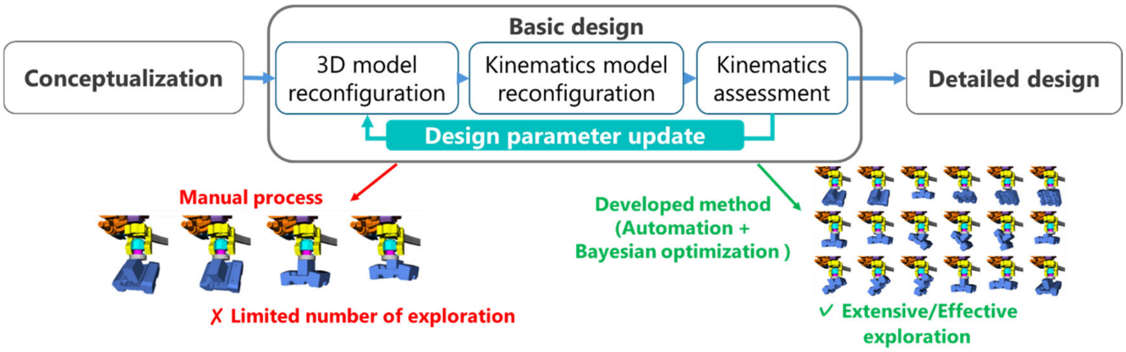


FIG. 3. Positioning of the developed method in the design workflow.

2.1. Definition of design parameters and objective function in this study

To facilitate the explanation in the following sections, we first define the design parameters and the objective function evaluated in this study.

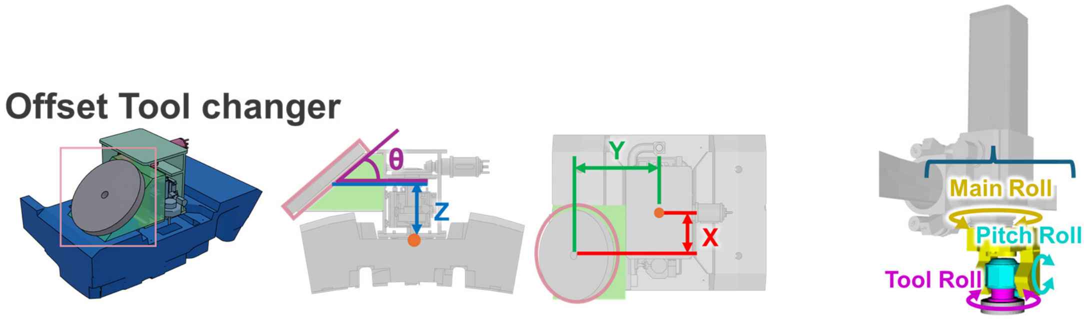
The design parameter $\boldsymbol{x} = (X, Y, Z, \theta)$, representing the offset structure of the SBG, is illustrated in Fig. 4 (*a*). Here, X, Y, Z are the translational distances from the center of the SB (indicated as an orange dot in the figure) to the mounting position of the tool changer, and θ is the mounting angle. These parameters define the offset structure between the tool changer and the main SBG structure which includes drive units. Given the constraints of transportation through the VV, it is desirable to maintain the compactness of the SBG, and thus the offset structure should be reasonably minimized. Furthermore, if a feasible path exists for a given offset structure, lower joint torque is desirable to ensure sufficient margins against the allowable torque limits of the robot joints and to facilitate correction of positioning errors caused by joint deflection.

Based on these considerations, the objective function $f(\boldsymbol{x})$, shown in (1), is defined to quantitatively evaluate the suitability of a design parameter $\boldsymbol{x}$. The function consists of two terms as follows.

— $S(\boldsymbol{x})$, which evaluates the compactness of the SBG with the offset structure, is defined as the normalized Euclidean norm of the translational distance parameters (X, Y, Z) with respect to the maximum value in the design space.
— $L(\boldsymbol{x})$, which represents the joint torque load, is calculated as a weighted sum of the maximum torque values observed at each joint during the planned path. In this study, the joints evaluated are the three axes of the wrist: Main Roll, Pitch Roll, and Tool Roll shown in Fig. 4 (*b*). The weights are defined as the inverse of the allowable torque for each axis, which are as follows: Main Roll: 50 kNm, Pitch Roll: 45 kNm, and Tool Roll: 7.5 kNm. Since an analytical formulation of $L(\boldsymbol{x})$ with respect to the design parameter $\boldsymbol{x}$ is difficult to derive, simulation-based evaluation is required to calculate $L(\boldsymbol{x})$. As a result, the function $L(\boldsymbol{x})$ and $f(\boldsymbol{x})$ become black-box functions. For this reason, BO was adopted to explore the design parameter space (as detailed in Section 2.3).

The optimized parameter $\boldsymbol{x}_{\mathrm{opt}}$ that minimizes $f(\boldsymbol{x})$ can be regarded as a rationalized design parameter as shown in (2).

If path planning fails for a given design parameter $\boldsymbol{x}$, the objective function $f(\boldsymbol{x})$ returns a value of 4. This return value corresponds to the worst design parameter case with excessive size and joint torque. Instead of applying an excessively large penalty (e.g., 10 or 100), the fixed return value of 4 is used for the following reason. The random sampling-based path planning algorithm used in the developed method cannot prove the nonexistence of a feasible path; in other words, a feasible path may still exist even when path planning fails. In particular, the boundary between feasible and infeasible regions where promising design parameters close to the Pareto front may exist, often results in path planning failures despite the actual feasibility of the design parameter $\boldsymbol{x}$. Assigning overly large penalties to such regions would cause the optimization algorithm to strongly avoid exploration around them, thereby leading to a degradation of optimization performance. Therefore, the fixed value equivalent to the worst case is adopted to provide a balanced treatment of failed cases and to preserve the robustness of the optimization process.



(a) Design parameters of the offset structure of the SBG *(b) Wrist joints*

FIG. 4. Design parameters and target joints evaluated in the objective function

$$f(\boldsymbol{x}) = \begin{cases} S(\boldsymbol{x}) + L(\boldsymbol{x}) & \text{if path exists} \\ 4 & \text{if path planning failed} \end{cases} \tag{1}$$

$$\boldsymbol{x}_{\mathrm{opt}} = \arg\min_{\boldsymbol{x}} (f(\boldsymbol{x})) \tag{2}$$

Where:

S : Term considering SBG size

$$S = s / \max(s)$$
$$s = \sqrt{X^2 + Y^2 + Z^2}$$

L : Term considering required joint torque

$$L = \sum_i w_i \tau_i$$

X : Translational offset X [mm]
Y : Translational offset Y [mm]
Z : Translational offset Z [mm]
θ : Angular offset [deg.]
i : Index of target joints
τ_i : The maximum torque of joint i in planned path
w_i : Weight factor of joint i

2.2. Automation of robot re-configurations

Automated re-configuration of the robot based on the specified design parameters is one of the most vital parts of the developed method. To enable evaluation of the four-dimensional parameter $\boldsymbol{x} = (X, Y, Z, \theta)$ without requiring manual CAD operations, a Python script was developed to automatically update the 3D model of the SBG with the offset structure according to the given parameter values. The updated geometry is then reflected in the kinematic model used for reachability assessment. This evaluation is performed using ROS and MoveIt, which are widely used frameworks in the robotics field. In this study, Rapidly-exploring Random Tree star (RRT*) [4] was applied as path planning method. Sampling-based planning methods such as RRT* are effective in generating feasible paths for robots operating in environments with obstacles. Since collision avoidance with other in-vessel equipment is critical for the manipulator, these methods are considered appropriate for its path planning. The entire evaluation process—comprising geometry update and reachability assessment—can be used as a black-box function that maps an input parameter $\boldsymbol{x}$ to an objective value $f(\boldsymbol{x})$ as shown in Fig. 5.

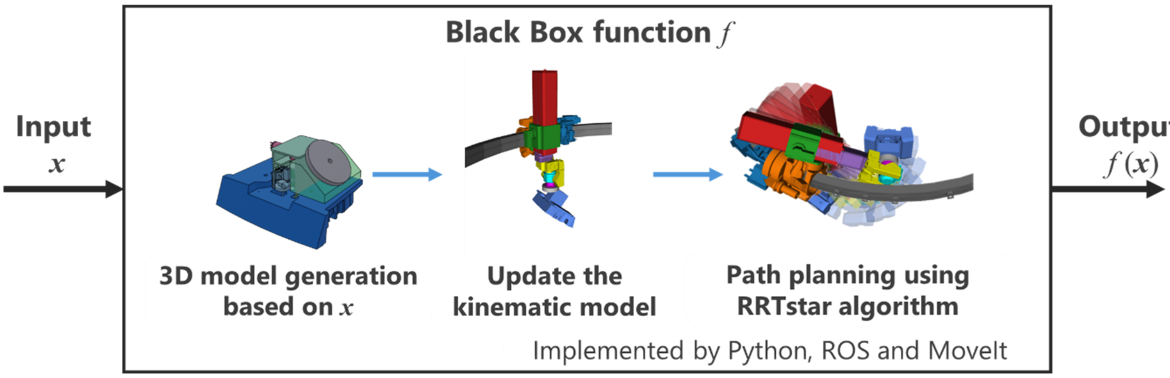


FIG. 5. Automated process implemented using Python, ROS, and MoveIt, serving as a black-box function

2.3. Bayesian Optimization for parameter exploration

This section explains the background of selecting BO and its behavior in the parameter exploration.

As discussed in Sections 2.1 and 2.2, since the objective function $f(\boldsymbol{x})$ is a black-box function, minimizing this function can be regarded as a black-box optimization problem. BO was selected to address the problem because of its effectiveness in balancing exploration and exploitation. This study aims not only to obtain the optimized parameters, but also to estimate the distribution of the objective function around promising region within a realistic timeframe. This is because understanding the distribution of the objective function around the optimized parameters enables the identification of multiple alternative design candidates, which can be used as backups in case design revisions become necessary during the detailed design phase. BO aligns well with this goal, as its exploration reduces uncertainty over a broad parameter space while its exploitation increases sampling density near the optimized parameters.

In BO, parameter exploration is conducted while estimating the distribution of the objective function based on observed results. The exploration progresses by determining the next sampling point from the estimated distribution. In this study, we applied the Tree-structured Parzen Estimator (TPE) sampler [5] as the surrogate model due to its scalability and computational efficiency. The following section briefly introduces the sampling strategy. If observations $D_n=\{(\boldsymbol{x}_1,f(\boldsymbol{x}_1)),(\boldsymbol{x}_2,f(\boldsymbol{x}_2)),\ldots,(\boldsymbol{x}_\mathrm{n},f(\boldsymbol{x}_\mathrm{n}))\}$ are obtained, the posterior distribution $P(f(\boldsymbol{x})|\boldsymbol{x}, D_n)$ is formulated as (3).

$$P\,(f(\boldsymbol{x})|\boldsymbol{x},\, D_n)=\frac{P(\boldsymbol{x}|\,f(\boldsymbol{x}),\, D_n)P(f(\boldsymbol{x})|\, D_n)}{P(\boldsymbol{x}|\, D_n)} \tag{3}$$

In the TPE sampler, $P(f(\boldsymbol{x})|\boldsymbol{x})$ is estimated by partitioning $f(\boldsymbol{x})$ values as (4) by the threshold y^*. Here, $l(\boldsymbol{x})$ and $g(\boldsymbol{x})$ represent the sum of parameter distributions estimated for each dimension using kernel density estimation based on observations D_n. The threshold y^* is determined such that it satisfies the relationship defined by the hyperparameter γ, as shown in (5). In this study, the hyperparameter γ is defined as shown (6). This definition is intended to use the top 10% of samples when the total number of samples is small, or the top 25 samples when the sample size is large, for partitioning into $l(\boldsymbol{x})$ and $g(\boldsymbol{x})$.

$$P(f(\boldsymbol{x})|\boldsymbol{x})=\begin{cases} l(\boldsymbol{x}) & \text{if } f(\boldsymbol{x})<y^* \\ g(\boldsymbol{x}) & \text{if } f(\boldsymbol{x})\geq y^* \end{cases} \tag{4}$$

$$\gamma = P\left(f(\boldsymbol{x})<y^*\right) \tag{5}$$

$$\gamma = \min(0.1,\, 25/n) \tag{6}$$

To select the next sampling point, Expected Improvement $EI(\boldsymbol{x})$ is defined as (7). In parameter exploration, the next sampling candidate $\boldsymbol{x}_{\mathrm{n}+1}$ is the point where $EI(\boldsymbol{x})$ is maximized.

$$EI(\boldsymbol{x})=\int_{-\infty}^{\infty}\max\left(y^*\text{-}f(\boldsymbol{x}),0\right)\,P(f(\boldsymbol{x})|\boldsymbol{x})\;dy \tag{7}$$

While the detailed derivation is omitted, substituting the expression of $P(f(\boldsymbol{x})|\boldsymbol{x})$ defined in (4) into (7) allows us to obtain the relationship for $EI(\boldsymbol{x})$ as (8).

$$EI(\boldsymbol{x}) \propto \left(\gamma + \frac{g(\boldsymbol{x})}{l(\boldsymbol{x})} (1-\gamma) \right)^{-1} \tag{8}$$

Within the defined parameter space, candidate points are globally sampled according to the distribution $l(\boldsymbol{x})$, then $g(\boldsymbol{x})/l(\boldsymbol{x})$ is evaluated for each. The point where $g(\boldsymbol{x})/l(\boldsymbol{x})$ is minimized is selected as the next sampling point $\boldsymbol{x}_{\mathrm{n}+1}$ as (9).

$$\boldsymbol{x}_{\mathrm{n}+1} = \arg\min_{\boldsymbol{x}} \left(\frac{g(\boldsymbol{x})}{l(\boldsymbol{x})} \right) \tag{9}$$

In this study, Optuna [6], an open source hyperparameter optimization framework, was used to conduct the exploration using the TPE sampler.

3. DEMONSTRATION

We performed numerical experiments to demonstrate applicability of the developed method for the BRHS component design rationalization. This section describes the experimental conditions and results.

3.1. Target SB and SBG

As the target SB to which the rationalized SBG design is applied, SB15NE was selected. SB15NE is located in a constrained area where the articulated rail and its supporting structure obstruct the manipulator. When a standard SBG is used, as shown in Fig. 6(*a*), the manipulator cannot reach the target SB because of collisions between the wrist part and the surrounding environment. To address this, an SBG design with the offset structure as shown in Fig. 6(*b*) had originally been considered. However, with this offset structure, the moment arms between the center of mass of the target SB and the wrist joints of the manipulator become significantly longer, resulting in excessive joint loads. Therefore, there was a clear need to rationalize the geometry of the SBG to balance reachability, joint load reduction, and component compactness.

(a) Spatial constraints to SB15NE *(b) Original design*

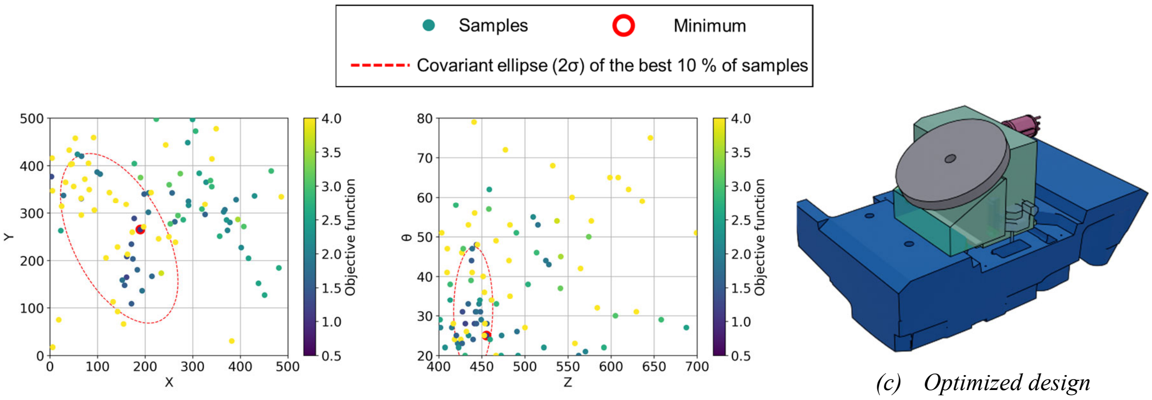
FIG. 6. Target SB location and original design of the corresponding SBG

3.2. Four-parameter exploration and comparison to exhaustive grid-search

To rationalize the SBG design for the SB15NE case, both a grid search and BO for the objective function defined in Section 2.1 were conducted using the framework described in Section 2.2. The design parameter exploration range is defined as (10). In the grid search, translational parameters were varied at 100 mm intervals and the angular parameter at 10-degree intervals, resulting in a total of 1008 iterations. This exhaustive search required about 48 hours of computation. In the exploration by BO, we did not impose discretization constraints as the grid search. The exploration by BO was conducted with up to 100 iterations and completed in about 5 hours.

$$\begin{gathered} 0 \le X \le 500 \\ 0 \le Y \le 500 \\ 400 \le Z \le 700 \\ 20 \le \theta \le 80 \end{gathered} \tag{10}$$

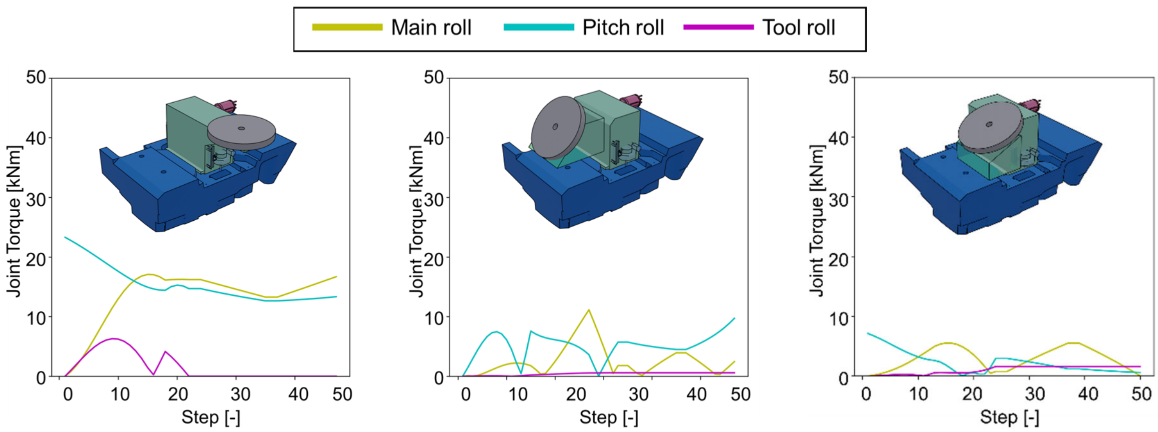
Sampled points evaluated during BO were projected onto the (X, Y) and (Z, θ) planes as shown in Fig. 7. Color intensity indicates the objective function value at each point, with darker blue representing better values. The sample corresponding to the optimized design is highlighted with a red border. The optimized design parameter was $\boldsymbol{x} = (189, 265, 455, 25)$ and the optimized objective function value was 1.04. The clustering of samples around the optimized one, while also covering other regions, demonstrates the expected behavior of the BO algorithm in balancing exploration and exploitation, as discussed in Section 2.3.



(a) Projection to XY Plane *(b) Projection to Zθ Plane* *(c) Optimized design by BO*

FIG. 7. Samples projected to 2D planes explored by BO

The wrist joint torque histories corresponding to each design during the target SB replacement are shown in Fig. 8. Table 1 shows design parameters, the maximum torque for each wrist joint, and the objective function values. Compared to the original design shown in Fig. 8(a), which is subjected to high joint torques, both the design by grid search (Fig. 8(b)) and the design by BO (Fig. 8(c)) significantly reduced the maximum torque—by more than 50% for the pitch and tool roll joints. This improvement is attributed to the rational repositioning of the tool changer by the optimized offset structure, which shortened the distance between the target SB's center of mass and the robot's joint. Moreover, the solution obtained through BO achieved a lower objective value than that of the grid search. While this is a natural outcome from exploring a continuous design space without discretization constraints, it is worth noting that achieving similar performance via grid search would require finer discretization, resulting in a substantial increase in the number of evaluations. Therefore, from the perspective of parameter exploration efficiency, the use of BO is demonstrated as an effective strategy for rationalizing the SBG design addressed in this study.



(a) Original design *(b) Optimized design by grid search* *(c) Optimized design by BO*

FIG. 8. Comparison of the wrist joint torque histories for SB15NE replacement for each design

TABLE 1. COMPARISON OF DESIGN PARAMETERS

Design type	Design parameters [mm, deg.]				Max. Torque [kNm]			$f(\boldsymbol{x})$
	X	Y	Z	θ	Main	Pitch	Tool	
Original design	600	0	482	0	17.1	23.3	6.3	2.48
Optimized design by grid search	0	500	400	50	11.1	9.7	0.6	1.16
Optimized design by BO	189	265	455	25	5.5	7.1	1.6	1.04

4. CONCLUSION

In this study, to achieve both the reduction of design effort and the improvement of design quality under challenging design constraints, the developed method integrates automation of the robot design evaluation process with BO for design parameter exploration. We successfully obtained the rationalized SBG design with approximately one-tenth the number of iterations compared to exhaustive grid search.

The proposed method is not limited to fusion maintenance robots such as the BRHS but can also be applied to robots operating in challenging environments, including decommissioning and rescue missions. In these applications, similar to the BRHS, the design must adapt to multiple constraints such as operation in confined spaces and limitations in actuator capacity. The proposed method could contribute to the efficient identification of feasible design.

REFERENCES

[1] NOGUCHI, Y., SAITO, M., MARUYAMA, T., TAKEDA, N., Design progress of ITER blanket remote handling system towards manufacturing, Fusion Eng. Des. **136** (2018) 722–728.

[2] QUIGLEY, M., CONLEY, K., GERKEY, B., FAUST, J., FOOTE, T., LEIBS, J., WHEELER, R., NG, A.Y., "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software (Proc. Int. Conf. Kobe, 2009), Kobe (2009) 5.

[3] COLEMAN, D., SUCAN, I., CHITTA, S., CORRELL, N., Reducing the barrier to entry of complex robotic software: a moveit! case study, arXiv preprint arXiv:1404.3785 (2014).

[4] KARAMAN, S., FRAZZOLI, E., Incremental Sampling-based Algorithms for Optimal Motion Planning, Robotics Science and Systems VI, **104**, 2, (2010) 267–274.

[5] BERGSTRA, J., BARDENET, R., BENGIO, Y., KÉGL, B., Algorithms for hyper-parameter optimization, Advances in neural information processing systems, **24** (2011).

[6] AKIBA, T., SANO, S., YANASE, T., OHTA, T., KOYAMA, M., "Optuna: A next-generation hyperparameter optimization framework", Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (2019) 2623–2631.